Trajectory Planning for the Shapeshifting of Autonomous Surface Vessels: Supplemetary Material

Banti Gheneti, Shinkyu Park, Ryan Kelly, Drew Meyers, Pietro Leoni, Carlo Ratti and Daniela Rus

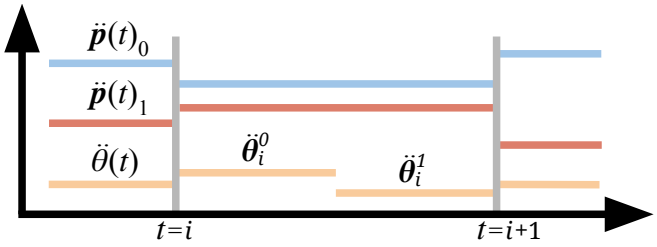


Fig. S1: Acceleration profile for $p(t)$ and $\theta(t)$

A. MIQP Cost Function

Equation (4) is derived from the following cost in (1).

$$\int_0^T \left[\ddot{p}(t)^T \ddot{p}(t) + w\ddot{\theta}^2(t)\right] dt \tag{S1}$$

We substitute $\ddot{p}(t)$ with the following, as shown by Usenko *et al.* [26], yielding the first term in the cost for (4).

$$\ddot{p}(t) = \begin{pmatrix}0\\0\\2\end{pmatrix}^T P \begin{pmatrix}\boldsymbol{\alpha}_{i-1}{}^T\\ \boldsymbol{\alpha}_i{}^T\\ \boldsymbol{\alpha}_{i+1}{}^T\end{pmatrix} \mid i \leq t < i+1 \tag{S2}$$

The second term in the cost for (4) directly follows from the definition of $\ddot{\theta}(t)$ in (3b).

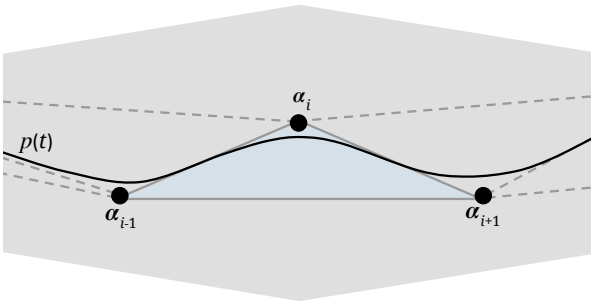
B. MIQP Collision Avoidance Constraints

In (6b) we use the big M method to constraint each line between control points to lay in one convex hull, where A_h and B_h define the h-th convex hull. The strong convex hull property guarantees that any point on the uniform quadratic B-spline $p(t)$ lies in the triangle of the three control points defining it. When two successive line segments $\overline{\boldsymbol{\alpha}_{i-1}\boldsymbol{\alpha}_i}$ and $\overline{\boldsymbol{\alpha}_i\boldsymbol{\alpha}_{i+1}}$ are contained in one convex hull this is sufficient to guarantee that the B-spline will stay in that convex hull for $t \in [i, i+1]$, as shown in in Fig. S2a.

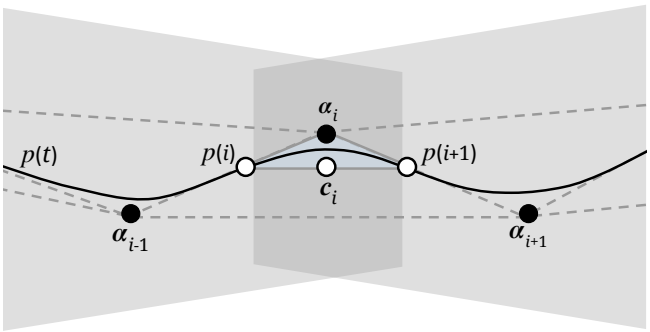
C. Polygon Transition Collision Avoidance Constraints

Additional constraints are added in (6c) for when two successive line segments are not in one convex hull. $p(t)$ is tangent to the line segments $\overline{\boldsymbol{\alpha}_{i-1}\boldsymbol{\alpha}_i}$ and $\overline{\boldsymbol{\alpha}_i\boldsymbol{\alpha}_{i+1}}$ respectively at $p(i)$ and $p(i+1)$. Its constant acceleration during this interval, according to (S2), yields a quadratic curve that stays above the line segment $p(i)p(i+1)$. By placing the additional constraint that the triangle $p(i)\boldsymbol{\alpha}_i p(i+1)$ must lay in the union of the convex hulls $\overline{\boldsymbol{\alpha}_{i-1}\boldsymbol{\alpha}_i}$ and $\overline{\boldsymbol{\alpha}_i\boldsymbol{\alpha}_{i+1}}$ lay in, we can efficiently ensure that the trajectory stays safe.

We define a point along the line segment $\overline{p(i)p(i+1)}$:



(a) Blue triangle $\alpha_{i-1}\alpha_i\alpha_{i+1}$ is in the grey convex polygon. This results in the trajectory $p(t)$, contained in this triangle for $t = i$ to $t = i+1$, to also stay in the convex polygon during that interval.



(b) $p(i)$ is in the first grey convex polygon, $p(i+1)$ is in the second polygon and α_i is in both. If $\boldsymbol{c}_i$ constrains the blue triangle to lay in the polygons' union spline $p(t)$ safely transitions between them.

Fig. S2: B-spline Collision Avoidance Constraints: (6b), depicted in (a), ensures trajectory segments in polygons remain safe. (6c), depicted in (b), ensures trajectory segments crossing between polygons remain safe.

$$\boldsymbol{c}_i = \boldsymbol{\beta}_i \frac{\boldsymbol{\alpha}_{i-1} + \boldsymbol{\alpha}_i}{2} + (1 - \boldsymbol{\beta}_i)\frac{\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_{i+1}}{2} \tag{S3}$$

$$\boldsymbol{\beta}_i \in [0, 1] \tag{S4}$$

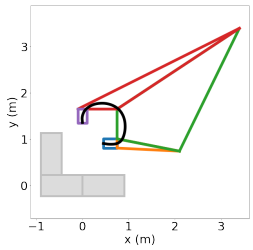
If $\boldsymbol{c}_i$ lays in the intersection of the two convex hulls, the triangles $p(i)\boldsymbol{c}_i\boldsymbol{\alpha}_i$ and $p(i+1)\boldsymbol{c}_i\boldsymbol{\alpha}_i$ respectively lay in the first and second convex hulls, as a whole ensuring that the trajectory is safe. This scenario is shown in Fig. S2b and is enforceable with the following constraint:

$$A_h \boldsymbol{c}_i \geq B_h \mid \boldsymbol{H}_h^{i-1} + \boldsymbol{H}_h^i > 0 \tag{S5}$$

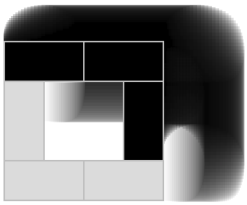
By setting $\boldsymbol{\beta}$ to 0.5, (S5) can be represented with the mixed integer linear constraint in (6c).

D. Simulation Trajectories

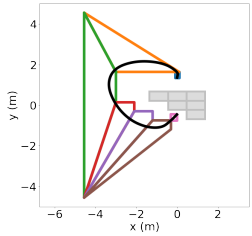
The resulting trajectories and simulation CVP movement for scenarios 4-5 are shown in Fig. S3. As is visible, the optimization finds tight continuous trajectories that stay in the configuration space. It is evident that our approach finds smooth and safe trajectories from an unlatch to a latch, which follows from the cost function and mixed integer constraints.



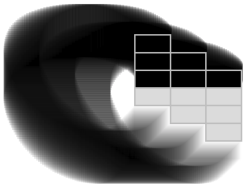
(a) Scenario 4 Trajectory



(b) Scenario 4 Boat Trail



(c) Scenario 5 Trajectory



(d) Scenario 5 Boat Trail

Fig. S3: Simulation Results for Scenarios 4 and 5: stationary CVP shown in grey. (a) and (c) show corridor regions in outlined polygons and the optimized trajectory in black. (b) and (d) show the CVP motion in black

E. Discussion on Time Complexity

The Minkowski sum is computed in $\mathcal{O}(nm)$ where n and m are the number of points defining the CVP shape. The Hertel-Melhorn algorithm partitions the space in $\mathcal{O}(m^3 + n^3)$. The resulting graph is produced in $\mathcal{O}(|\mathbb{V}|^2)$. As all the precomputation steps run in polynomial time, the time complexity of our approach is dominated by MIP which runs in time exponential to N.