The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

3D Path Following with no Bounds on the Path Curvature through Surface Intersection

Antonio Sgorbissa, Renato Zaccaria

***Abstract*— The article proposes a new feedback control model which is suited for path following in a 3 Dimensional Cartesian space. Differently from other methods in literature, the method proposed neither requires to compute a projection of the robot's position on the path, nor it needs considering a moving virtual target. In spite of this: i) it guarantees asymptotic stability for every 3D curve which can be represented through a couple of intersecting surfaces $f_1(X,Y,Z)=0$, $f_2(X,Y,Z)=0$; ii) it does not put any bounds on the initial position of the vehicle depending on the path's curvature.**

I. Introduction

The article proposes a new feedback control model for path following in the 3 Dimensional Cartesian space. The model is the extension of a model for 2D path following which has already been presented in [1]. In this work a solution is proposed for underactuated vehicles moving in the 3D space (e.g., Autonomous Underwater Vehicles and Autonomous Aerial Vehicles). It is important to anticipate that only kinematics aspects are considered in the paper. Therefore, even if AUVs are considered in the following as a possible application scenario, a complete discussion about dynamic aspects, including hydrodynamic and hydrostatic forces as well as environmental disturbances such as ocean currents, would be required for a real world implementation. For an introduction to the problem of path following for AUVs and AAVs see [3] or the more recent [4].

Path following has been deeply investigated in the literature. The assumption in early works dealing with wheeled vehicles [5][6] is that the vehicle's forward speed conforms to a prescribed speed profile, while the controller acts on the vehicle's orientation to steer it to the path. To achieve this, the orthogonal projection of the robot on the path is computed, and a distance and an angular error are consequently defined. Next, an asymptotically stable control law is proposed which minimizes both errors by controlling the vehicle's orientation. It is known that these methods require to put a bound on the initial vehicle configuration: in fact, depending on the path's curvature, this point cannot be too far from the curve in order to guarantee uniqueness of the projection. The approaches above have been extended to the 3D case for underwater vehicles [7], by inheriting the major shortcoming already present in the control strategy for wheeled robots.

A solution to this problem has been initially proposed in [8]. Path following is achieved by controlling explicitly the progression rate of a "virtual target" to be tracked along the path, thus bypassing the problems that arise when a projection of the actual vehicle onto that path must be computed. Since the publication of [8], the approach has been very popular, and variants with significant improvements have been proposed, both for wheeled vehicles [11], marine crafts [9] and AUVs [10][12][13][14]. The integration with path planning has been proposed in [15]. Approaches based on the virtual target have the minor drawback that they require additional computations to determine the target motion rate.

The major contribution of this work is to propose a novel approach to path following which works both in 2D and 3D, and solves the problem of the bounded path curvature [5][6]. The method neither requires to compute a projection of the robot's position on the path, nor it needs considering a moving virtual target. In spite of this: (i) it guarantees asymptotic stability for every 2D curve which can be represented through its implicit equation in the form $f(X,Y)=0$, and for every 3D curve which can be represented through a couple of intersecting surfaces $f_1(X,Y,Z)=0$, $f_2(X,Y,Z)=0$; (ii) it does not put any bounds on the initial position of the vehicle depending on the path curvature.

Section II summarizes the basic concepts of path following in 2D, outlining differences with the original approach presented in [1]; specifically, this work refers to the improved version of the same model proposed in [2]. Section III describes 3D path following. Simulated results, conclusions and future works are described in Sections IV and V.

II. Path Following in 2D

Assume a unicycle robot moving on the XY-plane of a fixed Cartesian frame. A state vector $\eta=[X,Y,\psi]^T$ is introduced, where X,Y, and ψ correspond to the vehicle's position and orientation. The state equations that describe the unicycle kinematics are:

$$\begin{aligned}\dot{X}&=u\cos\psi\\ \dot{Y}&=u\sin\psi\\ \dot{\psi}&=r.\end{aligned}\tag{1}$$

Inputs u and r correspond, respectively, to the translational and the rotational speed. The control law slightly varies depending on the shape of the path to follow, either it is a straight line or a generic curve expressed through its implicit equation in the form $f(X,Y)=0$.

A. Straight line

Let D be the signed distance to the line computed in the current robot position (X,Y), and let ψ_e be the difference between the robot orientation and the orientation of the

A. Sgorbissa, R. Zaccaria are with DIST, University of Genova, Via Opera Pia 13, 16145, Genova, Italy. Email:{sgorbiss,renato}@dist.unige.it.

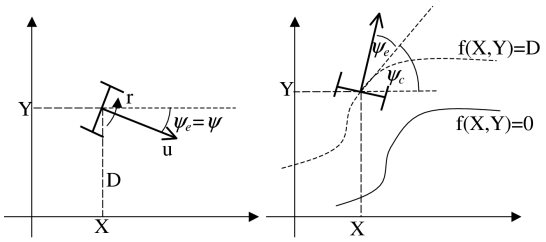

Fig. 1. Left: following the straight path $f(X,Y) = Y = 0$. Right: following a generic curve $f(X,Y) = 0$

line. Moreover, assume for simplicity that the path to follow corresponds to the X-axis of the reference system (Figure 1 on the left), and hence $D = Y$ and $\psi_e = \psi$. To guarantee asymptotic convergence to the path, we let:

$$\begin{aligned} &u = u(t), \quad \lim_{t\to\infty} u(t) > 0 \\ &r = K(-uS(D) - \dot{D}), \quad K > 0, \end{aligned} \tag{2}$$

where $S(D)$ is a C^n sigmoid function:

$$S(D) = \frac{D}{\sqrt{1+D^2}}. \tag{3}$$

In order to compute equilibrium points, consider the state equations for (D, ψ_e) from (1) and (2), by initially assuming that $u = \bar{u} > 0$ is a constant positive velocity:

$$\begin{aligned} \dot{D} &= \bar{u} \sin \psi_e \\ \dot{\psi}_e &= K(-\bar{u}S(D) - \dot{D}). \end{aligned} \tag{4}$$

Equation (4) describes a system which is regular and time–invariant: the trajectories of the system depend only on initial conditions. Equilibrium points are given by the solutions of:

$$\begin{aligned} 0 &= \bar{u} \sin \psi_e \\ 0 &= K(-\bar{u}S(D) - 0). \end{aligned} \tag{5}$$

Equilibrium points correspond to the set $\{(D = 0, \psi_e = k\pi) | k \in Z\}$, i.e., when the distance from the line is null and the robot is oriented along the line. In particular, points in the set $\{(D = 0, \psi_e = 2k\pi) | k \in Z\}$ are stable equilibrium points, whereas $\{(D = 0, \psi_e = (2k+1)\pi) | k \in Z\}$ are not (the robot is moving along the line in the wrong direction).

In [2], the asymptotical stability of points in the set $\{(D = 0, \psi_e = 2k\pi) | k \in Z\}$ is demonstrated. Moreover, it is shown that the system in (4) globally converges to one of the points in the set $\{(D = 0, \psi_e = 2k\pi) | k \in Z\}$, and that the difference between the vehicle heading in the initial position and the heading in equilibrium is less than 2π (in absolute value). These properties are confirmed by observing trajectories in the plane of phases in Figure 2[1].

Remark 1. When analyzing the system (4) in the plane of phases, it is required to compute, for every (D, ψ_e), a corresponding vector $(\dot{D}, \dot{\psi}_e)$ which is tangent to the trajectory in that point. Since both $\dot{D}$ and $\dot{\psi}_e$ are proportional to the velocity u in (4), this means that the norm of $(\dot{D}, \dot{\psi}_e)$ depends on u, but its direction does not. This in turn implies

[1]The analysis has been performed with the *pplane* Matlab toolbox, `http://math.rice.edu/~dfield`.

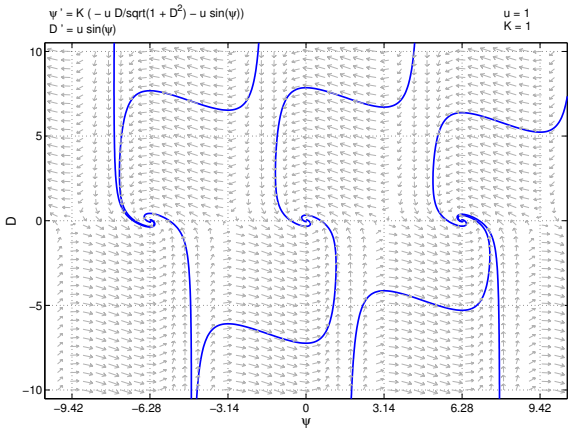

Fig. 2. Trajectories in the plane of phases.

that trajectories in the plane of phases have the same shape for every (not necessarily constant) velocity profile $u = u(t)$. In particular, if $u > 0$ the vehicle moves towards equilibrium points at speed u; if $u < 0$ the vehicle moves backward away from equilibrium points; if $u = 0$ the vehicle stops. If $\lim_{t\to\infty} u(t) > 0$ the system asymptotically converges to a point in the set $\{(D = 0, \psi_e = 2k\pi) | k \in Z\}$. □

The instability of equilibrium points in the set $\{(D = 0, \psi_e = (2k+1)\pi) | k \in Z\}$ can be easily proved, but – as for stability – it is not shown for sake of brevity.

Remark 2. To choose the heading of the vehicle along the straight line, it is sufficient to invert the sign of the distance D. This has the effect of switching stable equilibrium points to unstable, and viceversa. □

B. Generic curve

In the case of a generic curve in the plane, expressed through its implicit equation $f(X,Y) = 0$ (Figure 1 on the right), it is not trivial to express the distance from the curve in closed form. Even if it does not represent the Euclidean distance from the curve, a possibility is to use the function $D = f(X,Y)$ as a distance function. Notice that $Z = f(X,Y)$ defines a 3D surface in the Cartesian XYZ-space which, when intersected with the plane $Z = 0$, produces the desired path $f(X,Y) = 0$. When defined in this way, D has some good properties which make it appropriate for our purpose: (i) D is a scalar field; (ii) $D = 0$ when (X,Y) lies on the curve (by definition); (iii) given that the gradient of D is not null on the curve, D is positive / negative depending on which side of the plane (X,Y) is located with respect to the curve.

Let f_X, f_Y stand for the partial derivatives of f. The function f must be twice differentiable and $\|\nabla f\| = \sqrt{f_X^2 + f_Y^2} > 0$ in the chosen domain, that is the plane deprived of a neighbourhood of points where $\|\nabla f\| = 0$. The first and second order partial derivatives of D are:

$$\begin{aligned} D_X = \frac{\partial f(X,Y)}{\partial x}, \quad D_Y = \frac{\partial f(X,Y)}{\partial Y}, \\ D_{XX} = \frac{\partial^2 f(X,Y)}{\partial X^2}, D_{YY} = \frac{\partial^2 f(X,Y)}{\partial Y^2}, \\ D_{XY} = D_{YX} = \frac{\partial^2 f(X,Y)}{\partial X \partial Y}. \end{aligned} \tag{6}$$

To guarantee asymptotic convergence to the path, we let:

$$\begin{aligned} &u = u(t), \quad \lim_{t\to\infty} u(t) > 0, \\ &r = K(-\|\nabla D\| u S(D) - \dot{D}) + \dot{\psi}_c, \end{aligned} \tag{7}$$

where $\|\nabla D\| = \|\nabla f\|$, $K > 0$, and $S(D)$ is the sigmoid function in (3). The term ψ_c is defined as the angle between the X-axis and the vector $(D_Y, -D_X)$ normal to $\|\nabla D\|$ in (X, Y), i.e., tangent to the level curve (Figure 1 on the right). The angle ψ_c can be computed as $\psi_c = \tan^{-1}(-D_X/D_Y)$ when $D_Y \neq 0$, or $\psi_c = \cot^{-1}(-D_Y/D_X)$ when $D_X \neq 0$. In both cases:

$$\dot{\psi}_c = \frac{(D_X D_{YX} - D_Y D_{XX}) u \cos\psi + (D_X D_{YY} - D_Y D_{XY}) u \sin\psi}{\|\nabla D\|^2}. \tag{8}$$

By writing kinematics equations for (D, ψ) it follows that:

$$\begin{aligned} &\dot{D} = D_X u \cos\psi + D_Y u \sin\psi \\ &\dot{\psi} = K(-\|\nabla D\| u S(D) - \dot{D}) + \dot{\psi}_c. \end{aligned} \tag{9}$$

Finally, (9) is re–written for (D, ψ_e), where $\psi_e = \psi - \psi_c$ measures the error between the vehicle heading and the tangent to the level curve in (X, Y). After some computations (9) becomes (see [1][2] for details):

$$\begin{aligned} &\dot{D} = \|\nabla D\| u \sin\psi_e \\ &\dot{\psi}_e = K(-\|\nabla D\| u S(D) - \dot{D}). \end{aligned} \tag{10}$$

The system in (10) has the same expression as (4) by assuming a control input $u' = \|\nabla D\| u$ and therefore it has the same stable / unstable equilibrium points (see *Remark 1*): D and ψ_e tend asymptotically to a point in the set $\{(D = 0, \psi_e = 2k\pi) | k \in Z\}$. Obviously, the meaning of D is different from the straight line case, since now D represents the value of f in the current position, instead of the Euclidean distance to the curve. However, this is sufficient to guarantee that the system reaches an equilibrium state in which the robot position lies on the path, i.e., $f(X, Y) = 0$, and the robot heading is tangent to the path itself.

Differently from other approaches in the literature [5][6] there are no conditions on the path curvature. One could argue that, for some curves, it can be necessary to restrict the workspace to a subset of $\Re^2$ to meet the constraint $\|\nabla D\| > 0$ in (8) and (10). However, this constraint is not related to the path curvature: consider, for example, the sinusoidal profile:

$$f(X, Y) = Y - a \sin(bX) = 0, \tag{11}$$

for which $\|\nabla D\| > 0$ is always guaranteed, and whose maximum curvature in correspondence of $\{(Y = \pm a, X = \frac{1}{b}(2k\pi \pm \frac{\pi}{2})) | k \in Z\}$ can be arbitrarily increased by increasing b.

Experiments with a wheeled robot are described in [2].

III. Path Following in 3D

Kinematics equations must be properly reformulated for 3D path tracking. A notation similar to [4] is adopted, where $\eta = [X, Y, Z, \phi, \theta, \psi]^T$ is the vector describing North-East-Down positions in Earth-Fixed coordinates (*n*–frame) as well as Euler angles, and $v = [u, v, w, p, q, r]^T$ are the six DOF generalised velocities in body–fixed coordinates (*b*–frame).

It is assumed that it is possible to control only the linear velocity u along the x-axis of the *b*–frame, as well as its rotational velocity about the y- and the z-axis of the *b*–frame (q and r, referred to as pitch and yaw). This choice is very common in AUV design for energetic reasons. The kinematics equations can be written in component form as follows:

$$\begin{aligned} &\dot{X} = u \cos\psi \cos\theta \\ &\dot{Y} = u \sin\psi \cos\theta \\ &\dot{Z} = -u \sin\theta \\ &\dot{\phi} = q \sin\phi \tan\theta + r \cos\phi \tan\theta \\ &\dot{\theta} = q \cos\phi - r \sin\phi \\ &\dot{\psi} = q \tfrac{\sin\phi}{\cos\theta} + r \tfrac{\cos\phi}{\cos\theta} \\ &\qquad\qquad\qquad\qquad \theta \neq \pm\tfrac{\pi}{2}. \end{aligned} \tag{12}$$

In the following, it is assumed also that the AUV is stabilized in roll by a separate mechanism not described here, which guarantees that $\dot{\phi} = \phi = 0$. In this case, kinematics equations can be simplified as follows:

$$\begin{aligned} &\dot{X} = u \cos\psi \cos\theta \\ &\dot{Y} = u \sin\psi \cos\theta \\ &\dot{Z} = -u \sin\theta \\ &\dot{\phi} = 0 \\ &\dot{\theta} = q \\ &\dot{\psi} = r \tfrac{1}{\cos\theta} \\ &\qquad\qquad\qquad\qquad \theta \neq \pm\tfrac{\pi}{2}. \end{aligned} \tag{13}$$

A. Straight line in 3D

Assume now that the path–planner has produced, as the result of a planning phase, a path to be followed, expressed as the intersection of two properly chosen surfaces. As in the 2D case, it is first considered the situation in which the path is a straight line, i.e., the surfaces are planes in the 3D space, each plane being described through its implicit equation in the form $f_i(X, Y, Z) = 0$, for $(i = 1, 2)$. We anticipate however that, by properly defining a distance function D_i for $(i = 1, 2)$, the following discussion is still valid when $f_i(X, Y, Z) = 0$ are generic surfaces, and the resulting path is a generic curve in 3D. The straight line case is described first since it is simpler to be visualized. In the next subsection, the discussion will be straightfowardly extended to a generic curve.

In case of two non–parallel planes, the path is a straight line given by the ∞^1 solutions of the following system:

$$\begin{aligned} &f_1(X, Y, Z) = a_1 X + b_1 Y + c_1 Z + d_1 = 0 \\ &f_2(X, Y, Z) = a_2 X + b_2 Y + c_2 Z + d_2 = 0. \end{aligned} \tag{14}$$

Let f_{iX}, f_{iY}, f_{iZ} stand for the partial derivatives of f. Two distance functions $D_1 = D_1(X, Y, Z)$ and $D_2 = D_2(X, Y, Z)$ are introduced, describing – respectively – the distance from the first and the second plane. The distance is

taken as a signed value, which is positive on one side of the plane and negative on the other. That is:

$$D_i = \frac{a_i X + b_i Y + c_i Z + d_i = 0}{\sqrt{a_i^2 + b_i^2 + c_i^2}}, \quad (i = 1, 2), \tag{15}$$

whose partial derivatives are

$$\begin{aligned} D_{iX} &= \frac{\partial D_i(X,Y,Z)}{\partial X} = \frac{a_i}{\sqrt{a_i^2+b_i^2+c_i^2}} = \frac{f_{iX}}{\|\nabla f_i\|} \\ D_{iY} &= \frac{\partial D_i(X,Y,Z)}{\partial Y} = \frac{b_i}{\sqrt{a_i^2+b_i^2+c_i^2}} = \frac{f_{iY}}{\|\nabla f_i\|} \\ D_{iZ} &= \frac{\partial D_i(X,Y,Z)}{\partial Z} = \frac{c_i}{\sqrt{a_i^2+b_i^2+c_i^2}} = \frac{f_{iZ}}{\|\nabla f_i\|}. \end{aligned} \tag{16}$$

where $\|\nabla f_i\| = \sqrt{f_{iX}^2 + f_{iY}^2 + f_{iZ}^2}$ is the norm of the gradient of $f_i(X, Y, Z)$.

To simplify the discussion, we define:

$$\begin{aligned} \hat{D}_{1\hat{X}} &= D_{1X}\cos\psi + D_{1Y}\sin\psi \\ \hat{D}_{1\hat{Y}} &= -D_{1Z} \end{aligned} \tag{17}$$

and

$$\begin{aligned} \hat{D}_{2\hat{X}} &= \frac{D_{2X}D_{1Z} - D_{2Z}D_{1X}}{D_{1Z}} \\ \hat{D}_{2\hat{Y}} &= \frac{D_{2Y}D_{1Z} - D_{2Z}D_{1Y}}{D_{1Z}}. \end{aligned} \tag{18}$$

It is possible to show that the system described by kinematics equations in (13) can be forced to converge to the straight line defined by the system in (14) by setting control inputs $[u, q, r]^T$ as follows:

$$\begin{aligned} &u = u(t), \quad \lim_{t\to\infty} u(t) \neq 0 \\ &q = K_1(-\|\nabla\hat{D}_1\| u S(D_1) - \dot{D}_1) + \dot{\theta}_c \\ &r = K_2(-\|\nabla\hat{D}_2\| u\cos^2\theta S(D_2) - \cos\theta\dot{D}_2) + \dot{\psi}_c \\ &\qquad\qquad D_{1Z} \neq 0, \|\nabla D_2\| \neq 0. \end{aligned} \tag{19}$$

where, for $(i = 1, 2)$, $\|\nabla\hat{D}_i\| = \sqrt{\hat{D}_{i\hat{X}}^2 + \hat{D}_{i\hat{Y}}^2}$, $K_i > 0$, and $S(D_i)$ is the sigmoid function in (3).

The angle θ_c is computed as $\theta_c = \tan^{-1}(-\hat{D}_{1\hat{X}}/\hat{D}_{1\hat{Y}})$ when $\hat{D}_{1\hat{Y}} \neq 0$, or $\theta_c = \cot^{-1}(-\hat{D}_{1\hat{Y}}/\hat{D}_{1\hat{X}})$ when $\hat{D}_{1\hat{X}} \neq 0$; the former condition is guaranteed by the constraint $D_{1Z} \neq 0$, see (17) and (19).

The angle ψ_c is computed as $\psi_c = \tan^{-1}(-\hat{D}_{2\hat{X}}/\hat{D}_{2\hat{Y}})$ when $\hat{D}_{2\hat{Y}} \neq 0$, or $\psi_c = \cot^{-1}(-\hat{D}_{2\hat{Y}}/\hat{D}_{2\hat{X}})$ when $\hat{D}_{2\hat{X}} \neq 0$; one of the two conditions is guaranteed by the constraints $D_{1Z} \neq 0$ and $\|\nabla D_2\| \neq 0$, see (18) and (19).

It can be noticed that the constraints $D_{1Z} \neq 0$ and $\|\nabla D_2\| \neq 0$ guarantee that $f_1(X, Y, Z) = 0$ is not perpendicular to the XY-plane and $f_2(X, Y, Z) = 0$ is not parallel to the XY-plane.

Remark 4. The equations above are similar to (2). The pitch q is proportional to the difference between a reference value $-\|\nabla\hat{D}_1\| u S(D_1)$ and the approaching velocity $\dot{D}_1$ to the plane $f_1(X, Y, Z) = 0$ (i.e., the derivative of the distance D_1 with respect to time). Similarly, the yaw r is proportional to the difference between $-\|\nabla\hat{D}_2\| u\cos\theta(D_2)$ and $\dot{D}_2$. The term $\dot{\psi}_c$ in line 3 is reported since it is necessary in case of generic surfaces, but it can be easily verified from (18) that it is always null in case of two intersecting planes. □

As anticipated the basic idea is to decouple 3D path tracking into two disjoint problems. In particular, by controlling the pitch in order to converge to the surface $f_1(X, Y, Z) = 0$ and, simultaneously, controlling the yaw in order to converge to the surface $f_2(X, Y, Z) = 0$, the system is expected to converge to the desired path, i.e., expressed as the intersection of the two surfaces.

To show that this is possible, kinematics equations are re–written for a different state vector $[D_1, D_2, \theta, \psi]^T$, whereas ϕ is ignored (as allowed by *Remark 3*).

$$\begin{aligned} &\dot{D}_1 = D_{1X}u\cos\psi\cos\theta + D_{1Y}u\sin\psi\cos\theta - D_{1Z}u\sin\theta \\ &\dot{D}_2 = D_{2X}u\cos\psi\cos\theta + D_{2Y}u\sin\psi\cos\theta - D_{2Z}u\sin\theta \\ &\dot{\theta} = K_1(-\|\nabla\hat{D}_1\| u S(D_1) - \dot{D}_1) + \dot{\theta}_c \\ &\dot{\psi} = K_2(-\|\nabla\hat{D}_2\| u\cos\theta S(D_2) - \dot{D}_2) + \dot{\psi}_c \\ &\qquad\qquad \theta \neq \pm\tfrac{\pi}{2}, D_{1Z} \neq 0, \|\nabla D_2\| \neq 0. \end{aligned} \tag{20}$$

Initially consider only line 1 and 3 in the system above, and re–write them by using (17):

$$\begin{aligned} \dot{D}_1 &= \hat{D}_{1\hat{X}}u\cos\theta + \hat{D}_{1\hat{Y}}u\sin\theta \\ \dot{\theta} &= K_1(-\|\nabla\hat{D}_1\| u S(D_1) - \dot{D}_1) + \dot{\theta}_c. \end{aligned} \tag{21}$$

Assume that ψ is given: (21) corresponds to the situation that the AUV, with a fixed yaw, tends to converge to the surface defined by $f_1(X, Y, Z) = 0$ by controlling only the pitch. For D_1 to be in equilibrium, it must necessarily hold:

$$0 = \hat{D}_{1\hat{X}}u\cos\theta + \hat{D}_{1\hat{Y}}u\sin\theta. \tag{22}$$

In particular, as long as $D_{1Z} \neq 0$, this yields:

$$\sin\theta = \frac{-\hat{D}_{1\hat{X}}}{\hat{D}_{1\hat{Y}}}\cos\theta = \frac{D_{1X}\cos\psi + D_{1Y}\sin\psi}{D_{1Z}}, \tag{23}$$

and finally:

$$\theta = \tan^{-1}\left(\frac{-\hat{D}_{1\hat{X}}}{\hat{D}_{1\hat{Y}}}\right) + k\pi. \tag{24}$$

It is necessary to verify if, when ψ is given, the vehicle moves on a plane $(\hat{X}, \hat{Y})$ perpendicular to the XY-plane (Figure 3) until it lies on the surface $f_1(X, Y, Z) = 0$, i.e., until it reaches an equilibrium which depends on ψ (through the term $\hat{D}_{1\hat{X}}$) as stated in (24). Let us define $\hat{X}$ and $\hat{Y}$ the main axes of such plane, where $\hat{X}$ lies along the axis of nodes and $\hat{Y}$ lies along $-Z$ in the pitch–yaw–roll representation. It holds the following relationships between the vehicle's

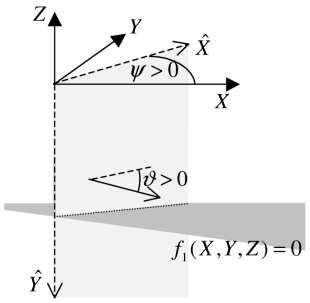

Fig. 3. A vehicle moving on the plane $(\hat{X}, \hat{Y})$ with fixed ψ.

configuration in (X, Y, Z) and the corresponding position and orientation $(\hat{X}, \hat{Y}, \hat{\theta})$ in the $(\hat{X}, \hat{Y})$ plane:

$$\begin{aligned} X &= \hat{X}\cos\psi \\ Y &= \hat{X}\sin\psi \\ Z &= -\hat{Y} \\ \theta &= \hat{\theta}. \end{aligned} \tag{25}$$

From (25) it can be derived that:

$$\begin{aligned} \frac{\partial D_1}{\partial \hat{X}} &= \frac{\partial D_1}{\partial X}\frac{\partial X}{\partial \hat{X}} + \frac{\partial D_1}{\partial Y}\frac{\partial Y}{\partial \hat{X}} = \hat{D}_{1\hat{X}} \\ \frac{\partial D_1}{\partial \hat{Y}} &= \frac{\partial D_1}{\partial Z}\frac{\partial Z}{\partial \hat{Y}} = \hat{D}_{1\hat{Y}}, \end{aligned} \tag{26}$$

which means that $\hat{D}_{1\hat{X}}$ and $\hat{D}_{1\hat{Y}}$ in (21) are the partial derivatives of D_1 with respect to $\hat{X}$ and $\hat{Y}$. That is, (21) states that the vehicle moves on the plane $(\hat{X}, \hat{Y})$ like a unicyle with speed u. By re–writing kinematics equations for (D_1, θ_e), where $\theta_e = \theta - \theta_c$, this yields after some computations (similarly to the 2D case shown in [2]):

$$\begin{aligned} \dot{D}_1 &= \|\nabla \hat{D}_1\| u \sin\theta_e \\ \dot{\theta}_e &= K_1(-\|\nabla \hat{D}_1\| u S(D_1) - \dot{D}_1). \end{aligned} \tag{27}$$

The equation above has the same form as (10) by assuming a control input $u' = \|\nabla \hat{D}_1\| u$, which means that the vehicle converges asymptotically to equilibrium points $\{(D_1 = 0, \theta_e = 2k\pi | k \in Z)\}$, while moving on the $(\hat{X}, \hat{Y})$ plane, exactly as it has been shown for the 2D case. The set $\{(D_1 = 0, \theta_e = 2(k+1)\pi | k \in Z)\}$ describes unstable equilibrium points. Convergence is guaranteed as long as $\lim_{t\to\infty} \|\nabla \hat{D}_1(X(t), Y(t), Z(t), \psi(t))\| \neq 0$: a sufficient condition for this to happen is $D_{1Z} \neq 0$.

Remark 4. The equation above implies that, for every ψ, the vehicle tends to lie on the surface $f_1(X, Y, Z) = 0$. The term $\dot{\theta}_c$ in (21) guarantees convergence however ψ varies in time. □

Consider now lines 2 and 4 of kinematics equations in (20). Assume that θ, after a transient behaviour, guarantees that the vehicle lies on the surface $f_1(X, Y, Z) = 0$. By substituting (23) into lines 2 and 4 of (20) and by collecting terms, the system becomes after some computations:

$$\begin{aligned} \dot{D}_2 &= \hat{D}_{2\hat{X}} u \cos\theta\cos\psi + \hat{D}_{2\hat{Y}} u \cos\theta \sin\psi \\ \dot{\psi} &= K_2(-\|\nabla \hat{D}_2\| u \cos\theta S(D_2) - \dot{D}_2) + \dot{\psi}_c. \end{aligned} \tag{28}$$

which corresponds to the situation that the AUV, with a pitch which lies on the surface $f_1(X, Y, Z) = 0$, tends to converge to the surface defined by $f_2(X, Y, Z) = 0$ by controlling only the yaw (i.e., symmetrical to the previous case). For D_2 to be in equilibrium, it must necessarily hold:

$$0 = \hat{D}_{2\hat{X}} u \cos\theta\cos\psi + \hat{D}_{2\hat{Y}} u \cos\theta \sin\psi, \tag{29}$$

which can be solved for ψ since we assumed $\theta \neq \pm\pi/2$, $D_{1Z} \neq 0$, and $\|\nabla D_2\| \neq 0$. For example, when $D_{2Y} \neq 0$, it can be computed:

$$\psi = \tan^{-1}\left(\frac{-D_{2\hat{X}}}{D_{2\hat{Y}}}\right) + k\pi. \tag{30}$$

An analogous expression can be computed when $D_{2X} \neq 0$.

By considering a plane $(\hat{X}, \hat{Y})$ which corresponds to the (X, Y) plane, the following relationship holds between (X, Y, Z) and the projection of the vehicle's position and orientation $(\hat{X}, \hat{Y}, \hat{\psi})$ on $(\hat{X}, \hat{Y})$, given that the vehicle lies on the surface $f_1(X, Y, Z) = 0$:

$$\begin{aligned} X &= \hat{X} \\ Y &= \hat{Y} \\ \frac{\partial Z}{\partial \hat{X}} &= \frac{\partial Z}{\partial D_1}\frac{\partial D_1}{\partial X} = \frac{D_{1X}}{D_{1Z}} \\ \frac{\partial Z}{\partial \hat{Y}} &= \frac{\partial Z}{\partial D_1}\frac{\partial D_1}{\partial Y} = \frac{D_{1Y}}{D_{1Z}} \\ \psi &= \hat{\psi}. \end{aligned} \tag{31}$$

From (31) it can be derived:

$$\begin{aligned} \frac{\partial D_2}{\partial \hat{X}} &= \frac{\partial D_2}{\partial X}\frac{\partial X}{\partial \hat{X}} + \frac{\partial D_2}{\partial Z}\frac{\partial Z}{\partial \hat{X}} = \hat{D}_{2\hat{X}} \\ \frac{\partial D_2}{\partial \hat{Y}} &= \frac{\partial D_2}{\partial Y}\frac{\partial Y}{\partial \hat{Y}} + \frac{\partial D_2}{\partial Z}\frac{\partial Z}{\partial \hat{Y}} = \hat{D}_{2\hat{Y}}, \end{aligned} \tag{32}$$

which means that $\hat{D}_{2\hat{X}}$ and $\hat{D}_{2\hat{Y}}$ are the partial derivatives of D_2 with respect to $\hat{X}$ and $\hat{Y}$ respectively. That is, (28) says that the projection of the vehicle's on the plane $(\hat{X}, \hat{Y})$ moves like a unicyle with velocity $u\cos\theta$.

By re–writing kinematics equations for (D_2, ψ_e), where $\psi_e = \psi - \psi_c$, this finally yields:

$$\begin{aligned} \dot{D}_2 &= \|\nabla \hat{D}_2\| u \cos\theta \sin\psi_e \\ \dot{\psi}_e &= K_2(-\|\nabla \hat{D}_2\| u \cos\theta S(D_2) - \dot{D}_2). \end{aligned} \tag{33}$$

The equation above has the same form as (10) by assuming a control input $u' = \|\nabla \hat{D}_2\| u\cos\theta$. This means that the vehicle converges asymptotically to equilibrium points $\{(D_2 = 0, \psi_e = 2k\pi | k \in Z)\}$, while its projection moves on the $(\hat{X}, \hat{Y})$ plane, exactly as it has been shown for the 2D case. On the opposite, $\{(D_2 = 0, \psi_e = 2(k+1)\pi | k \in Z)\}$ are unstable equilibrium points. Convergence is guaranteed as long as $\lim_{t\to\infty} \|\nabla \hat{D}_2(X(t), Y(t), Z(t))\| \neq 0$: a sufficient condition for this to happen is $D_{1Z} \neq 0$ and $\|\nabla D_2\| \neq 0$.

Remark 5. By considering (26) and (33) together it can be observed that the whole system has 4 equilibrium points for $\theta_e, \psi_e \in [-\pi, \pi)$, of which only $(\theta_e = 0, \psi_e = 0)$ is stable. However, due to the non–univocity of Euler angles to represent the vehicle's configuration, the same heading of the vehicle can now be represented with 2 different configurations (θ, ψ) and $(\theta+\pi, \psi+\pi)$. To put the additional constraint $\theta \in (-\frac{\pi}{2}, +\frac{\pi}{2})$ thus avoiding singularities, it is sufficient to properly choose the signs of the distance functions D_1 and D_2. □

B. Generic curve in 3D

As it is is done in the 2D case, it is possible to extend the system to a generic curve expressed as the intersection of two generic surfaces $f_1(X, Y, Z) = 0$, $f_2(X, Y, Z) = 0$ by setting:

$$D_i = f_i(X, Y, Z), \quad (i = 1, 2), \tag{34}$$

whose partial derivatives are

$$D_{iX} = \frac{\partial f_i(X,Y,Z)}{\partial X}, D_{iY} = \frac{\partial f_i(X,Y,Z)}{\partial Y}, D_{iZ} = \frac{\partial f_i(X,Y,Z)}{\partial Z}. \tag{35}$$

The system described by (13) can be forced to converge to the curve defined by $f_1(X,Y,Z)=0$, $f_2(X,Y,Z)=0$ by setting control inputs $[u,q,r]^T$ as in (19). In fact, the whole discussion above can be extended to the case of generic surfaces, by properly using the expression of D which has just been given. The reader can easily verify that, in the previous discussion, the assumption that $f_1(X,Y,Z)=0$, $f_2(X,Y,Z)=0$ define two planes has been used only to build the distance functions D_1 and D_2, and to help to visualize a vehicle that converges to the two surfaces separately. All of the equations from (17) to (33) can be reconsidered by assuming two generic surfaces.

Remark 6. The terms $\dot{\theta}_c$ and $\dot{\psi}_c$ in (19) take into accout the curvature of the surfaces. In particular: $\dot{\theta}_c$ guarantees that the vehicle, after a transient behaviour, lies on $f_1(X,Y,Z)=0$ however ψ varies in time; $\dot{\psi}_c$ guarantees that a vehicle lying on $f_1(X,Y,Z)=0$, after a transient behaviour, lies on $f_2(X,Y,Z)=0$. Notice also that, in experiments, we let the vehicle converge to both surfaces at the same time: that is, it is not necessary that the vehicle must move along the plane $(\hat{X},\hat{Y})$ shown in Figure 3 until it reaches $f_1(X,Y,Z)=0$, and later it moves towards $f_2(X,Y,Z)=0$.

Remark 7. As in 2D, there are no conditions on the path curvature. The constraints that have been put on partial derivatives $D_{1Z}\neq 0$, and $\|\nabla D_2\|\neq 0$ provide hints to choose the surfaces to describe the path. However, it can be easily verified that these constraints do not limit the curvature of the path: as an example, consider a couple of surfaces which have a similar expression as in (13), e.g., $f_1(X,Y,Z)=Z-a_1\sin(b_1X)=0$ and $f_2(X,Y,Z)=Y-a_2\sin(b_2X)=0$, whose maximum curvature can be arbitrarily increased by increasing b_1 and b_2. □

IV. Simulated Experiments

The system has been implemented and tested in the Matlab/Simulink environment, by using the Virtual Reality Toolbox for visualization. This Section describes simulated experiments which have been performed in the simulated submarine world shown in Figure 4, by introducing errors to simulate the presence of sea currents which affect the vehicle position and orientation. Since only kinematics is simulated, experiments do not aim at emulating the behaviour of a real AUV. Instead, they have the only purpose of validating the properties of the proposed control law.

Consider for example Figure 5: the path is composed of three different path segments, generated by the corresponding surfaces. In the first segment of the path, two intersecting planes allow to specify the heading as well as the immersion depth. In the second segment, a plane defines the heading whereas a cylinder defines the immersion depth by adhering to the profile of the canyon. In the third segment, a torus defines the immersion depth, whereas a cylinder defines the steering radius. Figure 4 shows snapshots taken during one of these experiments: to model the AUV we took inspiration from the Hydroid Remus[2]. The AUV moves with a constant linear speed $\bar{u}=1m/sec$, and has bounded pitch, yaw and turning radius.

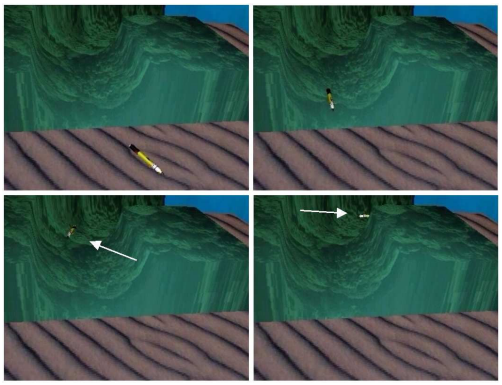

Fig. 4. Experiments in a simulated submarine world.

For all experiments the distances D_1 and D_2 are recorded and plotted: an example is shown in Figure 6. It can be seen that D_1 and D_2 converge to zero. It is also possible to notice that, when switching from path segment 1 to path segment 2, the absolute values of D_1 and D_2 temporarily increase. This is an effect of the bounded turning radius, and the fact that the path is not C^1 in the switching point: in fact, when switching to the second segment, the vehicle has a big error in orientation, which forces it to temporarily diverge from the path. However, D_1 and D_2 tend asymptotically to zero in subsequent simulation steps, thus guaranteeing convergence to the reference path.

More than 100 simulations have been performed with different start configurations, as well as different paths expressed as the intersection of planes, cylinders, and toruses. In all simulated experiments, the convergence to the path is guaranteed after a transient behavior, yielding a negligible positioning error which validates theory.

V. Conclusions and Discussion

The article has proposed a new feedback control model that allows path tracking in a 3 Dimensional Cartesian space, which relies on the general idea that a 3D path can be

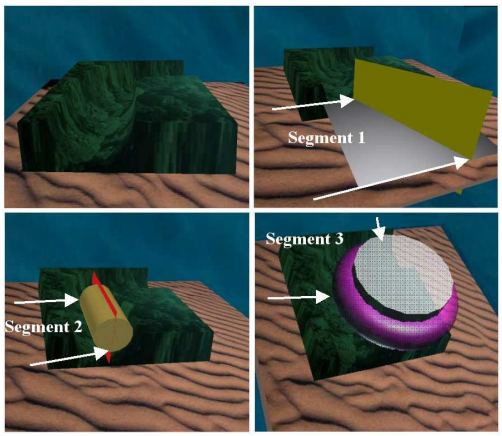

Fig. 5. Planning a path in a simulated submarine world.

[2]http://www.hydroidinc.com/

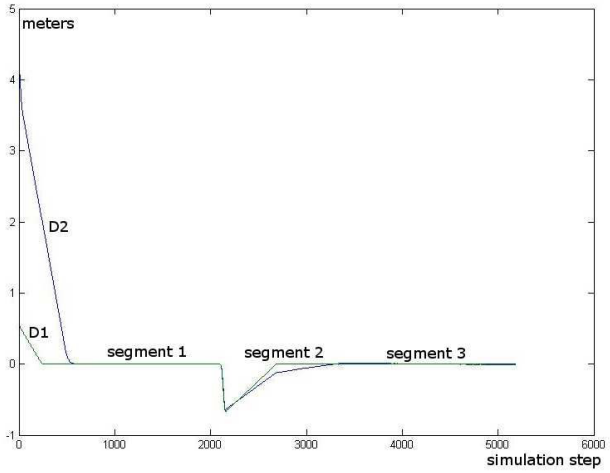

Fig. 6. Plot of D_1 and D_2.

represented as the intersection of two properly chosen surfaces. The approach offers many advantages, among which the possibility to re–use simple control laws that have been designed for 2D path following of non–holomic vehicles. In particular, the law proposed for path following [1][2] differs from all approaches in the literature, since neither it requires to compute a projection of the vehicle's position on the path [5][6], nor it simulates the motion of a virtual target [8][11]. In spite of this, it does not put any bounds on the initial position of the vehicle depending on the path curvature.

The way adopted for representing curves in 3D is not new in the autonomous vehicles literature, even if it is usually limited to intersecting planes [16]. Specifically, this appears to be particularly efficient when considering path planning (a problem that is not considered here, see [17]), which can be consequently handled in two decoupled steps: at every step, a constraint is added, choosing among a set of parameterized planning primitives [18] expressed as geometrical surfaces in the 3D Cartesian Space. For example, assume that a path is required which keeps an AUV as close as possible to the marine soil [19] while moving along a given direction, e.g., to monitor submarine cables. In the first step, a surface which guarantees that the vehicle maintains the proper distance from the marine soil can be computed, by possibly taking into account the slope and the topology of the soil, the desired immersion depth, etc. In the second step, another surface which determines the desired heading is computed: this can be a plane, or a curved surface if a steering path is required. The intersection of the two surfaces defines a path which is guaranteed not to collide with obstacles while, at the same time, it heads towards the final destination.

The proposed approach is also suited for a hybrid scheme in which some constraints are specified manually by the user through low–bandwidth acoustic communication (e.g., the heading), whereas other constraints are automatically provided by the path planner (e.g., to avoid collision with the marine soil). In addition, the control law adopted is suitable for real–time obstacle avoidance, since in most case it is possible to avoid sensed obstacles by locally modifying only one of the two surfaces (see [20] for the 2D case).

These latter aspects will be considered in future works.

REFERENCES

[1] A. Sgorbissa, R. Zaccaria, A Minimalist Feedback Control for Path Tracking in Cartesian Space, Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, MO, USA, 2009.

[2] Angelo Morro, Antonio Sgorbissa, Renato Zaccaria, Path Following for Unicycle Robots with Arbitrary Path Curvature, Technical Report, 2010. Available online: http://www.robotics.laboratorium.dist.unige.it/index.php?section=5

[3] Kaminer, I., A. Pascoal, E. Hallberg and C. Silvestre, Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control. J. of Guidance, Control, and Dynamic Syst. 21(1), 29–38.

[4] Breivik, M. and T.I. Fossen, Guidance Laws for Autonomous Underwater Vehicles. Chapter 4, In "Intelligent Underwater Vehicles. I-Tech Education and Publishing (A. V. Inzartsev, Ed.), Vienna, January 2009.

[5] A. Micaelli and C. Samson, Trajectory tracking for unicycle-type and two-steering-wheels mobile robots, Institut National de Recherche en Informatique, et en Automatique, Rapport de Recherche 2097, 1993.

[6] C. Canudas de Wit, H. Khennouf, C. Samson, and O.J. Sørdalen, Nonlinear control design for mobile robots, Recent trends in mobile robots, Y.F. Zheng Editor., World Scientific Series in Robotics and Automated Systems, Singapore, 1993.

[7] P. Encarnacão, A. Pascoal, 3D path-following for autonomous underwater vehicles. Proc. of the 39th IEEE Conf. on Decision and Control, CDC 2000. Sydney, Australia, 2000.

[8] G. Casalino, M. Aicardi, A. Bicchi, A. Balestrino, Closed loop steering and path-following for unicycle-like vehicles: a simple lyapunov function based approach. IEEE Robotics and Automation Magazine 2 (1), 2735, 1995.

[9] M. Aicardi, G. Casalino, G. Indiveri, P. Aguiar, P. Encarnacão, A. Pascoal. A planar path-following controller for underactuated marine vehicles. Proc. of the Ninth IEEE Mediterranean Conf. on Control and Automation, MED 2001. Dubrovnik, Croatia.

[10] L. Lapierre, D. Soetanto, Nonlinear path-following control of an AUV, Ocean Engineering 34 (2007) 17341744.

[11] D. Soetanto, L. Lapierre, A. Pascoal, Nonsingular path-following control of dynamic wheeled robos with parametric modeling uncertainty. Proc. of the 11th Int. Conf. on Advanced Robotics, ICAR 2003, Coimbra, Portugal.

[12] G. indiveri, A. A. Zizzari, Kinematics Motion Control of an Underactuated Vehicle: a 3D Solution with Bounded Control Effort. In: Proc. of the IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, IFAC NGCUV 2008, Killaloe, Ireland, 8 – 10 April 2008.

[13] L. Lapierre, B. Jouvencel, Robust Nonlinear Path-Following Control of an AUV, IEEE J. of Oceanic Engineering, Volume: 33, Issue: 2, pp. 89 - 102, 2008

[14] Y. Wang, W. Yan, B. Gao, R. Cui, Backstepping-based path following control of an underactuated autonomous underwater vehicle, Int. Conf. on Information and Automation, ICIA '09, pp. 466 - 471, 2009.

[15] O. Calvo, A. Sousa, A. Rozenfeld, G. Acosta, Smooth path planning for autonomous pipeline inspections, 6th Int. Multi-Conf. on Systems, Signals and Devices, 2009. SSD '09, Page(s): 1 - 9, 2009.

[16] S. van der Zwaan, M. Perrone, A. Bernardino, J. Santos-Victor, Control of an Aerial blimp based on visual input, 8th Int. Symp. on Intelligent Robotic Systems - SIRS200 - Reading, UK, July 2000.

[17] C. Pêtrès, Y. Pailhas, P. Patrón, Y. Petillot, Jo. Evans, and D. Lane, Path Planning for Autonomous Underwater Vehicles, IEEE Trans. on Robotics, Vol. 23, No. 2, April 2007

[18] S. Fleury, P. Soures, J.P. Laumond, R. Chatila, Primitives for smoothing mobile robot trajectories IEEE Int. Conf. on Robotics and Automation, Atlanta (USA), 2-6 Mai 1993, pp.832-839, IEEE Trans. on Robotics and Automation, Vol.11, N. 03, pp.441-448, Juin 1995.

[19] C. Silvestre, R. Cunha, N. Paulino, and A. Pascoal, A Bottom-Following Preview Controller for Autonomous Underwater Vehicles, IEEE Trans. on Control Systems Technology, Vol. 17, No. 2, March 2009

[20] A. Sgorbissa, A. Villa, A. Vargiu, R. Zaccaria, A Lyapunov-Stable, Sensor-Based Model for Real-Time Path-Tracking among Unknown Obstacles, Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, MO, USA, 2009.