A New Framework for Optimal Path Planning of Rectangular Robots Using a Weighted L_p Norm

Nak-seung Patrick Hyun, Patricio A. Vela, and Erik I. Verriest

***Abstract*—This letter introduces a new framework for modeling the optimal path planning problem of rectangular robots. Typically constraints for the safe, obstacle-avoiding path involve a set of inequalities expressed using logical OR operations, which makes the problem difficult to solve using existing optimization algorithms. Inspired by the geometry of the unit sphere of the weighted L_p norm, the authors find exact and approximate constraints for safe configurations using only logical AND operations. The proposed method does not require integer programming nor computation of a Minkowski sum in the configuration space. In particular, the authors analyze two different cases of obstacle geometry: circular obstacles and rectangular obstacles. Using the weighted L_p norm requires six inequalities to represent the exact constraints for collision avoidance of circular obstacles using AND operations, and eight inequalities for rectangular obstacles. Four shortest path planning examples are analyzed to validate the effectiveness of the proposed method.**

***Index Terms*—Collision avoidance, optimization and optimal control, motion and path planning.**

I. Introduction

THIS letter considers the problem of safe trajectory generation for a robot through the synthesis of cost minimizing optimal controls (e.g., arrival time or path length), also known as robot motion planning. A classical and well-known approach to robot motion planning is the *configuration space approach*, which converts the rigid body path planning problem into a simpler point mass robot problem in the higher dimensional space, denoted as the C-space [1]. Transforming the rigid body to a point mass equivalent involves applying the Minkowski sum, defined by $A \oplus B = \{q \in \mathbb{R}^n : q = x + y, \forall x \in A \wedge \forall y \in B\}$ for two given sets $A, B \subset \mathbb{R}^n$. The process creates obstacles representations in the C-space, so called C-obstacles, when computed via the Minkowski summation for each robot orientation. Once the C-obstacles are constructed, the set collision free configurations, called the C-free space, is obtained via the complement of the C-obstacles in C-space. An algorithm to find the C-free space for two polygon shaped robot and obstacle is shown in [2] and an algorithm for the exact Minkowski sum of convex polyhedra is given in [3]. More details of the motion planning on the configuration space, and the numerical algorithms can be found in [4]–[6].

Analytic expressions for the C-free space involve a set of inequalities combined by the logical OR operation when the robot or the obstacle are polygon shaped [5]. The use of the logical OR causes a major problem in applying general optimization algorithms since they are constructed based on AND operations applied to the constraints; constraints with OR operations are analytically intractable [7]. One way to avoid the OR operation for the rectangular obstacle with a point mass robot is to use mixed integer programming (MIP) by introducing artificial integer valued variables [8], [9]. However, integer programing is known to be NP-hard [10]. Recently, an efficient way of reducing the number of integers in the optimal path planning problem was given in [11]. The method improves on the scaling of the NP-hard problem at the expense of a decreased probability of finding a feasible solution.

This letter proposes an alternative method, without using integer variables, to find the constraints for safe configurations through the use of the logical AND operations for the case of a rectangular robot. The proposed framework to formulate the equivalent constraints of a safe configuration is inspired by the geometry of unit spheres for weighted L_p norms. The unit level set of the L_p norm with different p values forms different shapes that converge to the square as $p \to \infty$. Thus weighted L_p norms lead to exact and approximated implicit equations for the boundaries of rectangles. Taking advantage of this observation, this letter describes how to synthesize a collection of mixed L_p norms that implicitly define safe configurations.

Work related to the proposed framework exploring non-circular robots includes [12], which gives the analytic formulation of the exact boundary for the Minkowski sum of two ellipses through affine transformations. Follow-up work applied this formulation to the path planning problem with elliptic obstacles in the C-space by providing the exact boundaries for the C-obstacles [13]. Although ellipses arise through weighted L_2 norms, the approach here is essentially different from- and complementary to- [13]. First, the method here considers rectangular robots and a mixture of rectangular and circular obstacles (it also holds for circular robots). Second, the goal of current work is to find the implicit constraints for the center of the robot without constructing the Minkowski sum between the robot and the obstacles. As will be shown, the synthesis of these implicit constraints is computationally simple since all that is required

Manuscript received September 10, 2016; accepted February 3, 2017. Date of publication February 23, 2017; date of current version March 10, 2017. This letter was recommended for publication by Associate Editor E. Papadopoulos and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant CMMI-1400256.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: nhyun3@gatech.edu; pvela@ece.gatech.edu; erik.verriest@ece.gatech.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2673858

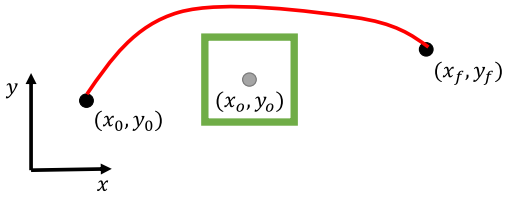

Fig. 1. Single square obstacle avoidance example.

is to define 8 different weighted L_p norms for the rectangular obstacle case, and 6 for the circular obstacle case. In addition, the primary interest is in solving the optimization problem by analytically setting up a Bolza type optimal control problem [14], rather than constructing C-obstacles.

II. Motivational Example

To motivate the solution, consider the shortest path problem for a point-mass unicycle robot which should travel from the starting configuration $g = (\mathbf{x_i}, \mathbf{y_i}, \theta_\mathbf{i})$ to the final position $(x, y) = (\mathbf{x_f}, \mathbf{y_f})$, irrespective of orientation, in the planar space $SE(2)$. As per Fig. 1, let there be a single square obstacle centered at (x_o, y_o) with edge length 2 between the initial and the final positions of the robot, so that a detour is needed to avoid collisions. A formulation of the shortest safe path problem is

$$\arg\min_{u_1,u_2} \int_0^T \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}dt \quad \text{subject to} \tag{1}$$

$$\begin{cases} \dot{x} = u_1(t)\cos(\theta(t)) \\ \dot{y} = u_1(t)\sin(\theta(t)) \\ \dot{\theta} = u_2(t) \end{cases}, \tag{2}$$

$$(x(0), y(0), \theta(0)) = (\mathbf{x_i}, \mathbf{y_i}, \theta_\mathbf{i}), \tag{3}$$

$$(x(T), y(T)) = (\mathbf{x_f}, \mathbf{y_f}), \tag{4}$$

$$|x - x_o| > 1 \quad \text{OR} \quad |y - y_o| > 1, \tag{5}$$

where (1) is the cost, and (2)–(5) are constraints of the optimization problem. The safe trajectory constraints (5) are a set of inequalities with the logical OR operation. However, as mentioned in Section I, the OR condition is analytically intractable. To solve with mathematical programming algorithms, we consider conversion to equivalent AND operations.

The idea is to find a function $V : \mathbb{R}^2 \to \mathbb{R}$ such that there exist a constant $c \in \mathbb{R}$, which makes the superlevel set $\{(x, y) = q \in \mathbb{R}^2 : V(x, y) \geq c\}$ equivalent to the OR conditioned constraint in (5). The function V for the square obstacle centered at (x_o, y_o) will be formulated by the L_∞ norm in L_p theory [15]. Define V by

$$\begin{aligned} V(x, y) :&= ||(x - x_o, y - y_o)||_\infty \\ &= \lim_{p\to\infty} (|x - x_o|^p + |y - y_o|^p)^{\frac{1}{p}}. \end{aligned} \tag{6}$$

Since the L_∞-norm unit sphere in $\mathbb{R}^2$ is a square centered at the origin, the inequality constraint $V(x, y) > 1$ replaces the OR conditions in (5).

III. Problem Statement

This section generalizes the motivational problem to consider a rectangular robot navigating through a space with rectangular and circular obstacles. The geometry of the rectangular body is described by $\sigma := (\sigma_1, \sigma_2)$, where σ_1 and σ_2 are the half lengths of each edge. There are N_c circular obstacles centered at $o_c^i := (x_o^i, y_o^i) \in \mathbb{R}^2$ with radius $r_i > 0$ for each $i \in \{1, \cdots, N_c\}$, and there are N_r rectangular obstacles whose frame is $g_r^j := (x_r^j, y_r^j, \theta_r^j) \in SE(2)$, where the coordinate frame is centered in the square. The rectangular obstacle half lengths are $\sigma^j := (\sigma_1^j, \sigma_2^j)$ for each $j \in \{1, \ldots, N_r\}$. Without loss of generality, the problem will be formulated using an initial $SE(2)$ configuration and a final position, $\mathbb{R}^2 \subset SE(2)$, as the path boundary conditions. The arrival time, $T_f > 0$, is fixed.

Here, $g(q) \in \mathbb{R}^2$ denotes the transformation of coordinates of $q \in \mathbb{R}^2$ according to $g \in SE(2)$. The inverse g^{-1} would perform the change of coordinates in the inverse direction.

The collision-free spaces for the circular and rectangular obstacles are

$$C_i := \{q \in \mathbb{R}^2 : \left|\left|q - o_c^i\right|\right|_2^2 > r_i^2\}, \text{ and} \tag{7}$$

$$R_j := \{q \in \mathbb{R}^2 : \left|\left|[g_r^j]^{-1}(q)\right|\right|_{\Sigma,\infty} > 1\}, \tag{8}$$

respectively, in the world frame for all i and j (see Section IV regarding $||\cdot||_{\Sigma,\infty}$). As the robot moves, the collision constraints in the robot's body frame will change. With g defining the robot's body frame relative to the world frame, the collision-free space in robot body coordinates is:

$$C_i^\mathcal{R} := \{q \in \mathbb{R}^2 : \left|\left|g(q) - o_c^i\right|\right|_2^2 > r_i^2\}, \text{ and} \tag{9}$$

$$R_j^\mathcal{R} := \{q \in \mathbb{R}^2 : \left|\left|[g_r^j]^{-1} \circ g(q)\right|\right|_{\Sigma,\infty} > 1\}. \tag{10}$$

Let $B \subset \mathbb{R}^2$ describe the subset of $\mathbb{R}^2$ containing the robot's full body in the robot's body frame, then the optimal safe path planning problem formulation is

$$\arg\min_u \int_0^{T_f} L(g, \dot{g}, u)dt \text{ subject to} \tag{11}$$

$$\begin{cases} \dot{g} = f(g, u), \\ g(0) = (\mathbf{x_i}, \mathbf{y_i}, \theta_\mathbf{i}), \\ \text{Pos}\,(g(T_f)) = (\mathbf{x_f}, \mathbf{y_f}), \end{cases} \tag{12}$$

$$\begin{cases} B \subset (\bigcap_{i=1}^{N_c} C_i^\mathcal{R}), \\ B \subset (\bigcap_{j=1}^{N_r} R_j^\mathcal{R}), \end{cases} \tag{13}$$

where Pos $(\cdot)$ gives the translation coordinates of $SE(2)$, $u(t) \in \mathbb{R}^2$ is the control of the robot, f are the control equations of motion, and $L(g, \dot{g}, u)$ is some physically meaningful cost such as control energy or path length. Since the constraints in (13) typically require OR operations, our goal is to first convert them into a new set of constraints with AND operations.

Remarks: Once the problem is reformulated with AND operations, numerical solutions to Bolza type path planning problems [14] no longer require explicit construction of the C-obstacles nor the C-free space. The problem in (11–13) generalizes to

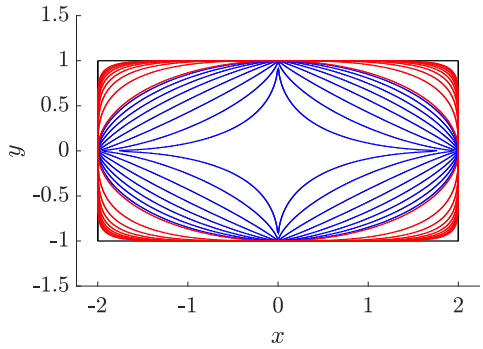

Fig. 2. Unit sphere of $(2,1)$-weighted L_p norms with different p values.

admit other constraints such as free final time or fixed final orientation. An example in Section VI-B contains the fixed final orientation and free final time for the generalization.

IV. Weighted L_p Norm

Let $x = (x_1, \cdots, x_n)$ be a vector in $\mathbb{R}^n$ for $n \in \mathbb{N}$, with x being positive (e.g., $x_i > 0$ for all i).

Definition 1 (Weighted L_p norm): Let $\sigma \in \mathbb{R}^n$ be a positive vector, and $0 < p < \infty$, then $||\cdot||_{\sigma,p} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the σ-weighted L_p norm such that

$$||x||_{(\sigma,p)} := \left(\sum_{i=1}^{n} (|x_i|/\sigma_i)^p \right)^{\frac{1}{p}},$$

for all $x \in \mathbb{R}^n$ where $\sigma = (\sigma_1, \ldots, \sigma_n)$. Furthermore the level set $\{x \in \mathbb{R}^n : ||x||_{(\sigma,p)} = 1\}$ is called the *unit sphere* of the $L_{\sigma,p}$ norm.

With a slight abuse of notation, if σ is a scalar but x is a vector, then the division by σ is applied across all coordinates. Another way of defining the σ-weighted L_p norm is by $||x||_{\Sigma,p} := ||\Sigma^{-1}x||_p$, where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, which shows that it is a generalization of the usual L_p norm. A major benefit of this generalization is that by choosing different σ or p, the unit sphere of usual L_p norm can be continuously deformed. For any given positive σ, as p increases, the unit sphere of $L_{(\sigma,p)}$ approaches the unit sphere of $L_{(\sigma,\infty)}$.

Fig. 2 plots the collection of unit spheres for $n = 2$ and $\sigma = (2,1)$, where each closed curve represents the unit sphere for different p values. The blue curves corresponds to the case when p changes from 0.5 (innermost) to 1.9 (outermost) with step size 0.2. Similarly, the red curves represent corresponding changes as p increases from 2 (innermost) to 10 (outermost) with step size 1. The black curve corresponds to $p = \infty$, where it is equivalent to the supremum norm.

V. Collision Avoidance Constraints

The motivational example, with a point mass robot and a square obstacle, demonstrated that the weighted L_p norm can be used to analytically define the free space with the implicit inequality constraint (using $p = \infty$). However, for non-point mass robots, imposing a single point-based L_∞ inequality constraint will not be sufficient. All boundary points of the robot must be considered to define a collision between the robot and an obstacle. Similar to the configuration space approach, the goal is to convert the collision-free space determination into an equation of the position of the robot frame or obstacle frame, but there is the additional objective to avoid explicit computation of the boundary points. Instead, this section seeks to define value functions as in (6) that provide implicit inequalities, and that compose using logical AND operations. Two cases are considered: between a circular object and a rectangular object, and between two rectangular objects.

A. Circular Obstacles

The case of circular obstacles will be examined by converting the Minkowski sum of a rectangle with a circle into a set of inequality constraints (or of super-level set definitions) that must all be met. Define the collision space for the circular obstacle given relative to its own frame by

$$\overline{C}_i^{\mathcal{O}} := \{q \in \mathbb{R}^2 : ||q||_2^2 \leq r_i^2\}. \tag{14}$$

A collision between the robot and the obstacle happens if the distance of the robot body to the circular obstacle is less than r_i. The boundary of Minkowski sum $B \oplus \overline{C}_i^{\mathcal{O}}$ is depicted in Fig. 3(a) by the outer boundary of the graphed shapes (coordinates in the robot's body frame). Since the robot is rectangular, the set B is a rectangular shape with half lengths $\sigma = (\sigma_1, \sigma_2)$. For the figure, $\sigma = (2,1)$ and $r = 1$. The figure also depicts 6 shapes whose union is equal to the Minkowksi sum. These sums define the 6 value functions to be constructed.

Using the σ-weighted L_p norm of Definition 1, the value functions for the ith circular obstacle are

$$\begin{aligned}
V_1^i(q;g) &:= ||g^{-1}(q)||_{(\sigma_1+r_i,\sigma_2),p} \\
V_2^i(q;g) &:= ||g^{-1}(q)||_{(\sigma_1,\sigma_2+r_i),p} \\
V_3^i(q;g) &:= ||g^{-1}(q) - (\sigma_1,\sigma_2)^T||_{r_i,2} \\
V_4^i(q;g) &:= ||g^{-1}(q) + (-\sigma_1,\sigma_2)^T||_{r_i,2} \\
V_5^i(q;g) &:= ||g^{-1}(q) + (\sigma_1,-\sigma_2)^T||_{r_i,2} \\
V_6^i(q;g) &:= ||g^{-1}(q) + (\sigma_1,\sigma_2)^T||_{r_i,2}
\end{aligned}$$

where g defines the current robot rigid body frame relative to the world frame. If g is omitted, then it is presumed to be the identity (e.g., the robot and world frames align).

Proposition 1: If $p = \infty$, then $B \oplus \overline{C}_i^{\mathcal{O}}$ is equal to $\bigcup_{j=1}^{6} \{q \in \mathbb{R}^2 : V_j^i(q) \leq 1\}$.

Proof. The proof is immediate by the construction of value functions, and so it is omitted. □

The proposition indicates that the proposed six constraints are exactly equivalent to the C-obstacles constructed by the Minkowski sum, given that $p = \infty$. Furthermore, the collision-free space is the negation of the union, which leads to intersections of super level-sets. Intersections of spaces are described through logical AND operators in the optimal control formulation. However, since the gradient of the maximum operator does not have an analytic formula, using L_∞ norm in the optimization

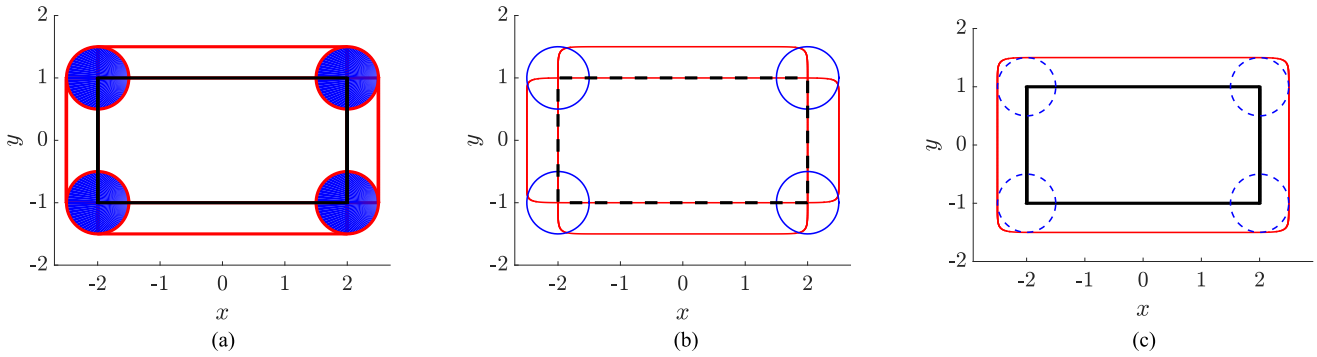

Fig. 3. Weighted L_p constraints method for a rectangular robot and a circular obstacle. (a) Exact constraints. (b) 6 constraints. (c) 1 constraint.

problem may not be a good choice. Differentiable, alternative or approximate formulations are sought.

The discussion of Section IV and visualization of unit L_p spheres showed that as p gets larger the unit sphere of L_p approaches that of L_∞. Although it is not a perfect rectangle, the benefit of having large p is an analytically computable gradient for the inequality constraint, and use it in the optimization problem. An example for the case $p = 20$ is shown in Fig. 3(b). The parameters in the examples are $\sigma = (2, 1)$ and $r = 1$. The black dotted line represents the robot body in its coordinate frame, and two horizontally long and vertically long rectangles represent the unit level sets of V_1^i and V_2^i, respectively. The four blue circles at each corner represents the unit level set of V_3^i, V_4^i, V_5^i and V_6^i. The approximate nature means that there will be some part of the collision-free space that is actually a collision, or conversely, some part of the collision space will return as collision-free. There are eight such regions, two at each corner. They are the dimples formed at the outer intersections of the approximately rectangular boundaries with the circular boundaries.

A simpler, conservative approximation of the Minkowski sum may be defined using a single value function instead of six. Let V_7^i be another value function defined as

$$V_7^i(q; g) := ||g^{-1}(q)||_{(\sigma_1+r_i+\epsilon,\sigma_2+r_i+\epsilon),p} \tag{15}$$

By choosing a large p and a small $\epsilon > 0$, the Minkowski sum is confined to the sub level-set $\{(x, y) : V_7^i(x, y) \leq 1\}$. An example for $p = 20$ and $\epsilon = 0.01$ is shown in Fig. 3(c), where red curve represents the unit level set of V_7^i, and dashed blue represents the circle with radius $r = 1$ at each corner. In this case, the opposite situation occurs. The extra collision regions at the rounded corners means that some parts of the collision-free space evaluate to collisions.

As noted earlier, the main result of this section is that the OR condition in (13) due to the rectangular body of the robot is now changed to the AND condition. In summary, the anti-collision condition for the circular obstacle may be replaced by one of the two logical expressions,

$$\wedge_{i=1}^{N_c} \wedge_{k=1}^{6} \{q \in \mathbb{R}^2 : V_k^i(q) > 1\} \tag{16}$$

$$\wedge_{i=1}^{N_c} \{q \in \mathbb{R}^2 : V_7^i(q) > 1\}, \tag{17}$$

where $\wedge$ stands for the logical AND operation. The number of inequalities required for the circular obstacles will be $6N_c$ if only the former expression is used, N_c if only the latter expression is used (or some mix if one of either two is selected per circular obstacle). The constraint (16) is more accurate in capturing the collision at the corner of the robot, while (17) is more beneficial numerically since the formulation has less inequality constraints.

B. Rectangular Obstacles

For two rectangular obstacles, collision space is trickier to compute with the Minkowski sum since the obstacles orientations modify the sum. In order to construct a full C-obstacle, it is necessary to compute the Minkowski sum at every orientation of the robot and the obstacles. Fortunately, the weighted L_p method to find the equivalent set of inequalities composed with logical AND operation is simpler than computing the Minkoswski sum in C-space.

The contact of two polygons in $\mathbb{R}^2$ is categorized into two types: one is the *robot vertex to obstacle edge* and the other is *robot edge to obstacle vertex* [5], which are denoted as *Type A* and *Type B* collisions, respectively, in [4]. Since the robot and the obstacle are both rectangular, there will be eight different types of possible contact.

Define the 8 corners from the robot and the jth obstacle, in the world frame, by

$$A = g\left(\begin{bmatrix} \sigma_1 & \sigma_1 & -\sigma_1 & -\sigma_1 \\ \sigma_2 & -\sigma_2 & -\sigma_2 & \sigma_2 \end{bmatrix}\right) \tag{18}$$

$$A_j = g_r^j\left(\begin{bmatrix} \sigma_1^j & \sigma_1^j & -\sigma_1^j & -\sigma_1^j \\ \sigma_2^j & -\sigma_2^j & -\sigma_2^j & \sigma_2^j \end{bmatrix}\right) \tag{19}$$

where each column of A and A^j represents the coordinate of corners of the robot and each obstacles, respectively. The obstacle corners in the robot frame are $A_j^{\mathcal{R}} = g^{-1}(A_j)$. Denote the columns of $A_j^{\mathcal{R}}$ at time t by $\psi_k(t) \in \mathbb{R}^2$ for each $k \in \{1, 2, 3, 4\}$. The anti-collision condition of *robot edge to obstacle vertex* contact consists of four AND operations,

$$\wedge_{k=1}^{4} \{||\psi_k(t)||_{(\sigma_1,\sigma_2),p} > 1\}. \tag{20}$$

The $L_{(\sigma_1,\sigma_2),p}$ distance from the center of the robot to the obstacle corners should be greater than one.

Now, consider the matrix of the four corner coordinates of the robot in the jth obstacle frame, $A^{\mathcal{O},j} = [g_r^j]^{-1}(A)$. Denote its columns at time t by $\psi_k^j(t) \in \mathbb{R}^2$ for each $k \in \{1,2,3,4\}$, then the anti-collision conditions of *robot vertex to obstacle edge* comprise of four AND operations,

$$\wedge_{k=1}^{4}\{||\psi_k^j(t)||_{(\sigma_1^j,\sigma_2^j),p} > 1\}. \tag{21}$$

The union of (20) and (21) gives eight inequality constraints,

$$\wedge_{j=1}^{N_2} \wedge_{k=1}^{4} \Big(\{||\psi_k(t)||_{(\sigma_1^j,\sigma_2^j),p} > 1\} \times \wedge \left(\{||\psi_k^j(t)||_{(\sigma_1,\sigma_2),p} > 1\}\right), \tag{22}$$

which represent the avoidance of both Type A and Type B collisions. For N_r rectangular obstacles, there will be $8N_r$ inequality constraints. They will be somewhat permissive, indicating that a configuration may be collision-free when it indeed collides. An ϵ-hedge gives conservative constraints.

VI. Optimal Path Planning

This section solves several optimal path planning problems for a unicycle robot with the rectangular body shape using the L_p constraints defined in the previous section. The cost function (11) choice is $L(g,\dot{g},u) := \sqrt{\dot{x}^2+\dot{y}^2}$, to give a shortest path problem. The control equations of motion are those from (2). Assume that we have control inputs, $u_1, u_2 \in C([0,T_f],\mathbb{R})$, bounded by $u_1(t) \in [\underline{u}_1, \overline{u}_1]$ and $u_2(t) \in [\underline{u}_2, \overline{u}_2]$. The final arrival time $T_f > 0$ is fixed in each problem except for Section VI-B.

The MATLAB-based numerical optimal control solver OPTRAGEN [16], converts the Bolza type optimal control problem into a nonlinear programming problem (NLP) by using the collocation method, and then we invoke IPOPT 3.12.6 [17] to solve the NLP with a sequential quadratic programming (SQP) algorithm. For all of the following solution figures, the red curves represent the obstacle boundaries, the blue curve represents the trajectory of the center position of the robot, and the black curves shows the shape of the robot at each sampled collocation time.

A. Circular Obstacles

1) Thin Rectangular Robot: This example involves a thin robot, $\sigma = (2,1)$, with two circular obstacles both of radius $r = 1$, centered at $(2,-1.6)$ and $(-1,1.5)$. The initial orientation is $\theta_{\mathrm{i}} = -\pi/4$. The direction of motion is in the direction of the shorter edge. The initial robot position is $(-3.11, 0.11)$, and the objective is to reach $(3.52,-0.22)$ at time $T_f = 3.667\pi$. The control upper and lower bounds are $u_1(t) \in [-2\pi, 2\pi]$ and $u_2(t) \in [-\pi/2, \pi/2]$. We use the simpler inequality constraint for the obstacles (2 constraints) using (17) with $p = 8$ for the approximation. The optimal control result depicted in Fig. 4 shows that the shortest path involved moving the robot through the narrow gap.

2) Wide Rectangular Robot: Here, the wide robot has $\sigma = (1,2)$. The obstacles are centered at the same location as in the skinny case but the radii are changed to $r = 0.8$. The initial orientation is $\theta_{\mathrm{i}} = 0$. The direction of motion is in the direction of the longer edge. The initial robot position is $(-2.11,-2.11)$,

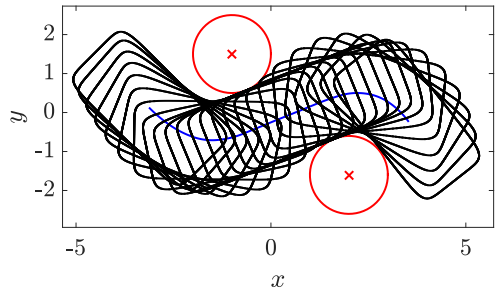

Fig. 4. Shortest path solution for the thin robot.

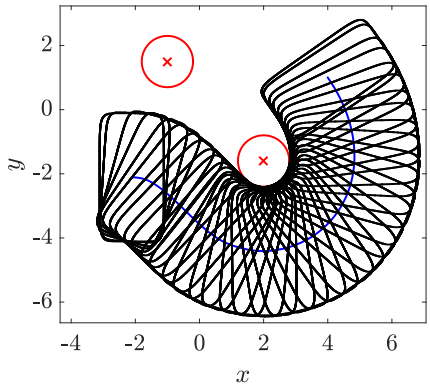

Fig. 5. Shortest path solution for the wide robot.

and the objective is to reach $(2.52, 2.22)$ at time $T_f = 7\pi$, with the same control bounds as for the thin robot. Again, the simpler inequality constraint (17) for the obstacles (2 constraints) are used with $p = 8$ for the approximation. The optimal control result depicted in Fig. 5 shows that the shorted path involved circumnavigating the two obstacles, as the gap was not wide enough.

B. Rectangular Obstacles

This example considers two rectangular obstacles whose frames are $g_r^1 = (2,-1.6,\pi/3)$ and $g_r^2 = (0,2.5,\pi/3)$, in $SE(2)$. Both rectangles have the shape half-lengths $\sigma^j = (5,1)$. The rectangular robot has $\sigma = (0.9, 0.3)$ with initial orientation $\theta_{\mathrm{i}} = -\pi/4$. In addition, we impose a final orientation constraint, $\theta_{\mathrm{f}} = -\pi/4$, and free T_f. The bounds for the controls are $u_1(t) \in [-5,5]$ and $u_2(t) \in [-0.5\pi, 0.5\pi]$. The initial position of the robot is $(-5,-2)$, with the final position $(6,4)$ at T_f. We use the L_p constraints for the safe path (8 constraints for each obstacle) using (22) with $p = 10$ for the approximation. The optimal control result is depicted in Fig. 6 and shows that the shortest path for the robot went through the hallway created by two rectangular obstacles.

C. Dual Problem: Circular Robot with Rectangular Obstacle

The last example demonstrates the dual problem of a circular robot navigating around rectangular obstacles. The example is included since it is easier to visualize the 6 constraints in this dual problem. The robot starts with the initial configuration $g(0) = (-3,-1,\pi/4)$, and has a radius of 0.5. The robot final position is $(5,1)$ with final time $T_f = 11.6\pi$. The obstacles

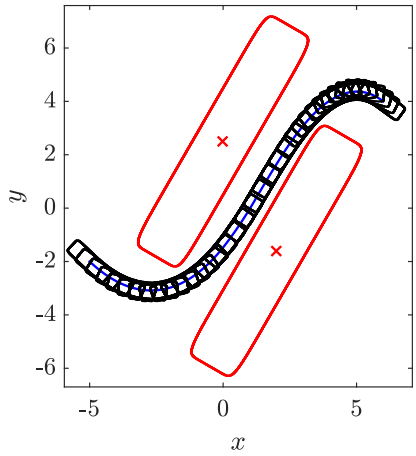

Fig. 6. Shortest path between rectangles for a thin robot with a fixed final orientation and free final time constraints.

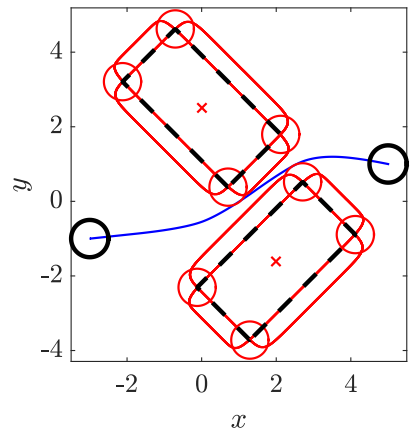

Fig. 7. Shortest path using 6 constraints obstacle.

are centered at the same location as for the thin rectangular robot case (Section VI-B), with half side lengths $(2, 1)$, and orientations $\pi/4$ and $-\pi/4$, respectively. The bounds for the controls are $u_1(t) \in [-15, 15]$ and $u_2(t) \in [-0.5\pi, 0.5\pi]$. We use the L_p constraints for the safe path (6 constraints for each obstacles) with a modified version of (16). The value $p = 10$ is chosen for the approximation.

The optimal control result is depicted in Fig. 7 where the two black circles represent the robot at the initial and the final position. Like Fig. 3(b), the unit spheres of the six L_p constraints are overlaid on the obstacles. Examination of the numerical output confirms that the robot trajectory does not enter the region generated by the six constraints for each obstacle.

VII. Conclusions and Future Work

A. Conclusion

This letter introduced a new method for finding an exact and approximated constraints of the safe configuration of a robot within an optimal control formulation. The constraints are given by inequalities that get composed using logical AND operations. The work was motivated by the geometry of the unit sphere of the L_p norm, and its variations using scaling factors. Given the rectangular robot, we have shown that 6 constraints are required for circular obstacles, and 8 constraints are required for rectangular obstacles to get the exact inequality constraints for the safe path. The method admits optimal control formulations that stay within a nonlinear programming framework without resorting to integer or discrete variables for the anti-collision constraints.

B. Future Work

This framework on motion planning with weighted L_p constraints has several immediate extensions. For the spatial or 3D case (cuboid robot), weighted L_p norms still apply. The number of constraints for sphere obstacle remains 6 for exact case, and 1 for the simple approximation. Similarly, for the cubic obstacles, *vertex to face* contact and *face to vertex* contact can be extended easily. However, modeling the *edge to edge* contact is still an open question.

In addition, detailed comparisons between the proposed framework and the existing mixed integer-programing approach should be done to show the difference in the scalability (computation time), and the feasibility (accuracy) of the solutions.

References

[1] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. C-32, no. 2, pp. 108–120, Feb. 1983.
[2] F. Avnaim and J.-D. Boissonnat, "Polygon placement under translation and rotation," in *Proc. Annu. Symp. Theoretical Aspects Comput. Sci.*, 1988, pp. 322–333.
[3] E. Fogel and D. Halperin, "Exact and efficient construction of Minkowski sums of convex polyhedra with applications," *Comput. Aided Design*, vol. 39, no. 11, pp. 929–940, 2007.
[4] J.-C. Latombe, *Robot Motion Planning*, vol. 124. New York, NY, USA: Springer Science & Business Media, 2012.
[5] S. M. LaValle, *Planning algorithms*. Cambridge U.K.: Cambridge Univ. Press, 2006.
[6] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1–4, pp. 65–100, 2010.
[7] H. A. Taha, *Operations Research: An Introduction*, 8th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2006.
[8] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. IEEE Eur. Control Conf.*, 2001, pp. 2603–2608.
[9] A. Richards and J. How, "Mixed-integer programming for control," in *Proc. IEEE Amer. Control Conf.*, 2005, pp. 2676–2683.
[10] R. M. Karp, *Reducibility Among Combinatorial Problems*. Berlin, Germany: Springer, 2010, pp. 219–241.
[11] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 42–49.
[12] Y. Yan and G. S. Chirikjian, "Closed-form characterization of the Minkowski sum and difference of two ellipsoids," *Geometriae Dedicata*, vol. 177, no. 1, pp. 103–128, 2015.
[13] Y. Yan, Q. Ma, and G. S. Chirikjian, "Path planning based on closed-form characterization of collision-free configuration-spaces for ellipsoidal bodies, obstacles, and environments," in *Proc. Int. Workshop Robot Learn. Plan.*, 2016, pp. 13–19.
[14] A. Bryson, *Applied Optimal Control: Optimization, Estimation and Control* (Halsted Press book). New York, NY, USA: Taylor & Francis, 1975.
[15] A. Naylor and G. Sell, *Linear Operator Theory in Engineering and Science* (Applied Mathematical Sciences). New York, NY, USA: Springer, 2000.
[16] R. Bhattacharya, "OPTRAGEN: A MATLAB toolbox for optimal trajectory generation," in *Proc. IEEE Conf. Decision Control*, 2006, pp. 6832–6836.
[17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.