IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

A General Approach to Path Planning Within C-Space Safe Corridors

Riccardo Abbati, Marco Riboli, Michel Rosselli, Alessandro Tasora, and Marco Silvestri

***Abstract*—This letter presents a novel strategy for collision-free path planning in robotic manipulators. The method operates in two stages: first, a sampling-based exploration of the configuration space is performed to construct a safe corridor composed of axis-aligned bounding boxes. Within this corridor, an optimisation-based trajectory generation phase addresses the channel problem by computing smooth joint trajectories as non-rational splines. A multi-objective cost function is minimised to reduce geometric acceleration along the path, while also maximising the distance from obstacles to improve safety margins. The proposed algorithm is general and applicable to a wide range of kinematic structures, and supports user-defined path degree and geometric continuity. Simulation results demonstrate superior performance compared to existing methods, and experimental validation further confirms its practical effectiveness. Our implementation is open-sourced and available on Github.**

***Index Terms*—Robot manipulator, collision avoidance, path planning, collision detection, quadratic programming, C-space.**

I. Introduction

PATH planning for robotic manipulators continues to represent a fundamental challenge, particularly in cluttered environments where industrial applications drive stringent demands on efficiency and reliability. Strategies for collision-free path planning can be broadly classified into sampling-based and optimization-based path planners [1]. Two established approaches falling under the first category include Probabilistic RoadMap methods (PRM) [2], [3] and Rapidly-exploring Random Trees (RRT) algorithms [4], with many variants proposed in the literature [5], [6], [7], [8], [9]. Both of these methods are based on the random exploration of the domain, but with substantially different approaches [10]. PRM initially generates a set of random points resulting in a graph structure representing the system (the roadmap), then a path-finding algorithm, such as Dijkstra or A*, is used to select the optimal polyline. In different way, the RRT algorithm starts from a single point in the domain and iteratively grows a tree by adding new random points and connecting the nearest one in its existing tree. However, both the mentioned approaches provide feasible polyline paths, and another stage in the process is needed to meet the smoothness requirements.

Optimisation-based planners formulate path planning as an optimisation to produce a feasible path with local or global optimality. They generally offer higher performance and motion quality than sampling-based methods, although formalising the problem to ensure stability and efficiency can be challenging. When the problem is convex, a unique optimum exists and local methods solve it efficiently. For non-convex problems with multiple local minima, two-stage schemes combining global exploration with local refinement are commonly used; however, these can be computationally demanding and may limit applicability in time-critical settings.

This letter presents a two-stage algorithm for collision-free path planning in manipulators. Stage one introduces a new C-space exploration algorithm that we call Goal-Biased Probabilistic Foam Cube-connect (GBPFC-connect). It performs goal-biased sampling and builds a safe corridor as a sequence of Axis-Aligned Bounding Boxes ($AABBs$) expanded asymmetrically along the C-space dimensions. Stage two generates a path within this corridor using an optimisation-based solution to the spline-based channel problem. The resulting joint-space trajectories are non-rational splines with user-defined degree and parametric continuity (C^3 in this work). The cost is a multi-objective function that balances minimising geometric acceleration to increase traversal speed and maximising obstacle clearance to enhance safety. Operating directly in C-space, the method applies to diverse kinematics; in this study, we use a 6-axis articulated arm. Superior performance over existing approaches is validated in simulation with the open-source Chrono environment [11], [12] and in real-world tests on a UR5e manipulator [13].

The full source code, including all examples in this letter, is available at: https://github.com/PathPlanningProject/pp-cs-sc.

II. Related Works

A. Bubbles of Free C-Space

A recent line of research in sampling-based C-space exploration focuses on multidimensional volumetric structures. An early systematic treatment appears in Sean Quinlan's PhD dissertation, which introduced elastic bands for real-time path modification [14], where a precompiled collision-free path is modelled as a flexible string that adapts to dynamic changes

Received 16 July 2025; accepted 2 November 2025. Date of publication 14 November 2025; date of current version 25 November 2025. This article was recommended for publication by Associate Editor Q. Lin and Editor L. Pallottino upon evaluation of the reviewers' comments. *(Corresponding author: Marco Riboli.)*

Riccardo Abbati and Alessandro Tasora are with the Department of Engineering for Industrial Systems and Technologies, University of Parma, 43124 Parma, Italy (e-mail: riccardo.abbati@unipr.it; alessandro.tasora@unipr.it).

Marco Riboli and Michel Rosselli are with the Department on Innovative Technologies, University of Applied Sciences and Arts of Southern Switzerland, CH-6962 Lugano, Switzerland (e-mail: marco.riboli@supsi.ch; michel.rosselli@supsi.ch).

Marco Silvestri is with the Department of Engineering for Industrial Systems and Technologies, University of Parma, 43124 Parma, Italy, and also with the Department on Innovative Technologies, University of Applied Sciences and Arts of Southern Switzerland, CH-6962 Lugano, Switzerland (e-mail: marco.silvestri@supsi.ch).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3632746, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3632746

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

while overlapping volumetric bubbles maintain safety. Formalised in [15], bubbles use minimum-distance data to define local free space and enable efficient collision-free adjustments, with the concept later revisited in [16] to construct exploratory structures in C-space. In the subsequent work [17], the Probabilistic Foam Method (PFM) planner was introduced to explore the C-space using overlapping bubbles. A goal-biased variant, Goal-biased Probabilistic Foam (GBPF), steers expansion towards the target [18], and subsequent extensions introduced radius-biased and heuristic-guided propagation [19]. These bubble-based planners build explicitly defined volumetric regions by expanding bubbles symmetrically around each configuration, and return polyline paths. In cluttered environments, this symmetry is conservative, producing undersized free-space estimates and a weak approximation of the free C-space. Burs of free C-space [20] mitigate this by expanding as star-shaped regions that define the Rapidly-exploring Bur Tree (RBT) planner, thereby increasing expansion flexibility, but they lack an explicit volumetric representation and still return polylines. This restricts trajectory optimisation that exploits the extent of the volumetric structures, as computationally efficient and robust encoding of collision-avoidance constraints requires an explicit geometric formalisation.

By contrast, the approach proposed in this work expands $AABBs$ asymmetrically along the C-space dimensions from a sampled configuration, and the resulting chain of overlapping boxes provides a geometrically formalised safe corridor specifically designed for the subsequent optimisation.

B. Spline-Based Channel Problem

By channel problem, we refer to the search for a function within a simply connected domain defined explicitly by geometric representations, aiming to optimise an objective function while satisfying specific constraints [21]. Various optimisation metrics have been investigated, including minimum path variation [22] and curvature [23], [24], [25], [26]. Based on the piecewise linear envelope algorithm introduced in [27], a highly efficient 1D formulation of the channel problem was presented in [28]. This solution relies on defining a spline within a linear envelope conforming to the channel's constraints. The robustness and practicality of this method have been demonstrated through industrial applications, particularly in pick-and-place application in sorting systems for laser cutting machine [25], [26]. Further work extended envelope methods to planar curve enclosures [29]. Notable applications include generating minimal-curvature planar paths for autonomous mining vehicles [22] and integrating incremental optimisation with D*-lite for dynamic UAV environments [23]. A significant contribution was introduced in [30], in which a linear-programming formulation was proposed to generate a non-rational spline constrained within a spatial channel. This approach proved practical validity by generating G^2-continuous paths for precision 5-axis machining [24].

Extending beyond conventional spline-based channel problem formulations, this work adapts the methodology to a C-space of arbitrary dimension, thereby addressing the challenges of high-dimensional trajectory optimisation.

III. Foundations of Spline Mathematics

A spline curve of degree d is a parametric function $\boldsymbol{r}(\tau) : \mathbb{R} \to \mathbb{R}^n$ composed of piecewise polynomials. When represented in B-form, it is expressed as a weighted combination of basis functions:

$$\boldsymbol{r}(\tau) = \sum_{i=1}^{m} B_{i,d,\boldsymbol{u}}(\tau)\, \boldsymbol{p}_i, \quad \tau \in [u_1, u_{m+d+1}], \tag{1}$$

where $\boldsymbol{p}_i = [p_{i,1} \ \dots \ p_{i,n}]$ are the control points, and $B_{i,d,\boldsymbol{u}}(\tau)$ are the B-spline basis functions associated with a non-decreasing knot vector $\boldsymbol{u} = [u_1 \ \dots \ u_{m+d+1}]$, which are defined according to the recursive Cox–de Boor formulation [31]. In the present study, we employ non-uniform and open (i.e., non-periodic) knot sequences that satisfy the endpoint conditions $u_1 = \cdots = u_{d+1}$ and $u_m = \cdots = u_{m+d+1}$, ensuring that the curve interpolates the first and last control points.

We define with $\boldsymbol{g} = [g_1 \ \dots \ g_m]$ the Greville abscissae, calculated by the averaging formula $g_i = \sum_{k=i+1}^{i+d} u_k/d$.

The k^{th} derivative of a B-spline basis function with respect to the parametric variable τ is provided by [32]:

$$B^{(k)}_{i,d,\boldsymbol{u}}(\tau) = d\left(\frac{B^{(k-1)}_{i,d-1,\boldsymbol{u}}(\tau)}{u_{i+d}-u_i} - \frac{B^{(k-1)}_{i+1,d-1,\boldsymbol{u}}(\tau)}{u_{i+d+1}-u_{i+1}}\right) \tag{2}$$

IV. C-Space Exploration With $AABBs$

To define the collision-free subset $\mathcal{C}_f$ of C-space, we use the collision-detection module of the open-source Chrono library [11]. Candidate interactions are first culled with simple bounding volumes, then the remaining pairs are tested with detailed geometry. Convex shapes are processed using standard collision-detection algorithms for convex polytopes [33], [34], while concave shapes are decomposed into convex parts and treated as convex. Further details are provided in [12], [35].

Once $\mathcal{C}_f$ has been computed, given the initial and goal configurations $\boldsymbol{q}_{\text{init}}, \boldsymbol{q}_{\text{goal}} \in \mathcal{C}_f$, we construct a safe corridor $\boldsymbol{\mathcal{C}} = \langle \mathcal{B}_1, \dots, \mathcal{B}_{\text{N}} \rangle$ comprising N sequentially overlapping axis-aligned bounding boxes ($\mathcal{B}_k \cap \mathcal{B}_{k+1} \neq \emptyset$) that connect $\boldsymbol{q}_{\text{init}}$ to $\boldsymbol{q}_{\text{goal}}$. A generic configuration $\boldsymbol{q}_k = [q_{k,1} \ \dots \ q_{k,n}]$ gathers the k^{th} values of the joint coordinates, while the vectors $\boldsymbol{q}_{\max}, \boldsymbol{q}_{\min} \in \mathbb{R}^n$ are, respectively, the lower and upper bounds of the axes. The safe corridor $\boldsymbol{\mathcal{C}}$ is built in two main steps:

1) A sampling-based exploration of the C-space, employing an RRT-connect logic [5] together with the GBPF algorithm [18], thereby yielding a provisional corridor.
2) Elimination of redundant $AABBs$ from the corridor, performed via a logic combining the shortcut and partial-shortcut techniques [36], and the final corridor definition.

A. Goal-Biased Probabilistic Foam Cube Connect

Algorithm 1 presents the pseudocode of the routine employed for C-space exploration and the generation of the safe corridor $\boldsymbol{\mathcal{C}}$, herein termed the Goal-Biased Probabilistic Foam Cube Connect (GBPFC-connect). In this algorithm, a generic axis-aligned bounding box $\mathcal{B}_k$ is defined by the structure:

```
AABB <struct>
+ RC:float[n]   ▷Reference Configuration
+ half_extent_min:float[n]
+ half_extent_max:float[n]
+ parent:int
```

Instances of this structure are initialized via the constructor AABB_INIT, which takes the above attributes in the same order. The RC attribute is the configuration from which the asymmetric expansion of the $AABB$ started, while the attributes half_extent_min and half_extent_max represent the half-lengths by which the $AABB$ extends in the negative and positive directions, respectively. For example, the total edge length along the first dimension is given by half_extent_min[1] + half_extent_max[1]. If the

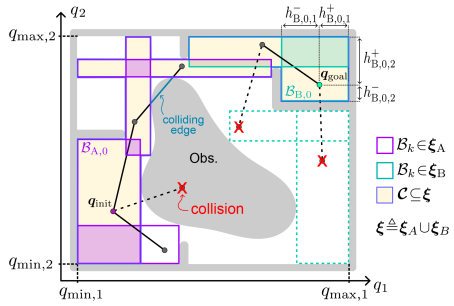


Fig. 1. Simplified 2D illustration of the GBPFC-connect algorithm.

Algorithm 1: GBPFC-CONNECT.

Require: $\boldsymbol{q}_{\text{init}}, \boldsymbol{q}_{\text{goal}}, \boldsymbol{q}_{\max}, \boldsymbol{q}_{\min}, \boldsymbol{h}_{\min}, \boldsymbol{h}_{\max}, l_{\text{try}}, \beta_0, \beta_{\max}, i_{\max}, \varepsilon$
Ensure: $\boldsymbol{\xi}$

$\boldsymbol{h}^-_{\text{A},0}, \boldsymbol{h}^+_{\text{A},0} \leftarrow$ AABB_EXPAND$(\boldsymbol{q}_{\text{init}}, \boldsymbol{q}_{\max}, \boldsymbol{q}_{\min}, \varepsilon, \boldsymbol{h}_{\max})$
$\boldsymbol{h}^-_{\text{B},0}, \boldsymbol{h}^+_{\text{B},0} \leftarrow$ AABB_EXPAND$(\boldsymbol{q}_{\text{goal}}, \boldsymbol{q}_{\max}, \boldsymbol{q}_{\min}, \varepsilon, h_{\max})$
$\boldsymbol{\xi}_{\text{A}} \leftarrow \{\, \mathcal{B}_{\text{A},0} \leftarrow$ AABB_INIT$(\boldsymbol{q}_{\text{init}}, \boldsymbol{h}^-_{\text{A},0}, \boldsymbol{h}^+_{\text{A},0}, \texttt{none})\,\}$
$\boldsymbol{\xi}_{\text{B}} \leftarrow \{\, \mathcal{B}_{\text{B},0} \leftarrow$ AABB_INIT$(\boldsymbol{q}_{\text{goal}}, \boldsymbol{h}^-_{\text{B},0}, \boldsymbol{h}^+_{\text{B},0}, \texttt{none})\,\}$
for $i \leftarrow 1$ **to** $i_{\max}$ **do**
  $r_v \leftarrow$ RAND() ▷uniform in$[0,1]$
  $\beta \leftarrow \beta_0 + (\beta_{\max} - \beta_0)\,\frac{i-1}{i_{\max}}$ (3)
  $\sigma \leftarrow \sigma_0\,(1.0 - \frac{i-1}{i_{\max}})$ ▷standard deviation
  if $r_v \le \beta$ **then**
    $\boldsymbol{q}_{\text{rand}} \leftarrow \boldsymbol{q}_{\text{goal}}$
  else
    for $\mathrm{d} \leftarrow 1$ **to** n **do**
      $r_v \leftarrow$ GAUSS$(\boldsymbol{q}_{\text{GOAL}}, \sigma)$ ▷normal distribution
      $q_{\text{rand,d}} \leftarrow \min(\max(r_v, q_{\min,\mathrm{d}}), q_{\max,\mathrm{d}})$ (4)
    end for
  end if
  $\mathcal{B}_{\text{A,new}} \leftarrow$ EXTEND_FOAM$(\boldsymbol{\xi}_{\text{A}}, \boldsymbol{q}_{\text{rand}}, \boldsymbol{h}_{\min}, \boldsymbol{h}_{\max}, l_{\text{try}}, \varepsilon)$
  if $\mathcal{B}_{\text{A,new}} = \texttt{fail}$ **then continue**
  else $\boldsymbol{\xi}_{\text{A}} \leftarrow \boldsymbol{\xi}_{\text{A}} \cup \{\mathcal{B}_{\text{A,new}}\}$
  $\boldsymbol{q}_{\text{A,new}} \leftarrow$ RC attribute of $\mathcal{B}_{\text{A,new}}$
  $\mathcal{B}_{\text{B,new}} \leftarrow$ EXTEND_FOAM$(\boldsymbol{\xi}_{\text{B}}, \boldsymbol{q}_{\text{A,new}}, \boldsymbol{h}_{\min}, \boldsymbol{h}_{\max}, l_{\text{try}}, \varepsilon)$
  if $\mathcal{B}_{\text{B,new}} = \texttt{fail}$ **then continue**
  else $\boldsymbol{\xi}_{\text{B}} \leftarrow \boldsymbol{\xi}_{\text{B}} \cup \{\mathcal{B}_{\text{B,new}}\}$
  if $\boldsymbol{\xi}_{\text{A}} \cap \boldsymbol{\xi}_{\text{B}} \neq \emptyset$ **then**
    $\boldsymbol{\xi} \leftarrow \boldsymbol{\xi}_{\text{A}} \cup \boldsymbol{\xi}_{\text{B}}$ ▷complete foam connecting $\boldsymbol{q}_{\text{init}}$ and $\boldsymbol{q}_{\text{goal}}$
    $\mathcal{C} =; \arg\min_{\mathcal{S} \subseteq \boldsymbol{\xi}} \{\, \text{card}(\mathcal{S}) \mid \exists\, \langle \mathcal{B}_1, \ldots, \mathcal{B}_{\text{N}} \rangle \subseteq \mathcal{S}, \boldsymbol{q}_{\text{init}} \in \mathcal{B}_0, ; \boldsymbol{q}_{\text{goal}} \in \mathcal{B}_{\text{N}}, ; \forall k :\ \mathcal{B}_k \cap \mathcal{B}_{k+1} \neq \emptyset \}$ (5)
    return $\mathcal{C}$ ▷safe corridor found
  end if
end for
No solution found

parent attribute is none, the instance corresponds to the first element of the sequence.

Employing an RRT-connect logic, the main idea is to define two distinct foams, denoted $\boldsymbol{\xi}_{\text{A}}$ and $\boldsymbol{\xi}_{\text{B}}$, originating respectively from the initial configuration $\boldsymbol{q}_{\text{init}}$ and the goal configuration $\boldsymbol{q}_{\text{goal}}$ (see Fig. 1). The algorithm terminates successfully once the two foams attain a non-empty intersection, at which point the safe corridor $\boldsymbol{\mathcal{C}}$ is selected as a subset of the union of $\boldsymbol{\xi}_{\text{A}}$ and $\boldsymbol{\xi}_{\text{B}}$. With this logic, Algorithm 1 begins by initialising these foams, defining the first $AABBs$ of the two sequences, namely $\mathcal{B}_{\text{A},0}$ and $\mathcal{B}_{\text{B},0}$. Assuming that neither the initial nor the goal configurations incur collisions (whether self-collisions or collisions with obstacles), the half_extent_min and half_extent_max attributes of $\mathcal{B}_{\text{A},0}$ and $\mathcal{B}_{\text{B},0}$, are computed by the AABB_EXPAND function detailed in Algorithm 2. The $AABB$ is expanded via an asymmetric binary search in both the positive and negative directions along each dimension using the EXTEND function, which takes as input the specific dimension $\mathrm{d} \in [1, \ldots, n]$ and the expansion sign s. The dichotomic search finds the maximal admissible expansion, thereby guaranteeing that the resulting $AABB$ describes a collision-free region of the C-space.

Referring back to the workflow of Algorithm 1, once the foams $\boldsymbol{\xi}_{\text{A}}$ and $\boldsymbol{\xi}_{\text{B}}$ have been initialised, the iterative expansion process begins. At the start of each iteration, a bias value $\beta \in [0,1]$ is defined according to (3), which employs the parameters β_0 and $\beta_{\max}$ (we use $\beta_0 \approx 0.1$ and $\beta_{\max} = 1$). Equation (3) updates the bias probability as the iterations progress: when $i = 0$, $\beta = \beta_0$, and when $i = i_{\max}$, $\beta = \beta_{\max}$. Thus, β increases with each iteration, steering the search progressively more towards the goal configuration. Next, β is compared with a random variable $r_v \sim \mathcal{U}(0,1)$. If $r_v < \beta$, the sample $\boldsymbol{q}_{\text{rand}}$ is set equal to the goal configuration $\boldsymbol{q}_{\text{goal}}$. Otherwise, a configuration is generated from a Gaussian distribution centred on the $\boldsymbol{q}_{\text{goal}}$. For each dimension d of the C-space the algorithm samples a zero-mean offset, and saturates the value within the admissible joint limits. The standard deviation σ starts from an initial value σ_0 and is gradually reduced as the iterations progress, so that the samples become increasingly concentrated around the goal. The configuration $\boldsymbol{q}_{\text{rand}}$ is used to determine the direction in which the foam $\boldsymbol{\xi}_{\text{A}}$ are to be extended. This operation is performed by computing $\mathcal{B}_{\text{A,new}}$ using the EXTEND_FOAM function detailed in Algorithm 3. This routine first selects the parent $\mathcal{B}_{\text{p}}$ from the input foam $\boldsymbol{\xi}$ by minimising the Euclidean distance between its RC attribute $\boldsymbol{q}_p$ and $\boldsymbol{q}_{\text{rand}}$. A unit vector $\hat{\boldsymbol{n}}$ is then computed to define the direction in which extension is attempted. An $AABB$ is initially generated with RC attribute given by (7), placing it at an offset determined by l_{p}, previously calculated by (6), and a trial edge length l_{try} (chosen here between 0.01 and 0.1 m according to the environment). Then the vectors $\boldsymbol{h}^-_{\text{new}}$ and $\boldsymbol{h}^+_{\text{new}}$ are calculated via the AABB_EXPAND routine using $\boldsymbol{q}_{\text{tmp}}$ as the attempt configuration. After that, it is checked that the obtained vectors are not less than the predefined threshold $\boldsymbol{h}_{\min}$. If the test is passed, a configuration $\boldsymbol{q}_{\text{new}}$ is computed by reapplying the (7) with l_{new}, calculated by (8), instead of l_{try}. Subsequently, the candidate configuration $\boldsymbol{q}_{\text{new}}$ is tested against existing foam $\boldsymbol{\xi}$. If it is within $\boldsymbol{\xi}$, it is discarded. Otherwise, $\mathcal{B}_{\text{new}}$ is instantiated by the AABB_INIT function. Following initialization, a further collision check is necessary to validate $\mathcal{B}_{\text{new}}$; in the absence of any detected collision the function returns $\mathcal{B}_{\text{new}}$, otherwise it returns fail. The EXTEND_FOAM function is used in Algorithm 1 first to compute $\mathcal{B}_{A,\text{new}}$. It is then applied again to extend the foam $\boldsymbol{\xi}_B$, this time using the RC attribute of $\mathcal{B}_{A,\text{new}}$ rather than $\boldsymbol{q}_{\text{rand}}$ to guide the expansion. In both cases, if EXTEND_FOAM returns fail the algorithm proceeds to the next

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 2: AABB_EXPAND.

Require: $\boldsymbol{q}$, $\boldsymbol{h}_{\max}$, $\boldsymbol{q}_{\min}$, $\boldsymbol{q}_{\max}$, ε
Ensure: $\boldsymbol{h}^-$, $\boldsymbol{h}^+ \in \mathbb{R}^n$
function EXTENTd, s
  $l \leftarrow 0$; $u \leftarrow h_{\max,\mathrm{d}}$; $q \leftarrow q_{\mathrm{d}} + s\,u$
  if not collision found for q **then return** u
  while $u - l > \varepsilon$ **do** ▷binary search loop
    $m \leftarrow (l+u)/2$; $q \leftarrow q_{\mathrm{d}} + s\,m$
    if collision found for q **then** $u \leftarrow m$
    else $l \leftarrow m$
  end while
  return l
end function
for $\mathrm{d} \leftarrow 1$ **to** n **do**
  $h^+_{\mathrm{d}} \leftarrow$ EXTENTd, $+1$; $h^-_{\mathrm{d}} \leftarrow$ EXTENTd, -1
end for
$\boldsymbol{h}^- \leftarrow \min(\boldsymbol{h}^-, \boldsymbol{q} - \boldsymbol{q}_{\min})$; $\boldsymbol{h}^+ \leftarrow \min(\boldsymbol{h}^+, \boldsymbol{q}_{\max} - \boldsymbol{q})$
$\boldsymbol{q}_{\text{lower}} \leftarrow \boldsymbol{q} - \boldsymbol{h}^-$; $\boldsymbol{q}_{\text{upper}} \leftarrow \boldsymbol{q} + \boldsymbol{h}^+$
if $\exists\, \boldsymbol{q} \in [\boldsymbol{q}_{\text{lower}}, \boldsymbol{q}_{\text{upper}}]$: collision found for q **then**
  return `fail` ▷ reject AABB
end if
return $\boldsymbol{h}^-$, $\boldsymbol{h}^+$

Algorithm 3: EXTEND_FOAM.

Require: $\boldsymbol{\xi}$, $\boldsymbol{q}_{\text{target}}$, $\boldsymbol{h}_{\min}$, $\boldsymbol{h}_{\max}$, l_{try}, ε
Ensure: $\mathcal{B}_{\text{new}}$
Select $\mathcal{B}_{\mathrm{p}} \in \boldsymbol{\xi}$ with RC attribute nearest to $\boldsymbol{q}_{\text{rand}}$.
$(\boldsymbol{q}_{\mathrm{p}}, \boldsymbol{h}^-_{\mathrm{p}}, \boldsymbol{h}^+_{\mathrm{p}}\, i_{\mathrm{p}}) \leftarrow$ attributes of $\mathcal{B}_{\mathrm{p}}$
$\hat{\boldsymbol{n}} \leftarrow (\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\mathrm{p}})/\|\boldsymbol{q}_{\text{target}} - \boldsymbol{q}_{\mathrm{p}}\|$
$$l_p \leftarrow \max(\boldsymbol{h}^-_p) + \max(\boldsymbol{h}^+_p) \tag{6}$$
$$\boldsymbol{q}_{\text{tmp}} \leftarrow (l_{\mathrm{p}} + l_{\text{try}})\,\hat{\boldsymbol{n}} \tag{7}$$
$\boldsymbol{h}^-_{\text{new}}, \boldsymbol{h}^+_{\text{new}} \leftarrow$ AABB_EXPAND$(\boldsymbol{q}_{\text{tmp}}, \boldsymbol{h}_{\max}, \boldsymbol{q}_{\max}, \boldsymbol{q}_{\min}, \varepsilon)$
$$l_{\text{new}} \leftarrow \max(\boldsymbol{h}^-_{\text{new}}) + \max(\boldsymbol{h}^+_{\text{new}}) \tag{8}$$
if $\boldsymbol{h}^+_{\text{new}} < \boldsymbol{h}_{\min}$ **or** $\boldsymbol{h}^-_{\text{new}} < \boldsymbol{h}_{\min}$ **then return** `fail`
$\boldsymbol{q}_{\text{new}} \leftarrow (l_{\mathrm{p}} + l_{\text{new}})\,\hat{\boldsymbol{n}}$
if $\boldsymbol{q}_{\text{new}} \in \boldsymbol{\xi}$ **return** `fail`
$\mathcal{B}_{\text{new}} \leftarrow$ AABB_INIT$(\boldsymbol{q}_{\text{new}}, \boldsymbol{h}^+_{\text{new}}, \boldsymbol{h}^-_{\text{new}}, i_{\mathrm{p}})$
if no collision found for $\mathcal{B}_{\text{new}}$ then **return** $\mathcal{B}_{\text{new}}$
else return `fail`

iteration; otherwise the newly generated $AABB$ is appended to the respective foam. The iteration concludes when $\boldsymbol{\xi}_A$ and $\boldsymbol{\xi}_B$ intersect in a non-empty set, indicating that a continuous corridor between $\boldsymbol{q}_{\text{init}}$ and $\boldsymbol{q}_{\text{goal}}$ has been found. The safe corridor $\boldsymbol{\mathcal{C}}$ is then extracted from the combined foam $\boldsymbol{\xi} = \boldsymbol{\xi}_A \cup \boldsymbol{\xi}_B$ by selecting the minimal subsequence of $AABBs$ according to (5).

B. Removal of Redundant $AABBs$

To reduce the number of $AABBs$ we adapt two techniques from [36]. We briefly summarise them below.

Shortcut: Select two $AABBs$, $\mathcal{B}_i$ and $\mathcal{B}_j$ ($0 \le i+1 < j <$ N), within the provisional corridor. If $\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset$, remove the intermediate boxes $\mathcal{B}_{i+1}, \dots, \mathcal{B}_{j-1}$, thereby reducing the number of $AABBs$. Repeat up to a preset iteration limit.

Partial-shortcut: Choose indices i, j ($0 \le i+1 < j <$ N) and a dimension d. Let $\boldsymbol{q}_i, \boldsymbol{q}_j$ be the RC attributes of $\mathcal{B}_i, \mathcal{B}_j$.

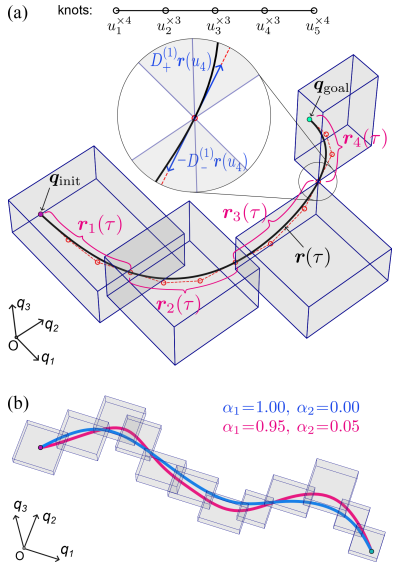


Fig. 2. (a) Path optimisation in a 3D C-space in the initial configuration. (b) Influence of the weights in the multi-objective cost function (9).

Linearly interpolate the d^{th} coordinate between $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$, holding the other dimensions fixed, to propose new reference configurations $\boldsymbol{q}_{i+1}, \dots, \boldsymbol{q}_{j-1}$. Rebuild $\mathcal{B}_{i+1}, \dots, \mathcal{B}_{j-1}$ starting from these configurations. Accept if all consecutive pairs intersect ($\mathcal{B}_k \cap \mathcal{B}_{k+1} \neq \emptyset$ for $k = i, \dots, j-1$), otherwise discard. Repeat up to a preset iteration limit.

V. Path Optimisation in C-Space Safe Corridor

A. Initial State and Correspondence

The optimisation problem is initialised by defining a knot sequence. As shown in Fig. 2(a), each internal knot u_j is assigned a multiplicity equal to the spline degree d, indicated as $u_j^{\times r}$, meaning that the j^{th} knot appears r times. With this configuration, the spline is initialised as a chain of Bézier segments joined at the interpolated control points. As better detailed at the end of Section V-B, a refinement phase may be required to ensure convergence of the method; therefore, in the rest of the letter spline segments are generally referred to as spline pieces.

Each spline pieces $\boldsymbol{r}_i(\tau)$ is associated with an $AABB_i$. This correspondence enables the formulation of non-interference constraints as simple bounds on the unknown control points. In other words, all control points defining $\boldsymbol{r}_i(\tau)$ are constrained to lie within the limits of $AABB_i$. However, due to the shared control point at the junction between two spline pieces, this point is simultaneously subject to the bounding constraints of both $AABB_i$ and $AABB_{i+1}$. Furthermore, at each curve junction, additional equality constraints must be introduced to impose the desired continuity.

B. Optimisation Problem Setting

Considering the following unknown vector

$$\boldsymbol{x} = \big[\underbrace{p_{1,1} \dots p_{1,n} p_{2,1} \dots p_{2,n} \dots p_{m,1} \dots p_{m,n}}_{nm \text{ control points values}}$$
$$\underbrace{z_{1,1} \dots z_{1,n} z_{2,1} \dots z_{2,n} \dots z_{m+\mathrm{N}-1,1} \dots z_{m+\mathrm{N}-1,n}}_{n(m+\mathrm{N}-1) \text{ slack variables}}\big],$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the optimisation problem can be summarised as a Linearly Constrained Quadratic Programming (LCQP) problem:

$$\underset{\boldsymbol{x}}{\text{minimise}} \quad f(\boldsymbol{x}) = \alpha_1 f_1(\boldsymbol{x}) + \alpha_2 f_2(\boldsymbol{x}), \tag{9}$$

$$\text{s.t.} \begin{cases} \textbf{Boundingconstraints}: \\ \quad \text{For each segment } i = 1, \ldots, \mathrm{N} \text{ with } AABB_i, \\ \quad \text{and each control point index } j = 1, \ldots, m: \\ \quad \text{if } j \notin \mathcal{J}, \text{ then } \forall\, k = 1, \ldots, n: \\ \quad\quad \texttt{lower}_{i,k} \le p_{j,k} \le \texttt{upper}_{i,k}, & (10) \\ \quad \text{if } j \in \mathcal{J}, \text{ then } \forall\, k = 1, \ldots, n: \\ \quad\quad p_{j,k} \ge \max\{\texttt{lower}_{i,k}, \texttt{lower}_{i+1,k}\}, & (11) \\ \quad\quad p_{j,k} \le \min\{\texttt{upper}_{i,k}, \texttt{upper}_{i+1,k}\}, & (12) \\ \textbf{Continuityconstraints}: \\ \quad \forall j \in \mathcal{J}, \forall\, k = 1, \ldots, \kappa: \quad D_-^{(k)} \boldsymbol{r}(u_j) = D_+^{(k)} \boldsymbol{r}(u_j). & (13) \end{cases}$$

where $\mathcal{J}$ denotes the set of knot indices corresponding to the junction control points of the spline segments $\boldsymbol{r}_i(\tau)$. In the initial state, it can be defined as $\mathcal{J} = \{\, k\,d + 1 \mid k = 1, 2, \ldots, \lfloor \frac{m-2}{d} \rfloor \,\}$. Referring to (13), we denote by $s{-}_j$ and $s{+}_j$ the left and righthand knot spans associated with the j^{th} control point, respectively. The corresponding left and righthand derivatives are given by

$$D_{\pm}^{(k)}\, \boldsymbol{r}(u_j) = \textstyle\sum_{i=s\pm_j-d}^{s\pm_j} B_{i,d,\mathbf{u}}^{(k)}(u_j)\, \boldsymbol{p}_i, \quad \text{with } s{-}_j = s{+}_j - d.$$

As reported in (9), the objective function consists of a linear combination of the two functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$, weighted by the coefficients α_1 and α_2. As a first contribution we have adopted the objective function originally used in [23]:

$$f_1(\boldsymbol{x}) = \textstyle\sum_{i=2}^{m-1} (\Delta_2 \boldsymbol{p}_i)^T (\Delta_2 \boldsymbol{p}_i) \tag{14}$$

with $\Delta_2 \boldsymbol{p}_i = \frac{\boldsymbol{p}_{i+1} - \boldsymbol{p}_i}{g_{i+1} - g_i} - \frac{\boldsymbol{p}_i - \boldsymbol{p}_{i-1}}{g_i - g_{i-1}}$.

Minimising (14) reduces the second derivative along the path, enhancing motion smoothness and enabling higher feed rate along the trajectory. However, this minimisation also tends to reduce the distance from the corridor boundaries during directional changes, consequently decreasing the safety margin from obstacles in Cartesian space. To counteract this effect, a second optimisation term is introduced:

$$f_2(\boldsymbol{x}) = \sum_{k=1}^{n} \sum_{i=2}^{m-1} \sum_{j \in \mathcal{J}_i} \left| -2\, p_{i,k} + \left(\texttt{upper}_{j,k} + \texttt{lower}_{j,k} \right) \right|, \tag{15}$$

where the set $\mathcal{J}_i$ is defined as

$$\mathcal{J}_i = \begin{cases} \{\, j, j+1 \,\}, & \text{if } i \in \mathcal{J}, \\ \{\, j \,\}, & \text{otherwise}. \end{cases}$$

Minimisation of (15) tends to keep the control points of $\boldsymbol{r}_i(\tau)$ as centred as possible within $AABB_i$, thus increasing the margin of safety. To linearise the non-linearity introduced by the absolute value, we introduce a slack variable $z_{q,k}$, with $q = 1, \ldots, m + \mathrm{N} - 1$, for each term of the summation in (15). Specifically, for each control point and each dimension k, we impose the following linear constraints:

$$2\, p_{i,k} + z_{q,k} \ge \left(\texttt{upper}_{j,k} + \texttt{lower}_{j,k} \right), \tag{16}$$

$$-2\, p_{i,k} + z_{q,k} \ge -\left(\texttt{upper}_{j,k} + \texttt{lower}_{j,k} \right), \tag{17}$$

with $p_{i,k}$ associated with $AABB_j$. These constraints ensure that $z_{q,k}$ attains the minimum non-negative value that bounds the expression, so that the absolute value is effectively replaced by the linear term $z_{q,k}$.

The influence of the weights α_1 and α_2 in the multi-objective cost function (9) is illustrated in Fig. 2(b). Increasing α_2 biases the optimisation towards penalising proximity to the $AABB$ boundaries, resulting in a spline that sacrifices smoothness to maintain a larger clearance from the corridor.

In light of the above considerations, we can mathematically formalise the LCQP problem in standard form:

$$\underset{\boldsymbol{x}}{\text{minimise}} \quad f(\boldsymbol{x}) = \alpha_1 \boldsymbol{x}^T \bar{\boldsymbol{P}} \boldsymbol{x} + \alpha_2 \boldsymbol{q}^T \boldsymbol{x}, \tag{18}$$

$$\text{subject to} \quad \boldsymbol{\ell} \le \boldsymbol{A}\boldsymbol{x} \le \mathbf{u}. \tag{19}$$

With reference to (18),

$$\bar{\boldsymbol{P}} = \left[\begin{array}{c|c} \begin{matrix} I_n \otimes \boldsymbol{P} \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \hline \begin{matrix} 0 & 0 \end{matrix} & \begin{matrix} 0 \end{matrix} \end{array}\right] \begin{matrix} nm \\ n(m+\mathrm{N}-1) \end{matrix}, \quad \boldsymbol{q} = [\underbrace{0 \ldots 0}_{nm}, \underbrace{1 \ldots 1}_{n(m+\mathrm{N}-1)}],$$

where I_n is the $n \times n$ identity matrix and $\otimes$ denotes the Kronecker product. The submatrix $\boldsymbol{P} \in \mathbb{R}^{m \times m}$ is pentadiagonal and symmetric. Moreover, it is straightforward to prove that $\boldsymbol{P} \succeq 0$. Hence, the resulting problem is convex and may be addressed efficiently with standard solvers. In this work, the OSQP solver [37] is used. For the definition of the coefficients of $\boldsymbol{P}$ and its proof, please refer to [21], [25].

The constraint matrix $\boldsymbol{A}$ and the boundary vectors $\boldsymbol{\ell}$ and $\mathbf{u}$ of (19) are summarised below:

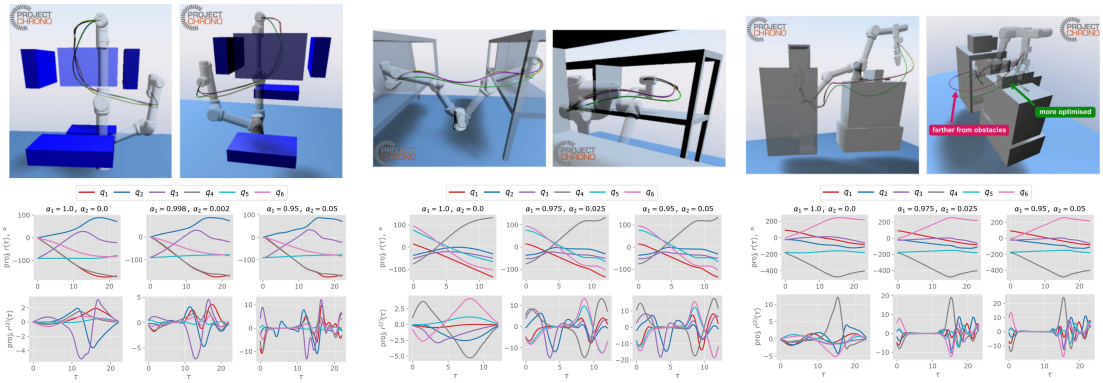
$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{\text{box}} & \mathbf{0} \\ \boldsymbol{A}_{\text{cont}} & \mathbf{0} \\ \boldsymbol{A}_{\text{bc}} & \mathbf{0} \\ \boldsymbol{A}_{\text{slack}}^{+} & \\ \boldsymbol{A}_{\text{slack}}^{-} & \end{bmatrix}, \quad \boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_{\text{box}} \\ \mathbf{0} \\ \boldsymbol{r}_{\text{bc}} \\ \boldsymbol{\ell}^{+} \\ \boldsymbol{\ell}^{-} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_{\text{box}} \\ \mathbf{0} \\ \boldsymbol{r}_{\text{bc}} \\ +\infty \\ +\infty \end{bmatrix}.$$

The submatrix $\boldsymbol{A}_{\text{box}} \in \mathbb{R}^{n(m+\mathrm{N}-1) \times nm}$ imposes the bounding constraints (10), (11), and (12). Each spline piece $\boldsymbol{r}_i(\tau)$ is defined by a subset of m_i control points. As the full spline is constructed by concatenating N spline pieces with shared control points at their junctions, the total number of control points m satisfies the relationship $\sum_{i=1}^{\mathrm{N}} m_i = m + (\mathrm{N} - 1)$. This determines the number of rows in the selector matrix $\boldsymbol{S}_{\text{box}} \in \mathbb{R}^{(m+\mathrm{N}-1) \times m}$, whose structure encodes which control points influence each spline piece. The full constraint matrix is then assembled as $\boldsymbol{A}_{\text{box}} = \boldsymbol{I}_n \otimes \boldsymbol{S}_{\text{box}}$. The corresponding lower and upper bound vectors, $\boldsymbol{\ell}_{\text{box}}$ and $\mathbf{u}_{\text{box}} \in \mathbb{R}^{n(m+\mathrm{N}-1)}$, are constructed by stacking the bounding limits $\texttt{lower}_{j,k}$ and $\texttt{upper}_{j,k}$ for each spline piece $j = 1, \ldots, N$ and dimension $k = 1, \ldots, n$, ordered first by piece index and then by coordinate.

The continuity constraints (13) are represented by the submatrix $\boldsymbol{A}_{\text{cont}} \in \mathbb{R}^{n\kappa(\mathrm{N}-1) \times nm}$. Specifically, for each knot index $j \in \mathcal{J}$, and for each derivative order $r = 1, \ldots, \kappa$, the condition enforces that the left- and right-hand derivatives of the spline match at the corresponding knot value. These constraints are implemented by evaluating the derivatives of the local basis functions at the shared parameter location u_j, in accordance with (2). The resulting row in the constraint matrix consists of the difference between the left and right derivative coefficients acting on the corresponding control points. The matrix $\boldsymbol{A}_{\text{cont}}$ is constructed block-wise across dimensions as $\boldsymbol{A}_{\text{cont}} = \boldsymbol{I}_n \otimes \boldsymbol{C}$, where each row of $\boldsymbol{C} \in \mathbb{R}^{\kappa(\mathrm{N}-1) \times m}$ contains non-zero entries only for the $d+1$ control points on the left and the $d+1$ on the right of each junction. The derivative coefficients are precomputed and inserted with opposite signs to enforce equality. The corresponding components in $\boldsymbol{\ell}_{\text{cont}}$ and $\mathbf{u}_{\text{cont}} \in \mathbb{R}^{n\kappa(\mathrm{N}-1)}$ are all set to zero.

The boundary constraints are represented by the submatrix $\boldsymbol{A}_{\text{bc}} \in \mathbb{R}^{2n \times nm}$, which enforces that the spline starts and ends at the specified initial and final positions. The matrix $\boldsymbol{A}_{\text{bc}}$ is constructed as $\boldsymbol{A}_{\text{bc}} = \boldsymbol{I}_n \otimes \boldsymbol{B}$, where $\boldsymbol{B} \in \mathbb{R}^{2 \times m}$ is a sparse

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Blocks. (b) Shelves. (c) Laser engraver.

Fig. 3. Three environments solved for three pairs of coefficient values α_1 and α_2. (a) – (c) Top: 3D views and the resulting end-effector path (green: $\alpha_1 = 1.0$, $\alpha_2 = 0.0$; magenta: $\alpha_1 = 0.95$, $\alpha_2 = 0.05$; yellow: otherwise). Bottom: joint trajectories and their second derivatives over the parametric variable τ.

matrix whose only nonzero entries are a 1in the first column of the first row and a 1in the last column of the second row; every other element is zero. The corresponding entries in the boundary vectors $\boldsymbol{\ell}$ and $\mathbf{u}$ are given by $\boldsymbol{r}_{\text{bc}} \in \mathbb{R}^{2n}$, which stacks the desired start and goal values for each dimension.

The set of constraints (16) and (17) is encoded by the matrices $\boldsymbol{A}^{+}_{\text{slack}}$ and $\boldsymbol{A}^{-}_{\text{slack}}$, which introduce additional conditions on the slack variables necessary to linearise the absolute value terms in function (15). The matrix $\boldsymbol{A}^{+}_{\text{slack}}$ contains rows with non-zero entries $[2 \ldots 1]$, while $\boldsymbol{A}^{-}_{\text{slack}}$ follows the same pattern with entries $[-2 \ldots 1]$. Each pair of values corresponds to a control point and its associated slack variable. These coefficients are arranged row-by-row for each control point and replicated across all dimensions. The corresponding entries in the lower bounds $\boldsymbol{\ell}^{+}$ and $\boldsymbol{\ell}^{-}$ are defined according to the $AABB$s data, while the upper bounds are set to $+\infty$.

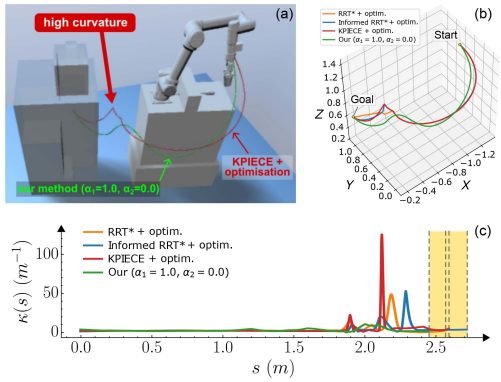


Fig. 4. Comparison of our trajectory and selected baselines. (a) Scene snapshot comparing our method with KPIECE + optimisation. (b) 3D view of the trajectories. (c) Estimated curvature $\kappa(s)$ as a function of arc length s.

C. Knots Refinement and Solution Feasibility

A single Bézier curve within each $AABB$ may not be sufficient to yield a feasible solution, especially in complex environments or under extra constraints (e.g. configuration interpolation). To ensure optimisation convergence, we increase the curve's cardinality by adding knots. In our setting of an nD channel formed by concatenated $AABB$s, refinement guarantees finite termination. As the knot sequence is refined, the control polygon moves closer to the spline. In particular, [28] shows that for a 1D uniform spline with equidistant knots, halving the knot spacing reduces the bound on the distance between the spline and its control polygon by a factor of 1/4, thus yielding geometric convergence. For an nD channel with bounds aligned with the axes, the same reasoning applies coordinate by coordinate, since the constraints decompose by dimension. Feasibility in every 1D projection then implies feasibility in the nD space. To preserve path quality during refinement, we use the knot distribution scheme of [25], which ensures quasi-uniform spacing of the Greville abscissae and thus avoids curvature spikes along the path.

VI. Simulation Studies & Real-World Applications

We evaluated the proposed path planning approach across three different environments: *Blocks*, *Shelves*, and *Laser engraver*, considering both simulated and real-world scenarios. Simulations were carried out using the open-source Chrono software [11], [12], employing a 6-axis articulated-arm model. Fig. 3(a)–(c) (top) shows 3D representations of each environment and corresponding end-effector paths. For real-world validation, the same planned trajectories were executed on a UR5e robotic manipulator [13] within our Mini-Factory educational laboratory [38], [39], [40]. In the *Blocks* and *Shelves* environments, validation focused solely on replicating the manipulator's motion, whereas the *Laser engraver* scenario included a complete reconstruction of the workspace. Specifically, this final case involved a practical demonstration in which the manipulator placed a mug within a confined capsule to enable personalised laser engraving. The supplementary material attached to this letter includes a video for each environment, showing both the simulation outputs and the real-world executions. Fig. 5 presents representative frames from two points of view in the *Laser engraver* environment, illustrating the real-world execution with the end-effector trajectory overlaid.

Table I summarises the obtained results. For all variants of the proposed algorithm, a spline of degree $d = 4$ with continuity C^3 is used. To evaluate the performance of the proposed method, several baseline planners from the OMPL C++ library (version 1.6.0) [41] were also evaluated across the three environments. To ensure equitable benchmarking, the raw polylines generated by these planners underwent sequential post-processing using

TABLE I
RESULTS OF SIMULATIONS AND COMPARATIVE ANALYSIS

Env.	Method	Successes[1](\10)	Path length (m)	Iterations	C-space exploration CPU time[2](s)	Optimisation CPU time[2](ms)
1 (Blocks)	Our (α_1=1.0, α_2=0.0)	10	**1.553±0.032**	201±111	**3.9±2.5**	**10±1**
	Our (α_1=0.998, α_2=0.002)	–	1.602±0.201	–	–	264±95
	Our (α_1=0.95, α_2=0.05)	–	1.640±0.144	–	–	331±54
	RRT [4] + optimisation[3]	8	2.220±0.064	432±167	171.7±59.7	2145±587
	PRM [2] + optimisation[3]	10	2.253±0.463	399±183	240.0±175.6	4576±1216
	RRT* [6] + optimisation[3]	5	2.007±0.335	620±92	522.0±124.3	2588±1098
	Informed RRT* [9] + optimisation[3]	7	1.732±0.293	271±54	438.2±196.9	2898±447
	KPIECE [42] + optimisation[3]	5	1.857±0.514	**153±110**	8.8±4.4	3579±1459
2 (Shelves)	Our (α_1=1.0, α_2=0.0)	10	**1.535±0.167**	58±24	**27.4±21.7**	**12±3**
	Our (α_1=0.975, α_2=0.025)	–	1.625±0.282	–	–	205±75
	Our (α_1=0.95, α_2=0.05)	–	1.630±0.324	–	–	226±71
	RRT [4] + optimisation[3]	10	1.677±0.032	**42±29**	121.4±81.2	3332±1901
	PRM [2] + optimisation[3]	7	1.815±0.161	209±139	407.9±226.9	2865±279
	RRT* [6] + optimisation[3]	4	1.757±0.124	204±84	513.8±197.9	2765±341
	Informed RRT* [9] + optimisation[3]	7	1.754±0.058	304±55	461.7±224.6	3318±811
	KPIECE [42] + optimisation[3]	9	1.727±0.094	83±63	108.2±98.9	2329±561
3 (Laser engraver)	Our (α_1=1.0, α_2=0.0)	8	**2.453±0.340**	**112±61**	**56.5±24.3**	**16±2**
	Our (α_1=0.975, α_2=0.025)	–	2.570±0.102	–	–	234±54
	Our (α_1=0.95, α_2=0.05)	–	2.613±0.057	–	–	221±45
	RRT [4] + optimisation[3]	3	2.725±0.064	355±321	277.2±178.5	2145±587
	PRM [2] + optimisation[3]	6	2.693±0.062	565±143	667.6±406.2	3367±1216
	RRT* [6] + optimisation[3]	8	2.660±0.081	230±77	363.8±126.9	3265±1168
	Informed RRT* [9] + optimisation[3]	6	2.782±0.066	363±124	582.5±128.0	1605±428
	KPIECE [42] + optimisation[3]	6	2.932±0.232	565±190	351.9±147.6	516±458

Best average value for each environment is shown in **bold**. A dash (–) denotes the value repeats from the row above. Reported values are averaged over successful runs; the corresponding counts are given in the Successes column (out of 10 trials). The "± value" denotes the full range observed across simulations. To facilitate replicability of the results, the parameter settings for each planner and environment are available in the project repository. [1]Trials exceeding 800 s of C-space exploration are counted as failures. [2]Computations performed with an Intel® Core™ i7-8550U. [3] This method is implemented with the OMPL C++ library (version 1.6.0) [41]. The optimisation is performed by first applying the *ropeShortcutPath()* and *partialShortcutPath()* functions, followed by smoothing using the *smoothBSpline()* function.

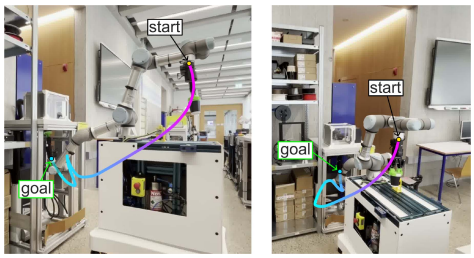


Fig. 5. Sample frames of the proposed trajectory in the *Laser Engraver* environment, showing the executed end-effector trajectory.

OMPL's shortcut and partial shortcut methods [36], followed by the optimiser provided by the *smoothBSpline()* function to enhance trajectory smoothness. Each planner was executed ten times per environment. To mitigate inherent stochasticity, path length and computational performance were averaged over successful runs only. A run was deemed a failure if no solution was found within 800 s of C-space exploration. The corresponding number of successes is also reported in Table I.

The results indicate that the proposed method typically outperforms the reference planners, yielding shorter, smoother and more geometrically continuous paths. In the *Blocks* environment, KPIECE [42] showed performance comparable to our method in terms of path length and C-space exploration time. Specifically, in one run it returned a shorter path, although the mean path length remained higher (KPIECE: 1.857± 0.514 m; ours: 1.553± 0.032 m), the success rate was lower, and the C-space exploration time was longer than ours, while still markedly faster than the other baselines and of a similar order to our method. In both of the other test environments we observe a substantial gap in favor of our method, even when compared with KPIECE, while for the remaining baselines our advantage is consistently confirmed. It is worth emphasising that, as shown in Fig. 4, our method typically provides higher motion quality, avoiding sharp corners along the path (Fig. 4(a), (b)) and suppressing curvature peaks, which in practice reduces points of arrest and feedrate fluctuations. Fig. 4(c) reports the estimated curvature $\kappa(s)$ as a function of the arc length s, where our profile shows lower peaks than the evaluated baselines and a shorter overall arc length. Moreover, our method exhibits superior computational efficiency during C-space exploration and, compared with the other tested algorithms, a substantially higher success rate. These performance advantages appear to arise from corridor-based exploration, which operates on $AABBs$ rather than polyline edges and thus preserves feasible states that a standard sampling-based method would discard when an edge collides. For example, the schematic representation in Fig. 1 shows a case in which a standard sampling-based planner operating on polylines would discard the configuration, owing to the edge linking it to the preceding configuration being in collision. This results in fewer discarded configurations and fewer dead-end branches during the search.

Fig. 3(a)–(c) (bottom) reports the joint trajectories and their second derivatives over the parameter τ, illustrating that even a small increase in α_2 rapidly degrades trajectory quality by producing pronounced acceleration spikes, indicating high sensitivity of the parameter. In particular, the use of only the (14) as an objective function (i.e., $\alpha_1 = 1.0$, $\alpha_2 = 0.0$) provided optimal results for both computational efficiency and path quality. However, assigning a non-zero value to α_2 introduced a margin of safety by increasing the distance to obstacles, thus providing a useful tool for trajectory design.

VII. DISCUSSION AND CONCLUSION

This letter introduces a novel path planning methodology integrating the GBPFC-connect algorithm for C-space exploration with spline-based convex optimisation, resulting in smooth, collision-free trajectories for manipulators.

The proposed approach was benchmarked against selected reference methods provided by the OMPL C++ library (version 1.6.0) [41], evaluating performance across three distinct scenarios. Comparative analysis demonstrated that our approach consistently yielded shorter paths and significantly reduced computational times compared to reference methods, while delivering a lower maximum curvature. Furthermore, the proposed method exhibited a higher success rate. The presented algorithm shows clear applicability potential beyond the examined kinematic. Furthermore, our problem formulation easily

supports additional linear constraints, including interpolation of user-defined configurations.

The primary limitation is the algorithm's current restriction to static environments. Nevertheless, the inherent computational efficiency afforded by convex optimisation presents promising opportunities for future enhancement, particularly through a buffered variant employing precomputed static roadmaps to facilitate dynamic obstacle handling. Subsequent studies will focus on efficient buffer management and real-time corridor adjustment strategies in order to extend the presented methodology to dynamic environments.

REFERENCES

[1] W. Thomas, B. Bernhard, M. Martin, H. N. Christian, and K. Andreas, "Optimization-based path planning framework for industrial manufacturing processes with complex continuous paths," *Robot. Comput.- Integr. Manuf.*, vol. 82, 2023, Art. no. 102516, doi: 10.1016/j.rcim.2022.102516.

[2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996, doi: 10.1109/70.508439.

[3] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. Robot. Automat.*, vol. 14, no. 1, pp. 166–171, Feb. 1998, doi: 10.1109/70.660866.

[4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Tech. Rep. 98-11, Oct. 1998.

[5] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia Proc.*, San Francisco, CA, USA, 2000, vol. 2, pp. 995–1001, doi: 10.1109/ROBOT.2000.844730.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011, doi: 10.1177/0278364911406761.

[7] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, 2014, pp. 2997–3004, doi: 10.1109/IROS.2014.6942976.

[8] S. Luo, S. Liu, B. Zhang, and C. Zhong, "Path planning algorithm based on Gb informed RRT* with heuristic bias," in *Proc. 36th Chin. Control Conf.*, Dalian, China, 2017, pp. 6891–6896, doi: 10.23919/ChiCC.2017.8028443.

[9] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 966-984, Aug. 2018, doi: 10.1109/TRO.2018.2830331.

[10] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014, doi: 10.1109/ACCESS.2014.2302442.

[11] A. Tasora et al., "Chrono: An open source multi-physics dynamics engine," in *Proc. High Perform. Comput. Sci. Eng.*, 2016, pp. 19–49, doi: 10.1007/978-3-319-40361-8_2.

[12] "Chrono: An open source framework for the physics-based simulation of dynamic systems," Accessed: Apr. 6, 2025. [Online]. Available: http://projectchrono.org

[13] "Universal robots," Accessed: Apr. 19, 2025. [Online]. Available: https://www.universal-robots.com

[14] S. Quinlan, "Real-time modification of collision-free paths," Ph.D. Dissertation. Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 1995.

[15] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, USA, 1993, vol. 2, pp. 802–807, doi: 10.1109/ROBOT.1993.291936.

[16] A. Ademovic and B. Lacevic, "Path planning for robotic manipulators via bubbles of free configuration space: Evolutionary approach," in *Proc. 22nd Mediterranean Conf. Control Automat.*, Palermo, Italy, 2014, pp. 1323–1328, doi: 10.1109/MED.2014.6961559.

[17] Y. S. Silveira and P. J. Alsina, "A new robot path planning method based on probabilistic foam," in *Proc. XIII Latin Amer. Robot. Symp. IV Braz. Robot. Symp.*, Recife, Brazil, 2016, pp. 217–222, doi: 10.1109/LARS-SBR.2016.43.

[18] L. B. P. Nascimento et al., "Goal-biased probabilistic foam method for robot path planning," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Competitions*, Torres Vedras, Portugal, 2018, pp. 199–204, doi: 10.1109/ICARSC.2018.8374183.

[19] L. B. P. Nascimento et al., "Safe path planning algorithms for mobile robots based on probabilistic foam," *Sensors*, vol. 21, no. 12, 2021, Art. no. 4156, doi: 10.3390/s21124156.

[20] B. Lacevic, D. Osmankovic, and A. Ademovic, "Burs of free C-space: A novel structure for path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Stockholm, Sweden, 2016, pp. 70–76, doi: 10.1109/ICRA.2016.7487117.

[21] M. Riboli, E. Manconi, R. Garziera, and A. Aimi, "A general approach for planning a smooth planar path within a channel using NURBS," *IEEE Robot. Automat. Lett.*, vol. 9, no. 7, pp. 6744–6751, Jul. 2024, doi: 10.1109/LRA.2024.3412636.

[22] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding Bspline paths for autonomous mining vehicles," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 167–172, Jan. 2010, doi: 10.1109/TASE.2009.2015886.

[23] D. Jung and P. Tsiotras, "On-line path generation for unmanned aerial vehicles using B-spline path templates," *J. Guidance, Control, Dyn.*, vol. 36, no. 6, pp. 1642–1653, 2013, doi: 10.2514/1.60780.

[24] E. Selinger and L. Linsen, "Efficient curvature-optimized G2-continuous path generation with guaranteed error bound for 5-axis machining," in *Proc. 13th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2018, pp. 59–70, doi: 10.5220/0006537400590070.

[25] M. Riboli, M. Jaccard, M. Silvestri, A. Aimi, and C. Malara, "Collision-free and smooth motion planning of dual-arm cartesian robot based on b-spline representation," *Robot. Auton. Syst.*, vol. 170, 2023, Art. no. 104534, doi: 10.1016/j.robot.2023.104534.

[26] M. Riboli, M. Jaccard, M. Silvestri, E. Manconi, A. Aimi, and R. Garziera, "Collision-free motion generation for dual-gantry robotic systems," *Robot. Auton. Syst.*, vol. 192, 2025, Art. no. 105031, doi: 10.1016/j.robot.2025.105031.

[27] D. Lutterkort and J. Peters, "Tight linear envelopes for splines," *Numerische Mathematik*, vol. 89, pp. 735–748, 2001, doi: 10.1007/s002110100181.

[28] D. Lutterkort and J. Peters, "Smooth paths in a polygonal channel," in *Proc. 15th Annu. Symp. Comput. Geometry*, 1999, pp. 316–321.

[29] D. Lutterkort and J. Peters, "Linear envelopes for uniform B-spline curves," in *Proc. Int. Conf. Curves Surfaces*, 1999, pp. 239–246.

[30] A. Myles and J. Peters, "Threading splines through 3D channels," *Comput.-Aided Des.*, vol. 37, no. 2, pp. 139–148, 2005, doi: 10.1016/j.cad.2004.04.004.

[31] C. de Boor, *A Practical Guide to Splines* (Applied Mathematical Sciences Series). New York, NY, USA: Springer, 2001.

[32] L. A. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY, USA: Springer, 1996.

[33] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988, doi: 10.1109/56.2083.

[34] S. Gary, "XenoCollide: Complex collision made simple," *Game Program. Gems*, vol. 7, pp. 165–178, 2008.

[35] "Bullet real-time physics simulation," Accessed: May 31, 2025. [Online]. Available: https://pybullet.org/wordpress/

[36] R. Geraerts and M. H. Overmars, "Creating high-quality paths for motion planning," *Int. J. Robot. Res.*, vol. 26, no. 8, pp. 845–863, 2007, doi: 10.1177/0278364907079280.

[37] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Computation*, vol. 12, no. 4, pp. 637–672, 2020, doi: 10.1007/s12532-020-00179-2.

[38] F. Daniele, M. Confalonieri, P. Pedrazzoli, A. Graf, A. Ferrario, and M. Foletti, "Implementation of a learning factory for research, education and training: The SUPSI mini-factory," in *Proc. Conf. Learn. Factories*, May 2021, doi: 10.2139/ssrn.3858404.

[39] F. Daniele et al., "THE innovative educational approach in the SUPSI mini-factory," in *Proc. Abstr. Proc. INTCESS 2022- 9th Int. Conf. Educ. Educ. Social Sci.*, 2022, pp. 457–466, doi: 10.51508/intcess.202265.

[40] S. Dell'Oca, E. Montini, V. Cutrona, D. Matteri, G. Landolfi, and A. Bettoni, "A scalable data-driven methodology for human intention prediction in diverse collaborative scenarios," in *Proc. IEEE 21st Int. Conf. Automat. Sci. Eng.*, Los Angeles, CA, USA, 2025, pp. 743–750, doi: 10.1109/CASE58245.2025.11163969.

[41] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Automat. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012, doi: 10.1109/MRA.2012.2205651.

[42] I. A. Sucan and L. E. Kavraki, "A sampling-based tree planner for systems with complex dynamics," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 116–131, Feb. 2012, doi: 10.1109/TRO.2011.2160466.