Globally Optimal Solution to Inverse Kinematics of 7DOF Serial Manipulator

Pavel Trutman[1,2], Mohab Safey El Din[3], Didier Henrion[4,1], Tomas Pajdla[2]

***Abstract*—The Inverse Kinematics (IK) problem is concerned with finding robot control parameters to bring the robot into a desired position under the kinematics and joint limit constraints. We present a globally optimal solution to the IK problem for a general serial 7DOF manipulator with revolute joints and a polynomial objective function. We show that the kinematic constraints due to rotations can be all generated by the second-degree polynomials. This is an important result since it significantly simplifies the further step where we find the optimal solution by Lasserre relaxations of nonconvex polynomial systems. We demonstrate that the second relaxation is sufficient to solve a general 7DOF IK problem. Our approach is certifiably globally optimal. We demonstrate the method on the 7DOF KUKA LBR IIWA manipulator and show that we are, in practice, able to compute the optimal IK or certify infeasibility in 99.9 % tested poses. We also demonstrate that by the same approach, we are able to solve the IK problem for any generic (random) manipulator with seven revolute joints.**

***Index Terms*—Kinematics, redundant robots, mechanism design.**

I. Introduction

THE Inverse Kinematics (IK) problem is one of the most important problems in robotics [1]. The solution to the IK problem finds robot control parameters to bring the robot into the desired position under the kinematics and collision constraints [2].

The IK problem has been extensively studied in robotics and control [3], [4]. The classical formulation [3] of the problem for 6 degrees of freedom (DOF) serial manipulators leads to solving a system of polynomial equations [5], [6]. This is a hard ("EXPSPACE complete" [7]) algebraic computational problem, but practical solving methods have been developed for 6DOF manipulators [3], [8], [9]. For more DOFs, the IK task is currently solvable only for nongeneric specially designed manipulators, such as *KUKA LBR IIWA* [10].

An important generalization of the IK problem aims at finding the optimal control parameters for a redundant mechanism, i.e. when the number of controlled joints in a manipulator is larger than six. Then, an algebraic computation problem turns into an optimization problem over an algebraic variety [5] of possible IK solutions. It is particularly convenient to choose a polynomial objective function to arrive at a semialgebraic optimization problem.

Semialgebraic optimization problems are, in general, nonconvex, but they can be solved with certified global optimality [11] using the Lasserre hierarchy of convex optimization problems [12]. Computationally, however, semialgebraic optimization problems are in general extremely hard and were often considered too expensive to be used in practice. In this work, we show that using "algebraic preprocessing" in semialgebraic optimization methods becomes practical in solving the IK problem of general 7DOF serial manipulators.

A. Our contributions

We prove that the variety of IK solutions of all generic 7DOF revolute serial manipulators can be generated by the second-degree polynomials only (Theorem 1). This considerably reduces the complexity of semialgebraic optimization and makes it computationally feasible.

We provide a method for computing a globally optimal solution to the IK problem for a general 7DOF serial manipulator with a polynomial objective function. The found solution is globally optimal w.r.t. the given manipulator, requested end-effector pose, and chosen objective function.

We employ techniques from algebraic geometry [5] and polynomial optimization [11] to solve the 7DOF IK problem exactly or certify the infeasibility when it happens.

We demonstrate that our approach works on a practical 7DOF *KUKA LBR IIWA* manipulator solving 99.9 % of configurations while the straightforward semi-algebraic optimization fails in approx. 28 % of cases.

We show that we can solve the IK problem for any generic (e.g., randomly generated) serial manipulator with seven revolute joints by the same approach.

We do not expect our method to be an online method as it is more time-consuming to find the global solution than to be satisfied with a feasible one. The local methods are more suitable for online usage as they are fast and sufficiently accurate. We see the application of our method in the design and exploration of the capabilities of manipulators. The offline method suits

Manuscript received: Novemeber, 29, 2021; Revised February, 18, 2022; Accepted March, 14, 2022.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. P. Trutman and T. Pajdla were supported by the EU SIF Oper. Prog. Res. Dev. and Edu. proj. IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468) and GA CTU Prague proj. SGS19/173/OHK3/3T/13. D. Henrion and M. Safey El Din were supported by the EU H2020 Marie Skłodowska-Curie grant agreement N°813211 (POEMA). M. Safey El Din was supported by the grants ANR-18-CE33-0011 Sesame, ANR-19-CE40-0018 De Rerum Natura, ANR-19-CE48-0015 ECARP. We are grateful to Stefan Richter for his helpful suggestions.

[1]P. Trutman and D. Henrion are with FEE, CTU in Prague, Czechia pavel.trutman@cvut.cz, henrion@laas.fr.

[2]P. Trutman and T. Pajdla are with CIIRC, CTU in Prague, Czechia pajdla@cvut.cz.

[3]M. Safey El Din is with Sorbonne Université and LIP6 CNRS, France Mohab.Safey@lip6.fr.

[4]D. Henrion is with LAAS-CNRS Toulouse, France.

Digital Object Identifier (DOI): see top of this page.

these tasks well as we are not limited by computation time. Our method can be, e.g., used when designing new 7DOF serial manipulators and optimizing their parameters, such as the manipulability. We develop our method for 7DOF serial manipulators as they are currently the most common redundant manipulators in the industry.

II. Previous work

The first breakthrough in solving IK problems was the global solution to IK for a general 6DOF serial manipulator, which was given in [13], [8]. It leads to solving a polynomial system with 16 solutions. Another important result was the solution to the forward kinematics problem of the Stewart-Gough parallel manipulator platform [14], leading to a polynomial system with 40 solutions. See recent work [15] to review local and other approximate techniques for solving IK problems. We next review only the most relevant work.

The closest previous works are related to solving IK for mechanisms, which are redundant only for fixed end-effector positions. The standard approach is to employ additional dynamics, time optimality, and collision constraints.

In [16], a technique for planning a dynamic whole-body motion of a humanoid robot has been developed. It solves IK as a part of motion planning by local optimization methods considering kinematics, dynamics, and collision models. The method requires a good initialization to converge, and, depending on its quality, it takes minutes to hours of running time. Our approach provides a globally optimal solution for kinematics subchains of more complex mechanisms and could be used to initialize the kinematics of motion planning.

Work [10] presented an IK solution for 7DOF manipulators with zero link offsets, e.g., the *KUKA LBR IIWA* manipulators. The solution uses special kinematics of its class of manipulators to decompose the general IK problem into two simpler IK problems that can be solved in a closed-form. The one-dimensional variety of self-motions becomes circular, and hence the paper proposes to parameterize it by the angle of a point of the circle. Our approach shows that it is feasible to solve the IK problem for completely general 7DOF manipulators and optimize over their self-motion varieties.

Paper [15] presents a global (but only approximate) solution to the IK for 7DOF manipulators. It formulates the IK problem as a mixed-integer convex optimization program. The key idea of the paper is to approximate the nonconvex space of rotations by piecewise linear functions on several intervals that partition the original space. This turns the original nonconvex problem into an approximate convex problem when a correct interval is chosen. Selecting the actual interval of approximation leads to the integer part of the optimization. This was the first practical globally optimal approach, but it is only approximate and delivers solutions with errors in units of centimeters and degrees. It also fails to detect about $5\ \%$ of infeasible poses. Our approach solves the original problem with sub-10^{-3} mm and degree error and can solve/decide the feasibility in all but $0.1\ \%$ of the tested cases. The computation times of [15] and our approach are roughly similar in units of seconds.

A global and precise solution to the IK problem for redundant serial manipulators was presented in [17]. It models the kinematic constraints as a distance geometry problem, which, alongside a novel formulation of the joint limit constraints, leads to quadratic constraints. The final configuration is found as the nearest configuration to the given one while satisfying the constraints. This quadratically constrained quadratic problem is solved by a semidefinite programming (SDP) relaxation with a global optimality certificate and infeasibility detection. Their implementation is fast (2.5 ms per pose) and accurate (sub-10^{-2} mm position error) with a failure rate of less than $0.4\ \%$. This formulation is restricted only to revolute joints for planar manipulators and spherical joints for spacial ones, not considering the full rotation of each link, which leads to simplified situations. In contrast, our method is general and applicable to any serial manipulator with revolute joints.

III. Problem formulation

Here we formulate the IK problem for 7DOF serial manipulators as a semialgebraic optimization problem.

The task is to find the joint coordinates of the manipulator in a way that the end-effector reaches the desired pose in space. If the manipulators have more DOF than they require to execute the given task, they are called redundant. This is our case, as the 7DOF manipulator is to reach a pose in space, which has 6 DOFs. The consequence is that the variety of IK solutions is one-dimensional for reachable generic end-effector poses for such manipulators.

The self-motion property of such manipulators makes them more versatile since it allows them to, e.g., avoid more obstacles in the path and avoid singularities. On the other hand, increasing the number of degrees of freedom increases the difficulty of the IK problem computation dramatically. The IK problem has no longer a finite number of solutions (while for serial manipulators with fewer DOFs, it has, in general, a finite number of solutions), and thus, it is natural to formulate it as a constrained optimization problem.

We next model the IK problem for 7DOF serial manipulators as a polynomial optimization problem (POP).

A. Inverse kinematics problem

We describe the manipulators by the Denavit-Hartenberg (D-H) convention [18] to construct transformation matrices $M_i(\theta_i) \in \mathbb{R}^{4\times 4}$ from link i to $i-1$ and parameterized by joint angles θ_i. Their product for i from 1 to 7 gives us the transformation matrix M representing the transformation from the end-effector coordinate system to the base coordinate system

$$\prod_{i=1}^{7} M_i(\theta_i) = M. \tag{1}$$

The matrix M consists of the position vector $t \in \mathbb{R}^3$ and the rotation matrix $R \in SO(3)$, which together represent the end-effector pose w.r.t. the base coordinate system.

Due to kinematic constraints, manipulators come with joint limits restricting the joint angles θ_i. Typically, their maximal θ_i^{High} and minimal θ_i^{Low} values are given satisfying $\theta_i^{Low} \leq \theta_i \leq \theta_i^{High}$ for $i = 1, \ldots, 7$.

To solve the IK problem, we set up the desired pose M and then solve Eqn. (1) for the joint coordinates θ_i.

For redundant manipulators, the polynomial system has an infinite number of solutions. By introducing an objective function, we select just one. Our choice is to prefer the solutions with minimal sum of distances of the joint angles $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_7]^\top$ from their preferred values $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \ldots, \hat{\theta}_7]^\top$

$$\min_{\boldsymbol{\theta} \in \langle -\pi, \pi \rangle^7} \sum_{i=1}^{7} w_i \left| \mathrm{angdiff}(\theta_i, \hat{\theta}_i) \right|, \tag{2}$$

where $w_i \geq 0$, $\sum_{i=1}^{7} w_i = 1$ and the function $\mathrm{angdiff}(\alpha, \beta)$ calculates the difference $\alpha - \beta$ and wraps it on the interval $\langle -\pi, \pi)$. Depending on how the preferred values $\hat{\boldsymbol{\theta}}$ are set, this objective function can be used to keep some links of the manipulator close to a required pose [19], minimize the distance from joints midrange [20], or minimize the total movement of the manipulator to reach the next pose.

In addition to joint limits, we get the following problem:

$$\begin{aligned} &\min_{\boldsymbol{\theta} \in \langle -\pi, \pi \rangle^7} \sum_{i=1}^{7} w_i \left| \mathrm{angdiff}(\theta_i, \hat{\theta}_i) \right| \\ &\text{s.t. } \textstyle\prod_{i=1}^{7} M_i(\theta_i) = M \\ &\qquad \theta_i^{Low} \leq \theta_i \leq \theta_i^{High} \qquad (i = 1, \ldots, 7) \end{aligned} \tag{3}$$

To use the technique of polynomial optimization, we need to remove the trigonometric functions that appear in Eqn. (1). We do that by introducing new variables $\boldsymbol{c} = [c_1, \ldots, c_7]^\top$ and $\boldsymbol{s} = [s_1, \ldots, s_7]^\top$, which represent the cosines and sines of the joint angles $\boldsymbol{\theta}$, respectively. Then, we rewrite Problem (3) in the new variables, and to preserve the structure, we need to add the trigonometric identities

$$q_i(\boldsymbol{c}, \boldsymbol{s}) = c_i^2 + s_i^2 - 1 = 0, \ i = 1, \ldots, 7. \tag{4}$$

Matrix Eqn. (1) contains 12 trigonometric equations, which rewritten in the newly introduced variables leads to 12 polynomial equations of degrees up to 7. To lower the maximal degree of the equations, we use fact that the inverse of a rotation matrix is its transpose, i.e., it is a linear function of the original rotation matrix, and rewrite Eqn. (1) as

$$\prod_{i=3}^{5} M_i(\theta_i) - \prod_{i=2}^{1} M_i^{-1}(\theta_i) M \prod_{i=7}^{6} M_i^{-1}(\theta_i) = 0. \tag{5}$$

It reduces the maximal degree of the polynomials in $\boldsymbol{c}$ and $\boldsymbol{s}$ to four. We denote these polynomials in Eqn. (5) as

$$p_j(\boldsymbol{c}, \boldsymbol{s}) = 0, \ j = 1, \ldots, 12. \tag{6}$$

Observe that the coefficients of the polynomials p_j depend on the entries of the transformation matrices M_i which define the 7DOF manipulators we consider. We further investigate *genericity* properties of polynomials p_j w.r.t. the entries of matrices M_i. Precisely, considering the entries of matrices M_i as parameters, we say that a property on the polynomials p_j holds generically if it does not fail for a larger set than an algebraic variety of the dimension strictly smaller than the dimension of the space. This is a way to encode in a rigid way that some property holds for randomly chosen values.

The next step is to change the objective function (2) to a polynomial in the variables $\boldsymbol{c}$ and $\boldsymbol{s}$. Instead of evaluating the distance between the joint angles and their preferred values, we can do the same in the space of their cosines and sines to reach the same goal, i.e., to get $\boldsymbol{\theta}$ as close as possible to $\hat{\boldsymbol{\theta}}$. Choosing the proper ℓ_p norm for the problem at hand may lead to a more straightforward solution to the problem (e.g., the ℓ_∞ norm is often used in multiple view geometry problems [21] to obtain a convex relaxation of the original problem). We use the squared ℓ_2 norm on the cosines and sines since it leads to a linear objective function:

$$\min_{\substack{\boldsymbol{c} \in \langle -1, 1 \rangle^7 \\ \boldsymbol{s} \in \langle -1, 1 \rangle^7}} \sum_{i=1}^{7} w_i \Big((c_i - \cos \hat{\theta}_i)^2 + (s_i - \sin \hat{\theta}_i)^2 \Big) \tag{7}$$

$$= \min_{\substack{\boldsymbol{c} \in \langle -1, 1 \rangle^7 \\ \boldsymbol{s} \in \langle -1, 1 \rangle^7}} \sum_{i=1}^{7} 2 w_i (1 - c_i \cos \hat{\theta}_i - s_i \sin \hat{\theta}_i). \tag{8}$$

After rewriting the joint limit inequalities into a polynomial form, we obtain the following final POP

$$\begin{aligned} &\min_{\boldsymbol{c} \in \langle -1, 1 \rangle^7, \ \boldsymbol{s} \in \langle -1, 1 \rangle^7} \sum_{i=1}^{7} 2 w_i (1 - c_i \cos \hat{\theta}_i - s_i \sin \hat{\theta}_i) \\ &\text{s.t.} \qquad p_j(\boldsymbol{c}, \boldsymbol{s}) = 0 \ \ (j = 1, \ldots, 12) \\ &\qquad q_i(\boldsymbol{c}, \boldsymbol{s}) = 0 \ \ (i = 1, \ldots, 7) \\ &\qquad -(c_i + 1) \tan \tfrac{\theta_i^{Low}}{2} + s_i \geq 0 \ \ (i = 1, \ldots, 7) \\ &\qquad (c_i + 1) \tan \tfrac{\theta_i^{High}}{2} - s_i \geq 0 \ \ (i = 1, \ldots, 7) \\ &\qquad c_i - \min\{\cos \theta_i^{Low}, \cos \theta_i^{High}\} \geq 0 \ \ (i = 1, \ldots, 7) \end{aligned} \tag{9}$$

We next show how to solve this POP in a general way such that any objective function can be chosen as long as it can be expressed as a low degree polynomial in sines and cosines of the joint angles. Although different objective functions will be chosen for different tasks, we demonstrate the presented approach with a classical objective function (8).

After solving Problem (9), we recover $\boldsymbol{\theta}$ from $\boldsymbol{c}$ and $\boldsymbol{s}$ by function atan2, which considers the signs of the arguments.

IV. Polynomial optimization

Here we describe the polynomial optimization methods we use to solve Problem (9).

In general, POPs are nonconvex, but they can be solved with global optimality certificates by convex optimization [11]. The idea consists of building a hierarchy of convex optimization problems of increasing size whose values converge to the value of the POP. The convergence proof is based on the results of real algebraic geometry, namely, on the representation of positive polynomials or Positivstellensatz. One of the most popular Positivstellensatz is due to Putinar [22], and it expresses a polynomial positive on a compact basic semialgebraic set as a sum of squares (SOS).

Finding this SOS representation amounts to solving an SDP problem, a particular convex optimization problem that can be solved efficiently numerically with interior point algorithms. By increasing the degree of the SOS representation, we increase the size of the SDP problem, thereby constructing a hierarchy of SDP problems. Dual to this polynomial positivity

problem is the problem of characterizing the moments of measures supported on a compact basic semialgebraic set. This admits an SDP formulation, called moment relaxations, yielding a dual hierarchy indexed by the so-called relaxation order.

The primal-dual hierarchy is called the moment-SOS hierarchy or the Lasserre hierarchy since it was first proposed in [12] in the context of POP with convergence and duality proofs. As the relaxation order increases, the Lasserre hierarchy generates a monotone sequence of superoptimal bounds on the global optimum of a given POP. Eventually, the result on the moment problem can be used to certify the exactness of the bound for the current relaxation order. This solves the original nonconvex POP at the price of solving a relaxed convex SDP problem of typically (quite) bigger size than was the original problem. A Matlab package GloptiPoly [23] has been designed to construct the SDP problems in the hierarchy and solve them with a general-purpose SDP solver.

In the moment-SOS hierarchy, we can certify global optimality by using flat truncation of the moment matrix. The rank of the moment matrix is then equal to the number of global optima. The global optima can be extracted by numerical linear algebra, as described in [24].

In many applications, the main limitation of the Lasserre hierarchy is its poor scalability w.r.t. the number of variables and the degree of the POP. This is balanced by the practical observation that, very often, global optimality is certified by the second or third-order relaxation. As our experiments reveal, for the degree 4 POP (9), the third-order relaxation is out of reach for off-the-shelf SDP solvers. It is hence critical to investigate reformulation techniques to reduce the degree.

V. Symbolic reduction of the POP

Here we provide the description of the algebraic geometry technique we use to reduce the degree of our POP problem to obtain a practical solving method. See [5] for algebraic-geometric notation and concepts.

Let us assume that our POP is constrained by the polynomial equations $f_1 = \cdots = f_s = 0$ of degree 4 in $\mathbb{Q}[x_1, \ldots, x_n]$. One can replace these polynomial equations with any other set of polynomial equations $g_1 = \cdots = g_t = 0$ as long as both systems have the same solution set. Natural candidates for g_i's are polynomials in the *ideal* generated by $f_1, \ldots, f_s$, i.e., in the set of algebraic combinations $I = \{\sum_i q_i f_i \mid q_i \in \mathbb{Q}[x_1, \ldots, x_n]\}$. It is clear that if all f_i's vanish simultaneously at a point, any polynomial g in this set I will vanish at that point.

The difficulty is how to understand the structure of this set and find a nice finite representation of it that would allow many algebraic operations (such as deciding whether a given polynomial lies in this set). Solutions have been brought by symbolic computation, aka computer algebra, through the development of an algorithm computing *Gröbner bases* (GB), which were introduced by Buchberger [5]. These are finite sets, depending on a monomial ordering [5], which generate I as input equations do, but from which the whole structure of I can be read off.

Modern algorithms for computing Gröbner bases ($F4$ and $F5$ algorithms), which significantly improved by several orders of magnitude the state-of-the-art, were introduced by J. C. Faugère [25], [26]. These latter algorithms bring a linear algebra approach to GB computations. In particular, noticing that the intersection of I with the subset of polynomials in $\mathbb{Q}[x_1, \ldots, x_n]$ of degree $\leq d$ is a vector space of finite dimension is the key to reducing GB computations to exact linear algebra operations.

Hence, Gröbner bases provide bases of such vector spaces when one uses monomial orderings, which filter monomials w.r.t. degree first. Finally, going back to our problem, a GB computation allows us to discover if I contains degree 2 polynomials (and is generated by such quadrics).

While this is never the case when starting with a generic POP of degree 4, observe that there are many relations between the coefficients of the degree 4 equations of our POP. Hence, we are not facing a generic situation here, and we will see further that a GB computation provides a set of quadrics that can replace our initial set of constraints. Note also that since GB algorithms rely on exact linear algebra, such a property holds for every generic instance of our POP if it holds for a randomly chosen one (the trace of the computation will always be the same, giving rise to polynomials of degree ≤ 2).

VI. Solving the IK problem

To solve the IK problem, we need to solve the optimization problem (9). First, we directly apply the implementation GloptiPoly [23] of the method described in Section IV.

A. Direct application of polynomial solver

Since the original Problem (9) contains polynomials of degree 4, we start with the smallest possible relaxation of order 2. The POP is in 14 variables, which leads to solving an SDP program with 3060 variables.

Solving this relaxation typically does not yield the solution, and therefore it is required to go higher in the hierarchy. Unfortunately, the relaxation order three for a polynomial problem in 14 variables leads to an SDP problem in $38\,760$ variables. Such a large problem is still solvable on contemporary computers, but it takes hours to finish.

B. Symbolic reduction

In view of the previous paragraph, we aim at simplifying the original POP to be able to obtain solutions even for the relaxation of order two, which takes seconds to solve.

Here is our main result. We claim that the polynomials p_j and q_i of degrees up to four in Problem (9) can be reduced to polynomials of degree two.

Theorem 1. *The ideal generated by the kinematics constraints (6) for a generic serial manipulator with seven revolute joints and for generic pose M with the addition of the trigonometric identities (4) can be generated by a set of degree two polynomials.*

Proof. The proof is based on *specialization properties* of Gröbner bases in [27] and is similar to the one in [28, App.

C]. As sketched above, we start by considering the coefficients of the transformation matrices as parameters (whose specializations define a given serial manipulator). This yields polynomials p_j depending symbolically on these parameters. Since Gröbner bases algorithms use only arithmetic operations $+, -, \times, /$, one can consider the generic Gröbner basis of a generic 7DOF serial manipulator, leaving the parameters as they are. We obtain a basis whose coefficients are rational fractions in these parameters. Specialization properties state that specializing these parameters actually yields a Gröbner basis of the considered 7DOF. Computing a Gröbner basis with randomly chosen values of the parameters (hence avoiding the vanishing of the denominators in the generic Gröbner basis) reveals the structure of this generic Gröbner basis. Doing so, one sees that the Gröbner basis contains polynomials of degree two and that these polynomials generate the whole Gröbner basis. □

C. Solving the reduced polynomial optimization problem

We exploit Theorem 1 to solve the IK problem followingly. First, we compute a Gröbner basis of the kinematics constraints (6) and (4). From the basis, we select only the degree two polynomials. Then, we construct Problem (9) with the degree two polynomial constraints instead of the original ones. We solve it by using the hierarchy of SDPs.

Reducing the degree of polynomials from four to two allows us to start with SDP relaxation of order one. The size of this SDP problem, in terms of the number of variables, is now 120. Practical experiments have shown that the first relaxation is not tight enough to yield the solution. On the other hand, the second relaxation gives a solution for almost all poses, see Tab. II.

D. Overcoming numerical issues

It may happen that the POP solver is not able to certify the global optimality. The certification is based on the evaluation of the rank of a matrix of floats returned by the SDP solver, and it may fail because of two reasons.

First, it may happen that the relaxation order was not high enough, and we have to go higher in the hierarchy. Secondly, the relaxation is actually tight, but the numerical rank of the matrix returned by the SDP is hard to compute correctly. This is because of the numerical issues of the SDP solver caused by the significant number of variables and constraints that typically have SDPs originated from the Lasserre hierarchies. We can not distinguish these two cases, and whenever we are unable to certify the optimum, we say that the method has failed for the given end-effector pose.

Naturally, we want to minimize the number of end-effector poses that our method fails to compute. For a given relaxation order, we can thus only reduce the number of failing poses by addressing the numerical issues of the SDP solver used.

From our experience and as we show in Section VII, SDPs with sharper objective function at the global minimum have a lower failure rate. To sharpen a function, we mean that we take its n-th power as long as we ensure that it is greater or equal to one. The new function has the same argument of the global minimum and a greater or equal absolute value of its second derivative at that minimum than the original one.

Our new objective function replacing Eqn. (8) is

$$\min_{\boldsymbol{c}\in\langle -1,1\rangle^7,\ \boldsymbol{s}\in\langle -1,1\rangle^7} \sum_{i=1}^{7} 2w_i(2 - c_i\cos\hat{\theta}_i - s_i\sin\hat{\theta}_i)^n \quad (10)$$

for any $n \in \mathbb{N}$. Obviously, Eqn. (10) has a different argument of the minimum than Eqn. (8) (for $n = 1$, they are the same), but still Eqn. (10) keeps the idea of the original non-polynomial objective function (2).

Since we use the second relaxation order in the Lasserre hierarchies, we can use an objective function of up to degree four without the need for enlargement of the relaxation order. Thus, we can use n up to four.

We show that taking higher power of the objective function improves the failure rate of our method in Section VII.

E. Rational approximation

The end-effector pose M consists of translation vector $t \in \mathbb{R}^3$ and rotation matrix $R \in SO(3)$, which are, as well as the D-H parameters of the manipulator, in practice given in their floating-point representation. This is a common approach as these values are typically an outcome of some planning algorithm (the end-effector pose) or measured and calibrated (the parameters of the manipulator).

When a numerical method is used to solve the IK task, everything works smoothly as long as the problem is well-conditioned. This is the case when we directly apply the POP solver, as described in Section VI-A.

On the other hand, symbolic methods require to compute exactly. Therefore, if we want to use a symbolic method, e.g., as in Section VI-C, we need to pass from floating-point numbers to exact rational numbers and ensure that all identities, following from sines, cosines, and rotations that have to hold, are valid.

The input for the symbolic reduction method (Section VI-C) is the D-H parameters of the manipulator and the end-effector pose M, which are floating points and need to be approximated by rational numbers. The D-H parameters are a) the lengths a_i and d_i, which we approximate by rounding them to 2κ digits to the right of the decimal point, where $\kappa \in \mathbb{N}$, and b) the angles α_i. In Eqn. (5), we only need rational values of $\sin\alpha_i$ and $\cos\alpha_i$ such that the trigonometric identities $\sin^2\alpha_i + \cos^2\alpha_i = 1$ hold. How to obtain such a rational representation in an optimal way w.r.t. bit size has been proposed, e.g., by [29]. For simplicity, we have used a nonoptimal approach. To provide the rational representation, we round $\tan\frac{\alpha}{2}$ to κ digits to the right of the decimal point, which we denote as τ_i. Then, the sines and cosines are replaced by their approximations as $\cos\alpha_i = \frac{1-\tau_i^2}{1+\tau_i^2}$ and $\sin\alpha_i = \frac{2\tau}{1+\tau_i^2}$, which are rational functions of τ_i, and therefore also rational.

We approximate the rotational part R, and the translational part t of the end-effector pose M independently. We approximate t element-wise by rounding it to 2κ digits to the right of the decimal point. For the rotation matrix R, we need to approximate it as a rational orthonormal matrix to ensure that

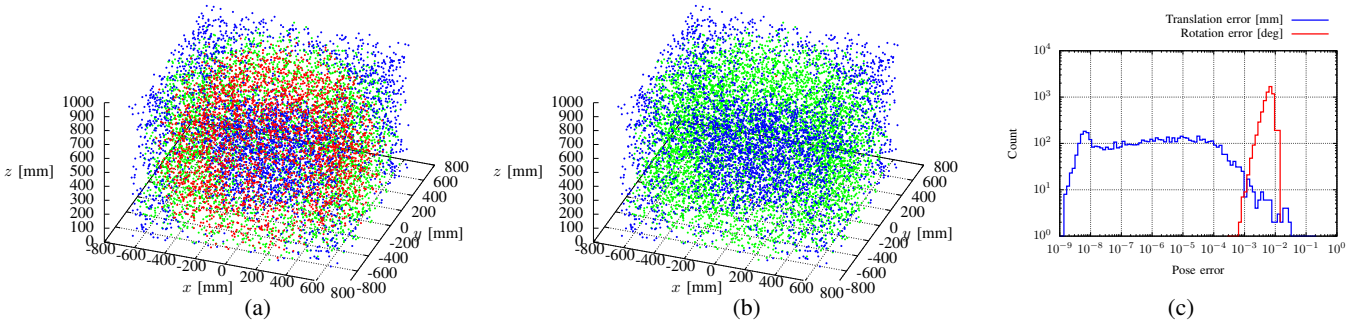

Fig. 1: Generated poses (their positional parts shown only) for the *KUKA LBR IIWA* manipulator. Green dots are poses marked as feasible, blue as infeasible, and for the red ones, the computation failed. (a) Naïve approach (Section VII-A2). (b) Our method with the symbolic reduction step (Section VII-A3). (c) Histogram of translation and rotation error for the same method as (b). Showed results are for the relaxation order $r = 2$ and the power of the objective function $n = 4$.

TABLE I: Overview of the execution times and accuracy of the presented methods. Results are for the relaxation order $r = 2$ and the power of the objective function (10) $n = 4$ and evaluated on the *KUKA LBR IIWA* manipulator.

Method	**Average execution time [s]**		**Mean error [mm, deg]**		**Failure rate [%]**
	Reduction step	**GloptiPoly**	**Translation**	**Rotation**	
Naïve (Section VII-A2)	—	14.1	$3.77 \cdot 10^{-4}$	$6.48 \cdot 10^{-5}$	27.6
With symbolic reduction (Section VII-A3)	2.6	4.4	$1.51 \cdot 10^{-4}$	$5.45 \cdot 10^{-3}$	0.02

TABLE II: Comparison of failure rates, average end-to-end execution time, and mean translation and rotation error of our proposed methods for different relaxation orders and powers of the objective function for the *KUKA LBR IIWA* manipulator.

Method	**Relaxation order**	**Power of the objective function (10)**							
		$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
		Mean translation error [10^{-3} mm]				**Mean rotation error [10^{-3} deg]**			
Naïve (Section VII-A2)	$r = 2$	1.68	0.94	0.60	0.38	0.28	0.17	0.12	0.06
With symbolic reduction (Section VII-A3)	$r = 1$	7.80	1.82	—	—	6.43	5.65	—	—
With symbolic reduction (Section VII-A3)	$r = 2$	2.52	0.83	0.24	0.15	5.64	5.51	5.46	5.45
		Average execution time [s]				**Failure rate [%]**			
Naïve (Section VII-A2)	$r = 2$	17.5	17.0	16.8	16.0	32.8	29.7	28.3	27.6
With symbolic reduction (Section VII-A3)	$r = 1$	5.3	5.2	—	—	31.9	28.1	—	—
With symbolic reduction (Section VII-A3)	$r = 2$	9.5	9.3	9.2	8.8	0.4	0.2	0.09	0.02

$R^\top = R^{-1}$ and $\det R = 1$. In work [30], several algorithms for optimal approximation of a rotation matrix w.r.t. elements bit size have been introduced. We use an easier-to-implement method. To find a rational approximation of R, we convert it to a quaternion q, which we round element-wise to κ digits to the right of the decimal point and denote it as $\bar{q}$. The rounding violates the condition $\|\bar{q}\| = 1$, and we can not divide $\bar{q}$ by non-rational $\|\bar{q}\|$ to get a rational quaternion. We overcome the issue by constructing a rotation matrix from non-unit rational $\bar{q}$, which we then divide by $\|\bar{q}\|^2$, obtaining a rational rotation matrix.

The output of the symbolic reduction method is the degree two polynomials. They have rational coefficients, so we evaluate them using floating-point arithmetic to convert them to floating-point numbers. Then, we can use them in the numerical method (GloptiPoly) to solve the IK task.

VII. Experiments

We demonstrate our method on the IK problem for the *KUKA LBR IIWA* arm, which is simple to solve. Then, we randomly generate a completely generic serial manipulator with seven revolute joints and solve the IK problem for it.

A. The KUKA LBR IIWA manipulator

The manipulator structure is designed in a special way such that the IK problem is simple to compute. There are three sequences of three consecutive revolute joints whose axes of motion intersect in a single point. Namely, they are the joints $(1, 2, 3)$, $(3, 4, 5)$, and $(5, 6, 7)$. Each of these triplets can be substituted by a single spherical joint. Moreover, the joint angle θ_4 is constant within the self-motion for a fixed end-effector pose. Such properties make the manipulator a very nongeneric serial manipulator.

1) POP for KUKA LBR IIWA: We parameterize Problem (9) by the D-H parameters of the *KUKA* manipulator. We set the weights equally to $w_i = \frac{1}{7}$, and the preferred joint angles $\hat{\theta}_i$ to zero. This leads to POP in 14 variables and with polynomials p_j of degrees up to 4.

2) Direct application of the POP solver: First, we solve Problem (9) with objective function (10) with powers n from

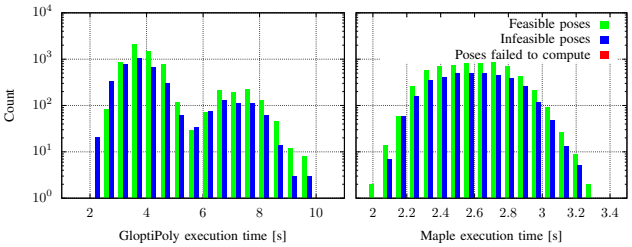

Fig. 2: Histogram of the execution time of (left) the online phase of GloptiPoly, (right) the symbolic reduction and elimination in Maple. Showed results are evaluated on the *KUKA LBR IIWA* manipulator by the method with the symbolic reduction step (Section VII-A3) for the relaxation order $r = 2$ and the power of the objective function $n = 4$.

1 to 4 by toolbox GloptiPoly [23] for relaxation order 2 with the use of MOSEK [31] as the SDP solver.

Our dataset consists of 10 000 randomly chosen poses within and outside of the working space of the manipulator, see Fig. 1a. For poses marked by red color, GloptiPoly failed to find the solution or report infeasibility. That is mainly due to the small relaxation order of the semidefinite relaxation. For the best choice of the power of the objective function $n = 4$ (see Tab. II), there is 27.6 % of such poses, which makes this approach quite impractical. Computations for the next relaxation of order three are still often feasible on contemporary computers but take hours to finish.

3) POP with symbolic reduction: Since the performance of GloptiPoly highly depends on the number of variables of the POP and the relaxation degree, which grows with the degree of the polynomials in the POP, we first symbolically reduce the polynomials p_j and q_i and then solve the new POP. Before we start, we approximate the inputs, which are given in the floating-point representation, by rational numbers with precision $\kappa = 4$ according to Section VI-E.

Firstly, we use the advantage of the simple structure of the *KUKA LBR IIWA* manipulator, i.e., that the joint angle θ_4 is constant within the self-motion. Therefore, it plays no role in the objective function (2). That allows us to eliminate the variables c_4 and s_4 from the equations. Secondly, we reduce the polynomials p_j and q_j with the use of Theorem 1.

In this way, we have reduced the number of variables from 14 to 12 and the degrees of the polynomials to 2, which significantly speeds up the SDP solver. Practical experiments showed that GloptiPoly is now able to compute the IK task for more poses with the same relaxation order than by the naïve approach used before, see Fig. 1b. Again, this approach performs best for the highest possible power of the objective function, i.e., $n = 4$ (see Tab. II). Now only 0.02 % of poses failed to be solved on the same dataset as in Section VII-A2.

4) Results: We have computed the translation and rotation error of the desired poses w.r.t. the poses computed by the forward kinematics task for the joint angles found by the proposed method. Their comparison between the proposed approaches for various values of the relaxation order r and the power of the objective function n can be found in Tab. II.

TABLE III: Overview of the execution times and accuracy of the presented methods evaluated on the randomly generated generic manipulator.

Method	Avg. execution time [s]		Mean error [mm, deg]		Failure
	Red. step	GloptiPoly	Translation	Rotation	rate [%]
Naïve	—	16.7	$5.41 \cdot 10^{-4}$	$4.68 \cdot 10^{-4}$	51.0
Sym. red. step	8127	12.6	$5.07 \cdot 10^{-5}$	$1.81 \cdot 10^{-5}$	0

The drop of precision between the naïve approach and the approach with the symbolic reduction step is mainly due to the approximation of the end-effector pose and the parameters of the manipulator. The histograms of the error using the best value $r = 2$ and $n = 4$ can be seen in Fig. 1c.

In Tab. II, we can see that it is worth taking higher powers of the objective function as explained in Section VI-D. We have shown that with its increasing power, both the failure rate and the execution time improve. Both methods perform best for the values $r = 2$ and $n = 4$. The overall comparison of the methods for these values can be seen in Tab. I.

For practical applications, the value of the execution time is essential. In Fig. 2, we show the histograms of the execution time of the online phase of GloptiPoly as well as of the symbolic reduction to degree 2 polynomials. We observe that our execution times are comparable to [15] when using off-the-shelf POP and GB computation tools. We plan to develop optimized solvers leading to considerable speedup, as it was done in computer vision [32].

B. A generic 7DOF serial manipulator

Here we show that we are able to solve the IK task of a randomly generated fully generic 7DOF serial manipulator.

We have randomly generated the D-H parameters of the manipulator. The values of d_i and a_i were generated as integers from 10 to 100 mm. We have set the allowance interval for the joint angles to $\langle -3, 3 \rangle$ rad. From the same interval, we have generated the angles α_i, for which we have found rational representations of their sines and cosines by the same approach as in Section VI-E with $\kappa = 1$.

The dataset of poses consists of 100 randomly generated poses, which were rounded to the rational representation as described in Section VI-E with $\kappa = 1$.

In this experiment, we have rounded the D-H parameters of the manipulator, and the end-effector poses to rational numbers in advance. The rounding with higher values of κ would lead to very long coefficients in the GB computation, which would significantly increase the execution time. However, doing the rounding in the symbolic method directly with $\kappa = 1$ would mean that we would be computing the IK problem for very different (rounded rational representation) kinematic parameters and end-effector poses than we are evaluating the errors for (the original floating-point representation). Therefore, we have decided to use a dataset with rational representations of the end-effector poses.

We again compare two approaches: direct solving of the IK task by the POP solver and symbolically reducing the degree of the polynomials and then solving it by the POP solver. Again, we set the weights to $w_i = \frac{1}{7}$, the preferred values

of the joint angles $\hat{\theta}_i$ to zero, relaxation order to $r = 2$, and the power of the objective function to $n = 4$. The average execution time, failure rate, and translation and rotation error for both approaches can be found in Tab. III.

The naïve approach is unable to solve 51.0 % of the end-effector poses. The approach with the symbolic reduction step is able to solve all poses from the dataset, but it takes more than 2 hours to symbolically preprocess the equations. In this case, the best way would be to try the naïve approach first, which takes seconds but has a high failure rate. And if it fails, then one would need to preprocess the equations by symbolic computation. This approach significantly reduces the computation time but still keeps the failure rate near zero.

VIII. Conclusions

We presented the first practical method for globally solving the 7DOF IK problem with a polynomial objective function. Our solution is accurate and can solve/decide infeasibility in 99.9 % of 10 000 cases tested on the *KUKA LBR IIWA* manipulator. We have shown that the method is general and, therefore, can be used to solve the IK problem for a generic 7DOF serial revolute manipulator. The code is open-sourced at https://github.com/PavelTrutman/Global-7DOF-IKT.

For future work, we consider two interesting directions. First, when the POP solver detects infeasibility, it would be desirable to return its certificate. It can be either numerical (obtained by solving the moment-SOS hierarchy with an SDP solver) or symbolic (obtained by the GB method). It can be obtained, e.g., by computing an SOS representation for the polynomial -1 on the quadratic module corresponding to the feasible set. See, e.g., [33] in the specific case of certifying emptiness of spectrahedra (SDP feasibility set).

Secondly, it would be interesting to exploit the specific structure of the POP studied in this paper to prove the exactness of the first or the second SDP relaxation in the moment-SOS hierarchy, i.e., that solving this relaxation always solves the original POP. For Euclidean distance POP arising in computer vision, this was achieved in [34] by arguing on the curvature properties of the Lagrangian and its SOS representation in the quadratic module.

References

[1] J. Shigley and J. Uicker, *Theory of machines and mechanisms*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1980.

[2] R. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. Springer, 2007.

[3] M. Raghavan and B. Roth, "Inverse kinematics of the general 6r manipulator and related linkages," 1993.

[4] ——, "Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators," 1995.

[5] D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2015.

[6] C. W. Wampler, A. P. Morgan, and A. J. Sommese, "Numerical continuation methods for solving polynomial systems arising in kinematics," 1990.

[7] E. W. Mayr and A. R. Meyer, "The complexity of the word problems for commutative semigroups and polynomial ideals," *Advances in Mathematics*, vol. 46, no. 3, pp. 305 – 329, 1982.

[8] D. Manocha and J. F. Canny, "Efficient inverse kinematics for general 6r manipulators," *IEEE Trans. Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.

[9] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Pittsburgh, PA, USA, 2010, aAI3448143.

[10] I. Kuhlemann, A. Schweikard, P. Jauer, and F. Ernst, "Robust inverse kinematics by configuration control for redundant manipulators with seven dof," in *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2016, pp. 49–55.

[11] J. B. Lasserre, *An introduction to polynomial and semi-algebraic optimization*. Cambridge University Press, 2015, vol. 52.

[12] ——, "Global optimization with polynomials and the problem of moments," *SIAM Journal on optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[13] M. Raghaven and B. Roth, "Kinematic analysis of the 6r manipulator of general geometry," in *The Fifth International Symposium on Robotics Research*. Cambridge, MA, USA: MIT Press, 1990, pp. 263–269.

[14] D. Lazard, "Generalized stewart platform: How to compute with rigid motions," 1993.

[15] H. Dai, G. Izatt, and R. Tedrake, "Global inverse kinematics via mixed-integer convex optimization," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1420–1441, 2019.

[16] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014.

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

[18] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices," *Journal of applied mechanics*, vol. 77, no. 2, pp. 215–221, 1955.

[19] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," in *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2010, pp. 1668–1674.

[20] A. Goldenberg, B. Benhabib, and R. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 14–20, 1985.

[21] F. Kahl and R. Hartley, "Multiple-view geometry under the L_∞-norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1603–1617, 2008.

[22] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana University Mathematics Journal*, vol. 42, no. 3, pp. 969–984, 1993.

[23] D. Henrion and J.-B. Lasserre, "Gloptipoly: Global optimization over polynomials with matlab and sedumi," *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 2, pp. 165–194, 2003.

[24] ——, "Detecting global optimality and extracting solutions in gloptipoly," in *Positive polynomials in control*. Springer, 2005, pp. 293–310.

[25] J.-C. Faugère, "A new efficient algorithm for computing gröbner bases (f4)," *Journal of Pure and Applied Algebra*, vol. 139, no. 1, 1999.

[26] J. C. Faugère, "A new efficient algorithm for computing gröbner bases without reduction to zero (f5)," in *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ser. ISSAC '02. New York, NY, USA: ACM, 2002, pp. 75–83.

[27] M. Kalkbrener, "On the stability of gröbner bases under specializations," *Journal of Symbolic Computation*, vol. 24, no. 1, pp. 51–58, 1997.

[28] B. Pascual-Escudero, A. Nayak, S. Briot, O. Kermorgant, P. Martinet, M. Safey El Din, and F. Chaumette, "Complete Singularity Analysis for the Perspective-Four-Point Problem," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1217–1237, Apr. 2021.

[29] J. Canny, B. Donald, and E. K. Ressler, "A rational rotation method for robust geometric algorithms," in *Proceedings of the eighth annual symposium on Computational geometry*, 1992, pp. 251–260.

[30] V. J. Milenkovic and V. Milenkovic, "Rational orthogonal approximations to orthogonal matrices," *Computational Geometry*, vol. 7, no. 1-2, pp. 25–35, 1997.

[31] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 8.0.*, 2016.

[32] V. Larsson, M. Oskarsson, K. Åström, A. Wallis, Z. Kukelova, and T. Pajdla, "Beyond grobner bases: Basis selection for minimal solvers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3945–3954.

[33] I. Klep and M. Schweighofer, "An exact duality theory for semidefinite programming based on sums of squares," *Mathematics of Operations Research*, vol. 38, no. 3, pp. 569–590, 2013.

[34] C. Aholt, S. Agarwal, and R. Thomas, "A QCQP approach to triangulation," in *European Conference on Computer Vision*. Springer, 2012, pp. 654–667.