2019 Third IEEE International Conference on Robotic Computing (IRC)

Convex Multi-Criteria Design Optimization of Robotic Manipulators via Sum-of-Squares Programming

Wankun Sirichotiyakul[1], Volkan Patoglu[2], and Aykut C. Satici[1]

***Abstract*— This paper presents a general framework for optimization of robotic manipulators via sums-of-squares (SoS) programming (semidefinite convex optimization) with multiple design objectives. Both kinematic and dynamic performance measures are discussed and an optimization problem for a proof-of-concept robotic manipulator has been formulated. SoS programming is shown to promise advantages as it can provide globally optimal results up to machine precision and scales much better with respect to the number of design variables than other methods which can obtain globally optimal solutions.**

I. Introduction

Design of robotic manipulators is often subjected to multi-criteria performance requirements, which are typically dominated by their geometrical parameters. Formulating and solving an optimization problem for these parameters are typically challenging due to the highly nonlinear, nonconvex nature of the kinematics and dynamics of most robotic manipulators and the considered performance metrics. Common undesirable characteristics that plague nonlinear and nonconvex optimization methods are the tendency to get stuck at a local optimum and the high computational demands.

The main contribution of this paper is the reformulation of the problem of design optimization of robotic manipulators as a convex optimization problem by invoking sum-of-squares (SoS) techniques [1]. This allows for global optimization of many performance indices efficiently up to machine precision. Moreover, SoS optimization scales exceptionally well as the number of design parameters increase, contrary to branch and bound methods whose computational complexity increases exponentially. The interested reader can refer to [2], [3] for more information about SoS optimization.

We consider the multi-criteria optimization of the global kinematic and dynamic isotropy indices, *GII* and *GDI*, respectively [4]. The Pareto-front curve is obtained by using the scalarization (weighted-sum) method to turn the multi-objective optimization problem into series of single-objective ones. We use SoS programming to solve each individual single-objective problem. A proof-of-concept case-study that formulates and solves the design optimization problem for the planar two-link manipulator is presented.

II. Kinematic and Dynamic Performance Indices

We use a performance index called the global isotropy index (*GII*), introduced in [4], to quantify the kinematic isotropy of robotic manipulators over the whole workspace. A manipulator with maximal *GII* corresponds to a design with best worst-case kinematic performance, increasing the efficiency of actuator utilization.

We choose to optimize the global dynamic index (*GDI*), also introduced in [4], to quantify dynamical performance. It measures the largest effect of mass on the dynamic performance. A manipulator with optimal *GDI* corresponds to a design with minimal inertial interference by the system. *GII* and *GDI* are expressed mathematically as

$$GII = \inf_{\theta \in W} \frac{\underline{\sigma}(J(\alpha,\theta))}{\bar{\sigma}(J(\alpha,\theta))}, \quad GDI = \inf_{\theta \in W} \frac{1}{1+\bar{\sigma}(H(\alpha,\theta))},$$

where W denotes the workspace, J the kinematic Jacobian matrix, H the mass matrix, $\bar{\sigma}, \underline{\sigma}$ are the largest and smallest singular values of the corresponding matrices, α is a known function of link lengths, and θ are configuration variables.

III. Polynomial Manipulator Equations

The two-link manipulator can be characterized by the lengths l_1, l_2 of its links. Here l_1 corresponds to the link fixed to the ground by a revolute joint, and l_2 corresponds to the link connecting the end effector to l_1 through a second revolute joint. Defining $x_i = \cos(\theta_i)$ and $y_i = \sin(\theta_i)$, where θ_i is the absolute angle of rotation of joint i measured from the horizontal axis, the kinematic Jacobian is computed to be

$$J = \begin{bmatrix} -l_1 \sin\theta_1 & -l_2 \sin\theta_2 \\ l_1 \cos\theta_1 & l_2 \cos\theta_2 \end{bmatrix} = \begin{bmatrix} -l_1 y_1 & -l_2 y_2 \\ l_1 x_1 & l_2 x_2 \end{bmatrix}, \quad (1)$$

from which the manipulability matrix is computed as $M = JJ^\top$, and is omitted from the paper due to space constraint.

Without loss of generality, we assume the center of mass of each link is situated at the geometric center of the link, and the mass matrix is given by

$$H = \begin{bmatrix} I_1 + \frac{1}{4} m l_1^2 & \frac{1}{2} m_2 l_1 l_2 (x_1 x_2 + y_1 y_2) \\ \frac{1}{2} m_2 l_1 l_2 (x_1 x_2 + y_1 y_2) & I_2 + \frac{1}{4} m_2 l_2^2 \end{bmatrix}, \quad (2)$$

where I_1, I_2 are the moments of inertia of the link about the axis perpendicular to the plane of operation, m_1, m_2 are the masses of each link and $m = (m_1 + 4m_2)$.

IV. Methods

To optimize kinematic and dynamic performance of a two-link planar manipulator, we formulate an optimization problem whose objective is to maximize *GII* and *GDI* simultaneously. Since SoS programming can be viewed as a special case of semidefinite programming, we need to formulate the optimization problem so that the objective function is a linear function of the decision variables and the constraints are linear matrix inequalities.

[1] Mechanical and Biomedical Engineering, Boise State University.
[2] Mechatronics Engineering, Sabanci University.

978-1-5386-9245-5/19/$31.00 ©2019 IEEE
DOI 10.1109/IRC.2019.00089

IEEE computer society

A. Unconstrained semidefinite optimization of GII and GDI

We use matrix norm minimization technique [5] to address the nonconvexity of *GII*. Let $\|\cdot\|_2$ denote the spectral norm, i.e. the maximum singular value. Defining the decision variables $\alpha_i = l_i^2$, $i = 1,2$ and using the fact that $\|J(\alpha;x,y)\|_2 \leq s$ if and only if $JJ^\top \preceq s^2 I$ (and $s \geq 0$), we can express the optimization that finds the global maximizer of *GII* as

$$\begin{aligned} \underset{(\alpha,s,t)\,\in\mathbb{R}^{n+2}}{\text{minimize}} \quad & s-t \\ \text{subject to} \quad & sI - M(\alpha;x,y) \text{ is SoS} \\ & M(\alpha;x,y) - tI \text{ is SoS.} \end{aligned} \tag{3}$$

From Section III we know that the manipulability matrix $M = JJ^\top$ is a linear function of the decision variables α and a polynomial function of the configuration variables x, y.

To incorporate the maximization of *GDI*, we notice that a natural linear function of the decision variables, whose minimization will also minimize $\bar{\sigma}(H)$, thus maximizing *GDI*, is given by the trace of H. Since $\mathrm{tr}(H) = \sum_i \lambda_i(H)$, minimizing $\mathrm{tr}(H)$ is equivalent to minimizing the maximum value of the sum of the eigenvalues of H over the workspace.

B. Formulating the constraints

The semidefinite program (3) is not yet a faithful optimization problem because it lacks the circle constraints $x_i^2 + y_i^2 = 1$, $\forall i = 1,2$, and the workspace constraints, G, defined as

$$G(\alpha,\beta,x,y) \;=\; \begin{bmatrix} -y_2 \\ x_1y_2 - x_2y_1 \\ x_1x_2 + y_1y_2 - \varepsilon \\ \varepsilon - x_1x_2 - y_1y_2 \\ c_i^\top \alpha + d_i \end{bmatrix} \leq 0,\; i = 1,\ldots,m,$$

where $\varepsilon = 0.5$ is the bound on $|\cos(\theta_1 - \theta_2)|$ and parameters for affine constraints on the decision variables α are given as $c_1 = \begin{bmatrix}1 & 1\end{bmatrix}^\top$, $c_2 = -\begin{bmatrix}1 & 1\end{bmatrix}^\top$ and $d_1 = -\frac{9}{70}\,\mathrm{m}^2$, $d_2 = \frac{1}{10}\,\mathrm{m}^2$.

The circle constraints, the elbow-out posture constraints, G_1, G_2 and nonsingularity constraints G_3 and G_4 of G are polynomial constraints, which can be embedded in SoS programming by invoking the *S-procedure* described in [1]. The last two elements of G define affine constraints on the decision variables and can be directly inserted into the final semidefinite program (4).

C. Multicriteria Optimization

Once the single criteria optimization problems have been cast as semidefinite programs, we combine the two using the weighted-sum approach to generate the Pareto-front. This involves solving the following semidefinite program for each value of the parameter $0 \leq \gamma \leq 1$.

$$\begin{aligned} \underset{(\alpha,s,t)\,\in\mathbb{R}^{n+2}}{\text{minimize}} \quad & \gamma c^\top \alpha + (1-\gamma)(s-t) \\ \text{subject to} \quad & sI - M(\alpha;x,y) \text{ is SoS} \\ & M(\alpha;x,y) - tI \text{ is SoS} \\ & G(\alpha;x,y) \;\leq 0 \\ & \alpha_l \;\leq \alpha \leq \alpha_u \\ & x_i^2 + y_i^2 \;= 1,\;\; \forall i = 1,2. \end{aligned} \tag{4}$$

where $\alpha_l = 0.1$ and $\alpha_u = 1.0$ stand for the lower and upper bounds on the design variables, respectively.

V. Results and Discussion

Table I presents the results of the SoS optimization algorithm for the single objective problems, for best kinematic (γ = 0) and best dynamic isotropy (γ = 1), respectively.

TABLE I: Results of independent optimizations

	Best Design for Kinematic Isotropy	Best Design for Dynamic Isotropy	Unit
GII	0.57735	0.28021	–
GDI	0.89354	0.95012	–
l_1	0.22361	0.10000	m
l_2	0.22361	0.30000	m

To characterize the trade-off between the single objective solutions, Pareto-front curve for the bi-objective optimization problem is constructed. This can be viewed through the link `https://bit.ly/2M7nTUc`. Schematics of the mechanism at the two extremes, i.e., best kinematic and dynamic isotropy, and at an approximately equal-trade-off-point are shown in the subplots within this figure.

With regards to computational demands, the multi-objective problem with 101 distinct values of γ only took approximately **2 minutes** to complete using a single *mobile* Intel $\mathrm{i7}-7700\mathrm{HQ}$ processor operating at 3.80GHz. These results serve as a strong indication that the application of framework presented in this case study to more complex mechanisms promises substantial improvements in accuracy and computational performance.

VI. Conclusions and Future Work

We cast design optimization of robotic manipulators, which was previously solved via nonlinear optimization techniques, as a convex optimization problem by employing sums-of-squares optimization. As a result, we are able to find the globally optimal solution for the single objective problems with arbitrarily high precision (up to machine precision). Interior-point methods are used to obtain the numerical solution to a SoS optimization problem very efficiently. They scale exceptionally well with increasing number of decision variables in contrast to other methods, such as branch-and-bound algorithms, which are able to find the globally optimal solution. In future work, we will use the same technique to optimally design parallel mechanisms, which additionally require the satisfaction of loop equations.

References

[1] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

[2] D. Henrion and A. Garulli, *Positive polynomials in control*. Springer Science & Business Media, 2005, vol. 312.

[3] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[4] L. Stocco, S. E. Salcudean, and F. Sassani, "Fast constrained global minimax optimization of robot parameters," *Robotica 16(6)*, pp. 595–605, 1998.

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