Efficient Polynomial Sum-Of-Squares Programming for Planar Robotic Arms

Daniel Keren[1], Amit Shahar[1], Roi Poranne[1]

***Abstract*— Collision-avoiding motion planning for articulated robotic arms is one of the major challenges in robotics. The difficulty of the problem arises from its high dimensionality and the intricate geometry of the feasible space. Our goal is to seek large convex domains in configuration space, which contain no obstacles. In these domains, simple linear trajectories are guaranteed to be collision free, and can be leveraged for further optimization. To find such domains, practitioners have harnessed a methodology known as Sum-Of-Squares (SOS) Programming. SOS programs, however, are notorious for their poor scaling properties, which makes it challenging to employ them for complex problems. In this paper, we explore a simple formulation for a two-dimensional arm, which results in smaller SOS programs than previous suggested ones. We show that this formulation can express a variety of scenarios in a unified manner.**

I. Introduction

Finding collision free trajectories for robotic arms is a fundamental problem in robotics. A wide variety of strategies are available, ranging from gradient-based trajectory optimization methods to random sampling techniques such as the Rapidly Exploring Random Tree approach. One notable approach aims to identify large convex domains in Configuration space (C-space) that are devoid of obstacles. A linear path between two points in such domains is guaranteed to be collision-free. By combining several such domains, it is possible to find paths between any two points. Furthermore, these domains can be leveraged in order to find initial trajectories for further optimization or for a more efficient sampling.

Our goal is to devise a strategy for finding large convex domains in C-space. Towards this end, researchers have previously formulated the problem as a Sum-Of-Squares [1], [2] (SOS) Program. In its most basic form, an SOS program asks whether a given polynomial can be expressed as a sum-of-squares of polynomials. The problem was conceived by Minkowski in his dissertation and inspired Hilbert's 17th problem. The SOS paradigm has later been expanded and led to a sequence of theorems known as *Positivstellensatz* (Psatz), which today form the foundations of SOS optimization. In particular, SOS programs have been found useful for determining whether objects along a trajectory collide. This is often referred to as a *certificate*, that is, an SOS decomposition of a certain polynomial (or lack thereof) provide a certificate of no collision.

While SOS programs provide a powerful tool for path planning, they are also notorious for their complexity and somewhat elusive nature. First, the required degree of the polynomial required for the certificate is generally unknown, although there exists a loose upper bound [3]. That means that while a low-degree polynomial might not be able to provide the certificate, a high-degree polynomial might. Second, the complexity of the problem scales like n^d, where n is the dimension (number of variables) of the polynomial, and d its degree. Hence, certificates with lower degrees are desirable.

Finding the minimal formulation is an open problem [3]. The focus of this manuscript is the simple case of articulated arms in two dimensions, as it more clearly reveals the fundamentals of this problem. We propose a methodology that requires fewer variables and can address polygonal and polynomial free-form obstacles, which are not necessarily convex. Given a robot configuration and a set of obstacles, we find the maximal ranges of motion for all joints such that no collisions occur. We evaluate our approach experimentally in simulation and demonstrate our favorable scaling properties.

To summarize, our formulation provides the following:

1) Requires only low degree polynomials.
2) Returns a collision-free convex region in joint space.
3) Can treat non-convex polygons and non-convex, free-form polynomial obstacles in a unified manner.

II. Related Work

Methods based on trajectory optimization, or random sampling, are among the more popular choices for motion planning problems. Indeed, they are generally simple to implement and provide satisfactory results. However, both have limitations. Gradient-based optimization methods [4] require an initial guess, and in some cases, this initial guess must be collision-free. To obtain a valid initialization, sampling-based methods are sometimes employed. These methods, e.g. random trees [5] and roadmaps [6], struggle in high dimensions and narrow C-spaces, where chances of accepting samples are low. This issue can be overcome by identifying and utilizing large *convex* regions in C-space. The approach can be traced back to the pioneering work of Lozano-Perez [7]. Sampling from a convex region is simpler and, in addition, a linear path from the current configuration to the sampled one is guaranteed to be collision-free. Finding convex regions in the C-space is, however, a daunting task. The problem can be approached from a geometric point of view, as in [8]–[10]. Another alternative, which we study in this paper, relies on SOS Programming, and we discuss a few notable examples here.

[1] The authors are with the Department of Computer Science, University of Haifa, Haifa, Israel. (dkeren,amit.uni17,roi.poranne)@ds.haifa.ac.il

SOS programming has been used in multiple contexts within robotics, such as inverse kinematics [11] and Lyapunov stability [12], [13]. It has also become a powerful tool in optimization and geometry processing [3], [14]–[18]. In recent years, SOS programming has been applied directly to collision avoidance [19], [20], with two recent papers [21], [22] particularly relevant to our work. Both papers formulate the SOS problem in a similar fashion, and we also follow an SOS-based paradigm, as described in detail in Sec. III. Briefly, [21] solves a simple problem: to find a maximal *square* region in the C-space for a two-link robot, with a linear obstacle, i.e. a *wall*. The paper only considers the end-effector for collisions, and in order to obtain a polynomial expression, the trigonometric functions in the forward kinematics formula are approximated by their Taylor expansions. In contrast, [22] tackles a more general problem, with more complex arms and obstacles in 3 dimensions. The algorithm also seeks a more general, *polyhedral* region in C-space. Following in the spirit of these papers, we propose a formulation that is more accurate and generic than [21], and more compact and direct than [22], in particular for the 2D case.

III. Method

A. Preliminaries

The SOS paradigm is a powerful technique for inference in polynomial systems [3], [23]. We begin with a simple, illustrative example to motivate the use of SOS programming for collision avoidance. Let $\mathcal{O}$ denote a set in the plane implicitly defined by a function p, i.e. $\mathcal{O} = \{(x,y)|p(x,y) \leq 0\}$, and let $\mathcal{C} = \{(g_1(t), g_2(t))\}$ denote a plane curve with parameter t. How can we check if $\mathcal{C}$ and $\mathcal{O}$ intersect, i.e., if $\mathcal{C} \cap \mathcal{O} = \emptyset$? One approach is to substitute $\mathcal{C}$ in $\mathcal{O}$, which yields the function $f(t) = p(g_1(t), g_2(t))$. If $f(t) > 0$ for every t, then the intersection is necessarily empty; see a schematic depiction in Fig. 1. A systematic approach to prove that $f(t)$ is positive for general functions is, of course, intractable. What if we could assume that p, g_1, g_2 are all polynomials? This case is still too broad, but it turns out that there is a weaker but much simpler condition; If

$$f(t) = \sum p_i^2(t),$$

that is, if $f(t)$ is a sum-of-squares (SOS) of polynomials, then clearly $f(t) \geq 0$. In practice, while testing polynomial positivity is NP-hard [24], testing whether a polynomial is SOS can be done by solving a Semi-Definite Program (SDP) [25]. If the problem is feasible, i.e. a feasible solution exists, then an SOS decomposition can be derived from it. However, the solution itself is rarely sought; it is its existence that truly matters.

Our focus is set on a class of problems slightly more generic. To illustrate, assume now that $\mathcal{C}$ is limited to $t \in [0,1]$. How can we incorporate this condition into the SOS framework? If, for example, we show that there exist SOS polynomials $\sigma_0(t), \sigma_1(t)$ such that

$$f(t) = \sigma_0(t) + t(1-t)\sigma_1(t), \tag{1}$$

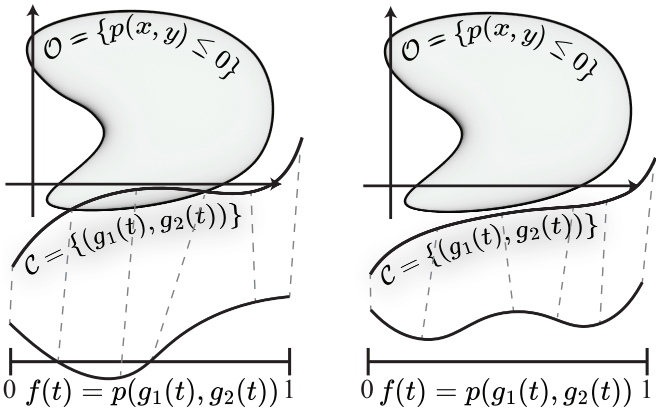

$\mathcal{O} = \{p(x,y) \leq 0\}$
$\mathcal{C} = \{(g_1(t), g_2(t))\}$
0 $f(t) = p(g_1(t), g_2(t))$ 1
$\mathcal{O} = \{p(x,y) \leq 0\}$
$\mathcal{C} = \{(g_1(t), g_2(t))\}$
0 $f(t) = p(g_1(t), g_2(t))$ 1

Fig. 1: Given an implicit domain $p(x,y) \leq 0$ and a curve $(g_1(t), g_2(t))$, finding if they intersect is as "simple" as determining whether $p(g_1(t), g_2(t))$ is always positive or not. We rely on SOS programming to achieve that.

then $f(t)$, although not necessarily positive for all t, is positive for $0 \leq t \leq 1$. Identities such as (1) are referred to as *Positivestellensatz* (PSatz) or, *certificates of positivity*. The question whether such σ_i's of certain degrees exist can also be transformed into an SDP. A well-known result, sometimes known as Putinar's PSatz, states that, under certain conditions, such certificates exist for every polynomial that is positive on a certain set.

Theorem 1 (Putinar, [26])**.** *Let a set W in Euclidean space $\mathbb{R}^m$ be defined by $W = \{x \mid f_i(x) \geq 0,\ g_j(x) = 0\}$, where f_i, g_j are polynomials and $x = (x_1, \ldots, x_m)$. Define the* quadratic module *associated with W by*

$$Q = \{\sigma_0 + \sum_i \sigma_i f_i + \sum_j \pi_j g_j\},$$

where σ_i are SOS polynomials, and π_j are arbitrary polynomials. If a positive real number L exists such that $L - \sum_{k=1}^m x_k^2 \in Q$ (a module satisfying this property is called Archimedean*), and $p(x) > 0$ on W, then $p(x) \in Q$.*

Intuitively, the theorem asserts that $p(x) > 0$ on W iff it can be written as a combination of the polynomials which define Q. In that case, we can say that we obtained a *certificate of positivity*. As mentioned, finding a solution can be done via SDP. Note that the minimal required degrees of σ_i and π_j is generally not known, although loose upper bounds exist [3]. This means that if we have found the SDP for a certain degree to be infeasible, an SDP derived from a higher degree might still be feasible. Thus, a common strategy is to begin testing with low order polynomials, which are faster to solve, and, if feasibility testing fails, increase the degree [22], [23]. We show in our experiments that with our formulation, the lowest possible degree polynomials are sufficient to determine a maximal region in C-space.

B. Overview

Our setting involves a single robotic arm in 2D, with n revolute joints, in an environment scattered with obstacles. Following [21], we parameterize the state of the robot with *absolute* angles. This somewhat unusual choice is due to forward kinematics being additive in this case, instead of multiplicative, which in turn has computational benefits (Sec. III-C). We denote the initial state by $\Theta^0 = (\theta_1^0, \dots, \theta_n^0)$. Given a collision-free state Θ_0, we seek the largest collision-free range of motion, in terms of joint angles. More precisely, we find the largest δ such that for any $\theta_i \in [\theta_i^0 - \delta, \theta_i^0 + \delta]$, the state $\Theta = (\theta_1, \dots, \theta_n)$ is also collision-free (Fig. 2). The rest of the paper is organized as follows. in Sec. III-D, we first treat polynomial obstacles, then convex polygonal objects in Sec. III-G, and finally line segments (which include the case of non-convex polygonal obstacles). In Sec. IV we evaluate our approach experimentally and discuss its performance. Finally, in Sec. V we offer conclusions and a glimpse of future work.

C. State representation

A first step towards an SOS formulation is to express any point on the arm, for any set of joint angles Θ, as a polynomial. We refer to joints and end-effector $\mathbf{R}_1, \dots, \mathbf{R}_{n+1}$, and to the links by $\mathbf{L}_i$, $i = 1, \dots, n$. For simplicity, we assume that all link lengths are equal to 1 and that the root joint $\mathbf{R}_1$ is positioned at the origin. In absolute angles, the position of $\mathbf{R}_j$, in its initial state, is simply given by

$$\sum_{i=1}^{j-1} (\cos(\theta_i^0), \sin(\theta_i^0)).$$

We note that with relative angles, the position is a more complex expression that involves products. Next, we make a change of variables:

$$c_i = \cos(\theta_i), \quad s_i = \sin(\theta_i), \quad \text{where} \quad c_i^2 + s_i^2 = 1, \tag{2}$$

and denote $c_i^0 = \cos(\theta_i^0)$, $s_i^0 = \sin(\theta_i^0)$. After rotating of every joint by θ_i we obtain:

$$\mathbf{R}_j(\mathbf{c}_{j-1}, \mathbf{s}_{j-1}) = \sum_{i=1}^{j-1} (c_i c_i^0 - s_i s_i^0, c_i s_i^0 + s_i c_i^0) = \sum_{i=1}^{j-1} \mathbf{p}_i(c_i, s_i),$$

where $\mathbf{c}_j = (c_1, \dots, c_j)$ and similarly for $\mathbf{s}_i$, and $\mathbf{p}_i(c_i, s_i)$ is defined accordingly. Any point on link $\mathbf{L}_j$ can be written as

$$\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j) = \mathbf{R}_j(\mathbf{c}_j, \mathbf{s}_j) + t_j \mathbf{p}_j(c_j, s_j), t_j \in [0, 1],$$

where we introduced the parameter t_j for that link. We note that the expression $\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j) \in \mathbb{R}^2$ involves two polynomials in $\mathbf{c}_j, \mathbf{s}_j, t_j$, with $2j+1$ variables overall, where the only non-linear term is the last *bilinear* one (note that t_j multiplies c_j and s_j).

Remark. We note that different representations have been proposed recently. We discuss this in Sec. III-F.

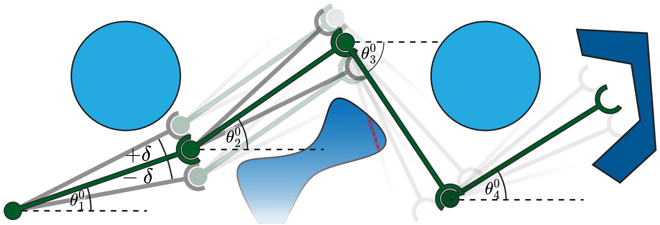


Fig. 2: An overview of our setting. Given a robot arm in a certain configuration, we seek the maximal range of motion δ for all joints, such that no collision occurs. The illustration shows many possible configurations of the arm as shadows. We describe the state in absolute angles, in order to obtain a more compact problem (Sec. III-C).

D. Polynomial obstacles

As in the schematic description in Sec. III-A, we define an obstacle implicitly by

$$C = \{\mathbf{x} = (x, y) \mid q(\mathbf{x}) \le 0\},$$

where $\mathbf{x} = (x, y)$ and $q(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ is a polynomial of degree d. Note that this representation allows free-form, *non-convex*, obstacles. A link $\mathbf{L}_j$ intersects C iff $q(\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j)) < 0$ for any $\mathbf{c}_j, \mathbf{s}_j, t_j$. Conversely, $\mathbf{L}_j$ never intersects C if

$$q\big(\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j)\big) > 0, \quad \forall \mathbf{c}_j, \mathbf{s}_j, t_j.$$

Again, note that $q\big(\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j)\big)$ is a polynomial, with degree $2d$ that originates in the single bilinear term of $\mathbf{L}_j(\mathbf{c}_j, \mathbf{s}_j, t_j)$.

Recalling our goal of finding the largest collision-free δ, the constraint we impose on θ_i is $|\theta_i| < \delta$. Given our change of variables in (2), we can write it equivalently as $T - s_i^2 > 0$, where $T = \sin^2(\delta)$. Moreover, we require that $c_i \ge 0$, so that rotations are restricted to $[-\pi/2, \pi/2]$; this is necessary, in order to maintain a range in which $\sin^2(\alpha)$ is monotonic in $|\alpha|$. Together with the constraints in (1) and (2), Putinar's PSatz for this problem is

$$\begin{aligned} q(\mathbf{L}_j) = {} & \sigma_0 + (1 - t_j) t_j \sigma_1 \\ & + \sum_{i=1}^{j-1} \left[(T - s_i^2) \sigma_2^i + c_i \sigma_3^i + (c_i^2 + s_i^2 - 1) \pi^i \right], \end{aligned} \tag{3}$$

where $\sigma_0, \sigma_1, \sigma_2^i, \sigma_3^i$ are SOS, and π^i are general polynomials, all in $\mathbf{c}_j, \mathbf{s}_j, t_j$. We chose the degree of σ_0 as $2d$, and of all the other polynomials as $2d - 2$.

Equating the coefficients of all monomials on the left and right hand sides of (3) results in a set of constraints $\mathcal{C}_j(\alpha) = 0$, where α is a vector containing all coefficients. With this, our optimization problem can finally be presented:

$$\begin{aligned} \max_{\alpha} \quad & T \\ \text{s.t.} \quad & \mathcal{C}_j(\alpha) = 0, \quad j = 1, \dots, n. \end{aligned} \tag{4}$$

See Sec. IV for a more detailed analysis of the cardinality of the coefficient set.

Lastly, we prove the problem at hand satisfies the property required by Putinar's Theorem.

Lemma 1. *The module corresponding to our SDP problem is Archimedean.*

Proof. First, we replace the parameters t_i over the range $[0, 1]$, by parameters over the range $[-1, 1]$. This replaces the condition $t_i(1 - t_i) \geq 0$ with $1 - t_i^2 \geq 0$. Next, We must show that there is a real number $L > 0$, SOS polynomials $\sigma_0, \sigma_1^i, \sigma_2^i, \sigma_3^i$, and general polynomials π^i, such that

$$\begin{aligned} L - \sum_i (t_i^2 + c_i^2 + s_i^2) = \sigma_0 + \sum_i (1 - t_i^2)\sigma_1^i + \\ + \sum_i \left[(T - s_i^2)\sigma_2^i + c_i \sigma_3^i + (c_i^2 + s_i^2 - 1)\pi^i \right] \end{aligned} \tag{5}$$

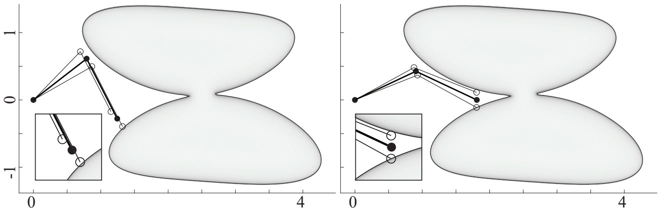
Choosing $L = 2n, \sigma_0 = \sigma_2^i = \sigma_2^i = 0, \sigma_1^i = 1, \pi^i = -1$, where n is the total number of summands, we obtain equality, completing the proof. □

We note that multiple polynomial obstacles can be addressed by adding more constraints in the form of 3. Indeed, since the non-intersection requirement is posed as a feasibility problem, a union of obstacles leads to a list of constraints. For instance, see in Fig. 5 an example with multiple circular obstacles.

E. Optimization

SOS Problems are generally solved via SDP [3], [23]. However, (4) is not truly in the correct form. The requirement is that the polynomials in (3) must be linear in their coefficients, which are the decision variables of the problem. This is not true for the term $(T - s_i^2)\sigma_2^i$, since both T and the coefficient of σ_2^i are variables, and hence this term is quadratic. To mitigate this, an option is to optimize T using a *binary search*. Essentially, we are looking for the maximal T such that (3) is feasible, by attempting to solve multiple SDPs. This is an effective approach, which we adopt in most experiments. Alternatively, we also experimented with removing σ_2^i from (3), to obtain a true SOS problem, which can be solved directly via a single SDP. The caveat is that the solution to this SDP may underestimate the maximal value, the reason being that due to the removal of σ_2^i, we lack the degrees of freedom it provides. We evaluate the differences between these approaches in Sec. IV.

The question of the degree necessary to set for the different SOS terms in (3) is important. Higher degrees provide more degrees of freedom, but may be redundant. As a rule of thumb, one should set the degree of each SOS polynomial in each term such that the degree of the product matches that of the l.h.s. of (3) as much as possible, noting the degree of SOS polynomials is always even. To illustrate, we provide an example of a 2-link arm and a 4-th degree non-convex polynomial obstacle in Fig. 3. $q(\mathbf{L}_j)$ is 8-th degree, and therefore we set the degrees of $\sigma_0, \sigma_1, \sigma_2^i, \sigma_3^i$ and π^i to 8,6,6,6,6, respectively. Note that the exception here is the degree of the term $c_i\sigma_3^i$, which is 7. While we could have set it to 9, we did not observe any limitations experimentally. In Sec. IV we also report on the influence of the degree of π_i on the precision of the result.



$$\{(x, y) | x^4 - 10.0x^3 + 37.5x^2 + 2.187xy - 62.65625x + 9.378906y^4 \\ - 2.679688y^3 - 15.02539y^2 - 3.294922y + 39.37134 \leq 0\}$$

Fig. 3: A free form, non-convex obstacle, defined by an implicit quartic polynomial. The initial configuration is shown in bold, with two other configurations that are on the boundary of the collision free C-space region.

F. Alternative representations

In order to apply SOS techniques to motion planning, it is necessary to represent the trigonometric functions by polynomials. We represent the sine and cosine functions as variables s, c, with the added condition $c^2 + s^2 = 1$. In [21], a truncated Taylor expansion was used, which is inaccurate (for large angles). In [22], a rational parameterization was used: $\sin(\alpha) = \frac{2t}{1+t^2}, \cos(\alpha) = \frac{1-t^2}{1+t^2}$, where $t = \tan(\alpha/2)$. While requiring less variables, this representation suffers from a "blow up" in the degrees of the polynomials required for the SOS program, since the $1 + t^2$ factor must be canceled out by multiplying all identities by the highest degree at which it appears in the denominators (as SOS techniques cannot directly handle rational functions). Since a polynomial of degree d in n variables has $\binom{n+d}{d}$ coefficients, we opted for a parameterization with a smaller d.

For example, assume we have to handle 10 rotation angles, and that the polynomial describing the problem is of degree 2. Our approach yields a quadratic in 20 variables, with 231 coefficients, and 10 quadratic constraints. However, the rational parameterization offered in [22] yields a rational function with denominators $(1 + t_i^2)^2, i = 1 \ldots 10$, all of which must be canceled out; hence, it is necessary to multiply the corresponding identities by a polynomial of degree $d = 40$, yielding polynomial identities with $1.03 \cdot 10^{10}$ coefficients.

G. Convex Polygonal obstacles

In Sec. III-D we mentioned that the approach readily works for unions of obstacles. Convex polygonal obstacles, however, are defined by *intersection* of half-planes, i.e. implicit polynomials of degree 1. In this section we describe a complementary approach, suitable for this case.

To make the exposition simpler we begin with the conditions for non-intersection of a fixed segment with a convex polygon $\mathcal{P}$. We describe $\mathcal{P}$ by intersection of half-planes,

$$\mathcal{P} = \bigcap_{i=1}^{n} \{(x, y) \mid L_i(x, y) \geq 0\}$$

and denote the endpoints of the segment by $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$. Consider the following feasibility problem:
There exist SOS polynomials $\sigma_0 \ldots \sigma_{n+1}$ in the variables x, y, t, such that the following equality holds:

$$\begin{aligned} \sigma_0 + \sum_{i=1}^{n} \sigma_i L_i(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \\ + \sigma_{n+1} t(1-t) + \varepsilon = 0, \end{aligned} \tag{6}$$

where ε is a positive constant, which can be made arbitrarily small. Suppose the feasibility problem in (6) can be satisfied, and let $\mathbf{x}$ be a point inside $\mathcal{P}$. Since $L_i(\mathbf{x}) \geq 0$ for every i, $\sigma_i \geq 0$ everywhere, and $t(1-t) \geq 0$ for the t range in question (the interval $[0,1]$), it follows that $\mathbf{x}$ cannot be on the segment, since the sum of positive numbers and ε cannot be equal to 0. Hence, (6) defines a "certificate for non-intersection" of the segment and the polygon. To guarantee non-intersection of a link L_j with $\mathcal{P}$, we proceed as in Section III-D, i.e handle the link as a segment whose start and end points are determined by the rotation angles of itself and the previous links. See an example in Fig. 4.

Remark. Non convex polygons can be treated by dividing them into several convex polygons and treating them as a union. Alternatively, they can also be treated as independent edges. To this end, we require an SOS formulation for edge-edge intersection, which we briefly sketch here. Let the segments be defined by their endpoints, p_1, p_2 and q_1, q_2. The certificate for non-intersection should imply the condition that for every $0 \leq s, t \leq 1$, it holds that $tp_1 + (1-t)p_2 \neq sq_1 + (1-s)q_2$. We therefore define the following feasibility problem, which obviously cannot hold if the segments intersect:

$$||tp_1 + (1-t)p_2 - sq_1 - (1-s)q_2||^2 - \sigma_0 - \sigma_1 t(1-t) - \sigma_2 s(1-s) - \varepsilon = 0, \tag{7}$$

where, again, ε is an arbitrarily small positive constant. To check intersection of a segment with a non-convex polygon, we simply check it vs. each of the polygon's edges. In fact, this formulation allows any arbitrary set of scattered line-segments, not necessarily polygons. Again, the generalization to links follows Sec. III-C. The drawback of this approach in comparison to the one above for convex polygons, is that the required degrees of the SOS polynomials are higher. For example s_0 is required to be of degree 4 instead of 2 for convex polygons.

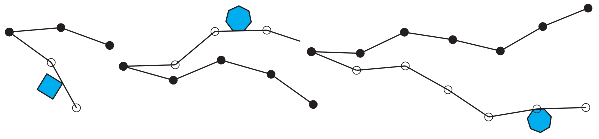
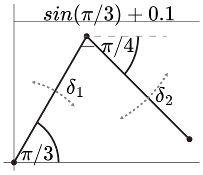


Fig. 4: An example showing various arms and polygonal obstacles. The initial configurations are shown with bold joints, and maximal ones touching the polygonal obstacles are shown with hollow joints

H. Anisotropic regions in C-space



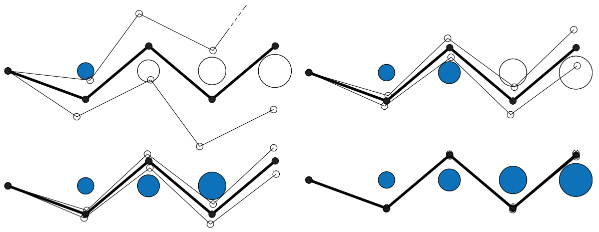


Fig. 5: An example showing multiple disc obstacles. The hollow discs are not present and are only shown for comparison. Treating multiple obstacles is straightforward (Sec. III-D).

We end this section with a preliminary discussion regarding anisotropic collision-free regions in C-space. So far, we have described how to obtain a maximal, symmetric *hypercube* in C-space. Clearly though, hypercubes are suboptimal in comparison to more general shapes [22]. For example, in the configuration in the inset, we may try and obtain a set with a larger volume, reflecting the fact that the second link has considerably more rotational freedom than the first. A natural extension is a hyperbox in C-space. Denoting the height of the "ceiling" obstacle by w, the rotational "slacks" of the first(second) link by $\delta_1(\delta_2)$, the cosines and sines of the initial angles of the first and second links by $c_{10}, s_{10}, c_{20}, s_{20}$, and $\cos(\delta_1) = c_1, \sin(\delta_1) = s_1, \cos(\delta_2) = c_2, \sin(\delta_2) = s_2$, and following as in Sec. III-C, we obtain the following intriguing optimization problem

$$\begin{aligned}
&\text{Maximize } T_1T_2 \text{ such that the following are feasible}\\
&\text{I) } w-(c_{01}s_1+s_{01}c_1) = p_1 + p_2(T_1 - s_1^2) + p_3c_1 + \\
&q_1(c_1^2+s_1^2-1)\\
&\text{II) } w-(c_{01}s_1+s_{01}c_1)-(c_{02}s_2+s_{02}c_2) = p_4 + \\
&p_5(T_1-s_1^2) + p_6(T_2-s_2^2) + p_7c_2 + \\
&q_2(c_1^2+s_1^2-1) + q_3(c_2^2+s_2^2-1),
\end{aligned} \tag{8}$$

where the p_i are SOS polynomials in the variables c_1, s_1, c_2, s_2, and the q_j are arbitrary polynomials in these variables.

The problem in (8) cannot be solved with convex optimization tools, and its solution can be approximated by iterative "alternating" between two convex problems, as in [22]. However, if we set in (8) $p_2 = p_5 = p_6 = 1$, then, while not obtaining the most general solution, we can use the following result to obtain a convex problem (i.e. minimizing a convex function over a convex region):

Lemma 2. *When setting $p_2 = p_5 = p_6 = 1$, the set of pairs (T_1, T_2) for which the problems I,II in (8) are feasible, is convex.*

We omit the proof, which proceeds rather directly by noting that the set of all polynomials, as well as SOS polynomials, are convex. Then, we use the standard trick of minimizing $-(\log(T_1) + \log(T_2))$, which is convex, over the convex region guaranteed by Lemma 2.

For the two-link problem above, the symmetric (square) shape in C-space found by the algorithm in Section III-G is of area 0.0043, the optimal non-symmetric (rectangle) found by exhaustive optimization 0.029, and by the convex optimization described here, 0.024. The method can readily be extended to any number of links. Initial experiments suggest that its improvement over the method in Section III-G is better when the degree of the implicit polynomials describing the obstacles is lower.

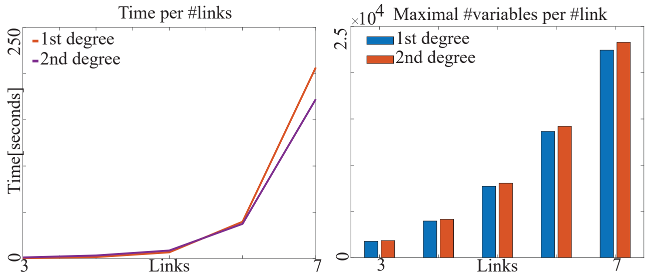


Fig. 6: Time to solve and number of variables used to describe the problem for an arm with n links and a single round obstacle, using 1st and 2nd degree polynomials for π^j.

IV. Results

We evaluate our formulation in terms of performance, complexity and accuracy, in the scenarios described in section III-D, on a i7-9750H machine clocked at 2.6GhZ with 16GB of RAM. We implemented our solver in Matlab, and SOSTOOLS [27] to implement the SOS program, and SeDuMi [28] as the underlying SDP solver. SOSTOOLS is a Matlab toolbox that allows describing SOS programs in high-level, and converts them to an SDP problem. We use the default parameters everywhere. The code is included with the submission and will be open sourced when published.

Evaluating the structure of the SDP derived from our SOS formulation is not straightforward. In this paper we focus on evaluating the complexity of the problem in term of the number of decision variables, e.g. the number of polynomial coefficients. This indirectly corresponds to complexity and timing, i.e. a problem with fewer variables will generally be solved faster. To obtain a positivity certificate, when it exists, we generally select the highest degree polynomials necessary, which will require the most time to compute. We can opt to relax the problem, and use lower degree polynomials. We illustrate this with two examples involving a polygonal obstacle and a disc obstacle. For both, we use only second degree SOS polynomial, except σ_0 for the disc case, since that is the minimal degree required to obtain a feasible solution. For the general polynomial π^j, we chose either 1st or 2nd degree and compared the results. It is expected that the optimal δ for the 1st degree case should underestimate the maximum, and be below the δ for the 2nd degree case. On the other hand, since fewer variables are present, performance should improve timewise. We show the difference in the value obtained in Fig. 8, which also includes a comparison to the direct optimization approach from Sec. III-E, which requires no binary search for the maximal T, hence is faster. The results were obtained by randomly placing obstacles and computing the optimal δ with each method. This was done 100 times. Next, the results were organized in deciles, and the averages are shown as a bar plot. Additional difference in time for the disc case are shown in Fig. 6, and in Fig. 7 for the polygonal case. We have also included a demonstration of the scaling of the problem for the polygonal case, shown in Fig. 7 as well.

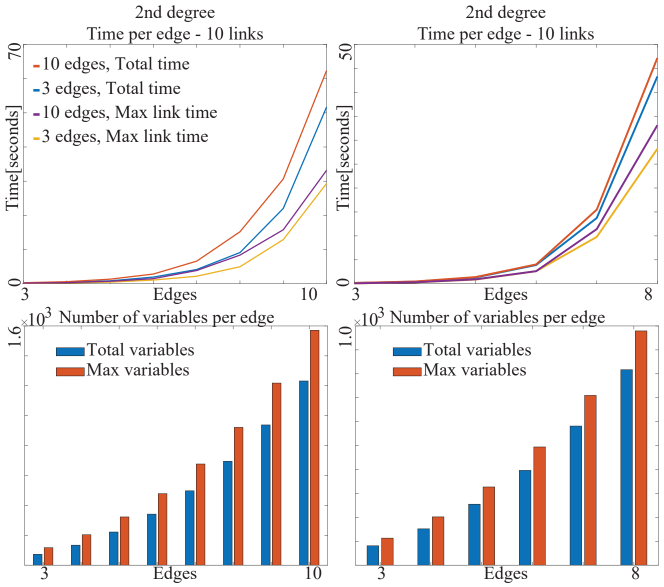


Fig. 7: Comparison of the running time and number of variables present when treating polygonal obstacles. The experiment was run with 3 and 10 links, and 3 and 10 polygonal sides. The number of variables shown refers to the total number existing in the problem, and variables related to the last link only.

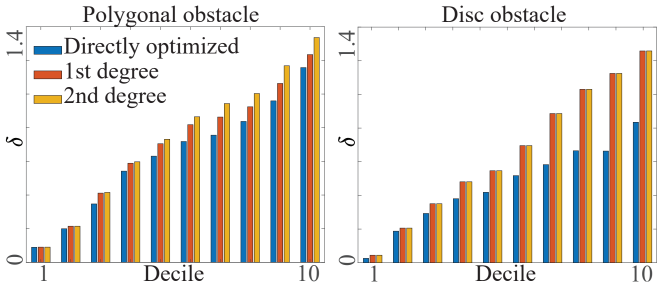


Fig. 8: The value of δ found with direct optimization (Sec. III-E), 1st and 2nd degree polynomials, for a polygonal and disc obstacle, with a 2-link arm. 100 random sample were arranged in deciles and averaged. As expected, the 2nd degree case achieves the highest δ, but suffer from the worst performance (compare with Fig. 6 and 7).

V. Conclusions

We presented an SOS-based formulation for finding a large and convex collision-free region in the C-space of a two-dimensional robotic arm. Our formulation is very direct, relying on substituting the links into an implicit expression defining the obstacles, and using the SOS paradigm to define and check that the result is positive for the entire link. This approach is general, and covers not only convex polygonal obstacles, but also free-form and non-convex obstacles. The solution is non-iterative, and uses only SOS polynomials of modest degrees. Future work will address convex regions in C-space with more general shapes, as well as three-dimensional and mobile robots.

References

[1] M. Choi, T. Lam, and B. Reznick, “Sums of squares of real polynomials,” *Proceedings of Symposia in Pure Mathematics*, vol. 58.2, pp. 103–126, 1995.

[2] Y. Alfassi, D. Keren, and B. Reznick, "The non-tightness of a convex relaxation to rotation recovery," *Sensors*, vol. 21, no. 21, p. 7358, 2021. [Online]. Available: https://doi.org/10.3390/s21217358

[3] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001. [Online]. Available: https://doi.org/10.1137/S1052623400366802

[4] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[5] S. M. LaValle, J. J. Kuffner, B. Donald *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, vol. 5, pp. 293–308, 2001.

[6] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[7] T. Lozano-Perez, "Spatial planning: A configuration space approach," in *IEEE Trans- actions on Computers*, vol. 100, 1983.

[8] J.-M. Lien and N. M. Amato, "Approximate convex decomposition of polyhedra and its applications," *Computer Aided Geometric Design*, vol. 25, no. 7, pp. 503–522, 2008, solid and Physical Modeling. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167839608000344

[9] A. Sarmientoy, R. Murrieta-Cidz, and S. Hutchinsony, "A sample-based convex cover for rapidly finding an object in a 3-d environment," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 3486–3491.

[10] M. Ortlieb, F.-M. Adolf, and F. Holzapfel, "Protected online path planning for uavs over congested areas within convex regions of obstacle-free space," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–9.

[11] 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)*, 2020, pp. 7101–7107.

[12] G. Franzè, "A nonlinear sum-of-squares model predictive control approach," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1466–1471, 2010.

[13] A. Papachristodoulou and S. Prajna, "On the construction of lyapunov functions using the sum of squares decomposition," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3, 2002, pp. 3482–3487 vol.3.

[14] Z. Marschner, P. Zhang, D. R. Palmer, and J. Solomon, "Sum-of-squares geometry processing," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 253:1–253:13, 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480551

[15] A. A. Ahmadi, G. Hall, A. Makadia, and V. Sindhwani, "Geometry of 3d environments and sum of squares polynomials," 2017.

[16] P. A. Parrilo and B. Sturmfels, "Minimizing polynomial functions," in *Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science, Proceedings of a DIMACS Workshop, Piscataway, NJ, USA, March 12-16, 2001*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, S. Basu and L. González-Vega, Eds., vol. 60. DIMACS/AMS, 2001, pp. 83–99. [Online]. Available: https://doi.org/10.1090/dimacs/060/08

[17] D. Keren and C. Gotsman, "Fitting curves and surfaces with constrained implicit polynomials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 1, pp. 31–41, 1999. [Online]. Available: https://doi.org/10.1109/34.745731

[18] P. Zhang, Z. Marschner, J. Solomon, and R. Tamstorf, "Sum-of-squares collision detection for curved shapes and paths," in *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, E. Brunvand, A. Sheffer, and M. Wimmer, Eds. ACM, 2023, pp. 76:1–76:11. [Online]. Available: https://doi.org/10.1145/3588432.3591507

[19] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," 2018.

[20] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," 2022.

[21] W. Zhao, S. He, and C. Liu, "Provably safe tolerance estimation for robot arms via sum-of-squares programming," *IEEE Control Systems Letters*, vol. 6, pp. 3439–3444, 2022.

[22] H. Dai, A. Amice, P. Werner, A. Zhang, and R. Tedrake, "Certified polyhedral decompositions of collision-free configuration space," *CoRR*, vol. abs/2302.12219, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.12219

[23] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000. [Online]. Available: https://resolver.caltech.edu/CaltechETD:etd-05062004-055516

[24] B. R. Barmish, P. S. Shcherbakov, S. R. Ross, and F. Dabbene, "On positivity of polynomials: The dilation integral method," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 965–978, 2009.

[25] P. Seiler, "Sosopt: A toolbox for polynomial optimization," *arXiv preprint arXiv:1308.1889*, 2013.

[26] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana Univ. Math. J.*, vol. 42, pp. 969–984, 1993.

[27] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parrilo, M. M. Peet, and D. Jagt, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, http://arxiv.org/abs/1310.4716, 2021, available from `https://github.com/oxfordcontrol/SOSTOOLS`.

[28] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.