Supplementary Material for "Convex Iteration for Distance Geometric Inverse Kinematics"

Matthew Giamou, Filip Marić, David M. Rosen, Valentin Peretroukhin,
Nicholas Roy, Ivan Petrović, and Jonathan Kelly

***Abstract*— Supplementary material for the 2021 IEEE Robotics and Automation Letters submission entitled "Convex Iteration for Distance Geometric Inverse Kinematics".**

I. Mathematics

In this section, we describe some of the mathematics behind our formulation of the SDP `CIDGIK` solves in greater detail than the main paper.

A. QCQP Formulation

Throughout this section we slightly abuse our notation by using e to refer both to a directed edge $e = (i, j) \in \mathcal{E}_{\text{eq}}$ as well as an integer ($e \in [|\mathcal{E}_{\text{eq}}|]$) corresponding to a fixed index of this same edge. The incidence matrix is

$$\mathbf{B}(\mathcal{E}_{\text{eq}})_{i,e} = \begin{cases} 1 \text{ if } e \in \delta(i)^+, \\ -1 \text{ if } e \in \delta(i)^-, \\ 0 \text{ otherwise,} \end{cases} \tag{1}$$

where $\delta(i)^-$ and $\delta(i)^+$ are the set of edges leaving and entering $i \in \mathcal{V}$, respectively. Thus, the columns

$$\mathbf{P}\mathbf{B}(\mathcal{E}_{\text{eq}})^{(e)} = \mathbf{P}^{(j)} - \mathbf{P}^{(i)} \tag{2}$$

each refer to relative position of vertices i and j. The diagonal elements of the product

$$\mathbf{B}(\mathcal{E}_{\text{eq}})^\mathsf{T}\mathbf{P}^\mathsf{T}\mathbf{P}\mathbf{B}(\mathcal{E}_{\text{eq}}) \in \mathbb{R}^{|\mathcal{E}_{\text{eq}}| \times |\mathcal{E}_{\text{eq}}|} \tag{3}$$

are therefore equal to

$$\ell(e) = \left\| \mathbf{P}^{(j)} - \mathbf{P}^{(i)} \right\|^2, \quad e = (i, j) \in \mathcal{E}_{\text{eq}}. \tag{4}$$

Recalling that $\mathbf{d}_e = \ell(e)^2$ leads us to the compact expression for our equality constraints found in the main paper:

$$\operatorname{diag}\left(\mathbf{B}(\mathcal{E}_{\text{eq}})^\mathsf{T}\mathbf{P}^\mathsf{T}\mathbf{P}\mathbf{B}(\mathcal{E}_{\text{eq}})\right) = \mathbf{d}. \tag{5}$$

B. SDP Constraints

After deriving the feasibility QCQP formulation of IK, we apply an SDP relaxation. The lifted matrix variable

$$\mathbf{Z}(\mathbf{X}) = [\mathbf{X} \ \mathbf{I}_d]^\mathsf{T}[\mathbf{X} \ \mathbf{I}_d] = \begin{bmatrix} \mathbf{X}^\mathsf{T}\mathbf{X} & \mathbf{X}^\mathsf{T} \\ \mathbf{X} & \mathbf{I}_d \end{bmatrix} \in \mathbb{S}^{2n+d}_+ \tag{6}$$

allows us to write a number of degree-2 or lower polynomial expressions in $\mathbf{X}$ as linear expressions of $\mathbf{Z}$. Since $\mathbf{Z}_{i,j} = \mathbf{x}_i^\mathsf{T}\mathbf{x}_j \ \forall i, j \leq 2n$, each equation in 5 that is between two variables $\mathbf{x}_k, \mathbf{x}_l$, $k \neq l$ can be written $\operatorname{tr}(\mathbf{A}\mathbf{Z}) = \ell((k, l))$, where

$$\mathbf{A}_{j,i} = \mathbf{A}_{i,j} = \begin{cases} 1 \text{ if } i = j \in \{k, l\}, \\ -1 \text{ if } i = k, \ j = l, \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

Similarly, the expression for the squared distance between a variable $\mathbf{x}_k$ and some anchor $\mathbf{w}$ (or obstacle centre $\mathbf{c}$) can be encoded with matrix $\mathbf{M} \in \mathbb{S}^{2n+d}$ where

$$\mathbf{M}_{k,k} = 1, \tag{8}$$

$$\mathbf{M}_{2n:2n+d,k} = -\mathbf{w}, \tag{9}$$

$$\mathbf{M}_{k,2n:2n+d} = -\mathbf{w}^\mathsf{T}, \tag{10}$$

$$\mathbf{M}_{2n+d,2n+d} = \|\mathbf{w}\|^2. \tag{11}$$

Likewise, linear constraints can be enforced with matrices manipulating the elements in $\mathbf{Z}_{2n:2n+d,1:2n}$ and its symmetric counterpart. These constraints can be collected in the linear maps

$$\mathcal{A}(\mathbf{Z}) = \mathbf{b}, \ \ \mathcal{A} : \mathbb{S}^{2n+d} \to \mathbb{R}^{m+d^2+|\mathcal{V}_{\text{p}}|}, \tag{12}$$

$$\mathcal{B}(\mathbf{Z}) \leq \mathbf{c}, \ \ \mathcal{B} : \mathbb{S}^{2n+d} \to \mathbb{R}^{|\mathcal{O}|}, \tag{13}$$

which appear in our SDP problem formulation. The final constraint worth noting, which contributes d^2 to the dimension of the codomain of $\mathcal{A}$, simply arises from the requirement that the bottom-right $d \times d$ diagonal of $\mathbf{Z}$ is equal to $\mathbf{I}_d$.

C. Geometric Interpretation

This section provides details on the toy SDP formulation in Section IV-C of the main paper. Recall the QCQP formulation for the IK problem of reaching a point $\mathbf{w} \in \mathbb{R}^2$ with the end-effector of a simple planar 2-DOF manipulator:

$$\begin{aligned} \text{find} \quad & \mathbf{x} \in \mathbb{R}^2 \\ \textit{s.t.} \quad & \|\mathbf{x}\|^2 = 1, \\ & \|\mathbf{x} - \mathbf{w}\|^2 = 1, \\ & \|\mathbf{x} - \mathbf{o}\|^2 \geq 0.25, \end{aligned} \tag{14}$$

where $\mathbf{o} = [1, \ 0]^\mathsf{T}$ is the position of a unit-diameter circular obstacle. For $\mathbf{w} = [1, \ 1]^\mathsf{T}$, the insets of Figure 3 in the main paper show that of the two candidate solutions to this problem, the "elbow down" configuration in the bottom right collides with the obstacle at $\mathbf{o}$ (partially depicted as a blue semicircle). Homogenizing (14) with $s^2 = 1$ and lifting to the rank-1 matrix variable

$$\mathbf{Z}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix} \begin{bmatrix} \mathbf{x}^\mathsf{T} & s \end{bmatrix} \tag{15}$$

lets us apply the SDP relaxation $\mathbf{Z} \succeq \mathbf{0}$ to yield:

$$\begin{aligned} \text{find} \quad & \mathbf{Z} \in \mathbb{S}^3_+ \\ \textit{s.t.} \quad & \text{tr}\,(\mathbf{A}_0\mathbf{Z}) = 1, \\ & \text{tr}\,(\mathbf{A}_1\mathbf{Z}) = 1, \\ & \text{tr}\,(\mathbf{A}_2\mathbf{Z}) = 1, \\ & \text{tr}\,(\mathbf{A}_3\mathbf{Z}) \geq 0.25, \end{aligned} \tag{16}$$

where

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{17}$$

describe the unit link length constraints and

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{18}$$

describe the homogenization equation ($s^2 = 1$) and the obstacle avoidance constraint, respectively.

D. Link and Self-Collision Avoidance

In order to implement collision avoidance on parts of the robot that are not represented by the points in $\mathbf{X}$, we can introduce any number of auxiliary variables $\mathbf{y} \in \mathbb{R}^d$ that are fixed between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ for some $(i, j) \in \mathcal{E}_{\text{eq}}$. We can parameterize the interior of the line segment connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\alpha \in (0, 1)$ and constrain $\mathbf{y}$ to lie at some point of our choosing on this line segment:

$$\mathbf{y} = (1 - \alpha)\mathbf{x}_i + \alpha\mathbf{x}_j. \tag{19}$$

This auxiliary point can be used in collision avoidance constraints between $\mathbf{y}$ and obstacles:

$$\|\mathbf{y} - \mathbf{c}_k\|^2 \geq l_k^2 \;\; \forall k \in \mathcal{O}. \tag{20}$$

Additionally, `CIDGIK` can easily incorporate self-collision constraints with distance inequalities between variables:

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \geq \epsilon_{i,j} \;\; \forall (i, j) \in \mathcal{E}_{\text{eq}}, \tag{21}$$

where $\epsilon_{i,j}$ is any user-defined threshold, ideally based on robot geometry. Equations of this form can also replace $\mathbf{x}_i$ or $\mathbf{x}_j$ with auxiliary variables defined in Equation (19). Since the constraints defined in this section are either linear or distance-geometric, they are supported by the "rank-d" semidefinite relaxation of Equation (6).

II. Additional Experiments

This section summarizes two small experiments not covered in the main paper.

A. Infeasibility Certification

Convex relaxations can often be used certify the infeasibility of a problem. To determine whether `CIDGIK` is capable of infeasibility certification, we conducted a simple experiment. Using the *Cube* environment and a closed-form analytic solver for the UR10, we gave `CIDGIK` 10,000 goal poses with the following properties:

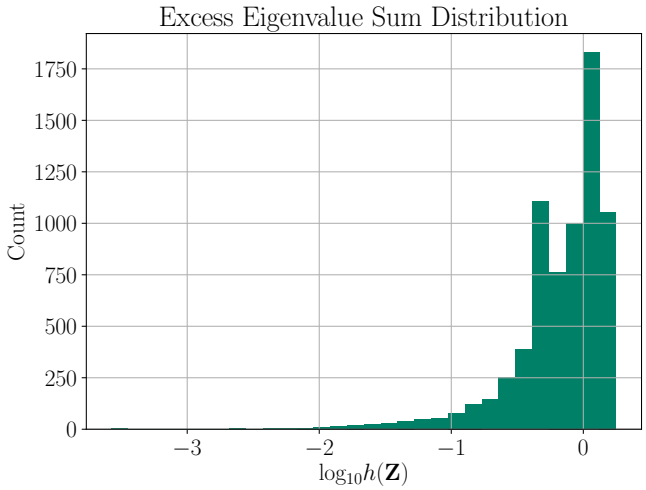

Fig. 1: Distribution of $\log_{10}(h(\mathbf{Z})$ over 7,000 runs of our SDP formulation with the nuclear norm heuristic ($\mathbf{C} = \mathbf{I}$).

1) the analytic UR10 solver deemed the goal pose *infeasible* by checking for collisions over all 8 possible solutions,
2) the end-effector was not in collision with an obstacle, and
3) the point attached to the end-effector's parent joint was not in collision with an obstacle.

Goal poses that are infeasible because of conditions 2) and 3) are easy for `CIDGIK` to certify as infeasible because the Euclidean distance constraints they violate are unavoidable, even in higher-rank solutions representing configurations embedded in $\mathbb{R}^{d'}$ where $d' > d$. Unfortunately, the infeasible cases that remain are much more challenging, with `CIDGIK` certifying a mere 152 out 10,000 (1.52%) as infeasible.

B. Nuclear Norm Heuristic

The first iteration of `CIDGIK` uses cost matrix $\mathbf{C} = \mathbf{I}$, which is equivalent to the popular nuclear norm heuristic. The nuclear norm is an effective surrogate for low-rank matrix recovery in other problems, but in this section we demonstrate that it is inappropriate for IK. Figure 1 shows the distribution of $\log_{10}(h(\mathbf{Z}))$ over 7,000 experiments with feasible UR10 goal poses in an obstacle-free environment, where

$$h(\mathbf{Z}) = \sum_{i=d+1}^{2n+d} \lambda_i(\mathbf{Z}) \tag{22}$$

is our "excess rank" heuristic. The convergence threshold for $h(\mathbf{Z})$ we used for our experiments with `CIDGIK` in the main paper was 10^{-6}, with some high quality solutions recovered for $h(\mathbf{Z}) \approx 10^{-3}$. As you can see in Figure 1, the nuclear norm heuristic led to very few instances with $h(\mathbf{Z}) < 10^{-2}$ and is therefore only appropriate as an initialization for `CIDGIK`, which typically required fewer than 10 iterations to converge below our threshold of 10^{-6}.

III. Limitations and Future Work

This section describes a number of promising research directions that can address some of the shortcomings and

limitations of CIDGIK.

A. Rank-1 Relaxation and Ellipsoids

Like SNL, our IK problem leads to a rank-d SDP relaxation because of the d-dimensional nature of our spatial problem. More concretely, we saw in the previous section that since we are only constraining the squared *distances* between points in $\mathbb{R}^d$, quadratic terms in these constraints appear as inner products between variables (e.g., $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$). In general, QCQPs do not contain this structure, and the lifted $\mathbf{Z}$ variable would be a "rank-1" lifting. For our problem, a rank-1 SDP relaxation, like the one used for the *pierogi* example would take the form

$$\mathbf{Z} = [\mathrm{vec}(\mathbf{X})^\mathsf{T}\ 1]^\mathsf{T}[\mathrm{vec}(\mathbf{X})^\mathsf{T}\ 1] \in \mathbb{S}_+^{2dn+1}, \qquad (23)$$

where $\mathrm{vec}(\cdot) : \mathbb{R}^{a\times b} \to \mathbb{R}^{ab}$ is a column-wise vectorization of its matrix argument (i.e., it "vertically" stacks b columns). As you can see, this $\mathbf{Z}$ variable is much larger, leading to greater computation and memory requirements in SDP solvers, and our rank-d relaxation, by contrast, exploits the spatial structure of our problem. However, since the generic rank-1 relaxation allows us to work with any quadratic constraint, in future work we are interested in exploring its ability to model constraints like arbitrary ellipsoidal obstacles:

$$(\mathbf{x}_i - \mathbf{c})^\mathsf{T}\mathbf{A}(\mathbf{x}_i - \mathbf{c}) \geq l^2. \qquad (24)$$

B. Chordal Sparsity

A graph is *chordal* if every cycle of length four or greater has a chord (an edge that is not part of the cycle but connects two vertices in the cycle). In our problem, the edges of $\mathcal{G}$ describe points whose distances are constrained. Since the SDP variable $\mathbf{Z}$ contains the dot product of point variables (i.e., $Z_{ij} = \mathbf{x}_i^\mathsf{T}\mathbf{x}_j$), $\mathcal{G}$ also describes the elements of $\mathbf{Z}$ that appear together in constraints. In this section, we demonstrate that $\mathcal{G}$ is chordal due to a sparsity pattern induced by the fact that distance constraints only affect points fixed to neighbouring joints. This "chordal sparsity" can be exploited to speed up the solution of large SDPs with the methods reviewed in [1]. In future work, we intend to use these methods within CIDGIK to quickly solve IK problems for tree-like robots with many redundant degrees of freedom.

Assume that $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathrm{eq}}, \ell)$ is the distance constraint graph for a tree-like robot (i.e., no loops in its joints). Since the robot's joints and links describe a tree, the only cycles in $\mathcal{G}$ occur in the cliques described by $\mathbf{p}_i$, $\mathbf{q}_i$, $\mathbf{p}_j$, and $\mathbf{q}_j$ for neighbouring joints i and j. Since the cycles occur within cliques, they are chordal (because cliques are complete by definition). In order to apply the methods in [1], we would additionally need to ensure that the workspace constraints and cost function matrix $\mathbf{C}$ also exhibit this sparsity pattern. Spherical and planar obstacle constraints clearly do not ruin the chordal sparsity, since each constraint only involves a single point $\mathbf{p}_i$. Finally, the work on optimal power flow in [2] describes a heuristic method for constructing $\mathbf{C}$ in a manner that preserves problem sparsity. The COSMO algorithm [3] has a freely available implementation that is able to automatically exploit chordal sparsity in conic optimization problems.

C. Joint Angle Limits

Quadratic joint angle limits for revolute manipulators can be defined in a manner similar to the method used in [4] on planar and spherical chains. Unfortunately, without the addition of auxiliary variables, joint angle ranges which are not symmetric about the "straightened" configuration are impossible to enforce. Crucially, this precludes the use of joints that behave like fingers or knees (i.e., only able to bend in one direction away from the straightened configuration) in IK problems we wish to solve with CIDGIK. This need for "directionality" in angular constraints can be addressed with the addition of constraints involving the cross product of differences between adjacent points, which we leave for future work.

D. "Tetrahedral" Robot Link Geometry

The work in this paper assumes that consecutive joints have axes of rotation that are coplanar (i.e., either parallel or intersecting). This requirement ensures that for consecutive joints i and j, the points $\mathbf{p}_i, \mathbf{q_i}, \mathbf{p}_j$, and $\mathbf{q}_j$ form a quadrilateral or triangle. When the axes are *not* coplanar, these points form a tetrahedron that we are specifying from distances alone. Generic tetrahedra are chiral in that their reflections cannot be reproduced by rigid transformations. Therefore, our method of enforcing a tetrahedral joint pair's structure with distances alone allows CIDGIK to search a feasible set that contains physically unrealizable configurations. Much like asymmetrical joint limits, this problem can be addressed with the addition of auxiliary variables and constraints involving cross-products, but the effect of these additions on the accuracy and performance of CIDGIK remains to be seen.

REFERENCES

[1] A. Majumdar, G. Hall, and A. A. Ahmadi, "A Survey of Recent Scalability Improvements for Semidefinite Programming with Applications in Machine Learning, Control, and Robotics," *arXiv:1908.05209 [cs, eess, math]*, Sept. 2019.

[2] W. Wang and N. Yu, "Chordal Conversion Based Convex Iteration Algorithm for Three-Phase Optimal Power Flow Problems," *IEEE Trans. on Power Systems*, vol. 33, no. 2, pp. 1603–1613, Mar. 2018.

[3] M. Garstka, M. Cannon, and P. Goulart, "COSMO: A conic operator splitting method for large convex problems," in *2019 18th European Control Conference (ECC)*, Naples, Italy, June 2019, pp. 1951–1956.

[4] F. Marić, M. Giamou, S. Khoubyarian, I. Petrović, and J. Kelly, "Inverse kinematics for serial kinematic chains via sum of squares optimization," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, Aug. 2020, pp. 7101–7107.