Resolution-Exact Planner for Thick Non-Crossing 2-Link Robots[⋆]

Chee K. Yap, Zhongdi Luo, and Ching-Hsiang Hsu

Department of Computer Science
Courant Institute, NYU
New York, NY 10012, USA
{yap,zl562,chhsu}@cs.nyu.edu

Abstract. We consider the path planning problem for a 2-link robot amidst polygonal obstacles. Our robot is parametrizable by the lengths $\ell_1, \ell_2 > 0$ of its two links, the thickness $\tau \geq 0$ of the links, and an angle κ that constrains the angle between the 2 links to be strictly greater than κ. The case $\tau > 0$ and $\kappa \geq 0$ corresponds to "thick non-crossing" robots. This results in a novel 4DOF configuration space $\mathbb{R}^2 \times (\mathbb{T}^2 \setminus \Delta(\kappa))$ where $\mathbb{T}^2$ is the torus and $\Delta(\kappa)$ the diagonal band of width κ.
We design a resolution-exact planner for this robot using the framework of Soft Subdivision Search (SSS). First, we provide an analysis of the space of forbidden angles, leading to a soft predicate for classifying configuration boxes. We further exploit the T/R splitting technique which was previously introduced for self-crossing thin 2-link robots.
Our open-source implementation in Core Library achieves real-time performance for a suite of combinatorially non-trivial obstacle sets. Experimentally, our algorithm performs significantly better than any of the state-of-art sampling algorithms we looked at, in speed and success rate.

1 Introduction

Motion planning is one of the key topics of robotics [9, 3]. The dominant approach to motion planning for the last two decades has been based on sampling, as represented by PRM [7] or RRT [8] and their many variants. An alternative (older) approach is based on subdivision [2, 20, 1]. Recently, we introduced the notion of **resolution-exactness** which might be regarded[1] as the well-known idea of "resolution completeness" with a suitable converse [16, 17]. Briefly, a planner is resolution-exact if there exists a $K > 1$ and its input includes, in addition to the usual inputs of path planning, a parameter $\varepsilon > 0$. It will output a path if there exists one with clearance $K\varepsilon$; it will output NO-PATH if there does not exist one with clearance K/ε. Note that its output is indeterminate if the clearance lies between K/ε and $K\varepsilon$. This provides the theoretical basis for exploiting the concept of **soft predicates**, which is roughly speaking the

[⋆] This work is supported by NSF Grants CCF-0917093 and CCF-1423228.

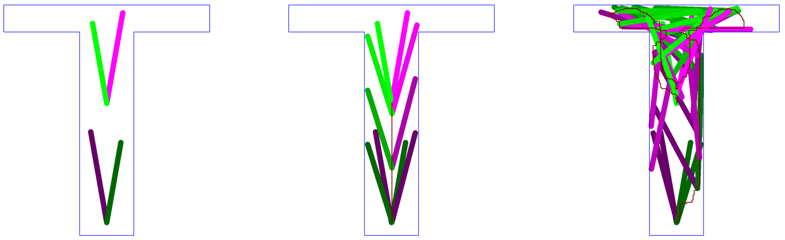
[1] In the theory of computation, a computability concept that has no such converse (e.g., recursive enumerability) is said to be "partially complete".

numerical approximation of exact predicates. Such predicates avoids the hard problem of deciding zero, leading to much more practical algorithms than exact algorithms. To support this new class of algorithms, and inspired by the success of the PRM framework, we introduce an algorithmic framework [17, 18] based on subdivision called **Soft Subdivision Search** (SSS). The present paper continues our exploration of algorithms in this framework.

Link robots offer a compelling class of non-trivial robots for exploring path planning (see [6, chap. 7]). In the plane, the simplest example of a non-rigid robot is the **2-link robot**, $R_2 = R_2(\ell_1, \ell_2)$, with links of lengths $\ell_1, \ell_2 > 0$. The two links are connected through a rotational joint A_0 called the **robot origin**. The 2-link robot is in the intersection of two well-known families of link robots, **chain robots** and **spider robots** (see [11]). One limitation of link robots is that links are unrealistically modeled by line segments. On the other hand, a model of mechanical links involving complex details may require algorithms that currently do not exist or have high computational complexity. As a compromise, we introduce **thick links** by forming the Minkowski sum of each link with a ball of radius $\tau > 0$ (**thin links** correspond to $\tau = 0$). To our knowledge, no exact algorithm for thick R_2 is known; for a single link R_1, an exact algorithm based on retraction follows from [14]. In this paper, we further parametrize R_2 by a "bandwidth" κ which constrains the angle between the 2 links to be strictly larger than κ ("self-crossing" links is recovered by setting $\kappa < 0$). Thus, our full robot model

$$R_2(\ell_1, \ell_2, \tau, \kappa)$$

has four parameters; our algorithms are uniform in these parameters.



(a) α (above), β (below)   (b) Self-crossing path   (c) Non-crossing path

Fig. 1: Path from configurations α to β in T-Room Environment

To illustrate the non-crossing constraint, we use the simple "T-room" environment in Figure 1. Suppose the robot has to move between the two indicated configurations in Figure 1(a): from start configuration α to goal configuration β. There is an obvious path from α to β as illustrated in Figure 1(b): the robot origin moves directly from its start to goal positions, while the link angles simultaneously adjust to their goal angles. However, such paths require the two links to cross each other. To achieve a "non-crossing" solution from α to β, we need a less obvious path such as found by our algorithm in Figure 1(c): the robot origin must first move *away* from the goal, towards the T-junction, in order to maneuver the 2 links into an appropriate relative order before it can move toward the goal configuration.

We had chosen $\varepsilon = 2$ in Figure 1(b,c); furthermore, $\kappa = 7$ for the non-crossing instance. But if we increase either ε to 3 or κ to 8, then the non-crossing instance would report NO-PATH. It is important to know that the NO-PATH output from resolution-exact algorithms is not never due to exhaustion ("time-out"). It is a principled answer, guaranteeing the non-existence of paths with clearance $> K \cdot \varepsilon$ (for some $K > 1$ depending on the algorithm). In our view, the narrow passage problem is, in the limit, just the **halting problem** for path planning: algorithms with narrow passage problems will also have problems with halting when there is no path. Our experiments suggests that the "narrow passage problem" is non-existent for our particular 4DOF robot, but we can easily create scenarios that is difficult for sampling approaches. But no amount of experimental data can truly express the conceptual gap between the various sampling heuristics and the *a priori* guaranteed methods such as ours.

The T-Room Environment has trivial combinatorial complexity, designed to illustrate the non-crossing phenomenon. But our algorithm scales well with the combinatorial complexity of the environment; all our solutions are "realtime". An interesting environment is Figure 2 with 100 randomly generated triangles.

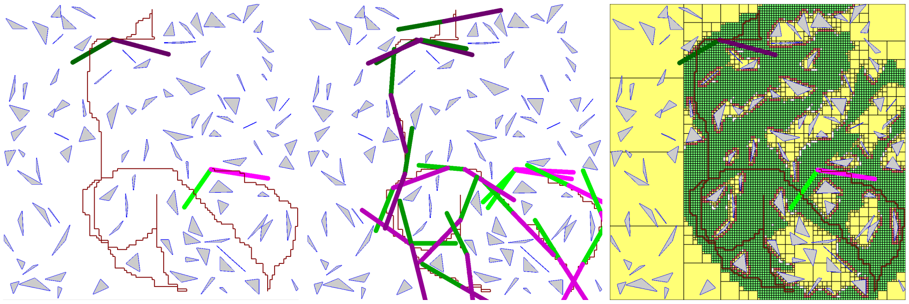

(a) Trace of the robot origin (b) Sub-sampled path (c) Subdivision boxes

Fig. 2: 100 Random Triangles Environment: non-crossing path found ($\kappa = 115°$)

Literature Review. Our theory of resolution-exactness and SSS algorithms apply to any robot system but we focus on algorithmic techniques to achieve the best algorithms for a 2-link robot. The main competition is from sampling approaches – see [4] for a survey. Besides PRM, we look at variants such Toggle PRM [4] as well as the lazy version [5]. RRT is another family of sampling methods with variants such as RRT-connect [8] or retraction-RRT [12]. The latter is comparable to Toggle PRM's exploitation of non-free configurations, except that retraction focuses on contact configurations. Salzman et al [13] introduce a "tiling technique" for handling non-crossing links in sampling algorithms that is quite different from our subdivision solution [11].

Overview of Paper. Section 2 describes our parametrization of the configuration space of R_2, and implicitly, its free space. Section 3 analyzes the forbidden angles of thick links. Section 4 shows our subdivision representation of the non-crossing configuration space. Section 5 describes our experimental results. Section 6 concludes the paper. Omitted proofs and additional experimental data are available as an appendix in the full version of this paper.

2 Configuration Space of Non-Crossing 2-Link Robot

The configuration space of R_2 is $C_{space} := \mathbb{R}^2 \times \mathbb{T}^2$ where $\mathbb{T}^2 = S^1 \times S^1$ is the torus and $S^1 = SO(2)$ is the unit circle. We represent S^1 by the interval $[0, 2\pi]$ with the identification $0 = 2\pi$. Closed angular intervals of S^1 are denoted by $[s, t]$ where $s, t \in [0, 2\pi]$ using the convention

$$[s,t] := \begin{cases} \{\theta : s \leq \theta \leq t\} & \text{if } \ s \leq t, \\ [s, 2\pi] \cup [0, t] & \text{if } \ s > t. \end{cases}$$

In particular, $[0, 2\pi] = S^1$ and $[2\pi, 0] = [0, 0]$. The standard Riemannian metric $d : S^1 \times S^1 \to \mathbb{R}_{\geq 0}$ on S^1 is given by $d(\theta, \theta') = \min\{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}$. Thus $0 \leq d(\theta, \theta') \leq \pi$.

To represent the non-crossing configuration space, we must be more specific about interpreting the parameters in a configuration $(x, y, \theta_1, \theta_2) \in C_{space}$: there are two distinct interpretations, depending on whether R_2 is viewed as a chain robot or a spider robot. We choose the latter view: then (x, y) is the **footprint** of the joint A_0 at the center of the spider and θ_1, θ_2 are the independent angles of the two links. This has some clear advantage over viewing R_2 as a chain robot, but we can conceive of other advantages for the chain robot view. That will be future research. In the terminology of [11], the robot R_2 has three named points A_0, A_1, A_2 whose **footprints** at configuration $\gamma = (x, y, \theta_1, \theta_2)$ are given by

$$A_0[\gamma] := (x, y), \quad A_1[\gamma] := (x, y) + \ell_1(\cos\theta_1, \sin\theta_1), \quad A_2[\gamma] := (x, y) + \ell_2(\cos\theta_2, \sin\theta_2).$$

The **thin footprint** of R_2 at γ, denoted $R_2[\gamma]$, is defined as the union of the line segments $[A_0[\gamma], A_1[\gamma]]$ and $[A_0[\gamma], A_2[\gamma]]$. The **thick footprint** of R_2 is given by $Fprint_\tau(\gamma) := D(\mathbf{0}, \tau) \oplus R_2[\gamma]$, the Minkowski sum $\oplus$ of the thin footprint with disc $D(\mathbf{0}, \tau)$ of radius τ centered at $\mathbf{0}$.

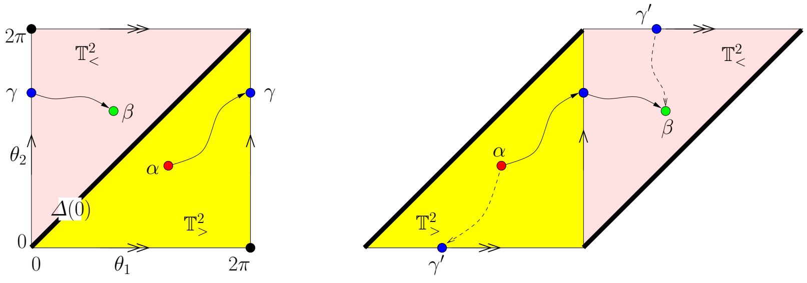
The **non-crossing configuration space** of bandwidth κ is defined to be

$$C_{space}(\kappa) := \mathbb{R}^2 \times (\mathbb{T}^2 \setminus \Delta(\kappa))$$

where $\Delta(\kappa)$ is the **diagonal band** $\Delta(\kappa) := \left\{(\theta, \theta') \in \mathbb{T}^2 : d(\theta, \theta') \leq \kappa\right\} \subseteq \mathbb{T}^2$. Note three special cases:

- If $\kappa < 0$ then $\Delta(\kappa)$ is the empty set.
- If $\kappa = 0$ then $\Delta(\kappa)$ is a closed curve in $\mathbb{T}^2$.
- If $\kappa \geq \pi$ then $\Delta(\kappa) = S^1$.

Configurations in $\mathbb{R}^2 \times \Delta(0)$ are said to be **self-crossing**; all other configurations are **non-crossing**. Here we focus on the case $\kappa \geq 0$. For our subdivision below, we will split $\mathbb{T}^2 \backslash \Delta(0)$ into two connected sets: $\mathbb{T}^2_< := \{(\theta, \theta') \in \mathbb{T}^2 : 0 \leq \theta < \theta' < 2\pi\}$ and $\mathbb{T}^2_> := \{(\theta, \theta') \in \mathbb{T}^2 : 0 \leq \theta' < \theta < 2\pi\}$. For $\kappa \geq 0$, the diagonal band $\Delta(\kappa)$ retracts to the closed curve $\Delta(0)$. In $\mathbb{R}^2$, if we omit such a set, we will get two connected components. In contrast, that $\mathbb{T}^2 \setminus \Delta(\kappa)$ remains connected. CLAIM: $\mathbb{T}^2 \setminus \Delta(\kappa)$ *is topologically a cylinder with two boundary components.* The point is that the non-crossing constraint has changed the topology of the configuration space. To see claim, consider the standard model of $\mathbb{T}^2$ represented by a square with opposite sides identified as in Figure 3(a) (we show the case $\kappa = 0$). By rearranging the two triangles $\mathbb{T}^2_<$ and $\mathbb{T}^2_>$ as in Figure 3(b), our claim is now visually obvious.



(a) (b)

Fig. 3: Paths in $\mathbb{T}^2 \setminus \Delta(0)$ from $\alpha \in \mathbb{T}^2_>$ to $\beta \in \mathbb{T}^2_<$

3 Forbidden Angle Analysis of Thick Links

Towards the development of a soft-predicate for thick links, we must first extend our analysis in [11] which introduced the concept of forbidden angles for thin links. Let $L(\ell, \tau)$ be a single link robot of length $\ell > 0$ and thickness $\tau \geq 0$. Its configuration space is $SE(2) = \mathbb{R}^2 \times S^1$. Given a configuration $(b, \theta) \in SE(2)$, the **footprint** of $L(\ell, \tau)$ at (b, θ) is

$$Fprint_{\ell,\tau}(b, \theta) := L \oplus D(\mathbf{0}, \tau)$$

where $\oplus$ denotes Minkowski sum, L is the line segment $[b, b + \ell(\cos\theta, \sin\theta)]$ and $D(\mathbf{0}, \tau)$ is the disk as above. When ℓ, τ is understood, we simply write "$Fprint(b, \theta)$" instead of $Fprint_{\ell,\tau}(b, \theta)$.

Let $S, T \subseteq \mathbb{R}^2$ be closed sets. An angle θ is **forbidden** for (S,T) if there exists $s \in S$ such that $Fprint(s,\theta) \cap T$ is non-empty. If $t \in Fprint(s,\theta) \cap T$, then the pair $(s,t) \in S \times T$ is a **witness** for the forbidden-ness of θ for (S,T). The set of forbidden angles of (S,T) is called the **forbidden zone** of S, T and denoted $\mathrm{Forb}_{\ell,\tau}(S,T)$. Clearly, $\theta \in \mathrm{Forb}_{\ell,\tau}(S,T)$ iff there exists a witness pair $(s,t) \in S \times T$. Moreover, we call (s,t) a **minimum witness** of θ if the Euclidean norm $\|s-t\|$ is minimum among all witnesses of θ. If (s,t) is a minimum witness, then clearly $s \in \partial S$ and $t \in \partial T$.

Lemma 1. *For any sets $S, T \subseteq \mathbb{R}^2$, we have*

$$\mathrm{Forb}_{\ell,\tau}(S,T) = \pi + \mathrm{Forb}_{\ell,\tau}(T,S).$$

Proof. For any pair (s,t) and any angle α, we see that

$$t \in Fprint(s,\alpha) \text{ iff } \; s \in Fprint(t, \pi+\alpha).$$

Thus, there is a witness (s,t) for α in $\mathrm{Forb}_{\ell,\tau}(S,T)$ iff there is a witness (t,s) for $\pi+\alpha$ in $\mathrm{Forb}_{\ell,\tau}(T,S)$. The lemma follows. **Q.E.D.**

¶1. The Forbidden Zone of two points Consider the forbidden zone $\mathrm{Forb}_{\ell,\tau}(V,C)$ defined by two points $V, C \in \mathbb{R}^2$ with $d = \|V - C\|$. (The notation V suggests a vertex of a translational box B^t and C suggests a corner of the obstacle set.) In our previous paper [11] on thin links (i.e., $\tau = 0$), this case is not discussed for reasons of triviality. When $\tau > 0$, the set $\mathrm{Forb}_{\ell,\tau}(V,C)$ is more interesting. Clearly, $\mathrm{Forb}_{\ell,\tau}(V,C)$ is empty iff $d > \ell + \tau$ (and a singleton if $d = \ell + \tau$). Also $\mathrm{Forb}_{\ell,\tau}(V,C) = S^1$ iff $d \leq \tau$. Henceforth, we may assume

$$\tau < d < \ell + \tau. \tag{1}$$

The forbidden zone of V, C can be written in the form

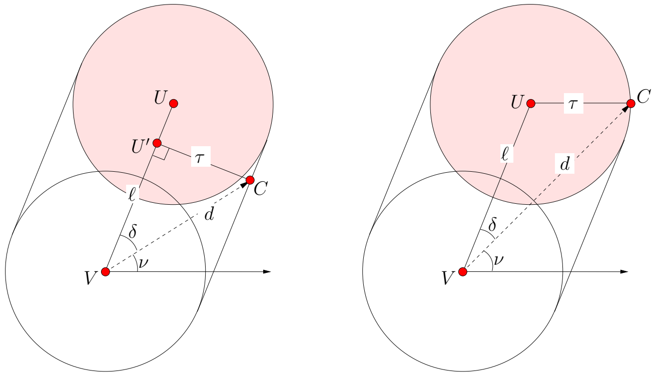
$$\mathrm{Forb}_{\ell,\tau}(V,C) := [\nu - \delta, \nu + \delta]$$

for some ν, δ. Call ν the **nominal angle** and δ the **correction angle**. By the symmetry of the footprint, ν is equal to $\theta(V,C)$ (see Figure 4).

It remains to determine δ. Consider the configuration $(V,\theta) \in SE(2)$ of our link $L(\ell,\tau)$ where link origin is at V and the link makes an angle θ with the positive x-axis. The angle δ is determined when the point C lies on the boundary of $Fprint(V,\theta)$. The two cases are illustrated in Figure 4 where $\theta = \nu + \delta$ and other endpoint of the link is U; thus $\|VU\| = \ell$ and $\|VC\| = d$, and $\delta = \angle(CVU)$. Under the constraint (1), there are two ranges for d:

(a) d is short: $d^2 \leq \tau^2 + \ell^2$. In this case, the point C lies on the straight portion of the boundary of the footprint, as in Figure 4(a). From the right-angle triangle $CU'V$, we see that $\delta = \arcsin(\tau/d)$.

(b) d is long: $d^2 > \tau^2 + \ell^2$. In this case, the point C lies on the circular portion of the boundary of the footprint, as in Figure 4(b). Consider the triangle



(a)

(b)

Fig. 4: $\mathrm{Forb}_{\ell,\tau}(V,C)$

CUV with side lengths of d, ℓ, τ. By the cosine law, $\tau^2 = d^2 + \ell^2 - 2d\ell\cos\delta$ and thus

$$\delta = \arccos\left(\frac{\ell^2 + d^2 - \tau^2}{2d\ell}\right).$$

This proves:

Lemma 2. *Assume $\|VC\| = d$ satisfies (1). Then*

$$\mathrm{Forb}_{\ell,\tau}(V,C) = [\nu - \delta, \nu + \delta]$$

where $\nu = \theta(V,C)$ and

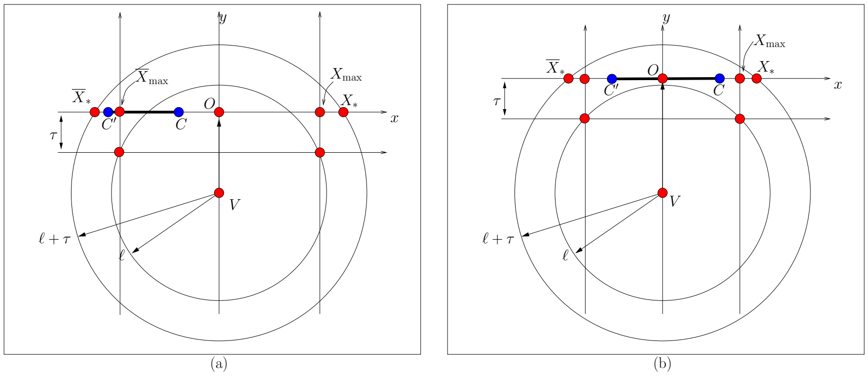
$$\delta = \delta(V,C) = \begin{cases} \arcsin(\tau/d) & \text{if } d^2 \le \tau^2 + \ell^2, \\ \arccos\left(\frac{\ell^2+d^2-\tau^2}{2d\ell}\right) & \text{if } d^2 > \tau^2 + \ell^2. \end{cases} \tag{2}$$

¶2. The Forbidden Zone of a Vertex and a Wall Recall that the boundary of a box B^t is divided into four **sides**, and two adjacent sides share a common endpoint which we call a **vertex**. We now determine $\mathrm{Forb}_{\ell,\tau}(V,W)$ where V is a vertex and W a wall feature. Choose the coordinate axes such that W lies on the x-axis, and $V = (0, -\sigma)$ lies on the negative y-axis, for some $\sigma > 0$. Let the two corners of W be C, C' with C' lying to the left of C. See Figure 5.

We first show that the interesting case is when

$$\tau < \sigma < \ell + \tau. \tag{3}$$

If $\sigma \ge \ell + \tau$ then $\mathrm{Forb}_{\ell,\tau}(V,W)$ is either a singleton ($\sigma = \ell + \tau$) or else is empty ($\sigma > \ell + \tau$). Likewise, the following lemma shows that when $\sigma \le \tau$, we are to point-point case of Lemma 2:



(a)

(b)

Fig. 5: Stop Analysis for $\mathrm{Forb}_{\ell,\tau}(V,W)$ (assuming $\sigma > \tau$)

Lemma 3. *Assume $\sigma \le \tau$. We have*

$$\mathrm{Forb}_{\ell,\tau}(V,W) = \begin{cases} S^1 & \text{if } \ D(V,\tau)\cap W \neq \emptyset, \\ \mathrm{Forb}_{\ell,\tau}(V,c) & \text{else} \end{cases}$$

where $c = C$ or C'.

Proof. Recall that we have chosen the coordinate system so that W lies on the x-axes and $V = (0,-\sigma)$. It is easy to see that $\mathrm{Forb}_{\ell,\tau}(V,W) = S^1$ iff the disc $D(V,\tau)$ intersects W. So assume otherwise. In that case, the closest point in W to V is c, one of the two corners of W. The lemma is proved if we show that

$$\mathrm{Forb}_{\ell,\tau}(V,W) = \mathrm{Forb}_{\ell,\tau}(V,c).$$

It suffices to show $\mathrm{Forb}_{\ell,\tau}(V,W) \subseteq \mathrm{Forb}_{\ell,\tau}(V,c)$. Suppose $\theta \in \mathrm{Forb}_{\ell,\tau}(V,W)$. So it has a witness (V,c') for some $c' \in W$. However, we see that the minimal witness for this case is (V,c). This proves that $\theta \in \mathrm{Forb}_{\ell,\tau}(V,c)$. **Q.E.D.**

In addition to (3), we may also assume the wall lies within the annulus of radii $(\tau, \tau+\ell)$ centered at V:

$$\|VC\|, \|VC'\| \in (\tau, \ell+\tau) \tag{4}$$

Using the fact that $V = (0,-\sigma)$ and W lies in the x-axis, we have:

Lemma 4. *Assume (3) and (4).*
Then $\mathrm{Forb}_{\ell,\tau}(V,W)$ is a non-empty connected interval of S^1,

$$\mathrm{Forb}_{\ell,\tau}(V,W) = [\alpha,\beta] \subseteq (0,\pi).$$

Our next goal is to determine the angles α, β in this lemma. Consider the footprints of the link at the extreme configurations $(V,\alpha), (V,\beta) \in SE(2)$. Clearly,

W intersects the boundary (but not interior) of these footprints, $Fprint(V,\alpha)$ and $Fprint(V,\beta)$. Except for some special configurations, these intersections are singleton sets. Regardless, pick any $A \in W \cap Fprint(V,\alpha)$ and $B \in W \cap Fprint(V,\beta)$. Since α is an endpoint of $\mathrm{Forb}_{\ell,\tau}(V,W)$, we see that $A \in (\partial W) \cap \partial(Fprint(V,\alpha))$. We call A a **left stop** for the pair (V,W) because[2] for any $\delta' > 0$ small enough, $A \in Fprint(V,\alpha+\delta')$ while $W \cap (V,\alpha-\delta') = \emptyset$. Similarly the point B is called a **right stop** for the pair (V,W). Clearly, we can write

$$\alpha = \theta(V,A) - \delta(V,A), \qquad \beta = \theta(V,B) + \delta(V,B)$$

where $\delta(V,\cdot)$ is given by Lemma 2. We have thus reduced the determination of angles α and β to the computation of the left A and right B stops.

We might initially guess that the left stop of (V,W) is C, and right stop of (V,W) is C'. But the truth is a bit more subtle. Define the following points $X_*, X_{\max}$ on the positive x-axis using the equation:

$$\begin{aligned}\|OX_*\| &= \sqrt{(\ell+\tau)^2 - \sigma^2},\\ \|OX_{\max}\| &= \sqrt{\ell^2 - (\sigma-\tau)^2}.\end{aligned}$$

These two points are illustrated in Figure 5. Also, let $\overline{X}_*$ and $\overline{X}_{\max}$ be mirror reflections of X_* and $X_{\max}$ across the y-axis. The points $X_*, \overline{X}_*$ are the two points at distance $\ell+\tau$ from V. The points $X_{\max}, \overline{X}_{\max}$ are the left and right stops in we replace W by the infinite line through W (i.e., the x-axis).

With the natural ordering of points on the x-axis, we can show that

$$\overline{X}_* < \overline{X}_{\max} < O < X_{\max} < X_*$$

where O is the origin. Since $\|VC\|$ and $\|VC'\|$ lie in $(\tau, \tau+\ell)$, it follows that

$$\overline{X}_* < C' < C < X_*.$$

Two situations are shown in Figure 5. The next lemma is essentially routine, once the points $X_{\max}, \overline{X}_{\max}$ have defined:

Lemma 5. *Assume (3) and (4).*
The left stop of (V,W) *is*

$$\begin{cases} C' & \text{if } X_{\max} \le C' \qquad (L1)\\ X_{\max} & \text{if } C' < X_{\max} < C \qquad (L2)\\ C & \text{if } C \le X_{\max} \qquad (L3)\end{cases}$$

The right stop of (V,W) *is*

$$\begin{cases} C & \text{if } C \le \overline{X}_{\max} \qquad (R1)\\ \overline{X}_{\max} & \text{if } C' < \overline{X}_{\max} < C \qquad (R2)\\ C' & \text{if } \overline{X}_{\max} \le C' \qquad (R3)\end{cases}$$

[2] Intuitively: At configuration (V,α), the single-link robot can rotate about V to the right, but if it tries to rotate to the left, it is "stopped" by A.

The cases (L1-3) and (R1-3) in this lemma suggests 9 combinations, but 3 are logically impossible: (L1-R1), (L1-R2), (L2-R1). The remaining 6 possibilities for left and right stops are summarized in the following table:

	(R1)	(R2)	(R3)
(L1)	*	*	(C', C')
(L2)	*	$(X_{\max}, \overline{X}_{\max})$	$(X_{\max}, C')$
(L3)	(C, C)	$(C, \overline{X}_{\max})$	(C, C')

Observe the extreme situations (L1-R3) or (L3-R1) where the the left and right stops are equal to the same corner, and we are reduced to the point-point analysis. Once we know the left and right stops for (V, W), then we can use Lemma 2 to calculate the angles α and β.

¶3. The Forbidden Zone of a Side and a Corner We now consider the forbidden zone $\text{Forb}_{\ell,\tau}(S, C)$ where S is a side and C a corner feature. Note that is complementary to the previous case of $\text{Forb}_{\ell,\tau}(V, W)$ since C and V are points and S and W are line segments. We can exploit the principle of reflection symmetry of Lemma 1:

$$\text{Forb}_{\ell,\tau}(S, C) = \pi + \text{Forb}_{\ell,\tau}(C, S)$$

where $\text{Forb}_{\ell,\tau}(C, S)$ is provided by previous Lemma (with C, S instead of V, W).

¶4. Cone Decomposition We have now provided formulas for computing sets of the form $\text{Forb}_{\ell,\tau}(V, W)$ or $\text{Forb}_{\ell,\tau}(S, C)$; such sets are called **cones**. We now address the problem of computing $\text{Forb}_{\ell,\tau}(B^t, W)$ where $B^t \subseteq \mathbb{R}^2$ is a (translational) box. We show that this set of forbidden angles can be written as the union of at most 3 cones, generalizes a similar result in [11]. Towards such a cone decomposition, we first classify the disposition of a wall W relative to a box B^t. There is a preliminary case: if W intersects $B^t \oplus D(0, \tau)$, then we have

$$\text{Forb}_{\ell}(B^t, W) = S^1.$$

Call this **Case (0)**. Assuming W does not intersect $B^t \oplus D(0, \tau)$, there are three other possibilities, **Cases (I-III)** illustrated Figure 6.

We first need a notation: if $S \subseteq \partial(B^t)$ is a side of the box B^t, let $H(S)$ denote the open half-space which is disjoint from B^t and is bounded by the line through S. Then we have these three cases:

(I) $W \subseteq H(S)$ for some side s of box B^t.

(II) $W \subseteq H(S) \cap H(S')$ for two adjacent sides S, S' of box B^t.

(III) None of the above. This implies that $W \subseteq H(S) \cup H(S')$ for two adjacent sides S, S' of box B^t.

Theorem 1. $\text{Forb}_{\ell,\tau}(B^t, W)$ *is the union of at most three thick cones.*

Sketch proof: we try to reduce the argument to the case $\tau = 0$ which is given in [11]. In that case, we could write

$$\text{Forb}_{\ell}(B^t, W) = C_1 \cup C_2 \cup C_3$$

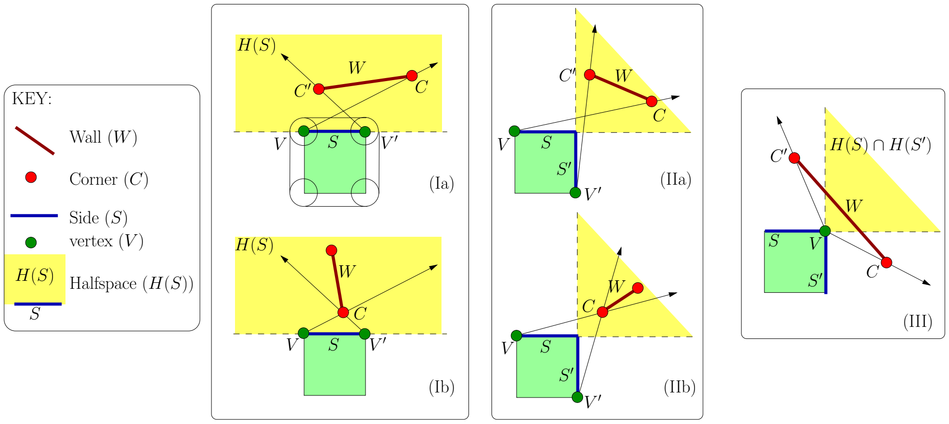


Fig. 6: Cases (I-III) of $\mathrm{Forb}_{\ell,\tau}(B^t, W)$

where each C_i is a thin cone or an empty set. In the non-empty case, the cone C_i has the form $\mathrm{Forb}_\ell(S_i, T_i)$ where $S_i \subseteq \partial B^t, T_i \subseteq W$. The basic idea is that we now "transpose" $\mathrm{Forb}_\ell(S_i, T_i)$ to the thick version $C'_i := \mathrm{Forb}_{\ell,\tau}(S_i, T_i)$. In case C_i is empty, C'_i remains empty. Thus we would like to claim that

$$\mathrm{Forb}_\ell(B^t, W) = C'_1 \cup C'_2 \cup C'_3.$$

This is almost correct, except for one issue. It is possible that some C_i is empty, and yet its transpose C'_i is non empty. In the full paper, we will fill in these detail.

Remark: in case of thin cones, the C_i's are non-overlapping (i.e., they may only share endpoints). But for thick cone decomposition, the cones will in general overlap.

4 Subdivision for Thick Non-Crossing 2-Link Robot

A resolution-exact planner for a thin self-crossing 2-link robot was described in [11]. We now extend that planner to the thick non-crossing case.

We will briefly review the ideas of the algorithm for the thin self-crossing 2-link robot. We begin with a box $B_0 \subseteq \mathbb{R}^2$ and it is in the subspace $B_0 \times \mathbb{T}^2 \subseteq C_{space}$ where our planning problem takes place. We are also given a polygonal obstacle set $\Omega \subseteq \mathbb{R}^2$; we may decompose its boundary $\partial\Omega$ into a disjoint union of corners (=points) and edges (=open line segments) which are called (boundary) **features**. Let $B \subseteq C_{space}$ be a box; there is an exact classification of B as $C(B) \in \{\mathtt{FREE}, \mathtt{STUCK}, \mathtt{MIXED}\}$ relative to Ω. But we want a soft classification $\widetilde{C}(B)$ which is correct whenever $\widetilde{C}(B) \neq \mathtt{MIXED}$, and which is equal to $C(B)$ when the width of B is small enough. Our method of computing $\widetilde{C}(B)$ is based on computing a set $\phi(B)$ of features that are relevant to B. A box $B \subseteq C_{space}$

may be written as a Cartesian product $B = B^t \times B^r$ of its translational subbox $B^t \subseteq \mathbb{R}^2$ and rotational subbox $B^r \subseteq \mathbb{T}^2$. In the T/R splitting method (simple version), we split B^t until the width of B^t is $\le \varepsilon$. Then we do a single split of the rotational subbox B^r into all the subboxes obtained by removing all the forbidden angles determined by the walls and corners in $\widetilde{\phi}(B^t)$. This "rotational split" of B^r is determined by the obstacles, unlike the "translational splits" of B^t.

¶5. Boxes for Non-Crossing Robot. Our basic idea for representing boxes in the non-crossing configuration space $C_{space}(\kappa)$ is to write it as a pair $(B, \mathtt{XT})$ where $\mathtt{XT} \in \{\mathtt{LT}, \mathtt{GT}\}$, and $B \subseteq C_{space}$. The pair $(B, \mathtt{XT})$ represents the set $B \cap (\mathbb{R}^2 \times \mathbb{T}^2_{\mathtt{XT}})$ (with the identification $\mathbb{T}^2_{\mathtt{LT}} = \mathbb{T}^2_{<}$ and $\mathbb{T}^2_{\mathtt{GT}} = \mathbb{T}^2_{>}$). It is convenient to call $(B, \mathtt{XT})$ an X**-box** since they are no longer "boxes" in the usual sense.

An angular interval $\Theta \subseteq S^1$ that[3] contains a open neighborhood of $0 = 2\pi$ is said to be **wrapping**. Also, call $B^r = \Theta_1 \times \Theta_2$ wrapping if either Θ_1 or Θ_2 is wrapping. Given any B^r, we can decompose the set $B^r \cap (\mathbb{T}^2 \setminus \Delta(\kappa))$ into the union of two subsets $B^r_{\mathtt{LT}}$ and $B^r_{\mathtt{GT}}$, where $B^r_{\mathtt{XT}}$ denote the set $B^r \cap \mathbb{T}^2_{\mathtt{XT}}$. In case B^r is non-wrapping, this decomposition has the nice property that each subset $B^r_{\mathtt{XT}}$ is connected. For this reason, we prefer to work with non-wrapping boxes. Initially, the box $B^r = \mathbb{T}^2$ is wrapping. The initial split of $\mathbb{T}^2$ should be done in such a way that the children are all non-wrapping: the "natural" (quadtree-like) way to split $\mathbb{T}^2$ into four congruent children has[4] this property. Thereafter, subsequent splitting of these non-wrapping boxes will remain non-wrapping.

Of course, $B^r_{\mathtt{XT}}$ might be empty, and this is easily checked: say $\Theta_i = [s_i, t_i]$ $(i = 1, 2)$. Then $B^r_{<}$ is empty iff $t_2 \le s_1$. and $B^r_{>}$ is empty iff $s_2 \ge t_1$. Moreover, these two conditions are mutually exclusive.

We now modify the algorithm of [11] as follows: as long as we are just splitting boxes in the translational dimensions, there is no difference. When we decide to split the rotational dimensions, we use the T/R splitting method of [11], but each child is further split into two X-boxes annotated by LT or GT (they are filtered out if empty). We build the connectivity graph G (see Appendix A) with these X-boxes as nodes. This ensures that we only find non-crossing paths. Our algorithm inherits resolution-exactness from the original self-crossing algorithm.

The predicate $\mathtt{isBoxEmpty}(B^r, \kappa, \mathtt{XT})$ which returns true iff $(B^r_{\mathtt{XT}}) \cap (\mathbb{T}^2 \setminus \Delta(\kappa))$ is empty is useful in implementation. It has a simple expression when restricted to non-wrapping translational box B^r:

Lemma 6.
Let $B^r = [a, b] \times [a', b']$ be a non-wrapping box.
(a) $\mathtt{isBoxEmpty}(B^r, \kappa, \mathtt{LT}) = \mathtt{true}$ *iff* $\kappa \ge b' - a$ *or* $2\pi - \kappa \le a' - b$.
(b) $\mathtt{isBoxEmpty}(B^r, \kappa, \mathtt{GT}) = \mathtt{true}$ *iff* $\kappa \ge b - a'$ *or* $2\pi - \kappa \le a - b'$.

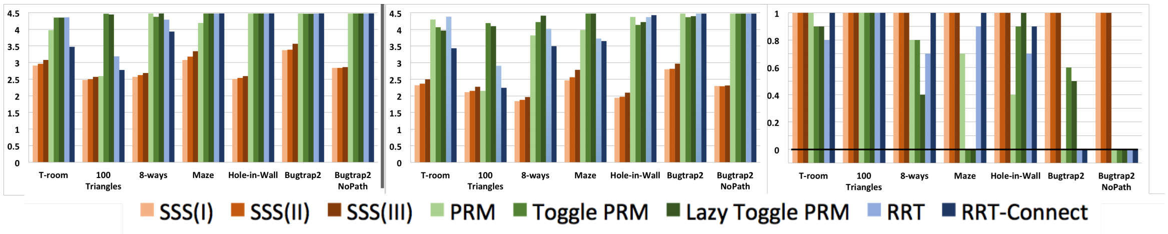
[3] Wrapping intervals are either equal to S^1 or has the form $[s, t]$ where $2\pi > s > t > 0$.

[4] This is not a vacuous remark – the quadtree-like split is determined by the choice of a "center" for splitting. To ensure non-wrapping children, this center is necessarily $(0, 0)$ or equivalently $(2\pi, 2\pi)$. Furthermore, our T/R splitting method (to be introduced) does not follow the conventional quadtree-like subdivision at all.

5 Implementation and Experiments

We implemented our thick non-crossing planner in `C++` and OpenGL on the Qt platform (the Qt part is new). A preliminary heuristic version appeared [11, 10]. Our code, data and experiments are distributed[5] with our open source `Core Library`. To evaluate our planner, we compare it with several sampling algorithms in the open source OMPL [15]. Based on a referee's suggestion, we also implemented the 2-link (crossing and non-crossing) versions of Toggle PRM and Lazy Toggle PRM (in lieu of publicly available code). We benefited greatly from the advice of Prof. Denny in our best effort implementation. The platform for our experiments is Mac OS X 10.12 (Sierra) on MacBook Pro. The processor is a 2.5 GHz Intel Core i7 with 16GB DDR3-1600 MHz RAM and 500GB Flash Storage.

Tables 1 and 2 summarize the results of two groups of experiments, which we call "Narrow Passages" and "Easy Passages". Each row in the tables represent an experiment (see below), and each column represents a planner. There are 8 planners: 3 versions of SSS, 3 versions of PRM and 2 versions of RRT. For reasons of space, only Table 1 is listed below. Our full paper has an appendix with Table 2 (as well as other experimental results). For visualization, we extract 3 bar charts from these tables in Figure 7, showing the average times of the planners in Narrow and Easy Passages, and also success rate on Easy Passages. Conclusion: (1) SSS is at least an order of magnitude faster than each sampling method, and (2) success rates of RRT-connect and Toggle PRM is best among sampling methods, but both are inferior to SSS.



(a) Narrow Passages: Average Time (msec) in Log10 Scale

(b) Easy Passages: Average Time (msec) in Log10 Scale

(c) Easy Passages: Success Rate

Fig. 7: Bar Charts of Average Times and Success Rates

Two general remarks are in order. First, as in our previous work, we implemented several search strategies in SSS. But for simplicity, we only use the Greedy Best First (GBF) strategy in our experiments; GBF is typically our best strategy. Next, OMPL does not natively support articulated robots such as R_2. So in the experiments of Tables 1 and 2, we artificially set $\ell_2 = 0$ for all the sampling algorithms (so that they are effectively one-link thick robots). In the

[5] http://cs.nyu.edu/exact/core/download/.

SSS versions, we set $\ell_2 = 0$ for SSS(I) but SSS(II) and SSS(III) represent (resp.) crossing and non-crossing robots where $\ell_2 > 0$.

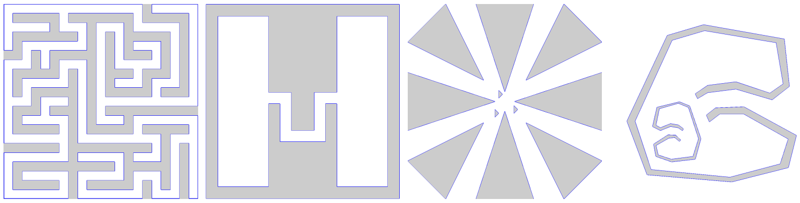

(a) Maze (b) Hole-in-Wall (c) 8-Way Corridor (d) Bugtrap2

Fig. 8: Some environments in our experiments

Reading the Tables: Each experiment (i.e., row) corresponds to a fixed environment, robot parameters, initial and goal configurations. Figure 8 depicts these environments (save for the T-Room and 100 triangles from the introduction). We name the experiments after the environment. E.g., column 1 for the Maze experiment, tells us that $\ell_1 = 16, \tau = 10$. The last 2 experiments use the "double bugtrap" environment, but one of them has parameters that ensures NO-PATH. For each experiment, we perform 40 runs of the following planners: PRM, RRT, RRT-connect (all from OMPL), SSS(I-III) and Toggle PRM and Lazy Toggle PRM (our implementation). Each planner produces 4 statistics:

Average Time / Best Time / Standard Deviation / Success Rate,

abbreviated as **Avg**/**Best**/**STD**/**Success**, respectively. Success Rate is the fraction of the 40 runs for which the planner finds a path (assuming there is one) out of 40 runs. But if there is no path, our SSS planner will always discover this, so its **Success** is 1; simultaneously, the sampling methods will time out and hence their **Success** is 0. All timing is in milliseconds (msec). Column 2 contains the **Record Statistics**, i.e., the row optimum for these 4 statistics. E.g., the Record Statistics for the T-Room experiment is **815.9/743.6/21.9/1**. This tells us the row optimum for **Avg** is 815.9 ms, for **Best** is 743.6 ms, for **STD** is 21.9 ms, and for **Success** is 1. "Optimum" for the first 3 (resp., last) statistics means minimum (resp., maximum) value. The 4 optimal values may be achieved by different planners. In the rest of the Table, we have one column for each Planner, showing the ratio of the planner's statistics relative to the Record Statistics. The best performance is always indicated by the ratio of 1. E.g., for T-Room experiment, the row maximum for **Success** is 1, and it is achieved by all SSS planners and RRT-Connect. The row minimum for **Avg**, **Best** and **STD** are achieved by SSS(I), RRT-Connect and SSS(II), resp. We regard the achievement of row optimum for **Success** and **Avg** (in that order) to be the main indicator of superiority. Table 1 (and Table 2) show that our planner is consistently superior to sampling planners.

Environment (ℓ_1, τ)	Record Statistics (Avg./Best/STD/Success)	Ratios Relative to Record Statistics							
		SSS (I) (ℓ_2: 0)	SSS (II) (ℓ_2: 20, 10, 10, 10, 10, 5, 5)	SSS (III) (non-crossing)	PRM	Toggle PRM	Lazy Toggle PRM	RRT	RRT-Connect
T-Room (120, 24)	815.9/743.6/21.9/1	1/1/1.3/1	1.1/1.1/1/1	1.5/1.5/1.4/1	11.6/1.4/414.7/0.9	27.6/3.6/468.1/0.4	27.4/1.8/539.1/0.3	28/1.1/571.9/0.3	3.6/1/213.5/1
100 Triangles (35, 20)	301.1/268.4/18.7/1	1/1/2/1	1/1.1/1/1	1.2/1.2/3.2/1	1.3/1.3/41.5/1	98.2/89.2/74.8/0.1	92.5/60.1/225.6/0.3	5.1/3.7/28.3/1	2/1.2/12.9/1
8-Ways (48, 11)	370.8/329.3/17.4/1	1/1/3.8/1	1.1/1.1/2.1/1	1.3/1.4/1/1	81.1/x/x/0	64.6/x/x/0	80.9/x/x/0	52.5/2.4/728.9/0.5	23/2.4/302.3/1
Maze (16, 10)	1193.2/1137.1/26.8/1	1/1/1/1	1.3/1.3/1.3/1	1.8/1.8/2.5/1	13.1/8.4/211.9/0	25.1/x/x/0	25.1/x/x/0	25.2/x/x/0	25.2/x/x/0
Hole-in-Wall (50, 12)	319.2/284.9/21.5/1	1/1/1.2/1	1.1/1.1/1/1	1.2/1.4/1.3/1	94.2/x/x/0.5	94/x/x/0	94/x/x/0	94.2/x/x/0	94.2/x/x/0
Bugtrap2 (36, 9)	2335/2213.2/75.2/1	1/1/1/1	1/1.1/1.2/1	1.6/1.5/4.9/1	12.9/x/x/0	12.5/9.5/27.9/0.2	12.5/11.2/22.9/0.3	12.9/x/x/0	12.9/x/x/0
Bugtrap2 NoPath (70, 8)	686.6/666.5/11.8/1	1/1/1.3/1	1/1.1/1/1	1.1/1.1/1.2/1	43.8/x/x/0	43.7/x/x/0	43.7/x/x/0	43.8/x/x/0	43.8/x/x/0

Table 1: Narrow Passages

Table 2 is basically the same as Table 1 except that we decrease the thickness τ in order to improve the success rate of the sampling methods. Our planner needs an ε parameter, which is set to 1 in Table 1 and 2 in Table 2 (this is reasonable in view of narrow passage demands). Sampling methods has many more tuning parameters; but we choose the defaults in OMPL because we saw no systematic improvements in our partial attempt to enhance their performance. In Toggle PRM, we use small k for k-nearest neighbors in the obstacle graph and the same default k as in OMPL in the free graph. We also set the time-out to be 30 seconds because the execution time for one table takes about 18 hours

6 Conclusion and Limitations

We have introduced a novel and efficient planner for non-crossing 2-link thick robots. Our work contributes to the development of practical and theoretically sound subdivision planners [16, 17]. It is reasonable to expect a tradeoff between the stronger guarantees of our resolution-exact approach versus a faster running time for sampling approaches. But our experiments suggest no such tradeoffs at all: SSS is consistently superior to sampling. Although we have been unable to improve the sampling planners by tuning their parameters, this is not an definitive conclusion. Unfortunately, there does not seem to be much basis for doing such tuning. Conventional wisdom maintains that subdivision will not scale to higher DOF's, and our current experiments are limited to at most 4DOF. We interpret this wisdom as telling us that new subdivision techniques (such as the T/R splitting idea) are needed to make higher DOF's robots perform in real-time. This is a worthy challenge for SSS which we plan to take up.

References

1. M. Barbehenn and S. Hutchinson. Toward an exact incremental geometric robot motion planner. In *Proc. IROS*, vol. 3, pp. 39–44, Pittsburgh, PA. 1995.

2. R. A. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. In *8th IJCAI – Vol. 2*, pp. 799–806, San Francisco, CA, USA, 1983. Morgan Kaufmann Pub. Inc.
3. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Boston, 2005.
4. J. Denny and N. M. Amato. Toggle PRM: A coordinated mapping of C-free and C-obstacle in arbitrary dimension. In *WAFR 2012*, volume 86 of *Springer Tracts in Advanced Robotics*, pp. 297–312. MIT, Cambridge, USA. June 2012.
5. J. Denny, K. Shi, and N. M. Amato. Lazy Toggle PRM: a Single Query approach to motion planning. In *Proc. ICRA*, pp. 2407–2414. 2013.
6. S. L. Devadoss and J. O'Rourke. *Discrete and Computatational Geometry.* Princeton University Press, 2011.
7. L. Kavraki, P. Švestka, C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics and Automation*, 12(4):566–580, 1996.
8. J. J. Kuffner Jr and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proc. ICRA*, vol. 2, pp. 995–1001. 2000.
9. S. M. LaValle. *Planning Algorithms.* Cambridge Univ. Press, Cambridge, 2006.
10. Z. Luo. Resolution-exact Planner for a 2-link planar robot using Soft Predicates. Master thesis, New York University, Courant Institute, Jan. 2014.
11. Z. Luo, Y.-J. Chiang, J.-M. Lien, and C. Yap. Resolution exact algorithms for link robots. In *Proc. 11th Intl. Workshop on Algorithmic Foundations of Robotics (WAFR '14)*, vol. 107 of *Springer Tracts in Advanced Robotics* , pp. 353–370, 2015. 3-5 Aug 2014, Istanbul, Turkey.
12. J. Pan, L. Zhang, and D. Manocha. Retraction-based rrt planner for articulated models. In *Proc. ICRA*, pages 2529–2536, 2010.
13. O. Salzman, K. Solovey, and D. Halperin. Motion planning for multi-link robots by implicit configuration-space tiling, 2015. CoRR abs/1504.06631.
14. M. Sharir, C. O'D'únlaing, and C. Yap. Generalized Voronoi diagrams for moving a ladder II: efficient computation of the diagram. *Algorithmica*, 2:27–59, 1987.
15. I. Şucan, M. Moll, and L. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.
16. C. Wang, Y.-J. Chiang, and C. Yap. On Soft Predicates in Subdivision Motion Planning. In *29th ACM Symp. on Comp. Geom.*, pages 349–358, 2013. SoCG'13, Rio de Janeiro, Brazil, June 17-20, 2013. Journal version in Special Issue of *CGTA*.
17. C. K. Yap. Soft Subdivision Search in Motion Planning. In A. Aladren et al., editor, *Proceedings, 1st Workshop on Robotics Challenge and Vision (RCV 2013)*, 2013. A Computing Community Consortium (CCC) **Best Paper Award**, Robotics Science and Systems Conference (RSS 2013), Berlin. In arXiv:1402.3213.
18. C. K. Yap. Soft Subdivision Search and Motion Planning, II: Axiomatics. In *Frontiers in Algorithmics*, volume 9130 of *Lecture Notes in Comp.Sci.*, pages 7–22. Springer, 2015. Plenary Talk at 9th FAW. Guilin, China. Aug 3-5, 2015.
19. J. Yu, C. Yap, Z. Du, S. Pion, and H. Bronnimann. Core 2: A library for Exact Numeric Computation in Geometry and Algebra. In *3rd Proc. Int'l Congress on Mathematical Software (ICMS)*, LNCS No. 6327, pages 121–141. Springer, 2010.
20. D. Zhu and J.-C. Latombe. New heuristic algorithms for efficient hierarchical path planning. *IEEE Transactions on Robotics and Automation*, 7:9–20, 1991.