

An Euclidean norm based criterion to assess robots' 2D path-following performance

Eleonora Saggini^{1,2}, Maria-Laura Torrente^{1,*}

¹ *Department of Mathematics, University of Genova, Italy*

² *Institute of Intelligent Systems for Automation, National Research Council, Genova, Italy*

Abstract. A current need in the robotics field is the definition of methodologies for quantitatively evaluating the results of experiments. This paper contributes to this by defining a new criterion for assessing path-following tasks in the planar case, that is, evaluating the performance of robots that are required to follow a desired reference path. Such criterion comes from the study of the local differential geometry of the problem. New conditions for deciding whether or not the zero locus of a given polynomial intersects the neighbourhood of a point are defined. Based on this, new algorithms are presented and tested on both simulated data and experiments conducted at sea employing an Unmanned Surface Vehicle.

2000 Mathematics Subject Classifications: 26C10, 15A60, 93C85, 62P99.

Key Words and Phrases: Euclidean norm, Weighted matrix norm, Crossing algorithm, Robotics, Good experimental methodologies, Path-following.

1. Introduction and motivation

We present new conditions for deciding whether or not the zero locus of a given polynomial intersects the neighbourhood of a point. This problem has already been addressed in [19] where a local analysis is performed, and both necessary and sufficient numerical crossing conditions are provided in terms of the polynomial's evaluation at the point and using $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms. Extending some results presented in [14], in this paper we exploit the approach of [19] to get new crossing conditions which use the more popular $\|\cdot\|_2$ norm. The main reason for this choice stands in the application of our methods to the robotic field, a true guideline of our work. The importance of developing and spreading Good Experimental Methodologies (GEMs) and standards for performance evaluation within the robotic community is recognized in many papers. A brief but comprehensive review of the issues related to measuring and comparing research results in robotics is provided in [5], which also discusses the need of defining benchmarks for specific sub-domains

*Corresponding author.

Email addresses: eleonora.saggini@ge.issia.cnr.it (E. Saggini), torrente@dima.unige.it (M. Torrente)

of robotics (visual servoing, grasping, motion planning,...) rather than benchmarks valid for all domains. At the same time, there are very few works in the literature that actively address these issues and even less in marine robotics, the subfield in which we are mainly interested. In [4] the author lists the main challenges in marine robotics, stressing the importance of obtaining a credible measurement for algorithms' performance versus the costs needed for organizing sea trials. Recent work by the authors deals with the definition of the identification of GEMs and practices to carry out repeatable experiments with Unmanned Surface Vehicles (USVs) [3] and with the definition and validation of performance indices in marine robotics applications [16].

Using the new $\|\cdot\|_2$ -norm crossing conditions (see Propositions 3, 4, 6 and 8), in the current paper we extend to the $\|\cdot\|_2$ norm the methodology presented in [15] for evaluating surface path-following experiments executed employing marine robots. A two step strategy for evaluating the capability of a robot to follow a reference path and an overall index of performance are defined. This is exploited for evaluating the control system mounted on the Charlie USV, a vessel developed in Genova by the Institute of Intelligent Systems for Automation (ISSIA) of the National Research Council (CNR).

The remainder of the paper is organized as follows: in Section 2 some background material used throughout the paper is introduced. In Section 3 the main theoretical results are presented: the necessary and sufficient crossing conditions (presented in Subsections 3.1 and 3.2) are gathered in the CROSSING Algorithm (CA-2) and Approximate CROSSING Algorithm (ACA-2), see Subsection 3.3. The behaviour of the crossing criteria under some affine transformations is analyzed in Subsection 3.4: namely, in the general case invariance under translation and rotation is proved, whereas invariance under uniform scaling holds for polynomials of special form. In Section 4 a two step methodology to be applied to path-following experiments in robotics' trials is set up, and results on real and simulated data are presented in Sections 5 and 6. Finally, in Section 7 the online applicability of the ACA-2 algorithm is supported by some simulated tests and corresponding computational time recorded.

2. Background material

We start this section recalling basic definitions and properties of matrices (see [8]).

Let m, n be positive integers; we denote by $\text{Mat}_{m \times n}(\mathbb{R})$ the set of $m \times n$ matrices with entries in \mathbb{R} ; if $m = n$ we simply write $\text{Mat}_n(\mathbb{R})$. For any $M \in \text{Mat}_{m \times n}(\mathbb{R})$, we denote by M^t its transpose. Let v be an element of $\text{Mat}_{n \times 1}(\mathbb{R})$. The ∞ -norm of v is defined by $\|v\|_\infty = \max_{i=1}^n |v_i|$; the 2-norm (also known as *Euclidean norm*) of v is defined by $\|v\|_2 = (\sum_{i=1}^n |v_i|^2)^{1/2}$. The well-known *Cauchy-Schwarz inequality* states that for each $v, w \in \text{Mat}_{n \times 1}(\mathbb{R})$ we have $|v^t w| \leq \|v\|_2 \|w\|_2$.

In the following definition we recall another very useful norm on $\text{Mat}_{n \times 1}(\mathbb{R})$, the *weighted 2-norm* (see [20]).

Definition 1. Let W be a positive diagonal matrix in $\text{Mat}_n(\mathbb{R})$.

1. The W -weighted 2-norm on $\text{Mat}_{n \times 1}(\mathbb{R})$ is defined by the formula

$$\|v\|_W := \|Wv\|_2$$

where $v \in \text{Mat}_{n \times 1}(\mathbb{R})$ and Wv denotes the usual product of matrices.

2. Let p be a point of \mathbb{R}^n ; the $(2, W)$ -unit ball centered at p , simply denoted by $\mathbf{B}_W(p)$, is the closed convex set defined as

$$\mathbf{B}_W(p) = \{\mathbf{x} \in \mathbb{R}^n \text{ such that } \|(\mathbf{x} - p)^t\|_W \leq 1\}.$$

In the following definition we recall two matrix norms on $\text{Mat}_{m \times n}(\mathbb{R})$ ($n > 1$): the matrix 2-norm induced by the 2-norm on $\text{Mat}_{n \times 1}(\mathbb{R})$ and the Frobenius norm.

Definition 2. Let $M = (m_{ij})$ be a matrix in $\text{Mat}_{m \times n}(\mathbb{R})$.

1. The matrix 2-norm is the norm on $\text{Mat}_{m \times n}(\mathbb{R})$ induced by the 2-norm on $\text{Mat}_{n \times 1}(\mathbb{R})$ and defined by the formula

$$\|M\|_2 := \max_{\|v\|_2=1} \|Mv\|_2,$$

where $v \in \text{Mat}_{n \times 1}(\mathbb{R})$.

2. The Frobenius norm is the norm on $\text{Mat}_{m \times n}(\mathbb{R})$ defined by

$$\|M\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |m_{ij}|^2}.$$

Thanks to the natural identification of $\text{Mat}_n(\mathbb{R})$ with $\text{Mat}_{n^2 \times 1}(\mathbb{R})$, a matrix $M \in \text{Mat}_n(\mathbb{R})$ can be viewed as an element, that we will denote by $M^{(\mathbf{v})}$ to avoid confusion, of $\text{Mat}_{n^2 \times 1}(\mathbb{R})$. Moreover,

$$\|M\|_F = \|M^{(\mathbf{v})}\|_2. \quad (1)$$

We recall a useful relation between the norms introduced above.

Proposition 1. For each $M \in \text{Mat}_{m \times n}(\mathbb{R})$ the following inequalities hold true:

$$\|M\|_2 \leq \|M\|_F \leq \sqrt{\min\{m, n\}} \|M\|_F.$$

Now, we collect some basic facts of analytic nature. We recall the following version of the Mean Value Theorem for vector valued real functions (for the proof we refer to [9]).

Proposition 2. Let $U \subseteq \mathbb{R}^n$ be a convex open set and let $p \in U$. Let $\phi: U \rightarrow \mathbb{R}^m$ be a differentiable vector valued function on U and denote by $D\phi(\mathbf{x})$ the $m \times n$ matrix of first order derivatives of each component of ϕ , that is,

$$D\phi(\mathbf{x}) = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \cdots & \frac{\partial \phi_m}{\partial x_n} \end{pmatrix}$$

Then, for each $\mathbf{x} \in U$, we have

$$\|(\phi(\mathbf{x}) - \phi(p))^t\|_2 < \sup_{0 < \nu < 1} \|D\phi(p + \nu(\mathbf{x} - p))\|_2 \|(\mathbf{x} - p)^t\|_2.$$

Let $U \subseteq \mathbb{R}^n$ be a convex open set, and let $M(\mathbf{x}) = (m_{ij}(\mathbf{x}))$ be a matrix whose entries are the evaluations at $\mathbf{x} \in U$ of differentiable vector valued functions $m_{ij}: U \rightarrow \mathbb{R}^n$. Hence, in particular, $M(p) \in \text{Mat}_n(\mathbb{R})$ for each given point $p \in U$. Following [19, Lemma 1.13], we will use the following special case of Proposition 2.

Lemma 1. *Let $U \subseteq \mathbb{R}^n$ be a convex open set. Fix a point p of U and let $M(\mathbf{x}) = (m_{ij}(\mathbf{x}))$ be a matrix as above. For each $\mathbf{x} \in U$, we have*

$$\|M(\mathbf{x})\|_2 < \sqrt{n}\|M(p)\|_2 + O(\|(\mathbf{x} - p)^t\|_2).$$

Proof. From Proposition 1 and equality (1) it follows that

$$\|M(\mathbf{x})\|_2 \leq \|M(\mathbf{x})\|_F = \|M^{(\mathbf{v})}(\mathbf{x})\|_2. \tag{2}$$

Consider the vector valued function $\phi = (M^{(\mathbf{v})})^t: U \rightarrow \mathbb{R}^{n^2}$ defined by $\phi(\mathbf{x}) := (M(\mathbf{x})^{(\mathbf{v})})^t$. Clearly, ϕ is differentiable on U , so we can apply Proposition 2 to get

$$\|M(\mathbf{x})^{(\mathbf{v})} - M(p)^{(\mathbf{v})}\|_2 < \sup_{0 < \nu < 1} \|D(M(p + \nu(\mathbf{x} - p)))^{(\mathbf{v})}\|_2 \|(\mathbf{x} - p)^t\|_2 = O(\|(\mathbf{x} - p)^t\|_2).$$

Combining the previous inequality with

$$|\|M(\mathbf{x})^{(\mathbf{v})}\|_2 - \|M(p)^{(\mathbf{v})}\|_2| \leq \|M(\mathbf{x})^{(\mathbf{v})} - M(p)^{(\mathbf{v})}\|_2$$

(a consequence of the usual triangular inequality), we obtain

$$\|M(\mathbf{x})^{(\mathbf{v})}\|_2 < \|M(p)^{(\mathbf{v})}\|_2 + O(\|(\mathbf{x} - p)^t\|_2) = \|M(p)\|_F + O(\|(\mathbf{x} - p)^t\|_2).$$

Applying again Proposition 1 we then find

$$\|M(\mathbf{x})^{(\mathbf{v})}\|_2 < \|M(p)\|_F + O(\|(\mathbf{x} - p)^t\|_2) \leq \sqrt{n}\|M(p)\|_2 + O(\|(\mathbf{x} - p)^t\|_2). \tag{3}$$

Combining (3) with (2) we are done. □

3. Crossing conditions

Following the approach of [19], in this section we provide both necessary and sufficient numerical conditions so that the zero locus of a polynomial crosses a bounded region containing a given point of \mathbb{R}^n .

We need to fix some notation which is borrowed from [13]. In particular, we let x_1, \dots, x_n be indeterminates and most of the times we use for simplicity the notation $\mathbf{x} = (x_1, \dots, x_n)$. The multivariate polynomial ring $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \dots, x_n]$ is denoted by P . Given $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$, we denote by $|\alpha|$ the number $\alpha_1 + \dots + \alpha_n$, by $\alpha!$ the

number $\alpha_1! \dots \alpha_n!$, by \mathbf{x}^α the power product $x_1^{\alpha_1} \dots x_n^{\alpha_n}$, and by $\frac{\partial^\alpha f}{\partial \mathbf{x}^\alpha} := \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$ the α -partial derivative of a polynomial $f = f(\mathbf{x}) \in P$.

Moreover, following the standard notation, we denote by $\text{Jac}_f(\mathbf{x}) := \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$ the *Jacobian* (or *gradient*) of f , and by $H_f(\mathbf{x}) := \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,n}$ the $n \times n$ symmetric *Hessian matrix* of f .

Let $f = f(\mathbf{x})$ be a polynomial of P and let $p = (p_1, \dots, p_n)$ be a point of \mathbb{R}^n . Let $\varepsilon_1, \dots, \varepsilon_n$ be positive real numbers. Set

$$\varepsilon := (\varepsilon_1, \dots, \varepsilon_n), \quad \varepsilon_{\min} := \min\{\varepsilon_1, \dots, \varepsilon_n\}, \quad \varepsilon_{\max} := \max\{\varepsilon_1, \dots, \varepsilon_n\},$$

and let $\mathcal{E} \in \text{Mat}_n(\mathbb{R})$ be the positive diagonal matrix with entries $1/\varepsilon_1, \dots, 1/\varepsilon_n$. Throughout this section we shall use the \mathcal{E} -weighted 2-norm on \mathbb{R}^n and we consider the corresponding closed unit ball $\mathbf{B}_\varepsilon(p)$ centered at p (see Definition 1).

3.1. Necessary crossing conditions

We start providing necessary conditions on $|f(p)|$ so that the locus $f = 0$ crosses $\mathbf{B}_\varepsilon(p)$. Such a condition is expressed in terms of the quantity (depending on $\mathbf{B}_\varepsilon(p)$)

$$\mathbf{H} := \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|H_f(\mathbf{x})\|_2. \tag{4}$$

Proposition 3. *Let $f = f(\mathbf{x})$ be a non-constant polynomial of P , let p be a point of \mathbb{R}^n , and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . If*

$$|f(p)| > \|\text{Jac}_f(p)\|_2 \varepsilon_{\max} + \frac{\mathbf{H}}{2} \varepsilon_{\max}^2 =: B_1(f, p, \varepsilon), \tag{5}$$

then the zero locus of f does not cross $\mathbf{B}_\varepsilon(p)$.

Proof. The proof runs parallel to the proof of [19, Proposition 2.1]: it is sufficient to replace $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms by $\|\cdot\|_2$ norm. \square

Since the quantity \mathbf{H} is hard to be computed, the use of Proposition 3 is sometimes not possible in the applications. Therefore, in the following we state another result which is analogous to Proposition 3 but it avoids the computation of \mathbf{H} , yet providing a non-crossing cell condition. The following statement holds in a second-order error analysis, so it is valid for small values of (the components of) the vector ε . To this purpose, for the rest of this section, we assume $\varepsilon_{\max} \ll 1$.

Proposition 4. *Let $f(\mathbf{x})$ be a degree ≥ 2 polynomial of P . Let p be a point of \mathbb{R}^n and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . If*

$$|f(p)| > \|\text{Jac}_f(p)\|_2 \varepsilon_{\max} + \frac{1}{2} \|H_f(p)\|_2 \varepsilon_{\max}^2 := B'_1(f, p, \varepsilon), \tag{6}$$

then the zero locus of f does not cross $\mathbf{B}_\varepsilon(p)$ neglecting contributions of order $O(\varepsilon_{\max}^3)$.

Proof. The proof runs parallel to the proof of [19, Proposition 2.5]: it is sufficient to replace $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms by $\|\cdot\|_2$ norm. \square

It is immediate to observe that the inequalities provided by Proposition 3 and 4 do not depend on the representation of f (that is, they are invariant under scalar multiplication of f).

We end this section by comparing the bounds $B_1(f, p, \varepsilon)$, $B'_1(f, p, \varepsilon)$ (given in Proposition 3 and 4) and the non-crossing bounds provided in [19].

Proposition 5. *Let $B_1(f, p, \varepsilon)$, $B'_1(f, p, \varepsilon)$ be the bounds as above. Further, let*

$$B_{1,\infty}(f, p, \varepsilon) := \|\text{Jac}_f(p)\|_1 \varepsilon_{\max} + \frac{H_\infty}{2} \varepsilon_{\max}^2$$

$$B'_{1,\infty}(f, p, \varepsilon) := \|\text{Jac}_f(p)\|_1 \varepsilon_{\max} + \frac{1}{2} \|H_f(p)\|_\infty \varepsilon_{\max}^2$$

where $H_\infty := \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|H_f(\mathbf{x})\|_\infty$. Then:

- (i) $B_1(f, p, \varepsilon) \geq B'_1(f, p, \varepsilon)$;
- (ii) $B_1(f, p, \varepsilon) \geq \frac{1}{\sqrt{n}} B_{1,\infty}(f, p, \varepsilon)$;
- (iii) $B'_1(f, p, \varepsilon) \geq \frac{1}{\sqrt{n}} B'_{1,\infty}(f, p, \varepsilon)$.

Proof. Item (i) follows from the definition of $B_1(f, p, \varepsilon)$ and $B'_1(f, p, \varepsilon)$, and from the inequality $H = \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|H_f(\mathbf{x})\|_2 \geq \|H_f(p)\|_2$. Item (iii) follows from the definition of $B'_1(f, p, \varepsilon)$ and $B'_{1,\infty}(f, p, \varepsilon)$, and from the inequalities $\|\text{Jac}_f(p)\|_2 \geq \|\text{Jac}_f(p)\|_1 \geq \frac{1}{\sqrt{n}} \|\text{Jac}_f(p)\|_1$ and $\|H_f(p)\|_2 \geq \frac{1}{\sqrt{n}} \|H_f(p)\|_\infty$. Finally, since $H \geq \frac{1}{\sqrt{n}} H_\infty$ item (ii) easily follows. \square

3.2. Sufficient crossing conditions

In this subsection we provide sufficient numerical conditions so that the zero locus of a polynomial crosses a bounded region containing a given point of \mathbb{R}^n .

We first need some technicalities. For each $\mathbf{x} = (x_1, \dots, x_n)$ such that $\text{Jac}_f(\mathbf{x})$ is not zero, we consider the *pseudo-inverse matrix* of $\text{Jac}_f(\mathbf{x})$, defined by

$$\text{Jac}_f^\dagger(\mathbf{x}) := \text{Jac}_f(\mathbf{x})^t (\text{Jac}_f(\mathbf{x}) \text{Jac}_f(\mathbf{x})^t)^{-1} = \frac{\text{Jac}_f(\mathbf{x})^t}{\|\text{Jac}_f(\mathbf{x})\|_2^2}.$$

Note that $\|\text{Jac}_f^\dagger(\mathbf{x})\|_2 = \frac{1}{\|\text{Jac}_f(\mathbf{x})\|_2}$.

For any positive real number R , set

$$\mathcal{D}(p, R) := \{\mathbf{x} \in \mathbb{R}^n \text{ such that } \|(\mathbf{x} - p)^t\|_2 < R\}.$$

Obviously, $\mathcal{D}(p, R) \subseteq \mathbf{B}(p)$ as soon as $R < \varepsilon_{\min}$.

The following lemma is useful for the main result of this subsection.

Lemma 2. *Let $f = f(\mathbf{x})$ be a degree ≥ 2 polynomial of P and let p be a point of \mathbb{R}^n such that the Jacobian $\text{Jac}_f(p)$ is nontrivial. Let R be a positive real number such that $R < \varepsilon_{\min}$. If $R < \frac{\|\text{Jac}_f(p)\|_2}{\mathsf{H}}$, then $\text{Jac}_f(\mathbf{x})$ is nontrivial for $\mathbf{x} \in \mathcal{D}(p, R)$.*

Proof. The proof runs parallel to the proof of [19, Lemma 3.1]: it is sufficient to replace $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms by $\|\cdot\|_2$ norm. \square

Proposition 6. *Let $f = f(\mathbf{x})$ be a degree ≥ 2 polynomial of P , let p be a point of \mathbb{R}^n such that $\text{Jac}_f(p)$ is not the zero vector, and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . Let R be a positive real number such that $R < \min\{\varepsilon_{\min}, \frac{\|\text{Jac}_f(p)\|_2}{\mathsf{H}}\}$ and let*

$$J := \sup_{\mathbf{x} \in \mathcal{D}(p, R)} \|\text{Jac}_f^\dagger(\mathbf{x})\|_2 = \frac{1}{\inf_{\mathbf{x} \in \mathcal{D}(p, R)} \|\text{Jac}_f(\mathbf{x})\|_2} \quad (7)$$

If

$$|f(p)| < \frac{2R}{J(2 + \mathsf{H}JR)} =: B_2(f, p, \varepsilon, R), \quad (8)$$

then the zero locus of f crosses $\mathbf{B}_\varepsilon(p)$.

Proof. The proof runs parallel to the proof of [19, Proposition 3.2] (to which we refer for more details) with some minor changes on the upper bound constants. For completeness we report it here.

If $f(p) = 0$ there is nothing to prove. From Lemma 2 we know that the Jacobian $\text{Jac}_f(\mathbf{x})$ is nonzero for $x \in \mathcal{D} := \mathcal{D}(p, R)$. Moreover, since $R < \varepsilon_{\min}$, one has $\mathcal{D} \subseteq \mathbf{B}_\varepsilon(p)$.

We now construct a sequence of points $\{p_k\}_{k \in \mathbb{N}}$ as follows. We let $p_0 = p$ and, for each $k \geq 0$, we define

$$s_k := -\text{Jac}_f^\dagger(p_k)f(p_k) = -\frac{f(p_k)}{\|\text{Jac}_f(p_k)\|_2^2} \text{Jac}_f(p_k)^t \quad \text{and} \quad p_{k+1} := p_k + s_k^t. \quad (9)$$

Obviously, $p = p_0 \in \mathcal{D}$. We prove by induction that the points p_k 's all lie in \mathcal{D} and satisfy the inequality

$$|f(p_k)| < |f(p_{k-1})| \quad \text{for each } k \geq 1. \quad (10)$$

Step I (The $k = 1$ case). From the definitions of s_0 and J we have

$$\|s_0\|_2 = \|\text{Jac}_f^\dagger(p)\|_2 |f(p)| \leq J |f(p)|.$$

Moreover, by assumption, it follows that $|f(p)| < B_2 < \frac{2R}{2J} = \frac{R}{J}$. Thus $\|s_0\|_2 < R$ showing that $p_1 \in \mathcal{D}$. Applying Taylor's theorem to $f(\mathbf{x})$ at p and then evaluating at p_1 we get

$$f(p_1) = f(p) + \text{Jac}_f(p)(p_1 - p)^t + \frac{1}{2}(p_1 - p)H_f(\xi)(p_1 - p)^t,$$

where ξ is a point of the line that connects p to p_1 . Therefore, by definitions (9), we get

$$f(p_1) = f(p) + \text{Jac}_f(p)s_0 + \frac{1}{2}s_0^t H_f(\xi)s_0$$

$$= |f(p)| \left(\frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2^4} \text{Jac}_f(p) H_f(\xi) \text{Jac}_f(p)^t \right).$$

Let us upper bound the absolute value of the quantity

$$Q := \frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2^4} \text{Jac}_f(p) H_f(\xi) \text{Jac}_f(p)^t.$$

To this end, use Cauchy-Schwarz inequality and recall the definitions of \mathbf{H} (see (4)) and \mathbf{J} to get:

$$\begin{aligned} |Q| &\leq \frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2^4} \|\text{Jac}_f(p)\|_2^2 \|H_f(\xi)\|_2 = \frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2^2} \|H_f(\xi)\|_2 \\ &= \frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2} \|\text{Jac}_f^\dagger(p)\|_2 \|H_f(\xi)\|_2 \leq \frac{1}{2} \frac{|f(p)|}{\|\text{Jac}_f(p)\|_2} \mathbf{JH}. \end{aligned}$$

By the assumption on R we thus obtain $|Q| < \frac{1}{2} |f(p)| \frac{\mathbf{J}}{R}$. On the other hand, $|f(p)| < B_2 < \frac{2R}{2\mathbf{J}} < \frac{2R}{\mathbf{J}}$. Therefore $|Q| < 1$, so that equality (11) reads $|f(p_1)| < |f(p)|$, showing condition (10) for $k = 1$.

Step II (The inductive step). Suppose that the points p, p_1, \dots, p_k of the sequence lie in \mathcal{D} and that $0 < |f(p_k)| < |f(p_{k-1})| < \dots < |f(p)|$. Hence, in particular, the points p, p_1, \dots, p_k are all distinct, so that, by definition, $\|s_{i-1}\|_2 \neq 0$ for $i = 1, \dots, k$.

First we show that $p_{k+1} \in \mathcal{D}$. For each $i = 1, \dots, k$, we apply Taylor's theorem to $f(\mathbf{x})$ at p_{i-1} and evaluate at p_i to get

$$f(p_i) = f(p_{i-1}) + \text{Jac}_f(p_{i-1})(p_i - p_{i-1})^t + \frac{1}{2}(p_i - p_{i-1})H_f(\xi_i)(p_i - p_{i-1})^t, \quad (11)$$

where ξ_i is a point of the line that connects p_{i-1} to p_i . On the other hand, by definition of s_{i-1} and recalling that $\text{Jac}_f(\mathbf{x})\text{Jac}_f^\dagger(\mathbf{x}) = 1$, we have $\text{Jac}_f(p_{i-1})s_{i-1} = -f(p_{i-1})$, whence

$$f(p_{i-1}) = -\text{Jac}_f(p_{i-1})s_{i-1} = -\text{Jac}_f(p_{i-1})(p_i - p_{i-1})^t. \quad (12)$$

By combining (11) and (12) with Cauchy-Schwarz inequality, we get

$$\begin{aligned} |f(p_i)| &= \frac{1}{2} |(p_i - p_{i-1})H_f(\xi_i)(p_i - p_{i-1})^t| \\ &\leq \frac{1}{2} \|H_f(\xi_i)\|_2 \|(p_i - p_{i-1})^t\|_2^2 \leq \frac{1}{2} \mathbf{H} \|s_{i-1}\|_2^2. \end{aligned} \quad (13)$$

Now, define $\tau_i := \frac{\|s_i\|_2}{\|s_{i-1}\|_2}$. Therefore inequality (13) gives

$$\|s_i\|_2 = \|\text{Jac}_f^\dagger(p_i)\|_2 |f(p_i)| \leq \mathbf{J} |f(p_i)| \leq \frac{1}{2} \mathbf{JH} \|s_{i-1}\|_2^2.$$

Thus

$$\tau_i = \frac{\|s_i\|_2}{\|s_{i-1}\|_2} \leq \frac{1}{2} \mathbf{JH} \|s_{i-1}\|_2 \leq \frac{1}{2} \mathbf{J}^2 \mathbf{H} |f(p_{i-1})| < \frac{1}{2} \mathbf{J}^2 \mathbf{H} |f(p)|. \quad (14)$$

Since $|f(p)| < B_2 < \frac{2}{J^2H}$, it must be $\tau_i < 1$ by the above inequality. Let $\tau := \max_{i=1,\dots,k} \{\tau_i\}$. We bound $\|(p_{k+1} - p)^t\|_2$ as follows:

$$\begin{aligned} \|(p_{k+1} - p)^t\|_2 &\leq \|s_0\|_2 + \|s_1\|_2 + \dots + \|s_k\|_2 \\ &= \|s_0\|_2 + \tau_1 \|s_0\|_2 + \tau_1 \tau_2 \|s_0\|_2 + \dots + \tau_1 \tau_2 \dots \tau_k \|s_0\|_2 \\ &= \|s_0\|_2 (1 + \tau_1 + \tau_1 \tau_2 + \dots + \tau_1 \tau_2 \dots \tau_k) \\ &\leq \|s_0\|_2 \sum_{i=0}^k \tau^i < \|s_0\|_2 \sum_{i=0}^{\infty} \tau^i = \frac{\|s_0\|_2}{1 - \tau} \leq \frac{J|f(p)|}{1 - \tau}. \end{aligned}$$

Then, by inequality (14) and the assumption $|f(p)| < B_2$, we find

$$\|(p_{k+1} - p)^t\|_2 < \frac{J|f(p)|}{1 - \frac{1}{2}J^2H|f(p)|} = \frac{2J|f(p)|}{2 - J^2H|f(p)|} < R,$$

therefore $p_{k+1} \in \mathcal{D}$.

Now, let us prove that $|f(p_{k+1})| < |f(p_k)|$. To this purpose we observe that relation (11) can be easily adapted to the pair of points p_k, p_{k+1} in the form

$$f(p_{k+1}) = |f(p_k)| \left(\frac{1}{2} \frac{|f(p_k)|}{\|\text{Jac}_f(p_k)\|_2^4} \text{Jac}_f(p_k) H_f(\xi_k) \text{Jac}_f(p_k)^t \right), \tag{15}$$

where ξ_k is a point of the line connecting p_k to p_{k+1} . Let us upper bound the absolute value of the quantity

$$Q_k := \frac{1}{2} \frac{|f(p_k)|}{\|\text{Jac}_f(p_k)\|_2^4} \text{Jac}_f(p_k) H_f(\xi_k) \text{Jac}_f(p_k)^t.$$

As previously done to upper bound the quantity $|Q|$, by using Cauchy-Schwarz inequality and the definition of H , we get

$$\begin{aligned} |Q_k| &\leq \frac{1}{2} \frac{|f(p_k)|}{\|\text{Jac}_f(p_k)\|_2^2} \|H_f(\xi_k)\|_2 \leq \frac{1}{2} \frac{|f(p_k)|}{\|\text{Jac}_f(p_k)\|_2^2} H \\ &\leq \frac{1}{2} |f(p_k)| J^2 H < \frac{1}{2} |f(p)| J^2 H, \end{aligned} \tag{16}$$

where the last inequality comes from the inductive hypothesis $|f(p_k)| < |f(p)|$. On the other hand, $|f(p)| < B_2 < \frac{2}{HJ^2}$. Thus we find $|Q_k| < 1$, so that equality (15) yields $|f(p_{k+1})| < |f(p_k)|$, as we want.

Step III (Conclusion). If there exists $k \in \mathbb{N}$ such that $f(p_k) = 0$ we are done. Otherwise, we know from Step II that $\tau_k := \frac{\|s_k\|_2}{\|s_{k-1}\|_2} < 1$ for $k \in \mathbb{N}$. Then, by D'Alembert criterion, the series $\sum_{k=1}^{\infty} \|s_k\|_2$ converges, so that $\lim_{k \rightarrow \infty} (\sum_{i=k+1}^{\infty} \|s_i\|_2) = 0$. Define $p^{*t} := p^t + \sum_{k=1}^{\infty} s_k$. Then, since $p_k^t = p^t + \sum_{i=1}^k s_i$, one has

$$\lim_{k \rightarrow \infty} \|(p_k - p^*)^t\|_2 = \lim_{k \rightarrow \infty} \left(\left\| \sum_{i=k+1}^{\infty} s_i \right\|_2 \right) \leq \lim_{k \rightarrow \infty} \left(\sum_{i=k+1}^{\infty} \|s_i\|_2 \right) = 0.$$

Thus the sequence of points $\{p_k\}_{k \in \mathbb{N}}$ converges to the point p^* . Since the p_k 's belong to \mathcal{D} , the point p^* belongs to the closure $\overline{\mathcal{D}} \subseteq \mathbf{B}_\varepsilon(p)$. We also know that $\|s_k\|_2 = \tau_1 \tau_2 \dots \tau_k \|s_0\|_2 < \tau^k \|s_0\|_2$, where $\tau = \sup_{k \in \mathbb{N}} \{\tau_k\}$. Therefore

$$\lim_{k \rightarrow \infty} \|s_k\|_2 < \lim_{k \rightarrow \infty} \tau^k \|s_0\|_\infty = 0.$$

From inequality (13), we then conclude that

$$|f(p^*)| = \lim_{i \rightarrow \infty} |f(p_i)| \leq \frac{1}{2} \mathbf{H} \lim_{i \rightarrow \infty} \|s_{i-1}\|_2^2 = 0.$$

This completes the proof. □

As already pointed out in the previous subsection, the quantities \mathbf{H} and \mathbf{J} are sometimes hard to be computed. As a consequence, applying Proposition 6 could be difficult in some applications. For this reason, in the following we state another result which is analogous to Proposition 6 but it avoids the computation of the quantities \mathbf{H} and \mathbf{J} , yet providing a sufficient crossing condition. Note that the statement holds in a first-order error analysis, so it is only valid for small values of (the components of) the vector ε .

We first need to state some more technical details. The following result yields an upper bound for the quantity $\mathbf{J} = \sup_{\mathbf{x} \in \mathcal{D}(p,R)} \|\text{Jac}_f^\dagger(\mathbf{x})\|_2$ introduced in Proposition 6.

Proposition 7. *Let $f(\mathbf{x})$ be a degree ≥ 2 polynomial of P and let p be a point of \mathbb{R}^n . Let R be a positive real number and suppose that the Jacobian $\text{Jac}_f(\mathbf{x})$ is nonzero for each $\mathbf{x} \in \mathcal{D}(p, R)$. Then*

$$\mathbf{J} < \frac{1}{\|\text{Jac}_f(p)\|_2} \left(1 + 3\sqrt{n} \frac{\|H_f(p)\|_2}{\|\text{Jac}_f(p)\|_2} R \right) + \mathbf{O}(R^2).$$

Proof. The proof runs parallel to the proof of [19, Proposition 4.1] (to which we refer for more details) with some minor changes on the upper bound constants. For completeness we report it here.

Let $\mathbf{x} \in \mathcal{D} := \mathcal{D}(p, R)$. Consider the vector valued function $(\text{Jac}_f^\dagger)^t: \mathcal{D} \rightarrow \mathbb{R}^n$. Since by hypothesis $\text{Jac}_f(\mathbf{x})$ has full row rank in \mathcal{D} , it follows that $(\text{Jac}_f^\dagger)^t$ is differentiable on the open convex set \mathcal{D} . We apply Proposition 2 to get

$$\|\text{Jac}_f^\dagger(\mathbf{x}) - \text{Jac}_f^\dagger(p)\|_2 < \sup_{0 < \nu < 1} \|D \text{Jac}_f^\dagger(p + \nu(\mathbf{x} - p))\|_2 \|\mathbf{x} - p\|_2. \tag{17}$$

Combining (17) with $|\|\text{Jac}_f^\dagger(\mathbf{x})\|_2 - \|\text{Jac}_f^\dagger(p)\|_2| \leq \|\text{Jac}_f^\dagger(\mathbf{x}) - \text{Jac}_f^\dagger(p)\|_2$ (the usual consequence of the triangular inequality), we have

$$\|\text{Jac}_f^\dagger(\mathbf{x})\|_2 < \|\text{Jac}_f^\dagger(p)\|_2 + \sup_{0 < \nu < 1} \|D \text{Jac}_f^\dagger(p + \nu(\mathbf{x} - p))\|_2 \|\mathbf{x} - p\|_2. \tag{18}$$

Applying Lemma 1 to the matrix $M(\mathbf{x}) = D \text{Jac}_f^\dagger(\mathbf{x}) \in \text{Mat}_n(\mathbb{R})$, one has

$$\sup_{0 < \nu < 1} \|D \text{Jac}_f^\dagger(p + \nu(\mathbf{x} - p))\|_2 < \sqrt{n} \|D \text{Jac}_f^\dagger(p)\|_2 + \mathbf{O}(R). \tag{19}$$

We explicitly express $D \text{Jac}_f^\dagger(\mathbf{x})$ by computing the partial derivatives of each component of $\text{Jac}_f^\dagger(\mathbf{x})$. That is,

$$D \text{Jac}_f^\dagger(\mathbf{x}) = \frac{1}{\|\text{Jac}_f(\mathbf{x})\|_2^4} (\|\text{Jac}_f(\mathbf{x})\|_2^2 H_f(\mathbf{x}) - 2 \text{Jac}_f(\mathbf{x})^t \text{Jac}_f(\mathbf{x}) H_f(\mathbf{x})).$$

We now upper bound $\|D \text{Jac}_f^\dagger(p)\|_2$ by

$$\begin{aligned} \|D \text{Jac}_f^\dagger(p)\|_2 &\leq \frac{1}{\|\text{Jac}_f(p)\|_2^4} (\|\text{Jac}_f(p)\|_2^2 + 2\|\text{Jac}_f(p)\|_2^2) \|H_f(p)\|_2 \\ &= \frac{3}{\|\text{Jac}_f(p)\|_2^2} \|H_f(p)\|_2 \end{aligned} \quad (20)$$

By combining (19) and (20) and recalling that $\|\text{Jac}_f^\dagger(p)\|_2 = \frac{1}{\|\text{Jac}_f(p)\|_2}$, inequality (18) yields

$$\|\text{Jac}_f^\dagger(\mathbf{x})\|_2 < \frac{1}{\|\text{Jac}_f(p)\|_2} + 3\sqrt{n} \frac{\|H_f(p)\|_2}{\|\text{Jac}_f(p)\|_2^2} R + O(R^2).$$

By definition of \mathbf{J} we are done. \square

We prove the following technical result, which is valid up to a first-order error analysis.

Lemma 3. *Let $f = f(\mathbf{x})$ be a degree ≥ 2 polynomial of P and let p be a point of \mathbb{R}^n such that both the Jacobian $\text{Jac}_f(p)$ and the Hessian matrix $H_f(p)$ are nontrivial. Let R be a positive real number such that $R < \varepsilon_{\min}$. If $R < \frac{\|\text{Jac}_f(p)\|_2}{\sqrt{n}\|H_f(p)\|_2}$, then, neglecting contributions of order $O(R^2)$, the Jacobian $\text{Jac}_f(\mathbf{x})$ is nonzero for each $\mathbf{x} \in \mathcal{D}(p, R)$.*

Proof. The proof runs parallel to the proof of [19, Lemma 4.2] (to which we refer for more details) with some minor changes on the upper bound constants.

Proposition 8. *Let $f = f(\mathbf{x})$ be a degree ≥ 2 polynomial of P , let p be a point of \mathbb{R}^n such that Jacobian $\text{Jac}_f(p)$ and the Hessian matrix $H_f(p)$ are nontrivial, and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . Let R be a positive real number such that $R < \min \left\{ \varepsilon_{\min}, \frac{\|\text{Jac}_f(p)\|_2}{\sqrt{n}\|H_f(p)\|_2} \right\}$ and set*

$$\Theta := \frac{1}{\|\text{Jac}_f(p)\|_2} \left(1 + 3\sqrt{n} \frac{\|H_f(p)\|_2}{\|\text{Jac}_f(p)\|_2} R \right).$$

If

$$|f(p)| < \frac{2R}{\Theta(2 + \sqrt{n}\|H_f(p)\|_2\Theta R)} =: B'_2(f, p, \varepsilon, R), \quad (21)$$

then the zero locus of f crosses $\mathbf{B}_\varepsilon(p)$ neglecting order $O(R^2)$ contributions.

Proof. The proof uses similar arguments to the proof of Proposition 6 and runs parallel to the proof of [19, Proposition 4.3] (to which we refer for more details) with some minor changes on the upper bound constants. For brevity we omit it here. \square

As for the necessary crossing conditions, it is immediate to observe that the inequalities provided by Proposition 6 and 8 do not depend on the representation of f (that is, they are invariant under scalar multiplication of f).

We conclude this subsection by comparing the bounds $B_1'(f, p, \varepsilon)$, $B_2(f, p, \varepsilon, R)$ and $B_2'(f, p, \varepsilon, R)$ given in Propositions 4, 6 and 8.

Proposition 9. *Notation and assumptions as above. Further assume $\varepsilon_{\max} \ll 1$ and let R be a positive real number such that*

$$R < \min \left\{ \varepsilon_{\min}, \frac{\|\text{Jac}_f(p)\|_2}{H}, \frac{\|\text{Jac}_f(p)\|_2}{\sqrt{n}\|H_f(p)\|_2} \right\}.$$

Then $B_2'(f, p, \varepsilon, R) + O(R^3) < B_2(f, p, \varepsilon, R) < B_1'(f, p, \varepsilon)$.

Proof. The proof runs parallel to the proof of [19, Proposition 4.5] (to which we refer for more details). For brevity we omit it here. \square

As conclusive remark, we observe that from Proposition 9 and Proposition 5, it follows that $B_1(f, p, \varepsilon) > B_2(f, p, \varepsilon, R)$. Therefore, there may be cases in which $B_2 < |f(p)| < B_1$. In such cases, with the only use of Proposition 3 and 6, we can draw no conclusion on the crossing of $f = 0$ around the given point p . Nevertheless, since the previous results hold locally, a more accurate analysis, performed by iteratively considering smaller balls, may overcome the problem.

3.3. The crossing algorithm

In this section, we gather the results obtained in Section 3.1 and 3.2 and introduce an algorithm for deciding whether or not the zero locus of a polynomial f intersects a ball centered at a given point p .

Let $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ be a vector of tolerances; let f be a polynomial and $\mathbf{B}_\varepsilon(p)$ the $(2, \varepsilon)$ -unit ball centered at a point p . Analogously to the Crossing Cell Algorithm (see [19]) for brevity called CA- ∞ here, our algorithm processes f and $\mathbf{B}_\varepsilon(p)$ and returns a value which describes the intersection of $f = 0$ with $\mathbf{B}_\varepsilon(p)$. Namely,

- 0 if $f = 0$ does not cross $\mathbf{B}_\varepsilon(p)$;
- 1 if $f = 0$ crosses $\mathbf{B}_\varepsilon(p)$;
- ξ (unknown) if neither Proposition 3 nor Proposition 6 applies.

Summarizing, we have:

The CROSSING Algorithm (CA-2)

Given a non-constant polynomial $f = f(\mathbf{x}) \in P$, a point $p \in \mathbb{R}^n$ such that both the Jacobian $\text{Jac}_f(p)$ and the Hessian matrix $H_f(p)$ are nontrivial at p , and a tolerance vector $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$, the algorithm returns an element of $\{0, 1, \xi\}$, namely 0 if $f = 0$ does not cross $\mathbf{B}_\varepsilon(p)$, 1 if $f = 0$ crosses $\mathbf{B}_\varepsilon(p)$, and ξ (unknown) if neither Proposition 3 nor Proposition 6 applies.

1. Compute $|f(p)|$ and the bounds $B_1(f, p, \varepsilon)$ (from Proposition 3).
If $|f(p)| > B_1(f, p, \varepsilon)$ return 0.
2. Else $R < \min \left\{ \varepsilon_{\min}, \frac{\|\text{Jac}_f(p)\|_2}{H} \right\}$ and compute $B_2(f, p, \varepsilon, R)$ (from Proposition 6).
If $|f(p)| < B_2(f, p, \varepsilon)$ return 1; else return ξ .

Note that, as for the Approximated CROSSING Algorithm (ACA- ∞ , see [19]), a variant of the CROSSING Algorithm is simply obtained by replacing the bounds $B_1(f, p, \varepsilon)$ and $B_2(f, p, \varepsilon, R)$ by $B'_1(f, p, \varepsilon)$ and $B'_2(f, p, \varepsilon, R)$ (defined in Proposition 4 and 8): in this way the variant, called the Approximated CROSSING Algorithm (ACA-2), works in a first-order error analysis. All the algorithms presented above and employed in Sections 5 and 6 (ACA-2, CA-2 and ACA- ∞) are written in Matlab code and available at <https://sites.google.com/site/eleonorasaggini/shared-files>.

3.4. Affine transformations

In this subsection we investigate on the behaviour of the bounds B_1 , B'_1 , B_2 , and B'_2 under some affine transformations. In Proposition 10 it is shown that B_1 , B'_1 , B_2 , and B'_2 are invariant under translation and rotation. The invariance under uniform scaling is proved in Proposition 11 for polynomials of special form.

Proposition 10. *Fix notation and assumptions as in Proposition 3, 4, 6, and 8. Then, the bounds B_1 , B'_1 , B_2 , and B'_2 are invariant under translation and rotation.*

Proof. Let $f = f(\mathbf{x})$ be a degree ≥ 2 polynomial of P , let p be a point of \mathbb{R}^n and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . Let $v \in \mathbb{R}^n$ be a vector and $W \in \text{Mat}_n(\mathbb{R})$ be an orthogonal matrix (that is $WW^t = I_n$). Let $T_v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $T_W : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined by

$$T_v(\mathbf{x}) = \mathbf{x} + v \quad \text{and} \quad T_W(\mathbf{x}) = W\mathbf{x},$$

which denote a translation and a rotation, respectively. Let $g_v = f \circ T_v$ and $g_W = f \circ T_W$ be polynomials of P and let $q_v = T_v^{-1}(p) = p - v$ and $q_W = T_W^{-1}(p) = Wp$ be points of \mathbb{R}^n . Further, let $\varepsilon_W = W\varepsilon$. Regarding the Jacobian vectors, we have:

$$\begin{aligned} \text{Jac}_{g_v}(\mathbf{x}) &= \text{Jac}_f(T_v(\mathbf{x})) \cdot \text{Jac}_{T_v}(\mathbf{x}) = \text{Jac}_f(\mathbf{x} + v) \\ \text{Jac}_{g_W}(\mathbf{x}) &= \text{Jac}_f(T_W(\mathbf{x})) \cdot \text{Jac}_{T_W}(\mathbf{x}) = \text{Jac}_f(W\mathbf{x}) \cdot W \end{aligned} \tag{22}$$

from which

$$\begin{aligned}\| \text{Jac}_{g_v}(q_v) \|_2 &= \| \text{Jac}_f(p) \|_2 \\ \| \text{Jac}_{g_W}(q_W) \|_2 &= \| \text{Jac}_f(p) \cdot W \|_2 = \| \text{Jac}_f(p) \|_2\end{aligned}\quad (23)$$

Regarding the Hessian matrices, we have:

$$\begin{aligned}H_{g_v}(\mathbf{x}) &= H_f(T_v(\mathbf{x})) = H_f(\mathbf{x} + v) \\ H_{g_W}(\mathbf{x}) &= W^t \cdot H_f(T_W(\mathbf{x})) \cdot W = W \cdot H_f(W\mathbf{x}) \cdot W\end{aligned}$$

from which

$$\begin{aligned}\| H_{g_v}(q_v) \|_2 &= \| H_f(p) \|_2 \\ \| H_{g_W}(q_W) \|_2 &= \| W H_f(p) \cdot W \|_2 = \| H_f(p) \|_2\end{aligned}\quad (24)$$

From the definition of B'_1 in Proposition 4 and equalities (23) and (24), it easily follows that

$$\begin{aligned}B'_1(g_v, q_v, \varepsilon) &= B'_1(f, p, \varepsilon), \\ B'_1(g_W, q_W, \varepsilon_W) &= B'_1(f, p, \varepsilon).\end{aligned}$$

Now, let $H(f, p, \varepsilon) = \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \| H_f(\mathbf{x}) \|_2$ be as defined in (4); we have:

$$\begin{aligned}H(g_v, q_v, \varepsilon) &= \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(q_v)} \| H_{g_v}(\mathbf{x}) \|_2 = \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(q_v)} \| H_f(\mathbf{x} + v) \|_2 \\ &= \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \| H_f(\mathbf{x}) \|_2 = H(f, p, \varepsilon) \\ H(g_W, q_W, \varepsilon_W) &= \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(q_W)} \| H_{g_W}(\mathbf{x}) \|_2 = \max_{\mathbf{x} \in \mathbf{B}_{\varepsilon_W}(q_W)} \| W H_f(W\mathbf{x}) W \|_2 \\ &= \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \| H_f(\mathbf{x}) \|_2 = H(f, p, \varepsilon)\end{aligned}\quad (25)$$

From the definition of B_1 in Proposition 3 and equalities (23) and (25), it easily follows:

$$\begin{aligned}B_1(g_v, q_v, \varepsilon) &= B_1(f, p, \varepsilon), \\ B_1(g_W, q_W, \varepsilon_W) &= B_1(f, p, \varepsilon).\end{aligned}$$

From Proposition 6, let $R < \min\{\varepsilon_{\min}, \frac{\| \text{Jac}_f(p) \|_2}{H(f, p, \varepsilon)}\}$ and $J(f, p, R) = \frac{1}{\inf_{\mathbf{x} \in \mathcal{D}(p, R)} \| \text{Jac}_f(\mathbf{x}) \|_2}$. From equalities (22) on the Jacobian vectors we have $J(g_v, q_v, R) = J(g_W, q_W, R) = J(f, p, R)$. These equalities combined with equalities (25) and the definition of B_2 in Proposition 6 yield:

$$\begin{aligned}B_2(g_v, q_v, \varepsilon, R) &= B_2(f, p, \varepsilon, R), \\ B_2(g_W, q_W, \varepsilon_W, R) &= B_2(f, p, \varepsilon, R).\end{aligned}$$

Finally, from Proposition 8, let $R < \min\{\varepsilon_{\min}, \frac{\| \text{Jac}_f(p) \|_2}{\sqrt{n} \| H_f(p) \|_2}\}$ and

$$\Theta(f, p, R) = \frac{1}{\| \text{Jac}_f(p) \|_2} \left(1 + 3\sqrt{n} \frac{\| H_f(p) \|_2}{\| \text{Jac}_f(p) \|_2} R \right)$$

From equalities (23) and (24) we have $\Theta(g_v, q_v, R) = \Theta(g_W, q_W, R) = \Theta(f, p, R)$. These equalities combined with (24) and the definition of B'_2 in Proposition 8 yield:

$$\begin{aligned} B'_2(g_v, q_v, \varepsilon, R) &= B'_2(f, p, \varepsilon, R), \\ B'_2(g_W, q_W, \varepsilon_W, R) &= B'_2(f, p, \varepsilon, R). \end{aligned}$$

This concludes the proof. \square

Proposition 11. *Fix notation and assumptions as in Proposition 3, 4, 6, and 8. Further assume that the polynomial $f \in P$ has degree $d \geq 2$ and it is of the form $f = f_d + f_0$, with $f_0, f_d \in P$ homogeneous polynomials of degree 0 and d respectively. Then, the bounds B_1, B'_1, B_2 , and B'_2 are invariant under uniform scaling.*

Proof. Let p be a point of \mathbb{R}^n and let $\mathbf{B}_\varepsilon(p)$ be the unit ball centered at p . Let $\gamma \in \mathbb{R}$ be a positive constant and $g_\gamma(\mathbf{x}) = f(\gamma\mathbf{x})$ be the polynomial of P obtained uniformly scaling the variables \mathbf{x} . By definition of f and g we have:

$$g(\mathbf{x}) = f(\gamma\mathbf{x}) = f_d(\gamma\mathbf{x}) + f_0(\gamma\mathbf{x}) = \gamma^d f_d(\mathbf{x}) + f_0.$$

For each $i = 1, \dots, n$, the first order derivative $\partial g_\gamma / \partial x_i$ of g can be expressed as:

$$\frac{\partial g_\gamma}{\partial x_i} = \gamma^d \frac{\partial f_d}{\partial x_i} + \frac{\partial f_0}{\partial x_i} = \gamma^d \frac{\partial f_d}{\partial x_i}.$$

Using the fact that $\frac{\partial f_d}{\partial x_i}$ is homogeneous of degree $d - 1$ we obtain

$$\frac{\partial g_\gamma}{\partial x_i}(\gamma^{-1}\mathbf{x}) = \gamma^d \frac{\partial f_d}{\partial x_i}(\gamma^{-1}\mathbf{x}) = \gamma^d (\gamma^{-1})^{d-1} \frac{\partial f_d}{\partial x_i}(\mathbf{x}) = \gamma \frac{\partial f_d}{\partial x_i}(\mathbf{x}) = \gamma \frac{\partial f}{\partial x_i}(\mathbf{x}).$$

It easily follows that

$$\text{Jac}_g(\gamma^{-1}\mathbf{x}) = \gamma \text{Jac}_f(\mathbf{x}). \quad (26)$$

Analogously, for each $j = 1, \dots, n$, the second order derivative $\partial^2 g_\gamma / \partial x_i \partial x_j$ of g can be expressed as:

$$\frac{\partial^2 g_\gamma}{\partial x_i \partial x_j} = \gamma^d \frac{\partial^2 f_d}{\partial x_i \partial x_j}$$

and using the fact that $\frac{\partial^2 f_d}{\partial x_i \partial x_j}$ is homogeneous of degree $d - 2$ we obtain

$$\frac{\partial^2 g_\gamma}{\partial x_i \partial x_j}(\gamma^{-1}\mathbf{x}) = \gamma^d \frac{\partial^2 f_d}{\partial x_i \partial x_j}(\gamma^{-1}\mathbf{x}) = \gamma^d (\gamma^{-1})^{d-2} \frac{\partial^2 f_d}{\partial x_i \partial x_j}(\mathbf{x}) = \gamma^2 \frac{\partial^2 f_d}{\partial x_i \partial x_j}(\mathbf{x}) = \gamma^2 \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}).$$

It easily follows that

$$H_g(\gamma^{-1}\mathbf{x}) = \gamma^2 H_f(\mathbf{x}). \quad (27)$$

From the definition of B'_1 in Proposition 4 and equalities (26) and (27) evaluated at p , it easily follows that $B'_1(g_\gamma, \gamma^{-1}p, \gamma^{-1}\varepsilon) = B'_1(f, p, \varepsilon)$.

Now, let $\mathbf{H}(f, p, \varepsilon) = \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|H_f(\mathbf{x})\|_2$ be as defined in (4). From (27) we have:

$$\begin{aligned} \mathbf{H}(g_\gamma, \gamma^{-1}p, \gamma^{-1}\varepsilon) &= \max_{\mathbf{x} \in \mathbf{B}_{\gamma^{-1}\varepsilon}(\gamma^{-1}p)} \|H_{g_\gamma}(\mathbf{x})\|_2 = \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|H_{g_\gamma}(\gamma^{-1}\mathbf{x})\|_2 \\ &= \max_{\mathbf{x} \in \mathbf{B}_\varepsilon(p)} \|\gamma^2 H_f(\mathbf{x})\|_2 = \gamma^2 \mathbf{H}(f, p, \varepsilon) \end{aligned} \quad (28)$$

From the definition of B_1 in Proposition 3 and equalities (26) evaluated at p and (28), it easily follows that $B_1(g_\gamma, \gamma^{-1}p, \gamma^{-1}\varepsilon) = B_1(f, p, \varepsilon)$.

From Proposition 6, let $R < \min\{\varepsilon_{\min}, \frac{\|Jac_f(p)\|_2}{\mathbf{H}(f, p, \varepsilon)}\}$ and $\mathbf{J}(f, p, R) = \frac{1}{\inf_{\mathbf{x} \in \mathcal{D}(p, R)} \|Jac_f(\mathbf{x})\|_2}$.

Using equality (26) we have $\mathbf{J}(g_\gamma, \gamma^{-1}p, \gamma^{-1}R) = \gamma^{-1}\mathbf{J}(f, p, R)$, which combined with (28) and the definition of B_2 in Proposition 6 yield $B_2(g_\gamma, \gamma^{-1}p, \gamma^{-1}\varepsilon, \gamma^{-1}R) = B_2(f, p, \varepsilon, R)$.

Finally, from Proposition 8, let $R < \min\{\varepsilon_{\min}, \frac{\|Jac_f(p)\|_2}{\sqrt{n}\|H_f(p)\|_2}\}$ and

$$\Theta(f, p, R) = \frac{1}{\|Jac_f(p)\|_2} \left(1 + 3\sqrt{n} \frac{\|H_f(p)\|_2}{\|Jac_f(p)\|_2} R \right).$$

From (26) and (27) evaluated at p it follows that $\Theta(g_\gamma, \gamma^{-1}p, \gamma^{-1}R) = \gamma^{-1}\Theta(f, p, R)$. This equality combined with (27) evaluated at p and the definition of B'_2 in Proposition 8 yields $B'_2(g_\gamma, \gamma^{-1}p, \gamma^{-1}\varepsilon, \gamma^{-1}R) = B'_2(f, p, \varepsilon, R)$. \square

Example 1. We consider the polynomial $f = x^3 - 2x^2y + y^3 - 8$ and the point $p = (1, 1)$. We let $\varepsilon = (1, 1)$ and consider the $(2, \varepsilon)$ -unit ball centered at p . Propositions 4 and 8 can not be applied since the components of ε are not small enough. According to Proposition 11 the problem can be uniformly scaled and crossing properties can be inferred by the new scaled data.

Let $\gamma = 2$ be the scaling factor. Let $g(x, y) = f(\gamma x, \gamma y) = 8x^3 - 8x^2y + 8y^2 - 8$, $q = \gamma^{-1}p = (1/2, 1/2)$ and $\varepsilon' = \gamma^{-1}\varepsilon = (0.5, 0.5)$. Propositions 4 and 8 are now applicable to g in the $(2, \varepsilon')$ -unit ball $\mathbf{B}_{\varepsilon'}(q)$ centered at q . From some computations we get

$$|g(q)| = 8, \quad B'_1 \approx 5.6503 \quad \text{and} \quad B'_2 \approx 0.0152$$

Since $|g(q)| > B'_1$, from Proposition 4 we conclude that the zero locus of g does not cross $\mathbf{B}_{\varepsilon'}(q)$. Applying Proposition 11 the same conclusion holds for the zero locus of f in $\mathbf{B}_\varepsilon(p)$, that is the zero locus of f does not cross $\mathbf{B}_\varepsilon(p)$.

4. Applications to marine robotics

The theory presented in Section 3 is exploited to define a methodology for evaluating the capability of an USV to follow a desired path. Let $\mathcal{R} = \{(x_{R,i}, y_{R,i}), i = 1, \dots, r\} \subset \mathbb{R}^2$ and $\mathcal{V} = \{(x_{V,i}, y_{V,i}), i = 1, \dots, v\} \subset \mathbb{R}^2$ be two sets identifying the reference and vehicle paths, respectively. The proposed method provides a quantitative evaluation of the closeness of a vehicle to the reference path through the following steps:

Step I computation of an algebraic curve $f = 0$ that approximates the points in \mathcal{R} within a tolerance ε_1 ;

Step II identification of the points in \mathcal{V} which are close to the path defined by the curve $f = 0$ by less than a tolerance ε_2 .

This method has been defined and tested for the first time in [15]; its feasibility has been proved on simulated data for the metric induced by the ∞ -norm. However, for the specific goal of the path-following performance evaluation, the common sense suggests to design the methodology using the Euclidean distance, that is employing the 2-norm. To this aim, the results illustrated in Section 3 are exploited. Using this new methodology we perform more simulations and experiments at sea, allowing comparison between the results achieved with the two different metrics (see Section 5). In the following, details about the two step methodology are presented and tests on real and simulated experiments are reported in Section 5. The general methods behind the two steps hold in higher dimension but we illustrate the basic ideas for $n = 2$ because of the application to $2D$ path-following.

4.1. Step I: approximation of a path by a polynomial curve

There are several methods in the literature to address the problem of approximating a path through a polynomial curve. An interesting class of recently developed algorithms relies on tools from Numerical Commutative Algebra [17, 2, 10, 6, 7]. For all these algorithms the input is a set of points possibly in n -dimensions and the output is a polynomial f in n -variables whose zero locus (which is a curve, or a surface, or more generally an algebraic variety) gives an approximation for the input points and can be interpreted as an implicit polynomial regression model [12, Ch 2]. The algorithm presented in [7] called Low-degree Polynomial Algorithm (LPA) is particularly interesting for our purposes because it returns a “simple” polynomial f whose zero locus “almost” contains the points. The “simplicity” of a polynomial f is measured by its total degree whereas a point is said to be “almost” contained in the zero locus of f if the ε_1 -ball (w.r.t. a given norm) centred at the point intersects the locus $f = 0$. In the LPA the ∞ -norm is used, and ε_1 represents the maximum error of the coordinates of a point. For further details on the algebraic perspective see [7].

4.2. Step II: identification of the points close to the reference path

In order to identify the points of the vehicle path close to the reference path we introduce the following rule based on the analytic results of Section 3. Let $f = 0$ be the algebraic plane curve computed in Step I and let ε_2 be a fixed bidimensional vector of tolerances for the closeness of a generic point $p = (x_V, y_V)$ of the vehicle path to $f = 0$. For each point, the value of $|f(p)|$ and the bounds $B_1(f, p, \varepsilon_2)$ and $B_2(f, p, \varepsilon_2, R)$ (refer to equations (5) and (8)) are computed for establishing whether a point is far from $f = 0$ by more than ε_2 . To this aim, the CROSSING algorithm (CA-2) introduced in Subsection 3.3 is applied to f and each point $p \in \mathcal{V}$.

As we have already observed, CA-2 requires the non trivial computation of H and J (defined in (4) and in (7)) for $B_1(f, p, \varepsilon_2)$ and $B_2(f, p, \varepsilon_2, R)$. For this reason, we decided to implement CA-2 adopting

$$\begin{aligned} H &:= \max\{\|H_f(\mathbf{x}_i)\|_2, i = 1, \dots, n_B : \mathbf{x}_i \text{ is a random point in } \mathbf{B}_\varepsilon(p)\} \\ J &:= 1/\min\{\|\text{Jac}_f(\mathbf{x}_i)\|_2, i = 1, \dots, n_D : \mathbf{x}_i \text{ is a random point in } \mathcal{D}(p, R)\}, \end{aligned}$$

where n_B and n_D are big enough. In order to decrease computation time, a variant of CA-2 which uses the approximated bounds $B'_1(f, p, \varepsilon_2)$ and $B'_2(f, p, \varepsilon_2, R)$ is also adopted (called ACA-2, see Section 3.3) in the experiments presented in Section 5. Note that the constraint $\varepsilon_2 < 1$ does not limit the applicability of the proposed method within the robotic community, since nowadays the majority of the vehicles are usually required to stay close to the path for less than 1 meter.

Given the output of our algorithm, a measure for evaluating the path-following manoeuvre from the starting time t^* until time t is given by the percentage $P_C(t)$ of points which are close to the path:

$$P_C(t) = 100 \frac{\text{Number of points close to the path in } [t^*, t]}{\text{Number of total points in } [t^*, t]}.$$

An online measure $P_{NC}(t)$ of the percentage of points of \mathcal{V} far from the path is defined analogously, and the percentage of undecided cases is $P_U(t) = 100 - P_C(t) - P_{NC}(t)$. The starting time t^* corresponds to the instant in which the vehicle can be declared to be on the path, namely it is said to be in *steady state*. Thus, in order to fairly evaluate the capability to follow a path, the transient state of the vehicle approaching the path has to be neglected; this depends on the type of experiments' execution and it will be specified in the examples presented in the following. Furthermore, a performance measure of the overall path-following manoeuvre is given by a unique value $P_C := P_C(t_{tot})$ computed at the end of the experiment. Similarly, the quantities P_{NC} and P_U are defined at time $t = t_{tot}$.

Remark 1. Let P_{NC}^{ACA-2} and P_{NC}^{CA-2} be the total percentage of points of \mathcal{V} far from the path, computed by using the ACA-2 and CA-2 algorithms respectively (see Section 3.3). From Propositions 3, 4 and 5 it directly follows that

$$P_{NC}^{ACA-2} \geq P_{NC}^{CA-2}.$$

5. Experiments

In this section we present and discuss the results achieved when applying the two step methodology defined in Section 4 to simulated and on field data collected employing the Charlie USV.

The available implementation of LPA (see Section 4.1) is done using the C++ language, the CoCoALib [1], and some routines of GSL - GNU Scientific Library [11]. All computations have been performed on an Intel Core i5 processor at 1.4 GHz. LPA assumes an

ordering of monomials: we use the degree lexicographic term ordering and consider both $x < y$ and $y < x$. Further, note that in Section 5 the coefficients of the polynomials are displayed as truncated decimals.

5.1. Experiments at sea

A test campaign was conducted in May 2015 in the Canale di calma di Prà, Genova, using two different CNR-ISSIA vessels: Charlie USV and Shark USSV (Unmanned Semi-Submersible Vehicle). The main aim consists to test a new software framework named *DeepRuler* for the automatic execution of path-following tasks, collecting experimental data and evaluating the performance of the control architecture through different performance indices defined by the authors. For details about the test campaign see [18]. Here we consider an example of execution of a sinusoidal path-following trial carried out by the Charlie USV, when the path is a sine wave with amplitude 10 *m* and 3 hemi-periods in 100 *m* (see Figures 1 and 3). Because of possible presence of uncontrollable variables or external disturbances that can influence the result of a single trial, *DeepRuler* has been implemented for executing each path both in forward and backward directions, and other repetitions can be added. Therefore, an analysis on the results is presented for both the directions of execution. An important feature to mention is that the discrimination between the transient state and the steady state is automatically decided by *DeepRuler* during the experiment execution. For this reason, given \mathcal{V} as the set for the vehicle motion during the steady state, we have immediately $t^* = 0$.

5.2. Forward path

In the forward case, the reference path \mathcal{R} is made up of $r = 26$ points, whereas the set \mathcal{V} is made up of $v = 1952$ points representing the GPS coordinates of the Charlie USV. Since the sampling rate is 8 *Hz*, we obtain that the experiment duration is about 4 minutes.

Step I. Different results can be achieved by applying LPA to \mathcal{R} and by varying the term ordering and the value of ε_1 . A “good” approximating curve $f_{FW}(x, y) = 0$ is obtained choosing the degree lexicographic term ordering with $y < x$ and $0.01 \leq \varepsilon_1 \leq 0.05$:

$$\begin{aligned} f_{FW}(x, y) = & x^5 + 1.90194x^4y - 3.14118x^3y^2 + 2.009x^2y^3 + 1.1038xy^4 + 0.578489y^5 \\ & + 77.6378x^4 - 19.0024x^3y + 90.0171x^2y^2 - 33.4575xy^3 + 11.3227y^4 \\ & - 1740.85x^3 - 3786.6x^2y + 1483.78xy^2 - 1047.06y^3 - 188247x^2 \\ & + 205996xy + 145757y^2 + 2.01272 \cdot 10^6x + 6.98548 \cdot 10^6y + 5.43407 \cdot 10^7. \end{aligned}$$

The time for this computation is 3 minutes and 58 seconds. The curve $f_{FW}(x, y) = 0$ is represented by the black line in Figures 1 and 2.

Step II. Let $\varepsilon_2 = (0.9, 0.9)$ be the tolerance vector. For each point of \mathcal{V} , we apply ACA-2 to f_{FW} with tolerance ε_2 : the result is shown in Figure 1, where the vehicle motion is displayed by dots differently coloured, that correspond to the three possible results of the

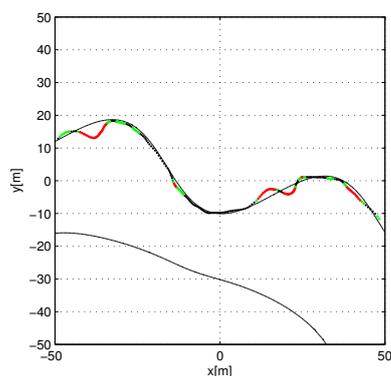


Figure 1: Charlie USV forward execution of a sinusoidal path with 3 half-periods and an amplitude of $10m$ within $100m$. The black line is the curve $f_{FW}(x, y) = 0$ while the coloured dots (black, red and green) correspond to the different outputs of the ACA-2 (cross, not cross and uncertainty), respectively.

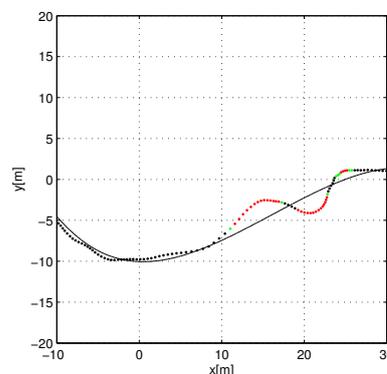


Figure 2: Figure 1 zoomed in the region $[-10, 30] \times [-20, 20]$.

decision rule (black-the vehicle is close to the reference path, red-the vehicle is far from the reference path, green-uncertainty). Figure 2 zooms on an interesting region, where the vehicle is far from the path and this is highlighted by the proposed closeness criterion. The results of an overall measure of the path-following performance are then reported in Table 1 (column 1) by means of the indices P_C , P_{NC} and P_U .

Table 1: Performance indices computed for the forward sinusoidal path depicted in Figure 1 employing different algorithms, together with the computation time needed.

Index	Algorithm		
	ACA-2	CA-2	ACA- ∞
P_C	50%	63%	13%
P_{NC}	29%	29%	33%
P_U	21%	8%	54%
Time [s]	85	902	75

Looking at Figures 1 and 2 and the values in Table 1, it is clear that the vehicle is able to stay on the path for the majority of the time, apart from three evident oscillations, that are all captured by the decision rule and depicted with red dots. The black dots on the curve in Figure 2 also highlight the prompt response of the algorithm when the vehicle crosses again the path, after being at a certain distance from it.

In addition to the ACA-2 algorithm, the CA-2 and ACA- ∞ versions have been tested. Results of the performance achieved by the vehicle together with the computation time are displayed in Table 1 (columns 2-3). The following is a list of remarks.

- (a) Looking at the performance evaluated with the 2-norm algorithms, it is clear that for the ACA-2 algorithm there is much more uncertainty in the classification with

respect to the CA-2 (columns 1-2). Furthermore, note that percentage of not crossing points is the same and this is supported by Remark 1.

- (b) There is a substantial difference between the values corresponding to the 2-norm and ∞ -norm algorithms (columns 1-2 and 3). In particular, in the latter case a great number of points are improperly unclassified, as suggested by the evidence that the value of $P_C^{\text{ACA}-\infty}$ is too small to properly represent the percentage of points close to the reference curve. This supports the initial conjecture about the choice of the 2-norm for a better correspondence to the common sense of closeness within path following applications. Furthermore, note also that the $P_{NC}^{\text{ACA}-\infty}$ index is higher than other cases.
- (c) A check on the computation time needed by each algorithm is important to actively discuss the online applicability of the proposed criterion, that refers to the possibility to monitor and evaluate the experiments' execution. In the following we make a few remarks in the case of path-following experiments (2 dimensional case) while an analysis on the computational time in higher dimensions is postponed in Section 7. Recalling that the experiment duration is about 4 minutes and checking the last row in Table 1, the conclusion is that the CA-2 algorithm can be exploited only in a post-processing of the data as an additional technique, since it requires about 15 minutes to process data collected in only 4 minutes. On the contrary, the other algorithms are suitable to be applied online. Further, note that the polynomial curve $f_{FW} = 0$ approximating \mathcal{R} is computed only once and is given as input for step II as soon as the vehicle starts the path-following.

The conclusion that can be drawn from the considerations above is that the ACA-2 introduced in this paper reveals to be a valuable algorithm for evaluating online the geometrical accuracy of path-following experiments, while more precision can be gained in a post processing analysis with the CA-2 algorithm.

5.3. Backward path

Similarly to the forward case (see Subsection 5.2), a set \mathcal{R} of $r = 26$ points is given as reference for the backward path, whereas the set \mathcal{V} is made up of $v = 1520$ points. The total experiment duration is about 3 minutes.

Step I. A plane curve $f_{BW}(x, y) = 0$ is computed with the LPA algorithm in 4 minutes and 9 seconds. The curve is obtained with degree lexicographic term ordering, $y < x$ and $0.01 \leq \varepsilon_1 \leq 0.05$:

$$\begin{aligned} f_{BW}(x, y) = & x^5 + 1.90194x^4y - 3.14118x^3y^2 + 2.009x^2y^3 + 1.1038xy^4 + 0.578489y^5 \\ & - 77.6378x^4 + 19.0024x^3y - 90.0171x^2y^2 + 33.4575xy^3 - 11.3227y^4 \\ & - 1740.85x^3 - 3786.6x^2y + 1483.78xy^2 - 1047.06y^3 + 188247x^2 \\ & - 205996xy - 145757y^2 + 2.01272 \cdot 10^6x + 6.98548 \cdot 10^6y - 5.43407 \cdot 10^7. \end{aligned}$$

Note that, there is an evident similarity with the expression of $f_{FW}(x, y)$ and a sort of “symmetry” between the black lines in Figures 1 and 3 that can be associated to the fact that the path to follow is essentially the same. However, a deeper investigation of this fact is not in the purposes of the present paper.

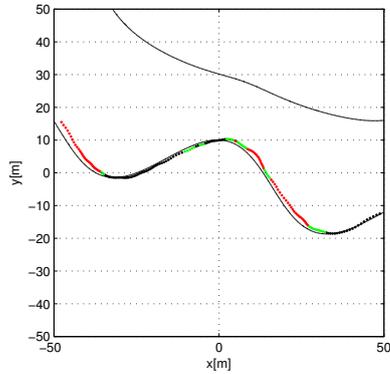


Figure 3: Charlie USV backward execution of a sinusoidal path with 3 half-periods and an amplitude of $10m$ within $100m$. For the legend the reader can refer to Figure 1.

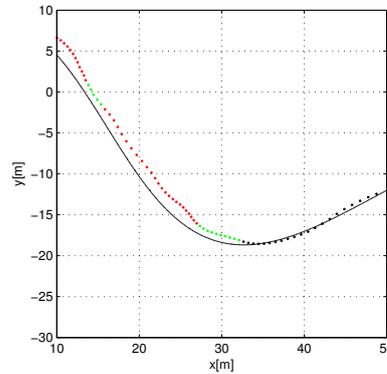


Figure 4: Figure 3 zoomed in the region $[10, 50] \times [-30, 10]$.

Step II. The norms and bounds adopted in this step correspond to those of the forward case and $\varepsilon_2 = (0.9, 0.9)$. The output of the ACA-2 algorithm is shown in Figure 3, while Figure 4 zooms on a region where the vehicle clearly passes from being on the path to being far from it (the interpretation of the backward path is right to left). There is an evident difference between the vehicle behaviour during the forward and backward executions: this can be imputed to external disturbances, because the difficulty in following the path is the same and engineers assume that a path-following performance remains unvaried during a test campaign. During the backward execution, in fact there are two significant departures from the line, that can suggest the presence of a heavy disturbance, e.g. a current or wind. The online monitoring of the path following performance could have alerted the human operator about such unexpected behaviours.

Finally, a comparison of the results achieved with different algorithms can be deduced from Table 2. The reader can note that the points (a)-(c) discussed in Subsection 5.2 also hold in the backward case.

6. Simulated experiments

In this section we present two examples of path following trials executed employing the Charlie USV simulator, a valuable tool for testing the vehicle behaviour while following a desired path and validating new control algorithms. For details on this architecture the reader can refer to [15].

Table 2: Performance indices computed for the backward sinusoidal path depicted in Figure 3 employing different algorithms, together with the computation time needed.

Index	Algorithm		
	ACA-2	CA-2	ACA- ∞
P_C	43%	50%	13%
P_{NC}	37%	37%	44%
P_U	20%	13%	43%
Time [s]	65	667	62

6.1. Path following on a closed path

A closed circuit is given as reference for the vehicle by means of $r = 32$ points, whereas the set \mathcal{V} is made up of $v = 2801$ points.

Step I. A plane curve $f = 0$ is computed with the degree lexicographic term ordering, $x < y$ and $0.75 < \varepsilon_1 < 0.95$:

$$\begin{aligned}
 f(x, y) = & y^3x + 1.5662y^2x^2 + 1.08512yx^3 + 0.159258x^4 - 23.448y^3 - 26.0963y^2x \\
 & - 5.17682yx^2 - 27.9674x^3 + 3499.08y^2 - 394.723yx - 2514.2x^2 \\
 & - 153897y - 26141.8x + 1.87511 \cdot 10^6.
 \end{aligned}$$

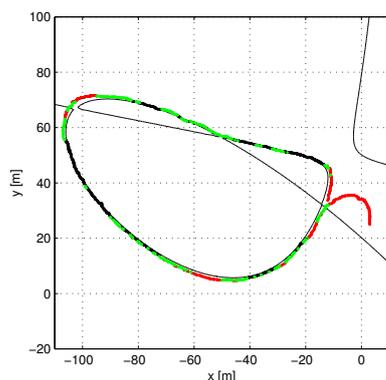


Figure 5: Simulation of the Charlie USV executing a circuit with coloured dots that refer to the output of the ACA-2 algorithm. For the legend the reader can refer to Figure 1.

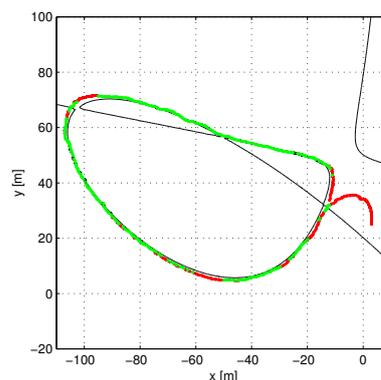


Figure 6: Simulation of the Charlie USV executing a circuit with coloured dots that refer to the output of the ACA- ∞ algorithm. For the legend the reader can refer to Figure 1.

Step II. The ACA-2 and ACA- ∞ algorithms have been tested by setting $\varepsilon_2 = (0.9, 0.9)$. From a quick comparison of the outputs displayed in Figures 5 and 6, it is clear how better the ACA-2 performs w.r.t. ACA- ∞ , that never assesses that the vehicle is on the path. Nevertheless, the numerous green points in Figure 5 suggest that many cases of uncertainty in the decision rule can happen, especially in the vicinity of the singularities. Finally, we observe that the red dots are limited to regions where the vehicle is definitely

far from the path, e.g. during the approach (region $[-12, 3] \times [24, 36]$) and during the turn (region $[-105, 95] \times [64, 72]$).

6.2. Path following on a path with a self-intersection

A path that exhibits a self-intersection is given as reference for the vehicle by means of $r = 33$ points, whereas the set \mathcal{V} is made up of $v = 2854$ points.

Step I. A plane curve $f = 0$ is computed with the degree lexicographic term ordering, $x < y$ and $0.01 < \varepsilon_1 < 0.05$:

$$\begin{aligned} f(x, y) = & y^4x + 0.286695y^3x^2 + 0.132059y^2x^3 - 0.0401925yx^4 + 0.0114993x^5 \\ & + 10.9289y^4 - 204.845y^3x - 66.1921y^2x^2 + 0.923376yx^3 + 4.59715x^4 \\ & - 4914.98y^3 + 12333.1y^2x + 6751.34yx^2 + 2581.01x^3 + 728885y^2 \\ & - 135438yx + 1433.46x^2 - 4.29953 \cdot 10^7y - 3.16167 \cdot 10^6x + 7.81355 \cdot 10^8. \end{aligned}$$

A “good” property of the polynomial curve above is that it properly reflects the particular geometry of the reference \mathcal{R} showing a singularity around the point $(-46, 52)$.

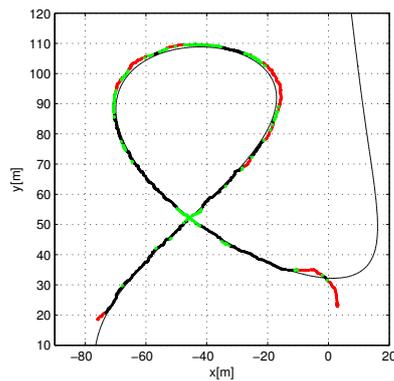


Figure 7: Simulation of the Charlie USV executing a path with a singularity, with coloured dots that refer to the output of the ACA-2 algorithm. For the legend the reader can refer to Figure 1.

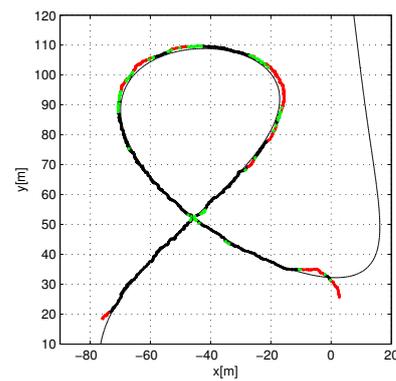


Figure 8: Simulation of the Charlie USV executing a path with a singularity, with coloured dots that refer to the output of the CA-2 algorithm. For the legend the reader can refer to Figure 1.

Step II. The ACA-2 and CA-2 algorithms have been tested on the vehicle set \mathcal{V} , by setting $\varepsilon_2 = (0.9, 0.9)$. As it is expected by Remark 1, the number of points classified as far from the path (red points) in Figure 7 is clearly greater than those in Figure 8. Furthermore, it is evident that for both the algorithms the point of singularity is associated to uncertainty, but the number of these points (that are green coloured) reduces when employing the most accurate computation, that is the CA-2 algorithm.

7. A simulation study for $n \geq 2$

In this section we discuss test examples in higher dimensions focussing on the computational time needed by the algorithms ACA-2 and CA-2. Our aim is not to present a long careful complexity analysis of the two algorithms (which, apart from the dimension and the number of points, would heavily depend on their configuration), rather to show via examples how the computational time scales as the dimension increases. Matlab code for running the following example is freely available at the following link <https://sites.google.com/site/eleonorasaggini/shared-files>.

Example 2. We consider three polynomials in 2, 4 and 8 variables:

$$\begin{aligned} f_2 &= -3x_1^3 + 5x_1^2 - x_1 + x_2 + 3 \\ f_4 &= x_4 + 3x_1^3 - 2x_2^3 + x_3^3 + 5x_1x_3^2 + x_1^2 - 4x_1x_2 + 5x_1x_3 - 10x_2x_3 + 7x_2 - 3x_3 + 9 \\ f_8 &= x_8 + 3x_1^3 - 2x_5^3 + x_4^3 + 5x_2x_3^2 + x_4^2 - 4x_5x_6 + 5x_6x_7 - 10x_1x_3 + 7x_4 - 3x_5 + 9 \end{aligned}$$

We consider three sets of points: $\mathbb{X}_2 \subset \mathbb{R}^2$, $\mathbb{X}_4 \subset \mathbb{R}^4$ and $\mathbb{X}_8 \subset \mathbb{R}^8$. For each $n = 2, 4, 8$ the set \mathbb{X}_n is made up of 100 points: 40 points are obtained by perturbing points lying on $f_n = 0$ by less than 0.001; 60 points are randomly chosen. Table 3 contains the results of the classification made by the algorithms ACA-2 and CA-2 and the relative computational time for different choices of the tolerance vector which, for simplicity, is a n -dimensional vector with all entries equal to $\varepsilon = 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$.

Table 3: Results of classifications and computational time for polynomials f_2 , f_4 and f_8 employing the ACA-2 and CA-2 algorithms and tolerance values ε .

n			$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-1}$
2	Index	P_C	11%	21%	40%	40%
		P_{NC}	89%	74%	60%	60%
		P_U	0%	5%	0%	0%
	Time [s]	ACA-2	5	6	6	6
CA-2		35	40	47	43	
4	Index	P_C	5%	27%	40%	40%
		P_{NC}	95%	71%	60%	60%
		P_U	0%	2%	0%	0%
	Time [s]	ACA-2	10	12	12	12
CA-2		370	396	388	362	
8	Index	P_C	3%	29%	40%	40%
		P_{NC}	96%	67%	60%	60%
		P_U	1%	4%	0%	0%
	Time [s]	ACA-2	19	19	19	19
CA-2		638	988	1051	1040	

If $\varepsilon \geq 10^{-2}$, in all the cases both ACA-2 and CA-2 make an exact classification of the points. Further, as expected, smaller values for ε lead to identify more points far from the

variety $f_n = 0$. The computational time does not depend on the choice of ε : this is evident for both the algorithms, especially for CA-2. On the other hand, the computational speed is strongly related to the dimension of the problem: in the considered examples the computational time of ACA-2 grows linearly with the dimension, whilst the time increase is more unpredictable in the case of CA-2. Just to give an idea, consider an online classification of points given sequentially with a sampling rate of 0.5 Hz: while the CA-2 can be adopted only for $n = 2$, the ACA-2 is applicable up to $n = 8$. Moreover, with the ACA-2 algorithm we can handle 20 point per second when $n=2$, and 5 points per second if $n = 8$. Again, these results confirm the large applicability of the ACA-2 algorithm, while for the CA-2 it heavily depends on the sampling rate.

Acknowledgments

We wish to thank Eva Riccomagno for encouraging us to undertake this collaboration and the colleagues at the CNR-ISSIA for making Charlie USV data available. We also thank the referees for their valuable comments and suggestions.

References

- [1] John Abbott and Anna M. Bigatti. CoCoALib: a c++ library for doing Computations in Commutative Algebra. Available at <http://cocoa.dima.unige.it/cocoalib>.
- [2] John Abbott, Claudia Fassino, and Maria-Laura Torrente. Stable border bases for ideals of points. *Journal of Symbolic Computation*, 43:883–894, 2008.
- [3] Massimo Caccia, Eleonora Saggini, Marco Bibuli, Gabriele Bruzzone, Enrica Zereik, and Eva Riccomagno. Towards good experimental methodologies for unmanned marine vehicles. In *Computer Aided Systems Theory-EUROCAST 2013*, pages 365–372. Springer, 2013.
- [4] Andrea Caiti. Underwater robot networking: the interplay between communication and cooperation. In *WMR2013 Workshop on Marine Robotics 2013: Looking into the Crystal Ball: 20 years hence in Marine Robotics - Canarie Island*, 2013.
- [5] Angel P. del Pobil. Why do we need benchmarks in robotics research? In *Lecture Notes for IROS 2006 Workshop on Benchmarks in Robotics Research*, Beijing, China, October 2006.
- [6] Claudia Fassino. Almost vanishing polynomials for sets of limited precision points. *Journal of symbolic computation*, 45:19–37, 2010.
- [7] Claudia Fassino and Maria-Laura Torrente. Simple varieties for limited precision points. *Theoretical Computer Science*, 479:174–186, 2013.
- [8] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Second edition edition, 1989.

- [9] William S. Hall and Martin L. Newell. The mean value theorem for vector valued functions: a simple proof. *Mathematics Magazine*, 52:157–158, 1979.
- [10] Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, and Hennie Poullisse. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation*, 44:1566–1591, 2009.
- [11] GNU Scientific Library. Gsl (release 1.15). Available at <http://www.gnu.org/software/gsl/>.
- [12] Giovanni Pistone, Eva Riccomagno, and Henry P. Wynn. *Algebraic Statistics*. Chapman and Hall/CRC, Boca Raton, 2000.
- [13] Lorenzo Robbiano and Martin Kreuzer. *Computational Commutative Algebra 1*. 2000.
- [14] Eleonora Saggini and Maria-Laura Torrente. A new crossing criterion to assess path-following performance for unmanned marine vehicles. Available at <http://www.dima.unige.it/%7Eerogantin/AS2015/AAAS2015Proc.pdf>.
- [15] Eleonora Saggini, Maria-Laura Torrente, Eva Riccomagno, Marco Bibuli, Gabriele Bruzzone, Massimo Caccia, and Enrica Zereik. Assessing path-following performance for unmanned marine vehicles with algorithms from numerical commutative algebra. In *Proceedings of the 22nd Mediterranean Conference on Control and Automation (MED'14)*, pages 752–757. IEEE, 2014.
- [16] Eleonora Saggini, Enrica Zereik, Marco Bibuli, Andrea Ranieri, Gabriele Bruzzone, Massimo Caccia, and Eva Riccomagno. Evaluation and comparison of navigation guidance and control systems for 2d surface path-following. Available at <http://www.sciencedirect.com/science/article/pii/S1367578815000486>, 2015.
- [17] Tomas Sauer. Approximate varieties, approximate ideals and dimension reduction. *Numerical Algorithms*, 45(1-4):295–313, 2007.
- [18] Andrea Sorbara, Andrea Ranieri, Elenora Saggini, Enrica Zereik, Marco Bibuli, Gabriele Bruzzone, Eva Riccomagno, and Massimo Caccia. Testing the waters: design of replicable experiments for performance assessment of marine robotic platforms. *IEEE Robotics and Automation Magazine*, 22(3):62–71, 2015.
- [19] Maria-Laura Torrente and Mauro C. Beltrametti. Almost vanishing polynomials and an application to the Hough transform. *Journal of Algebra and its Applications*, 13, 2014.
- [20] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. 1997.