Global inverse kinematics via mixed-integer convex optimization

As Published: 10.1177/0278364919846512

Publisher: SAGE Publications

Persistent URL: https://hdl.handle.net/1721.1/136519

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Global Inverse Kinematics via Mixed-Integer Convex Optimization

Hongkai Dai[1], Gregory Izatt[2] and Russ Tedrake[1,2*]

Abstract In this paper we present a novel formulation of the inverse kinematics (IK) problem with generic constraints as a mixed-integer convex optimization program. The proposed approach can solve the IK problem globally with generic task space constraints, a major improvement over existing approaches, which either solve the problem in only a local neighborhood of the user initial guess through nonlinear non-convex optimization, or address only a limited set of kinematics constraints. Specifically, we propose a mixed-integer convex relaxation on non-convex $SO(3)$ rotation constraints, and apply this relaxation on the inverse kinematics problem. Our formulation can detect if an instance of the IK problem is *globally* infeasible, or produce an approximate solution when it is feasible. We show results on a 7-joint arm grasping objects in a cluttered environment, and a quadruped standing on stepping stones. We also compare our approach against the analytical approach for a 6-joint manipulator. The code is open-sourced at `drake.mit.edu` [29].

1 Introduction

The inverse kinematics (IK) problem is one of the most fundamental problems in robotics. The problem is to find robot postures, so as to satisfy certain kinematic constraints. Traditionally the kinematic constraints are on the end-effectors, such as hand reaching a certain location with a given orientation. More recently applications have required more complex constraints in the task space, such as grasping a mug on the table while keeping the robot free from collision (Fig 1a).

Solving the IK problem is quite challenging, as it requires solving a set of non-linear equations involving products of trigonometric functions (sin/cos). There have been tremendous efforts to solve the inverse kinematics problem[11, 21, 24, 5], and they can be coarsely categorized as analytical or numerical. The analytical approach

* [1] Toyota Research Institute [2] Computer Science and Artificial Intelligence Lab, MIT, `hongkai.dai@tri.global, gizatt@mit.edu, russ.tedrake@tri.global`.

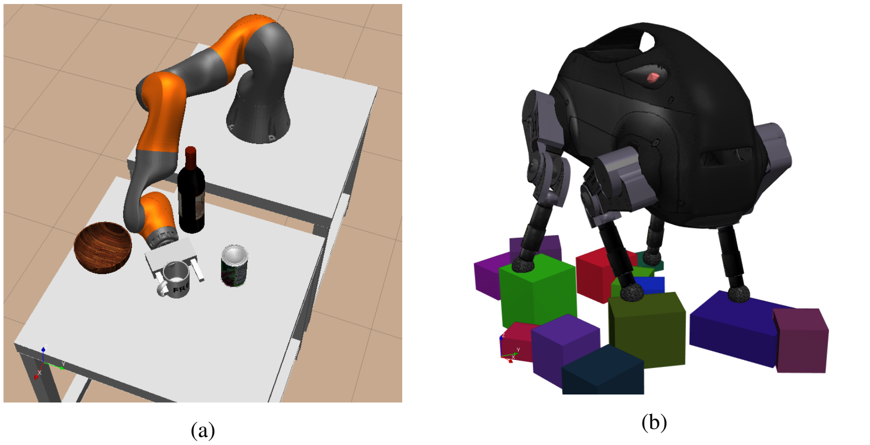

(a)
(b)

Fig. 1: Our IK solver finds a collision free grasping posture for the KUKA IIWA arm (left), and a posture for Little Dog standing on stepping stones (right).

solves the kinematics equations as polynomials of sine and cosine and can produce closed-form solutions. It is widely known that many manipulators with 6 Degree of Freedoms (DoFs) allow analytical solution for the end effector to reach a specified position with a given orientation [26, 20]. More generally, Diankov et al. introduced *IKfast*, which can find inverse kinematics solutions for more complicated robots, but the set of kinematics constraints it can handle is still limited [16]. Since the analytical approach solves the inverse kinematics problem by computing roots of polynomial equations, it does not permit inequality constraints on the link poses [2], which occur frequently when planning postures for certain tasks. This shortcoming makes it hard to handle general task space constraints involving inequalities, such as "putting the hand between the two boxes on the table".

On the other hand, the numerical approach can solve inverse kinematics problems for complicated robots with generic constraints. This approach formulates an inverse kinematics problem as a general non-convex nonlinear optimization problem, and calls gradient-based nonlinear solvers to handle these non-convex constraints [25, 13, 17, 4, 9]. The drawback of this approach is that the solution heavily relies on the user-supplied initial seed. Because the nonlinear optimization typically solves the problem locally around the initial seed, hence it can get trapped in some locally infeasible region and never reach the solution far away. As a result, when these non-convex nonlinear solver reports the problem is infeasible, it only asserts local infeasibility, providing no guarantee on the global solution [6]. We will show that our approach can significantly improve the robustness and success rate of the IK solver, over the gradient-based nonlinear optimization approach.

We shall use a numerical approach to the IK problem with generic constraints, while also obtain the global solution. Instead of formulating the problem through non-convex nonlinear optimization, we instead consider a mixed-integer convex optimization formulation, which does not require an initial seed, and can provide a global solution [8], such as a certificate of global infeasibility. The non-convexity of the IK problem originates from the non-convixity of $SO(3)$ constraints on rotation. If we choose to represent rotation with a matrix $R = [\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_3] \in \mathbb{R}^{3\times 3}$, this rotation matrix should satisfy the following constraints

$$\mathbf{u}_i^\mathrm{T}\,\mathbf{u}_i = 1 \text{ (Unit length)} \tag{1a}$$

$$\mathbf{u}_i^\mathrm{T}\,\mathbf{u}_j = 0 \text{ if } i \neq j \text{ (Orthogonality)} \tag{1b}$$

$$\mathbf{u}_i \times \mathbf{u}_j = \mathbf{u}_k, \tag{1c}$$

where $(i,j,k) = (1,2,3),(2,3,1)$ or $(3,1,2)$ in (1c). We choose the rotation matrix as the orientation representation, since the position of a point attached to a body is a *linear* expression of the body rotation matrix; on the other hand, the position would be a *nonlinear* expression, if we were to represent orientation using other forms, such as unit quaternions or angle-axis.

Obviously the orientation constraint $SO(3)$ (1a)-(1c) is non-convex. Take constraint (1a) as an example: geometrically, the convex combination of two points on the surface of a unit sphere, lies strictly in the interior of the unit sphere. Algebraically, an equality constraint is non-convex, if it includes quadratic terms such as $\mathbf{u}_i(1)^2$ in (1a), or bilinear terms like $\mathbf{u}_i(1)\mathbf{u}_j(1)$ in (1b).

Various convex relaxations for $SO(3)$ constraints have been proposed. Saunderson et al. proved that the convex hull of the set of rotation matrices can be described by a *positive semidefinite constraint*, a special type of convex constraint [27]. In [12] this convex relaxation is exploited to solve the IK problem through a sequence of convex optimizations. The drawback of this approach [12] is that the convex hull of the rotation matrix is a rather loose relaxation of $SO(3)$. For example, the matrix $\left[\mathbf{u}_1, \mathbf{0}, \mathbf{0}\right] = 0.5\left[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\right] + 0.5\left[\mathbf{u}_1, -\mathbf{u}_2, -\mathbf{u}_3\right]$ is in the convex hull for any rotation matrix $\left[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\right]$. Hence quite often the relaxed convex program is too loose to detect global infeasibility of the original IK problem. To overcome this, we seek a tighter mixed-integer convex relaxation of the $SO(3)$ constraint.

Our intuition is that instead of considering the convex hull of *all* rotation matrices as in [27], we divide the range of the rotation matrices into smaller intervals, and compute the convex hull of each small interval to obtain a tighter approximation. We can then constrain the approximated rotation matrix to be within *one of* the convex hulls. This approach is inspired by Deits et.al. [14], in which the orientation constraint $SO(2)$ in 2D is approximated by mixed-integer linear constraints. In a similar fashion, we replace (1a)-(1c) with mixed-integer convex (quadratic) constraints, as a relaxation of the original non-convex $SO(3)$ constraints. This relaxation allows the inverse kinematics problem to be formulated as a mixed-integer convex optimization program, which is amenable to the branch-and-bound algorithm by branching on integer variables, and can be solved efficiently by modern solvers [1]. Like convex optimization, mixed-integer convex optimization also does not rely on the initial seed, and warrants global solution [28, 7]. With the rapid advancement in numerical solvers, mixed integer convex optimization becomes increasingly popular in robotics [30, 22, 15]. In this paper we show that our approach can either produce an approximate solution to the inverse kinematics problem, or prove the problem is globally infeasible for generic constraints.

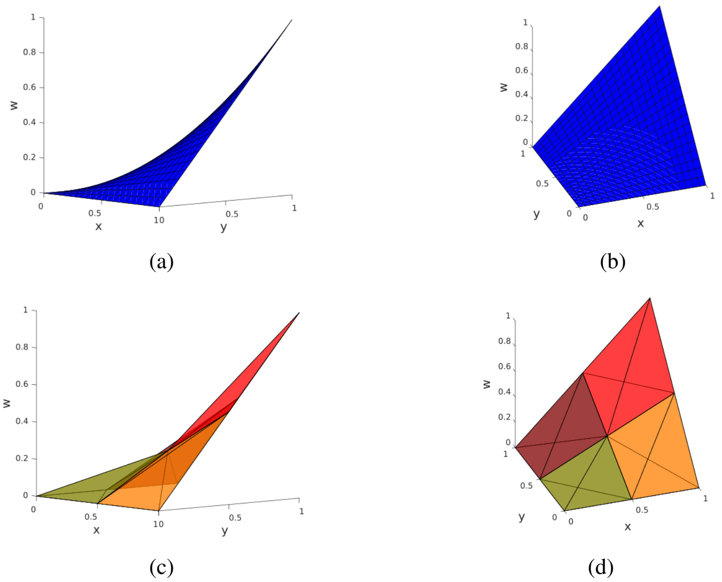
2 Background

In this section we briefly review how to relax a non-convex quadratic constraint as mixed-integer convex constraints, which will be used to relax the non-convex $SO(3)$ constraint on the rotation matrix (1a)-(1c).

Consider a non-convex quadratic constraint

$$\{\mathbf{z} | a \leq \mathbf{z}^{\mathrm{T}} Q\mathbf{z} + \mathbf{c}^{\mathrm{T}} \mathbf{z} \leq b\}, \tag{2}$$

where the matrix Q is not necessarily positive semidefinite. This quadratic constraint contains bilinear product terms like xy, visualized in Fig 2a -2b. It is NP-hard to obtain an exact global optimal solution to this non-convex problem, but there has been a lot of research on obtaining an *approximated* one through mixed-integer-convex optimization [23, 19]. The key idea is to partition the range of each variable into smaller intervals, and replace the bilinear term xy with a linear term which approximates xy inside each interval. To see this geometrically, we draw the surface of $w = xy$ in Fig 2 within $x \in [0,1], y \in [0,1]$. We then partition the range $[0,1]$ into two intervals, with interval i as $[\phi_i, \phi_{i+1}]$. The convex hull of $w = xy$ in interval $x \in [\phi_i, \phi_{i+1}], y \in [\phi_j, \phi_{j+1}]$ is a tetrahedron. We can then relax the constraint $w = xy$, with the constraint that the point (x, y, w) lies within one of the tetrahedrons. We will use some binary variables to determine within which tetrahedron the point lies.



(a)

(b)

(c)

(d)

Fig. 2: 2a,2b: $w = xy$ surface in the range $x \in [0,1], y \in [0,1]$ from two perspectives. In 2c,2d we divide the range $[0,1]$ into two intervals, with $(\phi_0, \phi_1, \phi_2) = (0, 0.5, 1)$, and compute the convex hull of the surface in each interval $x \in [\phi_i, \phi_{i+1}], y \in [\phi_j, \phi_{j+1}]$. Each convex hull is a tetrahedron drawn in distinct colors.

To enforce that (x,y,w) falls into one of the convex hulls, we rely on the *special ordered set of type 2* (sos2) constraint [3], defined as follows

Definition 1. Special ordered set of type 2: $\lambda = (\lambda_0,\dots,\lambda_n) \in \mathbb{R}^{n+1}$ is in sos2, if

$$\sum_i \lambda_i = 1, \lambda_i \geq 0 \tag{3a}$$

$$\exists j \in 0,...,n-1, \text{ s.t } \lambda_i = 0 \, \forall i \neq j, i \neq j+1. \tag{3b}$$

Namely at most two entries in λ can be strictly positive, these two entries must be consecutive in their ordering.

The sos2 constraint is often used to formulate a piecewise linear approximation to a nonlinear function. For example in Fig 3, we approximate $f(x)$ with a new variable w satisfying

$$\begin{bmatrix} x \\ w \end{bmatrix} = \sum_i \lambda_i \begin{bmatrix} \phi_i \\ f(\phi_i) \end{bmatrix}, \quad \lambda \text{ is in sos2} \tag{4}$$

(4) forces the point (x,w) to be on the black lines in Fig 3.

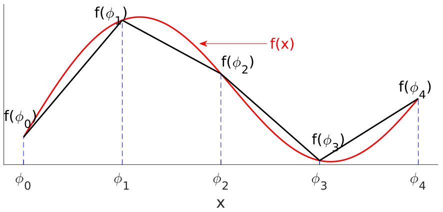


Fig. 3: A piecwise linear function (black lines) approximating the nonlinear function $f(x)$ (red curve).

We can formulate the sos2 (3) as mixed-integer linear constraints by introducing auxiliary binary variables. In this paper we adopt the formulation proposed in [31], which introduces $\log_2 n$ binary variables. We refer the readers to [31] for more details on sos2 constraint.

To impose the constraint that (x,y,w) is in one of the tetrahedron in Fig 2c-2d, we introduce auxiliary continuous variables $\gamma \in \mathbb{R}^{(n+1)\times(n+1)}, \alpha \in \mathbb{R}^{n+1}, \beta \in \mathbb{R}^{n+1}$ with the additional constraints

$$x = \sum_i \alpha_i \phi_i, \; y = \sum_j \beta_j \phi_j \tag{5a}$$

$$w = \sum_i \sum_j \gamma_{i,j} \phi_i \phi_j \tag{5b}$$

$$\sum_j \gamma_{i,j} = \alpha_i, \; \sum_i \gamma_{i,j} = \beta_j, \; \gamma_{i,j} \geq 0 \tag{5c}$$

$$\alpha, \beta \text{ are in sos2 }. \tag{5d}$$

Constraint (5d) is formulated as mixed-integer linear constraints with auxiliary binary variables.

To summarize, for a non-convex quadratic constraint (2), we can replace the bilinear term xy with w satisfying (5), and obtain mixed-integer convex constraints as a relaxation of the original non-convex constraint.

3 Approach

As explained in Section 1, the non-convexity of the inverse kinematics problem originates from the non-convexity of the $SO(3)$ constraint. In order to obtain a global solution to this non-convex problem, in sub-section 3.1, we first relax the $SO(3)$ constraint (1a)-(1c), to a set of mixed-integer convex constraints, using the technique described in Section 2. Then in sub-section 3.2, we formulate the inverse kinematics problem by searching over the link poses (position/orientation) satisfying both the kinematic constraints, and the relaxed $SO(3)$ constraint introduced in sub-section 3.1. Finally, we project the approximated rotation matrices to the $SO(3)$ manifold to obtain the angle of each joint within the joint limits.

3.1 Rotation constraint relaxation

We aim to find a matrix $\bar{R} \in \mathbb{R}^{3\times 3}$ satisfying a relaxation of the $SO(3)$ constraint (1a)-(1c). To this end, we cut the range $[-1,1]$ into n small intervals. The quadratic constraints (1b)-(1c) can be relaxed by replacing each bilinear term with a new variable approximating the bilinear term within each interval, described by (5).

To relax the unit length constraint $\mathbf{u}_i^{\mathrm{T}}\,\mathbf{u}_i = 1$ (1a), we first introduce auxiliary variables $\mathbf{w}_i \in \mathbb{R}^3, \lambda^{i,j} \in \mathbb{R}^{n+1}$ satisfying

$$\begin{bmatrix}\mathbf{u}_i(j)\\ \mathbf{w}_i(j)\end{bmatrix} = \sum_{k=0}^{n} \lambda_k^{i,j} \begin{bmatrix}\phi_k\\ \phi_k^2\end{bmatrix}, \quad \lambda^{i,j} \text{ is in sos2.} \tag{6}$$

Constraint (6) forces the point $(\mathbf{u}_i(j), \mathbf{w}_i(j))$ to be on the black lines in Fig 4, for $n = 4$ case. Namely $\mathbf{w}_i(j)$ is an upper-bound of $\mathbf{u}_i(j)^2$. We can thus relax the unit length constraint (1a) as

$$\mathbf{u}_i^{\mathrm{T}}\,\mathbf{u}_i \leq 1 \tag{7a}$$

$$\mathbf{w}_i(1) + \mathbf{w}_i(2) + \mathbf{w}_i(3) \geq 1. \tag{7b}$$

Geometrically, constraint (7a) forces the rows/columns of the rotation matrix to be within the unit sphere, while constraint (7b) pushes the row/column vector from inside the unit sphere, such that the Euclidean length of the row/column vector is bounded from below.

While the generic technique introduced in section 2 can already approximate the orthogonality constraint, it is desirable to introduce an additional relaxation, because enforcing the solution to live in the intersection of different relaxations can only improve its approximation quality. We thus impose the following convex (quadratic) constraints as a relaxation to the orthorgonal constraint (1b)

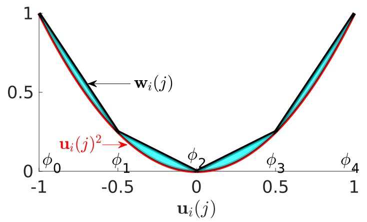


Fig. 4: The range of $\mathbf{u}_i(j)$ is cut into 4 intervals, with $(\phi_0,\phi_1,\phi_2,\phi_3,\phi_4) = (-1,-0.5,0,0.5,1)$. The adjacent two points (ϕ_i,ϕ_i^2) and $(\phi_{i+1},\phi_{i+1}^2)$ are connected with the black straight line. We introduce an auxiliary variable $\mathbf{w}_i(j)$ to be on the black lines, as an upper bound of the red curve $\mathbf{u}_i(j)^2$. The unit length constraint $\sum_j \mathbf{u}_i(j)^2 = 1$ is relaxed as $\sum_j z_i(j) = 1$, where $z_i(j)$ lies within the shaded cyan region, as a relaxation of the red curve $\mathbf{u}_i(j)^2$.

$$|\mathbf{u}_i \pm \mathbf{u}_j|_2^2 \le 2, \tag{8}$$

where $\mathbf{u}_i, \mathbf{u}_j$ are two distinctive rows/columns of the matrix $\bar{R}$, $|\bullet|_2$ is the l_2 norm of a vector. Constraint (8) is equivalent to $|\mathbf{u}_i^\text{T}\ \mathbf{u}_j| \le \frac{1}{2}(2 - \mathbf{u}_i^\text{T}\ \mathbf{u}_i - \mathbf{u}_j^\text{T}\ \mathbf{u}_j)$. When the vectors $\mathbf{u}_i, \mathbf{u}_j$ are "close to" the unit sphere (due to the constraint (7a)-(7b)), constraint (8) approximates the orthogonal constraint $\mathbf{u}_i^\text{T}\ \mathbf{u}_j = 0$. Similarly we can impose the convex constraint $|\mathbf{u}_1 \pm \mathbf{u}_2 \pm \mathbf{u}_3|_2^2 \le 3$ as an relaxation of the orthogonal constraint, where $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ are all three rows/columns of $\bar{R}$.

The non-convex $SO(3)$ constraint is thus relaxed to a set of mixed-integer convex quadratic constraints. These mixed-integer convex constraints are readily handled by modern numerical solvers such as Gurobi [1].

3.2 Inverse kinematics formulation

We aim to solve the inverse kinematics problem by finding the pose of each link. The pose of link i can be represented by rigidly attaching a frame to the link, with the position ${}^W\mathbf{p}_i \in \mathbb{R}^3$ and the rotation matrix ${}^W R_i$, where the right subscript i denotes it is link i's frame we are interested in, the left superscript ^{W} indicates the quantity is expressed in the *world* frame. In Fig 5, we illustrate the kinematics relationship between two links connected by a revolute joint. We will express this kinematics relation as constraints on the position ${}^W\mathbf{p}_i$ and orientation ${}^W R_i$ of link i relative to and expressed in the world frame, and the position/orientation of its parent link $i-1$. A joint frame J is rigidly attached to the parent link $i-1$, and coincides with the child link frame, when the joint angle $q_i = 0$. We denote the fixed translation of joint frame to the parent link frame as ${}^{i-1}\mathbf{p}_J$. The joint axis is expressed as ${}^{i-1}\hat{\mathbf{z}}_i$ in the parent link $(i-1)$'s frame, and ${}^i\hat{\mathbf{z}}_i$ in the child link i's frame. Since the position and direction of joint axis is invariant under rotation about this axis, we can impose the following constraints on the poses of the parent and child links

$${}^W\mathbf{p}_i = {}^W\mathbf{p}_{i-1} + {}^W R_{i-1}{}^{i-1}\mathbf{p}_J \tag{9a}$$
$${}^W R_{i-1}{}^{i-1}\hat{\mathbf{z}}_i = {}^W R_i{}^i\hat{\mathbf{z}}_i \tag{9b}$$

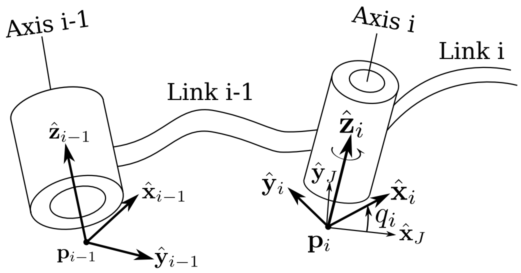


Fig. 5: The kinematics relation between adjacent link $i-1$ and i [11]. The frame with axes $\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i$ is rigidly attached to the child link i, the position of the frame origin is $\mathbf{p}_i$. Without loss of generality we can assume the rotation axis i is along the z axis $\hat{\mathbf{z}}_i$. Similarly for the parent link $i-1$. The joint frame J with axes $\hat{\mathbf{x}}_J, \hat{\mathbf{y}}_J, \hat{\mathbf{z}}_J$ is rigidly attached to the parent link $i-1$; this joint frame coincides with the child link frame when the joint angle $q_i = 0$.

(9) are linear constraints on the decision variables ${}^W R_i$, ${}^W R_{i-1}$, ${}^W\mathbf{p}_i$, ${}^W\mathbf{p}_{i-1}$.

We also handle the angle limits on the revolute joint. Without loss of generality, we assume that angle limits for joint i are $-\alpha \le q_i \le \alpha$. In order to enforce this joint limit constraint with only link poses as decision variables, we consider a unit length vector $\hat{\mathbf{x}}_J$, perpendicular to the joint axis $\hat{\mathbf{z}}_i$, and how it relates to itself before and after joint rotation. For example in Fig 5, rotating the i'th joint by q_i transforms the unit length vector $\hat{\mathbf{x}}_J$ to $\hat{\mathbf{x}}_i$. The joint limit constraint $|q_i| \le \alpha$ can be imposed as

$$|\angle({}^W\hat{\mathbf{x}}_J, {}^W\hat{\mathbf{x}}_i)| \le \alpha \Leftrightarrow {}^W\hat{\mathbf{x}}_J^{\mathrm{T}}\, {}^W\hat{\mathbf{x}}_i \ge \cos\alpha \Leftrightarrow \left|{}^W\hat{\mathbf{x}}_J - {}^W\hat{\mathbf{x}}_i\right|_2 \le 2\sin\frac{\alpha}{2}$$
$$\Leftrightarrow \left|{}^W R_{i-1}{}^{i-1}\hat{\mathbf{x}}_J - {}^W R_i{}^i\hat{\mathbf{x}}_i\right|_2 \le 2\sin\frac{\alpha}{2} \quad (10)$$

where $\angle(\cdot,\cdot)$ is the angle between two vectors. ${}^{i-1}\hat{\mathbf{x}}_J$ is the given unit length vector fixed and expressed in the parent $i-1$ frame, and ${}^i\hat{\mathbf{x}}_i$ is the given unit length vector fixed and expressed in the child i frame. Equation (10) is a convex (quadratic) constraint on the decision variables (parent and child link orientations). The joint limit constraint (10) would be tight if ${}^W R_i, {}^W R_{i-1}$ satisfied $SO(3)$ constraints exactly.

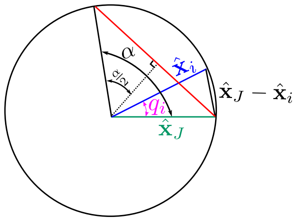


Fig. 6: The visual illustration of the joint limit constraint (10). The red chord is the difference between $\hat{\mathbf{x}}_J$ and another vector, obtained by rotating $\hat{\mathbf{x}}_J$ about an axis by angle α. The chord's length is $2\sin\alpha/2$. $\hat{\mathbf{x}}_J - \hat{\mathbf{x}}_i$ should have shorter length than the red chord, since the angle q_i between these two vectors is smaller than α.

To impose collision avoidance constraints, we use a similar formulation as [15]. We segment the free space into convex polytopes $\mathcal{P}_i, i = 1, \ldots, N_p$, with the vertices of $\mathcal{P}_i$ denoted as $\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,m_i}$. A point is collision free if it is in *one of* the polytopes. We impose the following mixed-integer convex constraints to enforce the point Q, on link k with coordinates ${}^k\mathbf{p}_Q$ in the link frame, to be collision free

$$ {}^{W}\mathbf{p}_k + {}^{W}R_k {}^{k}\mathbf{p}_Q = \sum_{i=1}^{N_p} \sum_{j=1}^{m_i} \lambda_{i,j} \mathbf{v}_{i,j} \tag{11a} $$

$$ z_i = \sum_{j=1}^{m_i} \lambda_{i,j}, \quad \sum_{i=1}^{N_p} z_i = 1 \tag{11b} $$

$$ \lambda_{i,j} \geq 0, \quad z_i \text{ is binary} \tag{11c} $$

z_i is the binary variable indicating within which polytope the link point Q stays.

With the mixed-integer convex relaxation on $SO(3)$ (sub-section 3.1), and the kinematic constraints in sub-section 3.2, we can solve the IK problem through mixed-integer convex (quadratic) optimization, to find the body positions and an approximated solution to the body orientations. If the mixed-integer convex program is infeasible, it proves that the original unrelaxed IK problem is *globally* infeasible.

3.3 Reconstruct joint angle by projecting back to $SO(3)$

After obtaining the approximated solution to the body orientations through the mixed-integer convex optimization, we need to recover the value of each joint angle. Since the solution to the body orientation does not satisfy the $SO(3)$ constraint exactly, we project the approximated solution to the rotation constraint, starting from the root link. If the link is floating, then the optimal projected solution is obtained as UV^{T}, where $\bar{R} = U\Sigma V^{\mathrm{T}}$ is the SVD of $\bar{R}$, as proved in [18]. On the other hand, if the link is connected to its parent link through a revolute joint, we project ${}^{W}R_{i-1}^{\mathrm{T}}\,{}^{W}\bar{R}_i$ onto $SO(3)$ with the given rotation axis and angle limits, where ${}^{W}R_{i-1}$ is the rotation matrix of the parent $i-1$ link, computed by doing forward kinematics using the recovered posture from the root to the $i-1$ link. To project ${}^{W}R_{i-1}^{\mathrm{T}}\,{}^{W}\bar{R}_i$, we find the joint angle q_i such that the joint rotation matrix $R({}^{i-1}\hat{\mathbf{z}}_i, q_i)$ has the minimal error to ${}^{W}R_{i-1}^{\mathrm{T}}\,{}^{W}\bar{R}_i$ under the joint limits (Fig 5). $R({}^{i-1}\hat{\mathbf{z}}_i, q_i) \in SO(3)$ means rotating by angle q_i about the axis ${}^{i-1}\hat{\mathbf{z}}_i$. Algebraically, the optimal q_i is the solution to the following program

$$ \min_{q_i} \left| R({}^{i-1}\hat{\mathbf{z}}_i, q_i) - {}^{W}R_{i-1}^{\mathrm{T}}\,{}^{W}\bar{R}_i \right|_F \tag{12a} $$

$$ \text{s.t } -\alpha \leq q_i \leq \alpha. \tag{12b} $$

In the objective function $|\cdot|_F$ is the matrix Frobenius norm, $|X|_F = \sqrt{\mathrm{trace}(X^{\mathrm{T}}X)}$. $R({}^{i-1}\hat{\mathbf{z}}_i, q_i)$ can be computed from the Rodriguez Formula below

$$ R({}^{i-1}\hat{\mathbf{z}}_i, q_i) = I_{3\times 3} + \sin q_i \lfloor {}^{i-1}\hat{\mathbf{z}}_i \rfloor_\times + (1-\cos q_i) \lfloor {}^{i-1}\hat{\mathbf{z}}_i \rfloor_\times^2, \tag{13} $$

where $\lfloor {}^{i-1}\hat{\mathbf{z}}_i \rfloor_\times \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix, representing the cross product with the vector ${}^{i-1}\hat{\mathbf{z}}_i$.

Substituting equation (13) into the optimization objective (12a), we can analytically obtain the optimal q_i as

$$\begin{cases} \text{if } \exists k \in \mathbb{Z}, -\alpha+\beta \leq \frac{\pi}{2}+2k\pi \leq \alpha+\beta, & q_i = \frac{\pi}{2}+2k\pi-\beta \\ \text{else:} & \\ \quad \text{if } \sin(-\alpha+\beta) \geq \sin(\alpha+\beta), & q_i = -\alpha \\ \quad \text{if } \sin(-\alpha+\beta) < \sin(\alpha+\beta), & q_i = \alpha. \end{cases} \tag{14}$$

where $\beta = \text{atan2}\left(-\text{trace}(M^{\text{T}}A^2), \text{trace}(A^{\text{T}}M)\right), A = \lfloor {}^{i-1}\hat{\mathbf{z}}_i \rfloor_\times, M = {}^W R_{i-1}^{\text{T}}\, {}^W \bar{R}_i$. Note that at value $\pi/2+2k\pi-\beta$ the derivative of the objective (12a) is zero, i.e., it is the optimal solution to (12a) without the joint limits (12b). The optimal solution is truncated to the joint bounds if $\pi/2+2k\pi-\beta$ is outside of the angle limits. q_i in (14) is the optimal joint angle within the joint limits, that minimizes the projection error from the mixed-integer convex optimization solution, to $SO(3)$ with a given rotation axis and angle bounds. We thus obtain a posture as the approximated solution to the inverse kinematics problem.

4 Results

We present the results on the $SO(3)$ relaxation and the inverse kinematics algorithm. The programs run on 64 bit Intel Xeon CPUs with Gurobi 7.5 [1] as the solver.

4.1 Tightness of rotation matrix relaxation

We first show the tightness of our mixed-integer convex relaxation of the $SO(3)$ constraint on the rotation matrix. If $\bar{R} \in \mathbb{R}^{3\times 3}$ satisfies the relaxed constraints, then we know that $|\bar{R}\mathbf{v}|_2 \approx |\mathbf{v}|_2 \; \forall \mathbf{v} \in \mathbb{R}^3$, and $\angle(\bar{R}\mathbf{v}_1, \bar{R}\mathbf{v}_2) \approx \angle(\mathbf{v}_1, \mathbf{v}_2)$. Geometrically the relaxation allows the transformation $\mathbf{v} \rightarrow \bar{R}\mathbf{v}$ to change the length of a vector, and to perturb the angle between two vectors. In this section we show how much shorter the transformed vector can be, and also derive a bound on the angle error.

We find the global optimum to two mixed-integer convex optimization programs, to determine the tightness of the relaxation. First we compute the global minimum of $\min_{\bar{R},i} |\bar{R}\mathbf{e}_i|_2$, where $\mathbf{e}_i$ is the unit length vector with $\mathbf{e}_i(i) = 1$. To determine the angle error, we compute the global minimum $\min_{\bar{R} \text{ in relaxation}, i\neq j} |\bar{R}\mathbf{e}_i + \bar{R}\mathbf{e}_j|_2$. If we denote this global minimum as d, we obtain loose bounds on the angle after transformation as $180^\circ - \arccos(d^2/2-1) \leq \angle(\bar{R}\mathbf{e}_i, \bar{R}\mathbf{e}_j) \leq \arccos(d^2/2-1)$. We compute the error bounds on the relaxation, by partitioning the range $[-1,1]$ into $N = 2, 4$ or 6 intervals, with $\phi_i = -1+2i/N$. The results are in Table 1. The last column with infinite number of binary variables is the ideal $\bar{R} \in SO(3)$ case without any relaxation. The bounds on $|\bar{R}\mathbf{e}_i|_2$ and $|\bar{R}\mathbf{e}_i + \bar{R}\mathbf{e}_j|_2$ are tight, but the bound on the angle in the last row of the table is not tight.

# of intervals	2	4	6	∞
$\min_{\bar{R},i} \lvert\bar{R}\mathbf{e}_i\rvert_2$	0.57735	0.90453	0.95784	1
$\min_{\bar{R},i,j} \lvert\bar{R}\mathbf{e}_i + \bar{R}\mathbf{e}_j\rvert_2$	0.60302	1.22474	1.32667	1.414
$\angle(\bar{R}\mathbf{e}_i, \bar{R}\mathbf{e}_j) - 90^\circ$	$[-54.90^\circ, 54.90^\circ]$	$[-14.47^\circ, 14.47^\circ]$	$[-6.89^\circ, 6.89^\circ]$	$[0^\circ, 0^\circ]$

Table 1: The tightness on the mixed-integer convex relaxation on $SO(3)$.

As the number of intervals increases, the convex hull of xy in each interval is closer to the bilinear product. As a result in Table 1, the minimal length of $\bar{R}\mathbf{e}_i$ increases, and the bound on the angle error shrinks, leading to a tighter relaxation of $SO(3)$. In the inverse kinematics problem, we choose to cut $[-1,1]$ into 4 intervals, as a compromise between computation speed and relaxation accuracy.

4.2 Inverse Kinematics

We first show that our inverse kinematics approach can find postures for complicated kinematics constraints including inequalities, such as those arising from the task “grasping a mug on the table in the cluttered environment” in Fig 7. We show two distinct postures found by Gurobi on the mixed-integer convex optimization programs. These two postures have different sets of active binary variables. We use a KUKA IIWA arm and a Schunk gripper with 7 total joints. The kinematics constraints are

- The middle point between the two fingers should be on the axis of the mug. The middle point should be above the table, and lower than the top rim of the mug.
- The y axis on the hand should be horizontal.
- The links and the gripper should be collision free.

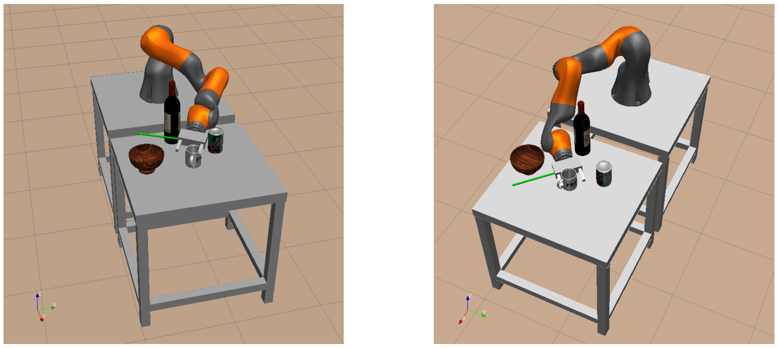


Fig. 7: KUKA IIWA arm grasping a mug on the table with two different postures. The green line is the y axis of the hand which should stay horizontal. The mesh files of the objects are obtained from Shapenet [10].

We also show that our approach can prove the global infeasibility of some kinematics constraints. In Fig 8, we impose the task constraint that the hand should grasp the wine bottle, by keeping its center on the green line segment, and its orientation aligned with the bottle’s longitudinal axis. In the left plot the mixed-integer program is solved successfully, while in the right plot with the fridge on the table, the additional collision avoidance constraint causes the mixed-integer convex program to be globally infeasible. The solver takes 0.75 seconds to detect the global infeasibility.

For more complicated robots like LittleDog (Fig 9), we show that our approach can find a posture of the robot with each leg on one of the stepping stones. This constraint is imposed similar to the collision avoidance constraint (11a)-(11c), but

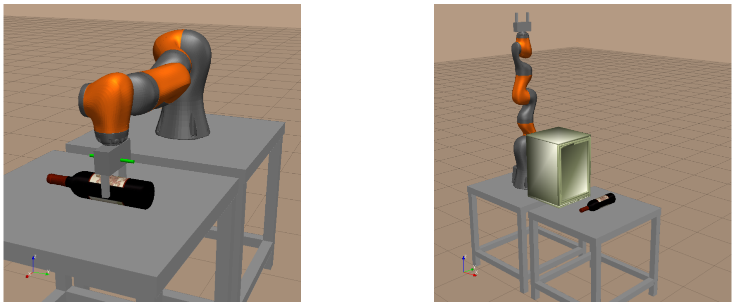
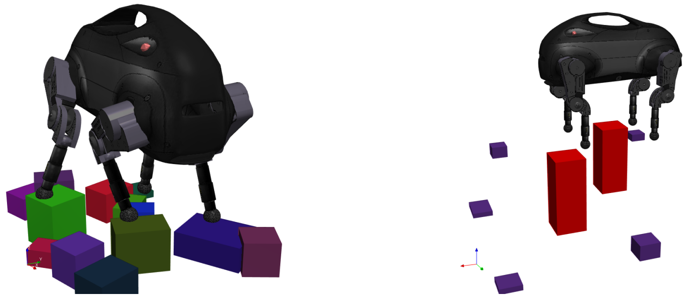


Fig. 8: KUKA IIWA arm grasping the wine bottle. On the left, the IK problem is solved successfully. The center of the hand is constrained to be on the green line segment. On the right, with the fridge on the table, the mixed-integer convex program proves there is no posture that can satisfy the same grasping constraint, while keeping the robot from colliding with the fridge.

replacing the vertices in each polytope with the top corners of each stepping stone. In Fig 9b the mixed-integer convex IK spends 0.25s to determine that there does not exist a posture for the robot to put each of its toes on one of the violet stepping stones, while avoiding the red obstacles.



(a) LittleDog stands on stepping stones. Computation takes 15s for this 18 DoF quadruped.

(b) Mixed-Integer Convex IK proves it infeasible to put each foot on one of the stepping stone (violet), while avoiding the red obstacles.

Fig. 9

To get some statistics on our mixed-integer convex IK approach, we test its performance on an ABB IRB140 arm with 6 DoF, for which the IK problem can be solved analytically [11]. In Fig 10, we show the results of running both analytical and mixed-integer convex IK on the robot. In each column of the figures, an IK problem is solved to determine whether the end effector will reach a sampled location with the given orientation. There are three categories of the solutions

- Green dot. Both analytical and mixed-integer convex IK find the solution.
- Blue dot. Both analytical and mixed-integer convex IK prove global infeasiblity.
- Red dot. Analytical IK proves the problem is infeasible, but mixed-integer convex IK thinks the problem is feasible, due to the relaxation.

We first analyze the tightness of the relaxation. The red dots, which imply loose relaxation on the $SO(3)$, form a thin layer between the unreachable blue dots, and the reachable green dots. Only 2.43% of all the samples are red. For all the kinematically unreachable points (red and blue), 94.72% of them are detected as *global infeasible* in our mixed-integer IK solver; on the other hand, SDP relaxation of the $SO(3)$ constraint, proposed in [27, 12], can detect only 42.23% of infeasibility. This huge difference demonstrates that our mixed-integer convex relaxation of $SO(3)$ constraint is a lot tighter than the SDP relaxation.

We also examine the computation time of the mixed-integer convex IK, together with the quality of the approximated solution. In Fig 11 we show the histogram of the mixed-integer convex IK computation time for the green dots (both IK approaches can find the solution). 95.1% of samples are solved within 10s, and the average computation time is 4.2s. In Fig 12 after reconstructing the posture from the mixed-integer convex IK, we compute the end effector pose error from the reconstructed posture to the sampled pose, to demonstrate how much relaxation error is brought into the approximated IK solution. Most of the solutions have less than 5cm of position error, and less than 2° of orientation error. The solutions with large pose error all have some joints with active joint limits. This large error occurs because the joint limit constraint (10) would be tight only if the link orientation matrices ${}^WR_{i-1}, {}^WR_i$ were on $SO(3)$. With our mixed-integer convex relaxation on $SO(3)$, the angle $\pi/2+2k\pi-\beta$ in (14), where the derivative of the objective function (12a) would be zero, falls outside of the joint limits. So the reconstructed joint angle is instead truncated to its angle limits $\pm\alpha$. This truncation can cause large cost in the projection error (12a), resulting in significant deviation in the end effector pose.

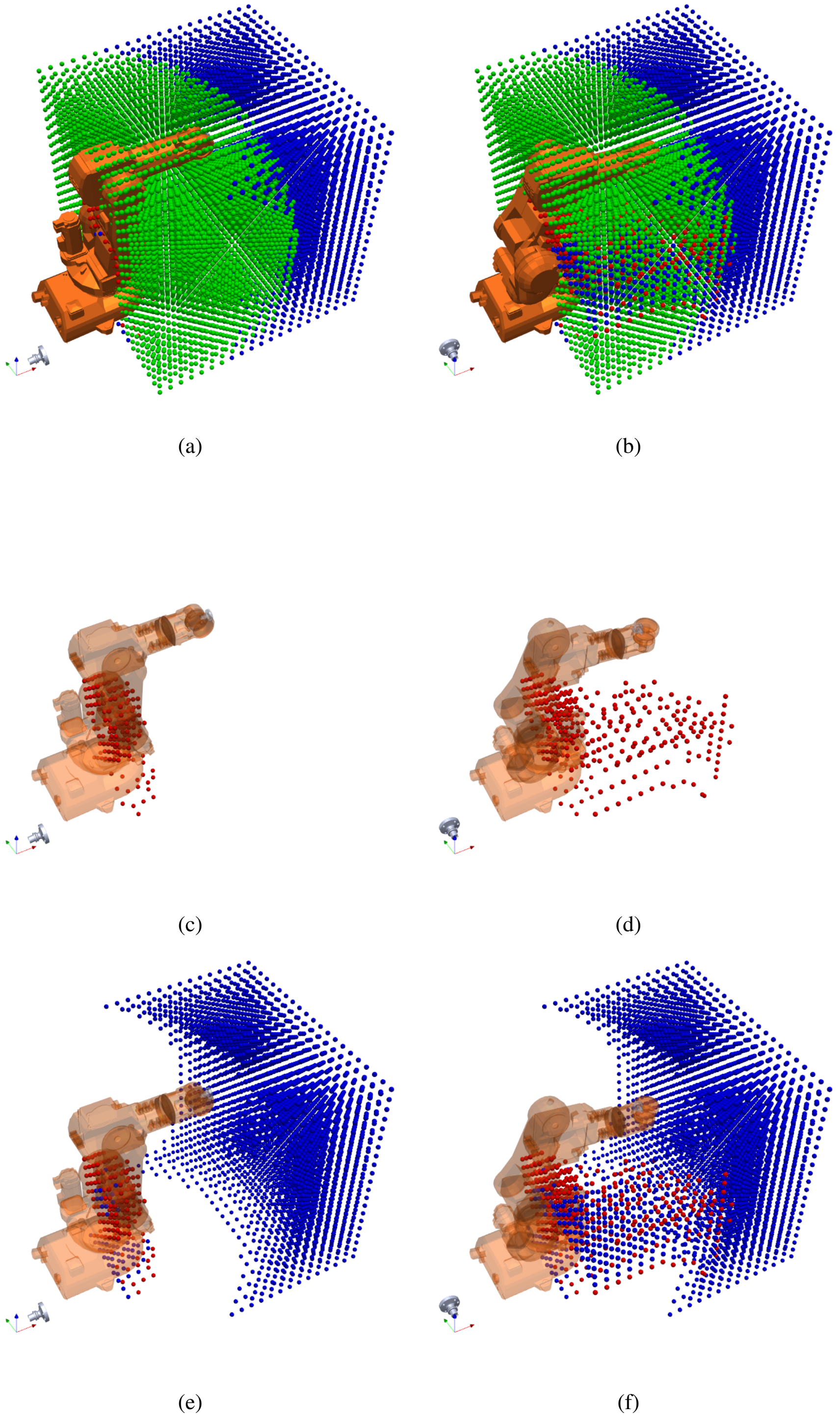
In Fig 13a, 13b we draw the histograms of the computation time, when the mixed-integer convex IK proves global infeasibility. In 86.05% cases the computation takes less than 0.1s.

Finally, we solve the inverse kinematics program for the samples in Fig 10a, 10b through nonlinear optimization [17]. We first set the initial seed as the postures in 10a and 10b, and then solve the nonlinear optimization again, but replace the initial seed with the reconstructed posture from the mixed-integer convex IK. The success rate of the nonlinear IK solver increases from 85.67% to 100%. This improvement demonstrates that the approximate solution from the mixed-integer convex IK can *always* serve as a good initial seed for the nonlinear optimization program.

5 Conclusion and Discussion

In this paper we propose a mixed-integer convex relaxation of the non-convex $SO(3)$ constraint, and formulate a mixed-integer convex optimization program to solve the inverse kinematics problem globally with generic constraints. We show that this relaxation is relatively tight, and our IK approach can either produce an approximate solution, or prove that the solution does not exist globally. We demonstrate results when applying our approach on manipulators and quadruped robots.

Currently the computation time is a lot slower than the gradient-based nonlinear optimization approach, and the reconstructed posture can violate the kinematic con-



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 10: The results of running both analytical and mixed-integer convex IK on an ABB IRB140 arm. We take 9321 sample points in an 1 m^3 cube. The robot end effector is required to reach each sampled location with two different given orientations. The desired orientation of the end effector is shown in silver color next to the coordinate axes in the bottom left of the figure. The green dots correspond to positions for which both IK approaches obtain the solution. The blue dots correspond to positions for which both IK approaches guarantee global infeasibility. The red dots correspond to the gap for which the analytical IK proves that the problem is infeasible, while the mixed-integer convex IK thinks the problem is feasible under relaxation. In the first row of the figures we show all three colors of dots. In the last two rows we highlight each color of the dots separately.

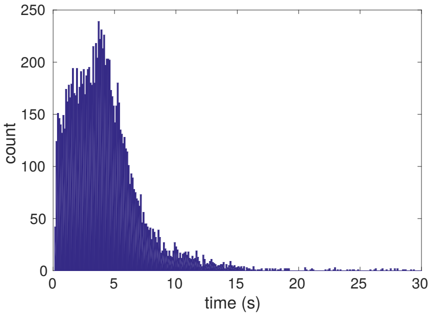


Fig. 11: Histogram on mixed-integer IK computation time, for all the green dots in Fig 10a, 10b

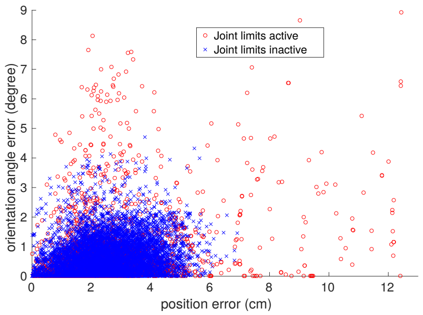
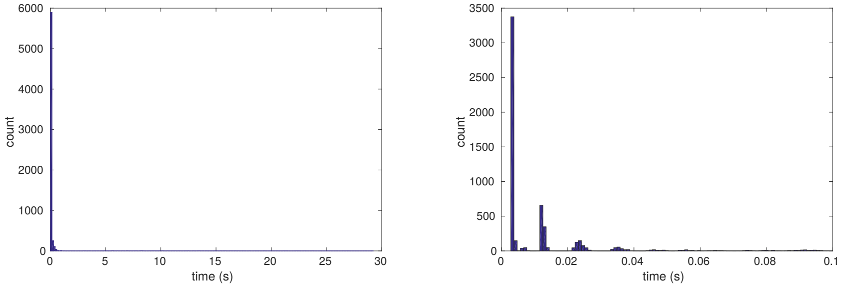


Fig. 12: The end effector pose error for the reconstructed posture from mixed-integer convex IK solver, for all the green dots in Fig 10a, 10b



(a) Histogram on mixed-integer IK computation time, when it proves global infeasibility.

(b) A zoomed in view of the histogram in Fig 13a for samples with computation time $\leq 0.1s$.

Fig. 13

straints with large errors. We are working on improving the accuracy of the solution, and reducing the computation time.

We would like to thank our colleagues at Toyota Research Institute and Robot Locomotion Group for the valuable discussions.

References

1. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, 2:1–3, 2012.
2. D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25. SIAM, 2013.
3. E. M. L. Beale and J. A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *OR*, 69(447-454):99, 1970.
4. P. Beeson and B. Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 928–935. IEEE, 2015.
5. D. Berenson, S. Srinivasa, and J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1460, 2011.

6. D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
7. D. Bertsimas and R. Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.
8. S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
9. S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.
10. A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
11. J. J. Craig. *Introduction to robotics: mechanics and control*, volume 3. 2005.
12. H. Dai, A. Majumdar, and R. Tedrake. Synthesis and optimization of force closure grasps via sequential semidefinite programming. In *International Symposium on Robotics Research*, 2015.
13. H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 295–302. IEEE, 2014.
14. R. Deits and R. Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014.
15. R. Deits and R. Tedrake. Efficient mixed-integer planning for uavs in cluttered environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015.
16. R. Diankov. Automated construction of robotic manipulation programs. 2010.
17. M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, et al. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 32(2):229–254, 2015.
18. R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446, 1989.
19. J. Huchette and J. P. Vielma. Small independent branching formulations for unions of v-polyhedra. *arXiv preprint arXiv:1607.04803*, 2016.
20. D. Manocha and J. F. Canny. Real time inverse kinematics for general 6r manipulators. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 383–389. IEEE, 1992.
21. M. T. Mason. *Mechanics of robotic manipulation*. MIT press, 2001.
22. D. Mellinger, A. Kushleyev, and V. Kumar. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 477–483. IEEE, 2012.
23. R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (miqcqp) through piecewise-linear and edge-concave relaxations. *Mathematical Programming*, pages 1–28, 2012.
24. R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
25. D. L. Peiper. The kinematics of manipulators under computer control. Technical report, 1968.
26. M. Raghavan and B. Roth. Kinematic analysis of the 6r manipulator of general geometry. In *International symposium on robotics research*, pages 314–320, 1990.
27. J. Saunderson, P. A. Parrilo, and A. S. Willsky. Semidefinite descriptions of the convex hull of rotation matrices. *SIAM Journal on Optimization*, 25(3):1314–1343, 2015.
28. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
29. R. Tedrake and the Drake Development Team. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2016.
30. A. K. Valenzuela. *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.
31. J. P. Vielma and G. L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1):49–72, 2011.