

Nonlinear output feedback for autonomous U-turn maneuvers of a robot in orchard headlands

E. Le Flécher, Adrien Durand Petiteville, Frédéric Gouaisbaut, Viviane Cadenat, Thierry Sentenac, Stavros Vougioukas

▶ **To cite this version:**

E. Le Flécher, Adrien Durand Petiteville, Frédéric Gouaisbaut, Viviane Cadenat, Thierry Sentenac, et al.. Nonlinear output feedback for autonomous U-turn maneuvers of a robot in orchard headlands. ICINCO 2019 - International Conference on Informatics in Control, Automation and Robotics, Jul 2019, Prague, Czech Republic. pp.355-362, 10.5220/0007918803550362 . hal-03192745

HAL Id: hal-03192745
https://hal.science/hal-03192745

Submitted on 8 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives| 4.0 International License

Nonlinear output feedback for autonomous U-turn maneuvers of a robot in orchard headlands

Le Flécher E.[1,2][a], Durand-Petiteville A[3], Gouaisbaut F.[1,2], Cadenat V.[1,2] Sentenac T.[1], Vougioukas S.[3]

[1]*CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*

[2]*Univ de Toulouse, UPS, F-31400, Toulouse, France*

[3]*Departement of Biological and Agricultural Engineering, University of California, Davis, CA, 95616, USA*

{*eleflech,cadenat,sentenac*}*@laas.fr*

Keywords: mobile robotics, sensor-based navigation, control theory, agricultural robotics, orchards

Abstract: This paper is devoted to the navigation of a robot in orchard headlands using embedded sensors such as lasers, Lidars or cameras. The main idea is to consider a differential robot model directly in polar coordinates and not in Cartesian coordinates which makes it possible to obtain simpler expressions of the outputs. Then two nonlinear output state feedback controllers are proposed to track two shapes based on spirals allowing to go from one row of fruit trees to another. These controllers are based on an input to output linearization and proved to be very efficient on simulations.

1 INTRODUCTION

According to (Foley et al., 2011), agriculture will need by 2050 to double its production to feed the increasing population. Mechanization has been identified as one of the best solutions to increase significantly the food production (Reid, 2011). For mowing, spraying, pruning, harvesting in fields or orchards, one of the main challenge consists in navigating. In other words, the robot has to autonomously drive from the entrance of a row to its exit, then navigate in the headlands to reach the entrance of the next row. This process is repeated to cover the area of interest. This problem has been addressed for many years as it is shown in the review proposed in (Li et al., 2009). Most of the presented solutions focus on open field navigation and rely on GPS. The presented work is included in the orchards navigation problem where GPS cannot be used because its signal is blocked by the dense canopy. For this reason, orchards navigation systems are based on sensors such as laser range finders, Lidars or cameras. For example, laser range finders and cameras are used to drive through rows (Subramanian et al., 2006), (Sharifi and Chen, 2015). The headland navigation problem is addressed in works such as (Andersen et al., 2010), (Zhang et al., 2014) and (Bayar et al., 2015) where sensory data (proprioceptive and/or exteroceptive) are used to localize the robot in a metric map of the orchard in order to follow a precomputed path. However, keeping an updated metric map seems challenging due to constant variation of the environment. Indeed, over the years, trees grow up, and are pruned; over the seasons, leaves grow then disappear; over the days, fruits grow, bend branches and finally fall. Moreover, metric localization can accumulate errors and lead to the navigation failure. To overcome these drawbacks, we have presented in (Durand-Petiteville et al., 2017) a solution relying on a topological representation of the environment coupled with a set of output feedback controllers. Especially, the headland navigation consists in a U-shape turn around the last tree of the row. It is performed thanks to an output feedback controller tracking a spiral, inspired by the work presented in (Boyadzhiev, 1999).

However, as presented in (Bochtis and Vougioukas, 2008) and in (Zhang et al., 2014), it is not always possible to perform a simple U-shape turn because of the minimum turning radius of the robot which might be too small to deal with narrow rows. It is then mandatory to consider different shapes for turns. Thus, in this paper, we address the headland navigation by designing controllers allowing to perform two different shapes of turns (see figure 1). The first one, the classical U-shape turn, is used when

[a] https://orcid.org/0000-0002-2683-8675

the robot turning radius is large enough to allow the vehicle directly reach the middle of the next row. When it is not the case, a more complex shape for the turn known as Ω-shape turn is chosen as proposed in (Bochtis and Vougioukas, 2008). This shape is classically used when the robot can only drive forward. The lack of sensors to detect hazards while driving backward usually motivates this choice.

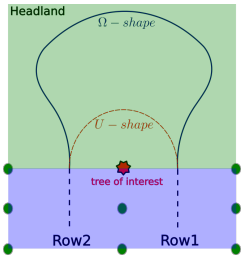


Figure 1: Dashed red: U-shape || Solid blue: Ω-shape

The problem considered in this paper is centered around control only (the planning level is not considered here). Thus, this work aims i) at creating the paths of reference to perform U-turn maneuvers (either Ω or U, built from spirals, following the ideas proposed in (Boyadzhiev, 1999)) and ii) at designing nonlinear controllers able to track the predefined shapes. The whole problem is stated in terms of polar coordinates instead of Cartesian ones. This formulation allows to design output feedback controllers which depend on the measurements and not only on the robot absolute position for instance as it has been done in (d'Andrea Novel et al., 1992; d'Andréa Novel et al., 1995). Therefore, unlike the works (Asif et al., 2016; Yang et al., 2016; Shi et al., 2016), there is no need to consider the design of observers to recover the whole state. In this way, metric localization and cumulative errors can be avoided. From a technical point of view, the proposed controllers rely on an input to output linearization based on the expression of the errors dynamics. A nonlinear diffeomorphism based only on the outputs allows to transform the nonlinear model into a linear system. This latter is then controlled using classical linear control laws. Finally, notice that both controllers induce a zero dynamic which is proved to be stable only.

This paper is organized as follows. The next section is devoted to the problem statement. An exogenous model of the spiral shape is first derived using polar information. Then, based on this reference, a nonlinear model of the error dynamics is established. The nonlinear output feedbacks controllers, based on input to state linearization techniques (Isidori, 2013), allowing to vanish these errors are designed in section 3. Finally, simulation results allowing to highlight the interest and the efficiency of the approach end the article.

2 ROBOT AND U-TURNS MODELING

In this section, the mathematical models used in this paper are introduced. First the model of a differential robot is presented. Next, the spiral model from (Boyadzhiev, 1999) is recalled. Finally, based on the geometry of the spiral, the two shapes of turns are modeled.

2.1 System modeling

We consider a differential robot driving in the headland by turning around the last tree of a row. Firstly, a global frame $F_w = (O_w, \vec{x}_w, \vec{y}_w, \vec{z}_w)$ represents the position of the tree of interest (see figure 2(a)). Next, the frame $F_r = (O_r, \vec{x}_r, \vec{y}_r, \vec{z}_r)$ is attached to the differential robot. The robot states in F_w are defined by $\chi(t) = [d(t)\ \beta(t)\ \alpha(t)]^T$, where $d(t)$ is the norm of the vector $\vec{d}$ connecting O_w and 0_r, $\beta(t)$ the angle between $\vec{x}_w$ and $\vec{d}$, and $\alpha(t)$ the angle between $\vec{x}_r$ and $\vec{d}$. $d(t)$ and $\beta(t)$ are the polar coordinates of the robot in F_w, and $\alpha(t)$ is its orientation. This choice for the state representation is motivated by the fact that both $d(t)$ and $\alpha(t)$ can be directly measured by embedded sensors such as laser range finders, stereo visions systems or Lidars. Thus, it is assumed that the measure is given by:

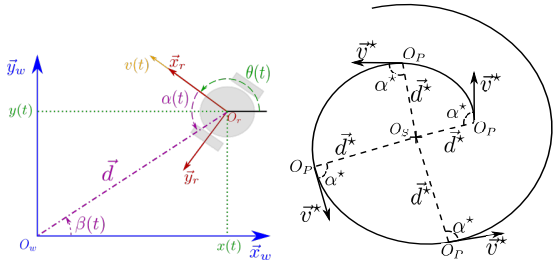
$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \chi(t) \tag{1}$$

Finally, the robot control input is defined by $[v(t)\ \omega(t)]^T$, with $v(t)$ the linear velocity along $\vec{x}_r$ and $\omega(t)$ the angular velocity around $\vec{z}_r$. Thus, the kinematic equations are given by:

$$\begin{bmatrix} \dot{d}(t) \\ \dot{\beta}(t) \\ \dot{\alpha}(t) \end{bmatrix} = \begin{bmatrix} -\cos(\alpha(t)) & 0 \\ \frac{-\sin(\alpha(t))}{d(t)} & 0 \\ \frac{\sin(\alpha(t))}{d(t)} & -1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \tag{2}$$

2.2 Spiral modeling

In this work, it is proposed to perform different shapes of turns by tracking paths created thanks to spirals, whose a model is given in (Boyadzhiev, 1999). Let define a point O_p moving on a plane with respect to a fixed point O_s (see figure 2(b)). From now on, O_s will be considered as the center of the spiral. $\vec{v}^\star$ is the velocity vector applied to O_p and its norm is denoted by $v^\star(t)$. Moreover $\vec{d}^\star$ is the vector connecting O_s to



(a) System model (b) Spiral model

Figure 2: Models

O_p whose norm is $d^\star(t)$. Finally $\alpha^\star(t)$ is defined as the oriented angle between $\vec{v}^\star$ and $\vec{d}^\star$. In (Boyadzhiev, 1999) it is shown that if both $v^\star(t)$ and $\alpha^\star(t)$ are constant then O_p describes a spiral whose center is O_s. For this reason they are respectively denoted $v^\star$ and $\alpha^\star$ from now on. Moreover the author shows that the dynamics of the distance $d^\star$ is defined as follows:

$$\dot{d}^\star = -v^\star \cos(\alpha^\star) \tag{3}$$

As it can be seen in this equation, the type of performed spiral depends on the sole parameter $\alpha^\star$. First if $0 < \alpha^\star < \pi$, O_p turns counter-clockwise with respect to O_s otherwise if $0 < \alpha^\star < -\pi$ it turns clockwise. Then if $0 \leq \alpha^\star < \pi/2$ or $0 \leq \alpha^\star < -\pi/2$, $d^\star(t)$ decreases with time. In other words, O_p is describing an inward spiral around O_s. If $\pi/2 < \alpha^\star \leq \pi$ or $-\pi/2 < \alpha^\star \leq -\pi$ then $d^\star(t)$ increases with time which means O_p is describing an outward spiral around O_s. Finally, if $\alpha^\star = \pi/2$ or $\alpha^\star = -\pi/2$, $d^\star(t) = d^\star(0)$. O_p then describes a circle of radius $d^\star(0)$ around O_s.

Equation (3) and its analysis highlight our interest in the spirals. Indeed, adapting the spiral model to our system, *i.e.,* the center of the spiral is the position of the tree of interest with $O_w = O_S$, defines a reference path, whose distance dynamics is known, solely based on one point. The reference path is thus defined in the robot sensor space.

2.3 U-shape turn modeling

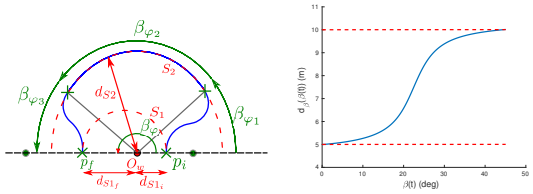
2.3.1 U-shape turn in the navigation problem

In the context of an orchard navigation, the robot has to drive from p_i, end of the current row, to p_f, beginning of the next row (see figure 3(a)). In this specific application, the width of each row is roughly known (*e.g.,* farmers data or Google map) and saved in the robot database. Thus, when the end of the row is detected by a dedicated data processing algorithm, the robot measures d_{S1_i} and extracts d_{S1_f} from its database. It is then possible to compute the spiral S_1 linking those two points, *i.e.,* to compute $\alpha^\star$ for a given $v^\star$. Moreover, for a given spiral, one can calculate the required turning radius. If this latter is smaller than the robot one, then a U-shape turn is tracked to enter the next row.

2.3.2 U-shape turn error

As previously explained, U-shape turns are performed by tracking a spiral. To do so, it is mandatory to respectively make $\alpha(t)$ and $d(t)$ converge towards $\alpha^\star$ and $d^\star(t)$. Thus, the problem of tracking a U-shape turn can be seen as the problem of finding a controller such that the error (4) converges to zero.

$$\begin{cases} e_d(t) = d(t) - d^\star(t) \\ e_\alpha(t) = \alpha(t) - \alpha^\star \end{cases} \tag{4}$$



(a) U-shape and Ω-shape turns (b) Dotted: d_{S1} & d_{S2} ‖ Solid: $d_\beta(\beta(t))$

Figure 3: U-shape, Ω-shape and distance profile $d_\beta(\beta(t))$

2.4 Ω-shape turn

2.4.1 Ω-shape turn in the navigation problem

In case the spiral S_1 computed in 2.3.1 requires a too large turning radius, the robot has to perform an Ω-shape turn. One recalls that this choice is motivated by the inability of the robot to go backward. An Ω-shape turn is made of three parts: i) moving from p_i to a spiral S_2 (φ_1), ii) tracking S_2 (φ_2), and iii) moving from S_2 to p_f (φ_3). For each part, the robot turning radius has to be large enough to track the reference. Moreover, one has to guarantee the Ω shape, and thus the success of the U-turn. Let define, β_φ the angular distance to switch from one row to the next one. Moreover, β_{φ_1}, β_{φ_2} and β_{φ_3} are respectively the angular distances of parts φ_1, φ_2 and φ_3. Because $\beta_{\varphi_1} + \beta_{\varphi_2} + \beta_{\varphi_3} = \beta_\varphi$, with $\beta_{\varphi_1} \geq 0$, $\beta_{\varphi_2} \geq 0$ and $\beta_{\varphi_3} \geq 0$, it is mandatory that:

$$\beta_{\varphi_1} + \beta_{\varphi_3} \leq \beta_\varphi \tag{5}$$

Next, let define S_{min}, the spiral centered on O_w with the smallest radius d_{min} such that the robot turning radius is large enough to track it. Thus, parts φ_1 and φ_3

have to make the robot move from/to spiral S_1 to/from a spiral S_2 centered on O_w with a radius $d_{S2} \geq d_{min}$ while respecting the constraint introduced by equation (5). For this reason, it is proposed to track a distance profile $d_\beta(\beta(t))$, guaranteeing that the robot reaches the spiral for a given angular distance.

2.4.2 Distance profile design

To design the distance profile $d_\beta(\beta(t))$, we first define normalized angle $\bar{\beta}(t)$, with $0 \leq \bar{\beta}(t) \leq 1$, as:

$$\bar{\beta}(t) = \frac{\beta(t) - \beta(t_0)}{\beta(t_f) - \beta(t_0)} \tag{6}$$

where t_0 and t_f are the initial and final time, and $\beta(t)$ is based on the encoders measurements. $\beta(t_0)$ and $\beta(t_f)$ represent therefore the starting angle and the final angle. In addition, $\bar{d_\beta}(\bar{\beta}(t))$ is defined as the normalized distance to track, with $0 \leq \bar{d_\beta}(\bar{\beta}(t)) \leq 1$. In order to obtain a distance profile dealing with the initial and final robot orientations, the path described by the profile is tangent to the spirals. Thus, it is proposed to define $\bar{d_\beta}(\bar{\beta}(t))$ such as:

$$\bar{d_\beta}(\bar{\beta}(t)) = k_0 \tan^{-1}(k_1(\bar{\beta}(t) + k_2)) + k_3 \tag{7}$$

with k_0, k_1, k_2 and k_3, scalar terms used to design the shape of the normalized function. Moreover the choice of those parameters has to guarantee that the distance profile to track does not require a too large turning radius. Finally, using equations (6) and (7), we obtain:

$$d_\beta(\beta(t)) = d(t_0) + \bar{d_\beta}(\bar{\beta}(t))[d(t_f) - d(t_0)] \tag{8}$$

An example of this distance profile is given in figure 3(b) with $k_0 = 0.365$, $k_1 = 10$, $k_2 = -0.5$, $k_3 = 0.5$, $\beta(t_0) = 0$ rad, $\beta(t_f) = \pi/4$ rad, $d(t_0) = 5$ m and $d(t_f) = 10$ m.

2.4.3 Ω-shape turn error

As previously explained, Ω-shape turns are made of three parts. φ_2 consists in a spiral tracking and thus requires to find a controller vanishing the error (4). Regarding parts φ_1 and φ_3, it is mandatory to make $d(t)$ converge towards $d_\beta(\beta(t))$. Thus, the problem of making the robot reach a spiral turns into the problem of finding a controller such that the error (9) converges to zero.

$$e_{d_\beta}(t) = d(t) - d_\beta(\beta(t)) \tag{9}$$

U-shape and Ω-shape turns can thus be performed by following a spiral and moving from one spiral to the other. At this stage, as the different turns have been expressed by introducing a set of errors, the next section will be dedicated to the design of controllers allowing to make them vanish.

3 CONTROLLERS DESIGN

This section proposes to design an output feedback control law which makes the errors (4) and (9) converge toward zero asymptotically. As the system is nonlinear, the main idea is to use an exact input to state linearization method proposed by (Isidori, 2013). We first present the controller allowing the tracking of a spiral. In a second subsection, an adaptation of this controller is presented to move from one spiral to another one. In this section, it is assumed that $v(t)$ is a constant input set at $v^\star$. The only remaining input is therefore $\omega(t)$.

3.1 Spiral tracking

The errors (4) dynamics are defined by:

$$\begin{cases} \dot{e}_d(t) = v^\star \left[\cos(\alpha^\star) - \cos(\alpha(t))\right] \\ \dot{e}_\alpha(t) = -\omega(t) + \frac{v^\star}{d(t)} \sin(\alpha(t)) \end{cases} \tag{10}$$

Which can be written as follows:

$$\begin{cases} \dot{e}_d(t) = v^\star \left[\cos(\alpha^\star) - \cos(e_\alpha(t) + \alpha^\star)\right] \\ \dot{e}_\alpha(t) = -\omega(t) + \frac{v^\star}{e_d(t) + d^\star(t)} \sin(e_\alpha(t) + \alpha^\star) \end{cases} \tag{11}$$

The main idea of this section is to linearize the error system. To this end, consider the transformation:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} e_d(t) \\ v^\star \left[\cos(\alpha^\star) - \cos(e_\alpha(t) + \alpha^\star)\right] \end{bmatrix} = T(e) \tag{12}$$

Notice that $T(0) = 0$ and in the domain $\mathcal{D}$ defined by:

$$\mathcal{D} = \left\{ (e_d, e_\alpha) \in \Re^2 \| e_d \in \Re, e_\alpha \in]-\alpha^\star, -\alpha^\star + \pi[\right\} \tag{13}$$

T defines a diffeomorphism. Applying this transformation to the error system leads to:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = v^\star \sin(e_\alpha(t) + \alpha^\star) \dot{e}_\alpha(t) \end{cases} \tag{14}$$

and

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = v^\star \sin(e_\alpha(t) + \alpha^\star)(-\omega(t) + \frac{v^\star}{e_d(t) + d^\star(t)} \sin(e_\alpha(t) + \alpha^\star)) \end{cases} \tag{15}$$

Taking:

$$\omega(t) = \frac{1}{v^\star \sin(e_\alpha(t)+\alpha^\star)}\omega_2(t) + \frac{v^\star}{e_d(t)+d^\star(t)}\sin(e_d(t)+d^\star) \quad (16)$$

we obtain :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = \omega_2(t) \end{cases} \quad (17)$$

where $\omega_2(t)$ is a new control law, which has to be designed. At this stage, as the system (17) is linear, a classical linear control law:

$$\omega_2(t) = -\lambda_1\, z_1(t) - \lambda_2\, z_2(t) \quad (18)$$

with $\lambda_1, \lambda_2 > 0$, allows to stabilize asymptotically system (17).

Therefore, we propose the following theorem:

Theorem 1. *Consider two positive scalars, λ_1, λ_2, the error system* (10) *in closed loop with the control law* (16)*,* (17)*,* (18)*, where $z = T(e)$ in* (12) *is locally asymptotically stable.*

Proof 1. *It is sufficient to notice that z converge asymptotically to zero and $z = T(e)$ defines a local diffeomorphism with $T(0) = 0$.*

Remark 1. *λ_1 and λ_2 are two gains used to tune the speed of convergence of the system in closed loop.*

Remark 2. *The relative degree of the chosen output e_d used for the linearization is two while the original state space dimension is three. Therefore, there exists a zero dynamic. Defining $\beta^\star$ the reference trajectory for β, the β error dynamics is driven by*

$$\dot{e}_\beta(t) = (-\frac{\sin\alpha}{d(t)} + \frac{\sin\alpha^\star}{d^\star})v^\star$$

Following (Isidori, 2013), it is straightforward to see that the zero dynamic defined by

$$\dot{e}_\beta(t) = 0$$

is stable only. As a consequence, in steady state, we may expect a static error between $\beta(t)$ and its expected value $\beta^\star(t)$.

3.2 Distance profile tracking

From now on, a path leading to the convergence toward a specific spiral is described and thus a function of $d_\beta(\beta(t))$ is proposed. Therefore, the error to be minimized can be rewritten as:

$$e_{d_\beta}(t) = d(t) - [d(\bar{\beta}(t_0)) + \bar{d_\beta}(\bar{\beta}(t))d_{gap}] \quad (19)$$

The controller designed to ensure the convergence of the error will use the same input to output feedback linearization method than the previous one. Let introduce the following new states

$$z_s = \begin{bmatrix} z_{1s} \\ z_{2s} \end{bmatrix} = \begin{bmatrix} e_{d_\beta}(t) \\ \dot{z}_{1s} \end{bmatrix}$$

and therefore the transformation T_s

$$z_s = \begin{bmatrix} z_{1s} \\ z_{2s} \end{bmatrix} = \begin{bmatrix} e_{d_\beta}(t) \\ -v^\star cos(\alpha) + d'_\beta(\beta(t))\frac{sin(\alpha)}{d} \end{bmatrix} = T_s(\chi(t)) \quad (20)$$

where $d'_\beta(\beta(t))$ stands for the derivative of d_β with respect to β. It can be proved that the transformation T_s defines a local diffeomorphism and applying this transformation leads to:

$$\dot{e}_{d_\beta}(t) = -v^\star\, cos(\alpha(t)) - k_0 k_1\, \frac{\dot{\bar{\beta}}(t) d_{gap}}{k_1^2(\bar{\beta}(t)+k_2)^2+1} \quad (21)$$

Calculating the derivative of z_{2s} along the trajectories of the original system gives:

$$\begin{aligned} \ddot{e}_{d_\beta}(t) =& v^\star\, \dot{\alpha}(t) sin(\alpha(t)) - \frac{k_0 k_1\, \ddot{\bar{\beta}}(t) d_{gap}}{(k_1^2(\bar{\beta}(t)+k_2)^2+1)} \\ &+ \frac{2k_0 k_1^3\, \dot{\bar{\beta}}^2 d_{gap}(\bar{\beta}(t)+k_2)}{(k_1^2(\bar{\beta}(t)+k_2)^2+1)^2} \end{aligned} \quad (22)$$

where, thanks to the Eq.6, $\dot{\bar{\beta}}(t)$ and $\ddot{\bar{\beta}}(t)$ are computed as:

$$\dot{\bar{\beta}}(t) = \frac{v^\star \sin(\alpha(t))}{d(t)\beta_{gap}} \quad (23)$$

$$\ddot{\bar{\beta}}(t) = -\frac{v^\star \sin(\alpha(t))\dot{d}(t)}{d^2(t)\,\beta_{gap}} + \frac{v^\star \dot{\alpha}(t) cos(\alpha(t))}{d(t)\beta_{gap}} \quad (24)$$

Hence, the z_s system can be rewritten as :

$$\begin{cases} \dot{z}_{1s} = z_{2s}, \\ \dot{z}_{2s} = f(\alpha(t), \beta(t), d(t)) + g(\alpha(t), \beta(t), d(t))\omega \end{cases} \quad (25)$$

where the function f is defined accordingly to (22),(23),(24).

Choosing the control law for ω such that

$$\omega(t) = \frac{1}{g(\alpha, \beta, d)}(f(\alpha, \beta, d) + \omega_3(t)) \quad (26)$$

where $\omega_3(t)$ is a new control allowing to obtain the following system defined by a simple double integrator:

$$\begin{cases} \dot{z}_{1s} = z_{2s}, \\ \dot{z}_{2s} = \omega_3(t) \end{cases} \quad (27)$$

We propose therefore the following control law for $\omega_3(t)$

$$\omega_3(t) = -\lambda_{1s}\dot{e}_{d_\beta}(t) - \lambda_{2s}e_{d_\beta}(t), \qquad (28)$$

where λ_{1s} and λ_{2s} are two positive scalars which ensure the asymptotic stability of the closed loop system (25) with the control (26), (28).

Therefore we propose the following theorem:

Theorem 2. *Consider two positive scalars, $\lambda_{1s}, \lambda_{2s}$, the error system* (25) *in closed loop with the control law* (26), (28)*, where $z_s = T_s(\chi)$ in* (20) *is locally asymptotically stable.*

Proof 2. *Omitted.*

Remark 3. *The controllers* (16), (26)*, shows two singularities, when $\alpha(t) = \pi$ and $\alpha(t) = 0$. These singularities are due to the polar coordinates model and the construction of the transformation T. Furthermore, these ones occurs when the robot is facing the last tree or let it on its back. Dealing with these configurations are then easily avoidable.*

As previously noted in the last subsection, 3.1, the system in closed loop exibits also a stable zero dynamic.

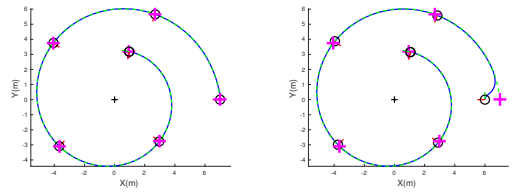
In this section, we have proposed two output state feedbacks controlling the convergence of the states toward its references. A first one follows a specific spiral based on $d^\star(t)$ and $\alpha^\star(t)$. It is used for the part φ_2 of our shapes of turn. The second controller allows performing the convergence toward a spiral with a given angle $\beta^\star$ through a arctan profile. It will be used during the part φ_1 and φ_3 of our turns.

4 SIMULATIONS

In this section, simulations realized with the MATLAB© software are presented. First, we focus on the previously designed controllers, then we couple them in order to simulate Ω-shape turns. For all the simulations: the sampling time is setup as $T_s = 0.1$ s, the spiral center is defined as $[0,0]$ and the linear velocity is given as $v = 0.15$ $m.s^{-1}$. Moreover, the robot is represented by a set of green and red lines, respectively $\vec{x}_r$ and $\vec{y}_r$.

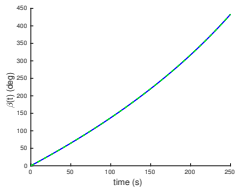
4.1 Spiral tracking

The first set of simulations shows the performance of the spiral tracking controller given by equation (16) setup with $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$. Here, two different situations are presented. The first column (*i.e.,* figures 4(a), 4(c), 4(e) and 4(g)) presents the behavior of the robot when the initial robot pose is on the spiral to be tracked. The robot starts at $\chi(t) = [7, 0, \pi/2]^T$ and tracks an inward spiral defined by $\alpha^\star(t) = 15\pi/32$ rad and $d^\star(t_0) = 7$ m. The second column (*i.e.,* figures 4(b), 4(d), 4(f) and 4(h)) presents the behavior of the robot when its initial pose is not on the spiral. The robot starts at $\chi(t) = [6, 0, \pi/2]^T$ and tracks the same spiral. In figure 4(a) it can be seen that the robot tracks accurately the spiral when its initial position belongs to it. In figure 4(b), the robot first converges toward the spiral and then follows it. However, the tracking is not fully accurate and there is an error between the current position and the desired one. As mentioned in section 3.1, it can be explained by a static error on $\beta(t)$ in steady state. Indeed, as it can be seen in figures 4(d) and 4(c) displaying the evolution of $\beta(t)$ and $\beta^\star(t)$, a static error exists on steady state when the initial robot state does not belong to the spiral. However,

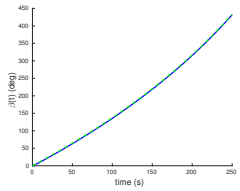


(a) Dotted: Reference spiral - Solid: Robot trajectory

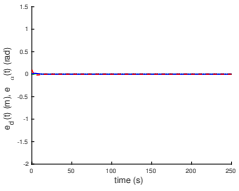
(b) Dotted: Reference spiral - Solid: Robot trajectory



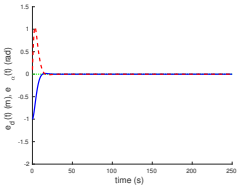
(c) solid: $\beta(t)$ spiral - Solid: $\beta(t)$ robot



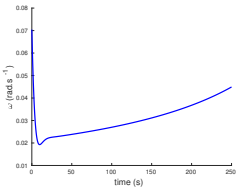
(d) solid: $\beta(t)$ spiral - Solid: $\beta(t)$ robot



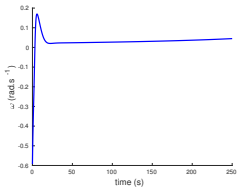
(e) Dotted: errors - Solid: $e_d(t)$ - Dashed: $e_\alpha(t)$



(f) Dotted: errors - Solid: $e_d(t)$ - Dashed: $e_\alpha(t)$



(g) $\omega(t)$



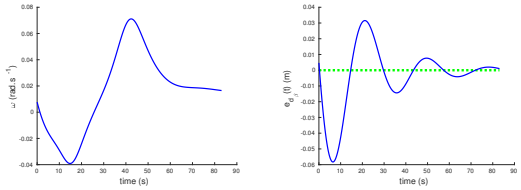
(h) $\omega(t)$

Figure 4: Simulation of spiral tracking

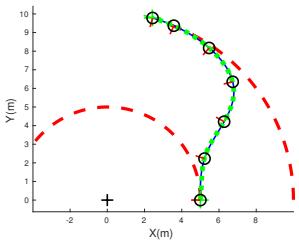
the static error on β does not modify the controller performances regarding $e_\alpha(t)$ and $e_d(t)$ which both converge towards zero (see figures 4(f), 4(e)). Finally, figures 4(h) and 4(g) show the evolution of $\omega(t)$ used to track the spiral.

4.2 Distance profile tracking

The second set of simulations presents the performances of controller given by equation (26), allowing to move toward a specific spiral. The controller is set up with $\lambda_{S1} = 0.05$, $\lambda_{S2} = 0.1$ and $d_\beta(\beta(t))$ is defined with $k_0 = 0.365$, $k_1 = 10$, $k_2 = -0.5$, $k_3 = 0.5$, $\beta(t_0) = 0$ and $\beta(t_f) = 3\pi/8$. The robot starts at $\chi(t) = [5, 0, \pi/2]$ and has to reach a spiral defined with $d^\star(t_0) = 10$ m and $\alpha^\star = \pi/2$ rad. As it can be seen in figure 5(c), the robot reaches the desired spiral at the given angle $\beta(t_f)$. Moreover, the distance profile is accurately tracked (5(b)) with a smooth command $\omega(t)$ (see figure 5(a)).



(a) $\omega(t)$ (b) $e_d(t)$

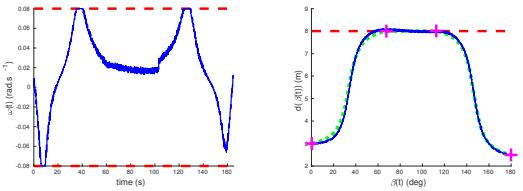


(c) Dashed: Spirals of reference - Dotted: desired trajectory - Solid: Robot trajectory

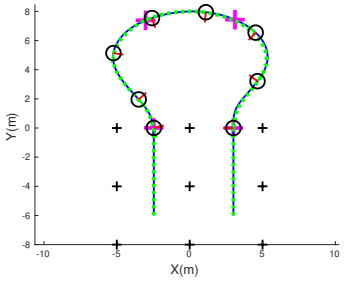
Figure 5: Distance profile tracking

4.3 Ω-shape turn

The third set of simulations presents an example of Ω-turn in the context of a headland navigation. First, a row following task (see (Durand-Petiteville et al., 2017) for more details) is performed, followed by an Ω-turn driving the robot to the next row. Finally, a new row following task is started. For this simulation, the turn is performed with $\omega_{MIN-MAX} = \pm 0.08\ rad/s^{-1}$ and it requires both controllers given by equations (16), (18), (26), and (28). The parameters $\lambda_1 = 0.01$, $\lambda_2 = 0.05$ and $\lambda_{s1} = 0.01$, $\lambda_{S2} = 0.05$ are used for their respective controller. A profile is created for φ_1 and φ_3 with $k_0 = 0.365$, $k_1 = 10$, $k_2 = -0.5$ and $k_3 = 0.5$. The robot starts at $\chi(t) = [3, 0, \pi/2]$ and needs to reach a spiral defined at $d_{S2} = 8$m and $\alpha^\star = \pi/2$ rad with a respective angle of reference of $\beta(\varphi_1) = 3\pi/8$ rad. Finally, $d_{s1_f} = 2.5$m with an angle of reference of $\beta(\varphi_3) = 3\pi/8$ rad. In addition, to assess the sensitivity of the controllers to errors, a centered Gaussian noise has been added to $d(t)$ and $\alpha(t)$, with a respective amplitude of 0.05m and 1°. As shown in 6(c), the robot successfully performs a Ω-shape turn by using alternatively both controllers presented above. Indeed the designed distance profiles, as well as the spiral, are accurately tracked by the robot (see figure 6(b)). Moreover, it should be noticed that the static error on steady state during the spiral tracking does not affect the turn. Indeed, the distance profile for φ_1 is designed to allow to start the spiral tracking when the robot state belongs to the spiral. Thus, despite this drawback, the controller given by (16) provides suitable performances for our specific navigation problem. It can also be noticed that the noise added to the measures does not disturb the behavior of the robot. Finally, an appropriate tuning of parameters defining the distance profiles and spiral allows to perform an Ω-shape turn while dealing with a maximal robot turning radius.



(a) Dashed: $\omega_{MIN-MAX}$ - Solid: $\omega(t)$ (b) Dashed: $d_{S2}(t)$ - Dashed: $d^\star(\beta(t))$ - Solid: $d(\beta(t))$



(c) Dashed: Reference path - Solid: Robot trajectory - Pink crosses: controller switches - Black crosses: trees

Figure 6: Ω-shape turn

5 CONCLUSION

In this work, the problem of driving in headlands during an orchard navigation has been addressed. First, U-shape and Ω-shape turns have been modeled using polar coordinates in order to generate the paths to follow when driving in the headland. Then, two nonlinear output state feedbacks controllers have been designed to track the designed paths. Their design was based on input to state linearization techniques.

Simulations have highlighted the suitable performances of the controllers and their usefulness to efficiently drive the robot in orchards headlands. The obtained results are encouraging and must be integrated on our testbed. To do so, it is planned to couple the controllers with a perception system allowing to detect the trees in the surroundings of the robot. Thus, it will be possible to evaluate the efficiency and robustness of the proposed approach.

Moreover, a couple of challenges have to be addressed to obtain a fully autonomous system. The first one consists in automatically generating the spiral and distance profile parameters based on the *a priori* known and on-line acquired data related to the orchard structure. Moreover, the controller sensitivity to robot state has to be investigated. Thus, it might be required to design a recursive estimation process based on the acquired data to improve the accuracy of the state knowledge. Finally, it seems relevant to guarantee the continuity of the control law when the robot switches from one controller to the other one.

REFERENCES

Andersen, J. C., Ravn, O., and Andersen, N. A. (2010). Autonomous rule-based robot navigation in orchards. *IFAC Proceedings Volumes*, 43(16):43–48.

Asif, M., Memon, A. Y., and Junaid Khan, M. (2016). Output feedback control for trajectory tracking of wheeled mobile robot. *Intelligent Automation & Soft Computing*, 22(1):75–87.

Bayar, G., Bergerman, M., Koku, A. B., and ilhan Konukseven, E. (2015). Localization and control of an autonomous orchard vehicle. *Computers and Electronics in Agriculture*, 115:118–128.

Bochtis, D. and Vougioukas, S. (2008). Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems engineering*, 101(1):1–12.

Boyadzhiev, K. N. (1999). Spirals and conchospirals in the flight of insects. *The college mathematics Journal*, 30(1):23.

d'Andrea Novel, B., Bastin, G., and Campion, G. (1992). Dynamic feedback linearization of nonholonomic wheeled mobile robots. In *Robotics and automation, 1992.*, pages 2527–2532. IEEE.

d'Andréa Novel, B., Campion, G., and Bastin, G. (1995). Control of nonholonomic wheeled mobile robots by state feedback linearization. *The International journal of robotics research*, 14(6):543–559.

Durand-Petiteville, A., Le Flecher, E., Cadenat, V., Sentenac, T., and Vougioukas, S. (2017). Design of a sensor-based controller performing u-turn to navigate in orchards. *International Conference on Informatics in Control, Automation and Robotics*, 2:172–181.

Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., OConnell, C., Ray, D. K., West, P. C., et al. (2011). Solutions for a cultivated planet. *Nature*, 478(7369):337.

Isidori, A. (2013). *Nonlinear control systems*. Springer Science & Business Media.

Li, M., Imou, K., Wakabayashi, K., and Yokoyama, S. (2009). Review of research on agricultural vehicle autonomous guidance. *International Journal of Agricultural and Biological Engineering*, 2(3):1–16.

Reid, J. F. (2011). The impact of mechanization on agriculture. *Bridge*, 41(3):22–29.

Sharifi, M. and Chen, X. (2015). A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards. In *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, pages 251–255. IEEE.

Shi, S., Yu, X., and Khoo, S. (2016). Robust finite-time tracking control of nonholonomic mobile robots without velocity measurements. *International Journal of Control*, 89(2):411–423.

Subramanian, V., Burks, T. F., and Arroyo, A. (2006). Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Computers and electronics in agriculture*, 53(2):130–143.

Yang, H., Fan, X., Xia, Y., and Hua, C. (2016). Robust tracking control for wheeled mobile robot based on extended state observer. *Advanced Robotics*, 30(1):68–78.

Zhang, J., Maeta, S., Bergerman, M., and Singh, S. (2014). Mapping orchards for autonomous navigation. In *2014 Montreal, Quebec Canada July 13–July 16, 2014*, page 1. American Society of Agricultural and Biological Engineers.