Geometric Primitives-Based AUV Path Planning In Cluttered Waterspaces ⋆

Matko Barisic * **Nikola Miskovic** * **Zoran Vukic** *
Antonio Vasilijevic *

* *University of Zagreb, Faculty of Electrical Engineering and Computing, Laboratory for Underwater Systems and Technologies, Unska 3, HR-10000 Zagreb, Croatia (e-mail: matko.barisic@fer.hr, nikola.miskovic@fer.hr, zoran.vukic@fer.hr).*

Abstract: This paper describes a procedure that utilizes a series of off-line algorithms for pre-planning a two-dimensional path for an AUV in a cluttered waterspace. The trajectory is planned as a clothoidal spline or interpolation based on the work of Shin and Singh (1990) between a sequence of *objective points*. The former are calculated by a method proposed by the authors, based on the traversal of vertices of apriori known obstacles modeled on a concise set of geometrical primitives. The objective points are first sequenced and then extraneous ones are pruned. Finally, the technique of Shin and Singh (1990) and numerical approximation of clothoid parameters is applied to calculating the final path through the obstacle-strewn waterspace.

Keywords: Navigation, guidance and control of unmanned marine vessels (surface and underwater); Autonomous and remotely operated (surface and underwater) marine vessels; Surveillance and supervision systems in marine applications

1. INTRODUCTION

The technology of autonomous underwater vehicles has been experiencing a steady progress in the last 30 years, and is at the verge of being accepted as a mature field of engineering (Bildberg (2009)). Most marketed AUVs come equipped with software for *human* operator / mission director assisted path or trajectory planning. The most often used methodology is providing the human operator with tools allowing the drafting of a mission plan consisting of a piecewise-affine path connecting a sequence of "clickable" set-points defined by latitude and longitude, augmented with reference depths (or altitudes off the bottom). Reference cruise speeds are specified for each affine segment. To increase situation awareness that contributes to the human operator's reasoning while drafting a mission plan, different sensor records (e.g. side-scan sonar, ADCP readings, interferometric sonar, multibeam sonar) can be layered on top of the basic geographic mission view in order to augment the *a priori awareness of a static waterspace* to as near as perfect as possible (Brutzman (1994), Lee (2004), deArruda (2009)).

The authors propose an algorithm of human-unassisted creation of a missions to be cruised. For this paper, the proposed solution maintains the limitations originally imposed on the *human* planner being emulated – namely, perfect or near-perfect situational awareness of the waterspace. For the benefit of the computer, this knowledge is assumed to be encoded in a database of obstacles, classified as instances of one of two fundamental types: a rectangle or a circle.

Section 2 defines the relevant terms used in the formulation of the obstacles in the 2D waterspace. Section 3 develops the algorithm for the interpolation of the objective points of the mission. Section 4 describes the process of the calculation of pairs or triplets of clothoidal curves used to span the sparse sequence of objective points rendered on the basis of the geometry of clutter of the waterspace to be navigated. The sequentialized algorithms in the former and the latter section represent the proposed algorithm. Section 5 concludes the paper.

2. GEOMETRICAL PRIMITIVES MODELING THE OBSTACLES

The waterspace, represented by $\mathbb{R}^2$ is supposed to be populated by obstacles that represent the instances of two classes: generalized rectangles and generalized circles. The perfect a priori knowledge of the pose (i.e. position of the geometric center and angular offset) of the rectangular obstacles is given by (1), according to figure 1.a). The circular obstacles are described by (2), according to figure 1.b).

$$A_{rect} = \left(\mathbf{z}_{rect},\ [a\ b]^{\mathrm{T}}\right) \tag{1}$$

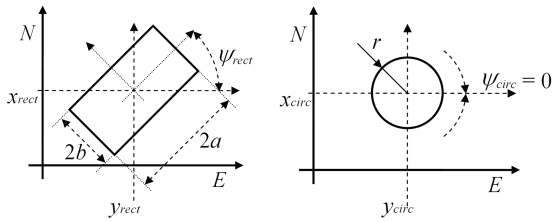
$$A_{circ} = \left(\mathbf{z}_{circ},\ r\right) \tag{2}$$

$$\tag{3}$$

Where:
- $\mathbf{z}_{rect} = [x_{rect} y_{rect} \psi_{rect}]^{\mathrm{T}}$ is the pose of the rectangle, separable into the position of the rectangle center $(x_{rect},\ y_{rect}) \in \mathbb{R}^2$ and the orientation of the rectangle-fixed x-axis w.r.t. the northward-pointing x-axis of the NE

⋆ This research was made possible by the EU 2008-1-REGPOT grant for the "Developing the Croatian Underwater Robotics Research Potential" project, grant agreement no. 229553.

Earth-fixed frame, cf. figure 1.a),
- $[ab]^{\mathrm{T}}$ is the vector of rectangle dimensions in the direction of the x- (length) and y-axes (width) of the rectangle-fixed frame, cf. figure 1.a),
- $\mathbf{z}_{circ} = [x_{circ} y_{circ} 0]^{\mathrm{T}}$ is the pose of the circle, wherein only the position of the circle center $(x_{circ}, y_{circ}) \in \mathbb{R}^2$ w.r.t. the NE Earth-fixed frame is non-trivial, and the orientation is identically equal to zero, due to the pointwise symmetry of circles, cf. figure 1.b),
- r is the radius of the circle, cf. figure 1.b).



(a) Rectangular obstacle (b) Circular obstacle

Fig. 1. The representations of obstacles with perfect knowledge of parameters.

3. INTERPOLATION OF OBJECTIVE POINTS

When considering the problem of trajectory planning, the solution procedure can be approached by a two-stage strategy:

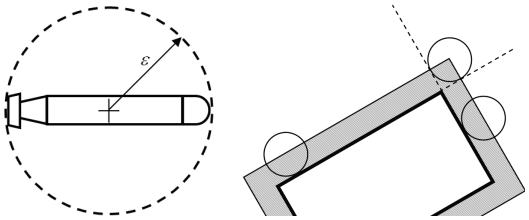
(1) Identifying safe, minimally distant, efficiently placed and objective-fulfilling objective points between which line of sight and line of navigation is guaranteed, interspersed throughout a cluttered or unsafe waterspace.
(2) Generating in a piece-wise manner the reference signals for the actuators of the AUV that guarantee stable, accurate navigation in between sequential pairs of points chosen from the above generated set.

The first task on the above list will be further subdivided into first generating a dense set of candidate objective points and then pruning of the set to arrive at a sparse sequence of points to be used in the second task. The generation of the dense sequence (tuple) of candidate objective points $\mathcal{O}_c = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_n)$ will be based on a P-complex, halting algorithm incorporating straightforward geometrical rules, presented in table 1.

Prior to running Algorithm 1, all obstacles are extended by a safety-radius at least equal to (or greater than) the radius of the AUV-circumscribing circle, as presented in figure 2. Rectangular obstacles are extended using ∞-norm, preserving the rectangular shape in lieu of circularly beveled corners (as seen in figure 2).

The result of Algorithm 1 is presented in figure 4. The denseness and failure in achieving optimality in $\mathcal{O}_c$ is evident in the existence of points additional to those defining the least number of uncluttered lines of navigation.

A good example of this suboptimal behavior of Algorithm 1 is in the *asterisk- (*) marked* procedure used to assure circumnavigation of circular obstacles. The if clause produces the manner of circumnavigation displayed in figure 3.



(a) AUV-circumcircle (b) Extension of obstacles by AUV-circumcircle

Fig. 2. The principle of obstacle extension by AUV-circumcircle.

Table 1. The candidate objective point interpolation algorithm.

```
set initial point o_b
set final point to o_e
push onto O_c ← o_b
o_c = o_b
while o_c ≠ o_e
    if o_c on a rectangle and o_c ≠ arg min d(vertices, o_e)
        o_n = the vertex of rectangle adjacent to o_c that
            decreases d(·, o_e)
        if both adjacent vertices decrease d(·, o_e)
            o_n = the one closest to o_b o_e
        end if
    else if o_c not on a circle
        o_n = boundary point of the obstacle closest to o_c
            (notwithstanding the obstacle that o_c is on or in)
            among the set of all obstacles closer to o_e than o_c
    else
        o_n = boundary point of the obstacle closest to
            the center of the circle on which perimeter o_c lies
            (notwithstanding the obstacle that o_c is on or in)
            among the set of all obstacles closer to o_e than o_c
        x_3 = the point on circle boundary
            intersected by line from circle center to o_n
        α = the bisector of azimuths of o_c and x_3
            w.r.t. the circle center
        x_0 = point of circle perimeter with azimuth α
        k_1 = slope of tangent to circle at o_c
        k_0 = slope of tangent to circle at x_0
        k_2 = slope of tangent to circle at x_3
        x_1 = intersection of [1 k_1]^T λ + o_c
            and [1 k_0]^T μ + x_0 .
        x_2 = intersection of [1 k_0]^T μ + x_0
            and [1 k_2]^T ν + x_3 .
        push onto O_c ← x_1, x_2, x_3
    end if
    push onto O_c ← o_n
    o_c = o_n
end while
```

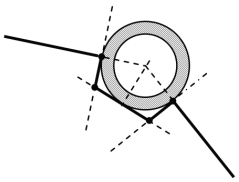


Fig. 3. The circumnavigation of circular obstacles.

The produced sequence of candidate objective points $\mathcal{O}_c$ is finally pruned using Algorithm 2 to produce the final set of objective points $\mathcal{O}_f$ presented in figure 5. This represents the final sequence containing the least number of points that are navigable in sequence along non-occluded lines in the 2D waterspace, from the given initial point to the given final point.

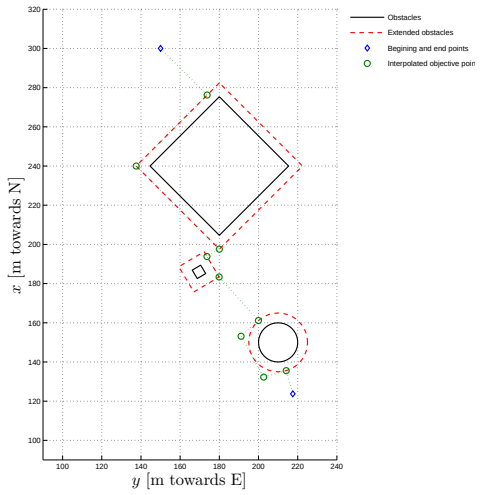


Fig. 4. The interpolated sequence of objective points $\mathcal{O}_c$.

Table 2. The pruning algorithm.

```
n = length(O_c)
while n≥2
    j = n-2
    del = n
    while j≥1
        if O_c(j) visible from O_c(n)
              del = j+1
        end if
        j=j-1
        end while
    delete from O_c elements from i-1 through del
        (if del≤i-1)
    i = del-1
end while
```

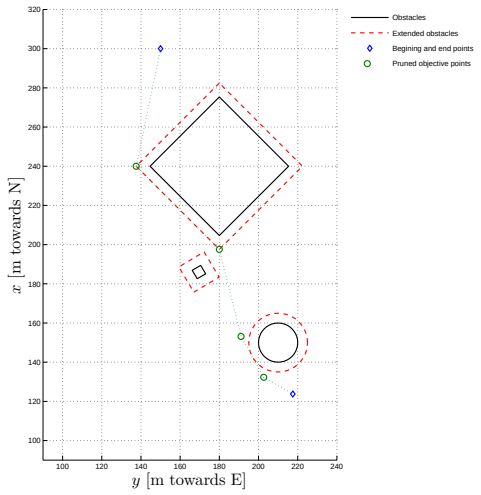


Fig. 5. The final pruned sequence of objective points $\mathcal{O}_f$.

Such a finalized sequence of objective points, $\mathcal{O}_f$ is used as initial data for the algorithms of clothoid fitting described in the next section.

4. CLOTHOID-SPLINE PATHS

A clothoid, Euler spiral or Cornu spiral (Abramowitz and Stegun (1964)) is a parametric bi-spiral curve, with the property that the curvature varies linearly between $-\infty$ and ∞ in the limits. The measure of differential increase in the curvature $C(s)$ across the path of the curve, called *sharpness*, k, is constant, producing for a vehicle traversing a clothoid-shaped path with constant forward speed a directly proportional and constant angular acceleration. This class of curves accordingly represents an ideal candidate for a spline used to plan posture-continuous trajectories in mobile robotics. The research in the application of clothoids for the calculation of posture-continuous paths is proceeding and the most proliferate research area in robotics that explores the application of piecewise-clothoid path segments is 2D path-planning (Wilde (2009), Yossawee et al. (2002), Shimizu et al. (2006), Zelinsky and Dowson (1992)). Research is also ongoing in 3D path-planning using clothoids (Guiqing et al. (2001), Liu et al. (2007)).

The difficulty with clothoid splines is that the parametric forms in terms of clothoid curve length s for the coordinates in the $\mathbb{R}^2$ waterspace of discourse, $(x,\, y)$ are *Fresnel integrals* for which no closed-form solution is obtainable. All clothoids, regardless of initial curvature and initial coordinates are reducible by a variety of scaling transformations to the two basic Fresnel integrals in (4, 5).

$$x(s) = \int_0^s \cos\left(\sigma^2\right)\, d\sigma \tag{4}$$

$$y(s) = \int_0^s \sin\left(\sigma^2\right)\, d\sigma \tag{5}$$

Additionally, for a general problem of linking a pair of poses $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{SE}^2$, which are positions augmented by a reference orientation, $\mathbf{x}_1 = [\mathbf{o}_1 \big| \psi_1]^{\mathrm{T}}$, $\mathbf{x}_2 = [\mathbf{o}_2 \big| \psi_2]^{\mathrm{T}}$, Shin and Singh (1990) have shown that minimally three clothoid segments of the same *amount* (and possibly differing signs) of shaprness are required.

Prior to the numerical calculation of the parameters of the three (or two) clothoid segments comprising the trajectory, all $\mathbf{o}_i \in \mathcal{O}_f$ need to be augmented by reference directions. Also, as starting points for the determination of the lengths $(s_1^{(i)},\, s_2^{(i)},\, s_3^{(i)})$ and sharpness $k^{(i)}$ of each triplet (or pair) of clothoids, initial and final curvatures, $(C_i^{(i)},\, C_f^{(i)})$ need to be determined from $\mathcal{O}_f$.

4.1 Determination of Heading and Curvature

In order to allow for the numerical calculation of triplets (or pairs) of clothoids comprising a trajectory of the AUV, the sequence of objective points $\mathcal{O}_f$, obtained by algorithms described in Section 3 need to be augmented by reference headings and reference path curvatures. This will be performed based on the method presented by Shin and Singh (1990).

The headings (ψ_i) will be determined on the bases of a circumcircle of a triangle $\triangle \mathbf{o}_{i-1,\, i,\, i+1}$, with the center specified by (7) and radius by (8). The headings correspond to the angles of the circle tangent through $\mathbf{o}_i \in$

$\mathcal{O}_f$, $\dim \mathcal{O}_f = n$, $i \neq 1, n$, whose unit direction vector is given in (12). The headings are given in (13). The initial curvatures at each $\mathbf{o}_i$, $i \neq 1, n$ will be the reciprocals of the circumcircular radii given in (8), according to (14). The posture-continuity of the required trajectory dictates that the final curvature $c_f^{(i)}$ for each triplet of clothoids will be $\forall i = 1 \ldots n$, $c_f^{(i)} \stackrel{\text{id}}{=} c_i^{(i+1)}$. In the ensuing formulas, a slight abuse of notation is used by adopting the $\arctan_2(\frac{y}{x})$ four-quadrant arc-tangent that returns an angle in the range of $[0,\, 2\pi)$ depending on the signs of the numerator expression (x) and denominator expression (y).

$$\begin{aligned}
\dim \mathcal{O}_f &= n \\
\mathcal{O}_f &= (\mathbf{o}_i),\ i = 1 \ldots n \\
\forall i\, \mathbf{o}_i &= \begin{bmatrix} x_i \\ y_i \end{bmatrix} && (6) \\
\forall i\, \boldsymbol{c}_i &= \begin{bmatrix} x_c^{(i)} \\ y_c^{(i)} \end{bmatrix} && (7) \\
x_c^{(i)} &= \frac{1}{D}\Big[\|\mathbf{o}_{i-1}\|(y_i - y_{i+1}) + \|\mathbf{o}_i\|(y_{i+1} - y_{i-1}) \\
&\qquad + \|\mathbf{o}_{i+1}\|(y_{i-1} - y_i)\Big] \\
y_c^{(i)} &= \frac{1}{D}\Big[\|\mathbf{o}_{i-1}\|(x_{i+1} - x_i) + \|\mathbf{o}_i\|(x_{i-1} - x_{i+1}) \\
&\qquad + \|\mathbf{o}_{i+1}\|(x_i - xi - 1)\Big] \\
D &\stackrel{\text{def}}{=} 2\Big[x_{i-1}(y_i - y_{i+1}) + x_i(y_{i+1} - y_{i-1}) \\
&\qquad + x_{i+1}(y_{i-1} - y_i)\Big] \\
\forall i\, r_c^{(i)} &= \frac{\|\mathbf{o}_{i+1} - \mathbf{o}_i\| \cdot \|\mathbf{o}_i - \mathbf{o}_{i-1}\| \cdot \|\mathbf{o}_{i-1} - \mathbf{o}_{i+1}\|}{2\sqrt{s(s - \|\mathbf{o}_{i+1} - \mathbf{o}_i\|)(s - \|\mathbf{o}_i - \mathbf{o}_{i-1}\|)\cdot \\ \cdot (s - \|\mathbf{o}_{i-1} - \mathbf{o}_{i+1}\|)}} && (8) \\
s &\stackrel{\text{def}}{=} \frac{1}{2}\Big(\|\mathbf{o}_{i+1} - \mathbf{o}_i\| + \|\mathbf{o}_i - \mathbf{o}_{i-1}\| \\
&\qquad + \|\mathbf{o}_{i-1} - \mathbf{o}_{i+1}\| \Big) \\
\hat{\boldsymbol{n}}_i &\stackrel{\text{def}}{=} \frac{\boldsymbol{c}_i - \mathbf{o}_i}{r_c^{(i)}} && (9) \\
\Delta\mathbf{o}_i &\stackrel{\text{def}}{=} \frac{\mathbf{o}_{i+1} - \mathbf{o}_i}{\|\mathbf{o}_{i+1} - \mathbf{o}_i\|},\ \|\Delta\mathbf{o}_i\| \stackrel{\text{id}}{=} 1 && (10) \\
\hat{\boldsymbol{u}}_i &= \begin{bmatrix} u_x^{(i)} \\ u_y^{(i)} \end{bmatrix} && (11) \\
&= \text{sign}\left(\Delta\mathbf{o}_i^{\mathrm{T}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \hat{\boldsymbol{n}}_i \right) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \hat{\boldsymbol{n}}_i && (12) \\
\forall i\, \psi_i &= \arctan_2 \frac{u_y^{(i)}}{u_x^{(i)}} && (13)
\end{aligned}$$

$$C_i^{(i)} = \frac{1}{r_c^{(i)}} \qquad (14)$$

$$C_f^{(i)} \stackrel{\text{id}}{=} C_i^{(i+1)} = \frac{1}{r_c^{(i+1)}} \qquad (15)$$

Where:
- n is the number of objective points,
- $\boldsymbol{c}_i = [x_c^{(i)}\, y_c^{(i)}]^{\mathrm{T}}$ is the center of the circumcircle of the triangle $\triangle\mathbf{o}_{i-1,\,i,\,i+1}$,
- $r_c^{(i)}$ is the radius of the circumcircle of $\triangle\mathbf{o}_{i-1,\,i,\,i+1}$,
- $\hat{\boldsymbol{n}}_i$ is the unit inward-facing normal vector to the circumcircle of $\triangle\mathbf{o}_{i-1}\mathbf{o}_i\mathbf{o}_{i+1}$ at $\mathbf{o}_i$,
- $\hat{\boldsymbol{u}}_i = [u_x^{(i)} u_y^{(i)}]^{\mathrm{T}}$, $\sqrt{u_x^{(i)\,2} + u_y^{(i)\,2}} \stackrel{\text{id}}{=} 1$ is the unit tangent vector to the circumcircle of $\triangle\mathbf{o}_{i-1,\,i,\,i+1}$, at $\mathbf{o}_i$, facing towards $\mathbf{o}_{i+1}$.

For the initial point $\mathbf{o}_1 \stackrel{\text{id}}{=} \mathbf{o}_b$, and the final point $\mathbf{o}_n \stackrel{\text{id}}{=} \mathbf{o}_e$ the heading and curvature settings are given below:

$$\hat{\boldsymbol{u}}_1 \stackrel{\text{id}}{=} \Delta\mathbf{o}_i \qquad (16)$$

$$\hat{\boldsymbol{u}}_n \stackrel{\text{id}}{=} \Delta\mathbf{o}_{n-1} \qquad (17)$$

$$\psi_{1,\,n} = \arctan_2 \frac{u_y^{(1,\,n)}}{u_x^{(1,\,n)}} \qquad (18)$$

$$C_1 = C_n \stackrel{\text{id}}{=} 0 \qquad (19)$$

The figure 6 displays the $\mathcal{O}_f$ (cf. fig. 5) with the augmentations of (ψ_i), (C_i) according to (12,13,14).

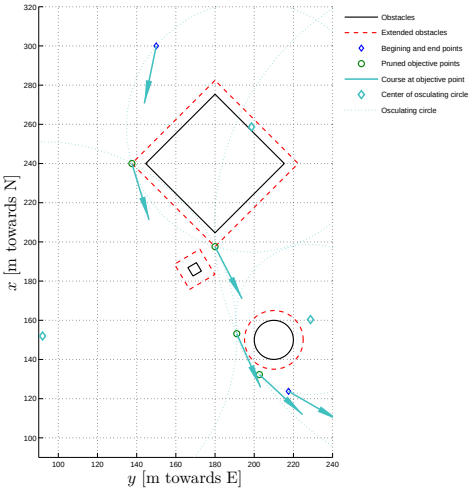


Fig. 6. Objective points $\mathcal{O}_f$ augmented by (ψ_i), (C_i).

4.2 Numerical Approximation of Clothoid Arcs

As explained previously, clothoid curves cannot be solved for $(x,\, y) \in \mathbb{R}^2$ in closed form. Therefore, a heuristic numerical approximation method is adopted that uses the heuristic based on the (ψ_i), (C_i) sequences in the previous subsection.

Consider the equations for a triplet of directed clothoid arcs of identical amount of sharpness, k_i, albeit with a changeover in sign for each segment in the form of a triplet of sharpness parameters $(k_i, -k_i, k_i)$, of path lengths $(s_1^{(i)}, s_2^{(i)}, s_3^{(i)})$, and with the objective point index i omitted for clarity:

$$\forall s \in [0, s_1 + s_2 + s_3)$$

$$C(s) = C_i \tag{20}$$

$$+ \begin{cases} s \le s_1 : & ks \\ s_1 < s \le s_1 + s_2 : & k(2s_1 - s) \\ s_2 < s \le s_1 + s_2 + s_3 : & k(s - 2s_2) \end{cases} \tag{21}$$

$$C_f \overset{\text{id}}{=} C_i + k(s_1 - s_2 + s_3) \tag{22}$$

$$\psi(s) = \psi_i \tag{23}$$

$$+ \begin{cases} s \le s_1 : & C_i s + \frac{k}{2} s^2 \\ s_1 < s \le \sum_i^2 s_i : & (C_i + 2ks_1)s \\ & -\frac{k}{2}(2s_1^2 - s^2) \\ \sum_i^2 s_i < s \le \sum_i^3 s_i : & C_i s - 2ks_2(s - s_1) \\ & +\frac{k}{2}(2s_2^2 + s^2) \end{cases} \tag{24}$$

$$\psi_f \overset{\text{id}}{=} \psi_{i+1} \overset{\text{id}}{=} \psi_i + C_i(s_1 + s_2 + s_3) + k(s_1 s_2 - s_2 s_3 + s_1 s_3) + \frac{k}{2}(s_1^2 - s_2^2 + s_3^2) \tag{25}$$

$$x(s) = \int_0^s \cos\psi(\sigma)\, d\sigma \tag{26}$$

$$y(s) = \int_0^s \sin\psi_(\sigma)\, d\sigma \tag{27}$$

The equations (22 – 27) form a system of 4 equations in 4 unknowns - (s_1, s_2, s_3, k), that can be used for the fitting of clothoid arcs to a pair of objective points ($\mathbf{o}_i$, $\mathbf{o}_{i+1}$). Since (26, 27) have no closed form solution, a successive approximation numerical method will include an optimization over (s_1, s_2) $\in \mathbb{R}^{+2}$. The initial pair of values will be based on the length of circular arcs of osculating circles. The osculating circles ($\boldsymbol{C}_i$) are defined by their respective centers in (7) and radii in (8). From those, $s^{(0)}_{1,2,3}$ can be initialized. In the case of differing signs of clothoid segments, when connection by a pair of clothoid segments is possible, $s^{(0)}_{1\,2}$, $s^{(0)}_3 \overset{\text{id}}{=} 0$ are initialized to half the average of the length of the arcs connecting a pair of points ($\mathbf{o}_i$, $\mathbf{o}_{i+1}$) measured along the two osculating circles (of curvatures C_i, C_{i+1}) through the pair. Otherwise, $s^{(0)}_1$, $s^{(0)}_2$ are initialized to thirds of the average arc length. This is summarized in equations (28 – 35).

$$\forall i = 1, \ldots n$$

$$\varphi_i^{(i)} = \arctan_2 \frac{(\mathbf{o}_i - \boldsymbol{c}_i)\hat{\boldsymbol{\jmath}}}{(\mathbf{o}_i - \boldsymbol{c}_i)\hat{\boldsymbol{\imath}}} \tag{28}$$

$$\varphi_{i+1}^{(i)} = \arctan_2 \frac{(\mathbf{o}_{i+1} - \boldsymbol{c}_i)\hat{\boldsymbol{\jmath}}}{(\mathbf{o}_{i+1} - \boldsymbol{c}_i)\hat{\boldsymbol{\imath}}} \tag{29}$$

$$\varphi_i^{(i+1)} = \arctan_2 \frac{(\mathbf{o}_i - \boldsymbol{c}_{i+1})\hat{\boldsymbol{\jmath}}}{(\mathbf{o}_i - \boldsymbol{c}_{i+1})\hat{\boldsymbol{\imath}}} \tag{30}$$

$$\varphi_{i+1}^{(i+1)} = \arctan_2 \frac{(\mathbf{o}_{i+1} - \boldsymbol{c}_{i+1})\hat{\boldsymbol{\jmath}}}{(\mathbf{o}_{i+1} - \boldsymbol{c}_{i+1})\hat{\boldsymbol{\imath}}} \tag{31}$$

$$l_i^{(i)} = r_c^{(i)} \Delta\varphi^{(i)} = r_c^{(i)}(\varphi_{i+1}^{(i)} - \varphi_i^{(i)}) \tag{32}$$

Table 3. The fitting algorithm.

```
for i = 1:number of objective points-1
    initialize s_1^(0;i), s_2^(0;i) according to (28 - 35)
    do
        solve (22, 25) to obtain s_3^(0;i), k_i^(0)
        solve by means of 1000-step Euler solver eqns. (26, 27)
        increment no. of iteration j
    until ||[x(s_1^(j;i) + s_2^(j;i) + s_3^(j;i)) y(s_1^(j;i) + s_2^(j;i) + s_3^(j;i))]^T − o_{i+1}|| <
        half the greatest dimension of the AUV
end for
```

$$l_{i+1}^{(i)} = r_c^{(i+1)} \Delta\varphi^{(i+1)} = r_c^{(i+1)}(\varphi_{i+1}^{(i+1)} - \varphi_i^{(i+1)}) \tag{33}$$

$$\text{sign}(k_i) \neq \text{sign}(k_{i+1}) \quad \Rightarrow \quad s_1^{(0;\,i)} = s_2^{(0;\,i)} = \frac{l_i^{(i)} + l_{i+1}^{(i)}}{4} \tag{34}$$

$$\text{otherwise} \quad \Rightarrow \quad s_1^{(0;\,i)} = s_2^{(0;\,i)} = \frac{l_i^{(i)} + l_{i+1}^{(i)}}{6} \tag{35}$$

Where:
- $\varphi^{(i)}_{i,\,i+1}$ are the angles subtended by the radii through points $\mathbf{o}_{i,\,i+1}$ on the osculating circle $\boldsymbol{C}_i$,
- $\varphi^{(i+1)}_{i,\,i+1}$ are the angles subtended by the radii through points $\mathbf{o}_{i,\,i+1}$ on the osculating circle $\boldsymbol{C}_{i+1}$,
- $l^{(i)}_{i,\,i+1}$ are the arc lengths between points $\mathbf{o}_{i,\,i+1}$ measured on the osculating circles $\boldsymbol{C}_{i,\,i+1}$, respectively.

Using the initialization described in (28 - 35), the equations (22), 25) to obtain (s_3, k) on the basis of heuristics for (s_1, s_2), and a 1000-point based Euler solver to solve (26, 27), the fitting algorithm in table 3 is used to connect the objective points $\mathcal{O}_f$ in a posture-continuous manner dictated by (ψ_i, C_i).

The path resulting from the objective point fit and augmentation by (ψ_i, C_i) as depicted in figure 6, using the algorithm specified in table 3, is depicted in figure 7.

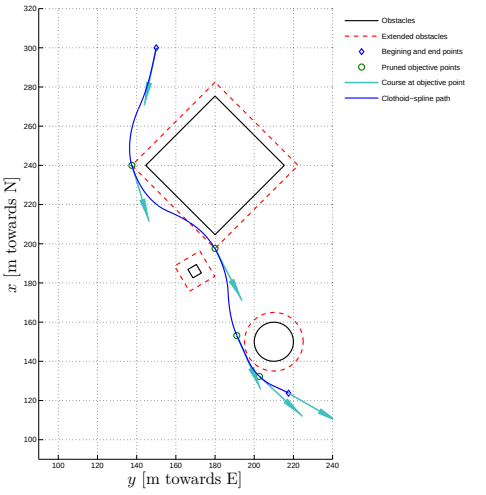


Fig. 7. Path planned for an AUV in a congested environment, through objective points specified in $\mathcal{O}_f$.

5. CONCLUSIONS AND FURTHER WORK

5.1 Conclusions

An algorithm for the design of a posture-continuous path for an AUV navigating in a cluttered waterspace has been presented. The algorithm relies on a priori knowledge of a static and stationary situation within the waterspace, in terms of the location and models of obstacles, drawn from a pair of geometric primitives – rectangles and circles. The algorithm functions swiftly, robustly, and is able to deal with a large degree of congestion in the waterspace. These features make it an ideal choice for inclusion in a subsequent automated trajectory generation algorithm, which can be used for designing on-the-fly trajectories for AUVs performing inspection, monitoring or observation around colonnades of piers, supports, girders or other periodical and interspersed clutter.

5.2 Further Work

The algorithm shall be tested on the Iver 2 AUV operated by the authors' laboratory (Barisic et al. (2010)) in the spring–summer of 2010. The leverage that modern parallel programming paradigms (cf. CUDA^{TM} by AMD, and the on-going work by de P. Veronese and Krohling (2009), Karunadasa and Ranasinghe (2009), Kumar et al. (2009) and Breitbart (2009)) can provide to this type of algorithm will be investigated. Brezak (2010) gives a good treatment of the optimization necessary to allow this type of function to be performed by an embedded computer in soft real-time.

ACKNOWLEDGEMENTS

This work has relied on infrastructure provided by the EU FP7 Capacities REGPOT-2008-1 project "*Developing the Croatian Underwater Robotics Research Potential*", Grant Agreement no. 229553 [1] . The work has been partly financed by the MSES [2] research grant no. 036-0362975-2999, *RoboMarSec – Underwater Robotics in Sub-Sea Protection and Maritime Security*, and a UKF [3] 2A Gaining Experience doctoral students' visits grant awarded for *A Distributed System for Coordinated Control of a Fleet of Autonomous Submersibles* [4] .

REFERENCES

Abramowitz, M. and Stegun, I.A. (1964). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 9. dover printing, 10. GPO printing edition.

Barisic, M., Vukic, Z., Miskovic, N., and Nagy, G. (2010). Developing the Croatian underwater robotics research potential. In *7th IFAC Symposium on Intelligent Autonomous Vehicles, IAV 2010.*, on CD.

Bildberg, R. (2009). Editor's foreword. In *Proceedings of the 16th International Symposium on Unmanned Untethered Submersibles Technology*, on CD.

Breitbart, J. (2009). Cupp - a framework for easy CUDA integration. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, 1 –8.

Brezak, M. (2010). *Localization, Motion Planning and Control of Mobile Robots in Intelligent Spaces*. Ph.D. thesis, University of Zagreb, Faculty of Electrical Engineering & Computing, Unska 3, HR-10000 Zagreb, Croatia.

Brutzman, D. (1994). *A Virtual World for an Autonomous Underwater Vehicle*. Ph.D. thesis, Naval Postgraduate School, 700 Dyer Rd., Monerey, CA 93943-5000, USA.

de P. Veronese, L. and Krohling, R. (2009). Swarm's flight: Accelerating the particles using C-CUDA. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, 3264 –3270.

deArruda, J. (2009). *VectorMap Mission Planner for Iver 2 Autonomous Underwater Vehicle, a technical manual*. OceanServer Technology Inc., 151 Marine St., Fall River, MA, USA.

Guiqing, L., Xianmin, L., and Hua, L. (2001). 3D discrete clothoid splines. In *Computer Graphics International 2001. Proceedings*, 321 –324.

Karunadasa, N.P. and Ranasinghe, D.N. (2009). Accelerating high performance applications with CUDA and MPI. In *Industrial and Information Systems (ICIIS), 2009 International Conference on*, 331 –336.

Kumar, N., Satoor, S., and Buck, I. (2009). Fast parallel expectation maximization for Gaussian mixture models on GPUs using CUDA. In *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, 103 –109.

Lee, C.S. (2004). *NPS AUV Workbench: Collaborative Environment for Autonomous Underwater Vehicles (AUV) Mission Planning and 3D Visualization*. Master's thesis, Naval Postgraduate School, 700 Dyer Rd., Monerey, CA 93943-5000, USA.

Liu, C., Cheng, W., and Hong, Z. (2007). A trajectory generator for a mobile robot in 3D pathplanning. In *Automation and Logistics, 2007 IEEE International Conference on*, 1247 –1251.

Shimizu, M., Kobayashi, K., and Watanabe, K. (2006). Clothoidal curve-based path generation for an autonomous mobile robot. In *SICE-ICASE, 2006. International Joint Conference*, 478 –481.

Shin, D.H. and Singh, S. (1990). *Path Generation For Robot Vehicles Using Composite Clothoid Segments*. Carnegie-Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA.

Wilde, D. (2009). Computing clothoid segments for trajectory generation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2440 –2445.

Yossawee, W., Tsubouchi, T., Sarata, S., and Yuta, S. (2002). Path generation for articulated steering type vehicle using symmetrical clothoid. In *Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on*, volume 1, 187 – 192.

Zelinsky, A. and Dowson, I. (1992). Continuous smooth path execution for an autonomous guided vehicle (AGV). In *TENCON '92. "Technology Enabling Tomorrow : Computers, Communications and Automation towards the 21st Century.' 1992 IEEE Region 10 International Conference.*, 871 –875 vol.2.

[1] *http://cure.fer.hr*

[2] Ministry of Science, Education and Sports of the Republic of Croatia.

[3] Unity Through Knowledge Fund of the Republic of Croatia.

[4] *http://labust.fer.hr/mbarisic*