Citation: M. Aranda, G. López-Nicolás, C. Sagüés. Sinusoidal input-based visual control for nonholonomic vehicles. Robotica, vol. 31, no. 5, pp. 811-823, 2013

Sinusoidal input-based visual control for nonholonomic vehicles

M. Aranda, G. López-Nicolás and C. Sagüés
Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain
Email: {marandac, gonlopez, csagues}@unizar.es

***Abstract*— This paper proposes a new visual control approach based on sinusoidal inputs to be used on a nonholonomic robot. We present several contributions: in our method, developed considering a unicycle kinematic model, sinusoids are used in such a way that the generated vehicle trajectories are feasible, smooth and versatile. Our technique improves previous sinusoidal-based control works in terms of efficiency and flexibility. As further contributions, we present the analytical expressions for the evolution of the robot's state, and propose a new state-feedback control law based on these expressions. All the information used in the control scheme is obtained from omnidirectional vision by means of the 1D trifocal tensor. Stability analysis of the proposed approach is presented, and its performance is illustrated through experiments.**

I. Introduction

Autonomous navigation and control of mobile robots using vision sensors is a widely researched topic [1]–[3]. Although other sensors can be used for control or localization purposes [4], vision is a highly regarded sensing modality due to its relatively low cost and the rich information it provides. In particular, significant interest has been put in recent years in omnidirectional cameras [5] which, due to their wide field of view, facilitate navigational tasks. In visual control methods, a control loop using image data (image-based approaches), 3D pose information (position-based methods) or a combination of both (hybrid techniques) is employed to drive the robot to the target location, which is defined by an image previously acquired from it.

Part of the visual control methods available in the literature do not consider the particular motion constraints of the vehicle on which they are to be used. In general, it is relevant to take into account the nonholonomic kinematic constraints usually associated to wheeled mobile robots if one aims to generate feasible motions for them [6], [7]. Due to its great interest, the control of nonholonomic vehicles has long focused the attention of researchers. A wide variety of works have been presented in this area, addressing such tasks as motion planning [8], pose stabilization [9]–[11], path following [12], [13] or trajectory tracking [14].

In order to perform motion control of a nonholonomic robot, using sinusoidal velocity inputs can be a good choice, due to the characteristic smoothness of sinusoidal functions, their well-known mathematical properties and their potential to generate flexible and analyzable trajectories. The use of sinusoidal control inputs was introduced in [8], where they were employed to steer the class of nonholonomic systems that can be expressed in chained form (which applies to many real systems). In [15] sinusoidal velocities at multiple frequencies are used in a similar fashion, and the trajectories generated by them are noted to be particularly suited for parallel parking manoeuvres. [16] proposes the use of sinusoids for feedback control to asymptotically stabilize systems in chained form. In [17] a modification of such sinusoidal feedbacks which makes them homogeneous is shown to provide faster convergence rates (exponential stability).

None of these works addresses the design of the sinusoidal inputs themselves. As a consequence, the trajectories and control performance obtained with these methodologies may not be satisfactory. In contrast, the starting point of our sinusoidal-input based control method is the design of the linear and angular velocity waves, which are defined with smoothness, flexibility, feasibility and safety considerations in mind. The aforementioned control approaches were developed for general classes of systems; instead, we focus on the particular nonholonomic constraints of a unicycle vehicle, and obtain analytical expressions of the evolution of the system through the integration of the sinusoidal velocities.

This work addresses the problem of visual control of mobile robots, an open field of research which has stayed very active in recent years. Some examples of related works where vision sensors are employed for mobile robot control tasks are [18] for structure and motion for navigation, [19] for vision-based control, [20] for car platooning or [21] for control with a visual memory. A particular task closely related with the approach we propose in this paper is visual homing, where a vehicle's position is stabilized to a location defined with an image. Different works have addressed the solution to this problem using information from the angles of image features [22], image distances [23] or frequency components [24], performing image warping [25], or utilizing the scale of image features [26], their distances [27] or their vertical displacement [28]. In [29], visual homing is carried out considering car-like kinematic constraints.

In the method we propose, the estimation of the system state is obtained from the visual information provided by the images acquired with an omnidirectional camera. A well-known way to improve the robustness of vision-based computations to image feature matching errors is through the use of the geometric models that relate multiple views of a scene. These models can be computed linearly from sets of feature matches between the different views [30] and may be used to estimate 3D parameters for their use in

This work was supported by Ministerio de Ciencia e Innovación/European Union (projects DPI2009-08126 and DPI2012-32100), by Ministerio de Educación under FPU grant AP2009-3430, and by DGA-FSE (group T04).

control [3]. [31] is an early example of the use of the two-view model expressed by the epipolar geometry for control purposes. Nonholonomic visual control methods have been proposed using this model [32] and the trifocal tensor [33], [34], which encapsulates the geometric constraints between three views [30] and provides improved robustness. [35] introduces the use of the 2D trifocal tensor tensor to control a 6-DOF manipulator. When the camera motion is planar and omnidirectional vision is employed, the best-suited multi-view model available is the 1D trifocal tensor, introduced in [36]. Within the field of mobile robot control, the 1D trifocal tensor has been exploited by using its elements directly in a control loop [37], or as a tool to extract geometric parameters of interest [38]. The work [33], which presents a control method for a nonholonomic mobile robot using three-view geometry, has the following main differences with our approach: we use the 1D tensor (which is simpler to compute and requires fewer point matches than the 2D one) and an omnidirectional camera without specific calibration, while [33] employs the 2D tensor and a calibrated perspective camera. Also, our method performs control to a goal location by estimating the system's state, and is characterized by providing great knowledge of the trajectories, whereas the other work performs the control task by tracking the desired evolutions of the trifocal tensor elements, not relying on state estimation and providing less knowledge about the actual robot trajectories.

In the present work, the computation of the 1D trifocal tensor between the current, initial and target views is used to estimate the state of the robot, providing our method with the robustness properties associated with this model. The target image, taken previously at the desired location, defines the goal of the control task. An important advantage of our approach with respect to the related works in the visual control field is given by the adopted motion strategy: when compared to typical vision-based control methods, our approach based on sinusoidal inputs generates particularly smooth and versatile trajectories. Furthermore, in comparison with previous works on 1D trifocal tensor-based control [37], [38], our method has the benefit of providing greater control on the path followed by the vehicle.

We can summarize the properties of our approach, in a qualitative manner, as follows: it is more flexible and efficient than related works in the field of nonholonomic motion planning and control, and it provides increased smoothness, trajectory control and robustness with respect to comparable visual control methods. In addition, our technique is versatile and can be easily adapted and extended to other tasks with different requirements, such as static obstacle avoidance or parking manoeuvres.

Figure 1 displays an overview of the control loop. The stabilization of the nonholonomic vehicle is performed in two sequential phases. The sinusoidal input-based control, which aligns the vehicle with the target location, is followed by a second step consisting in a straight-line trajectory to carry out depth correction using the 1D trifocal tensor elements directly.

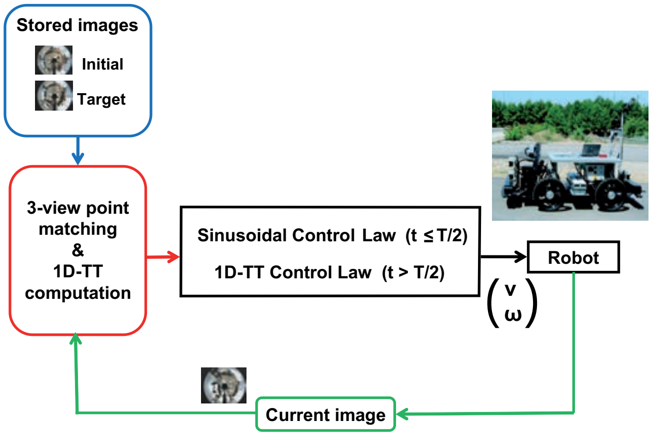
Stored images
Initial
Target
3-view point matching & 1D-TT computation
Sinusoidal Control Law (t ≤ T/2)
1D-TT Control Law (t > T/2)
Robot
v
ω
Current image

Fig. 1. Overview of the visual control loop. The sinusoidal input-based control law (step 1) operates until a fixed time $T/2$, where T is the period of the sinusoid. From that instant depth correction is performed using a 1D trifocal tensor-based control law (step 2).

The contents of the paper are organized as follows: Section II describes the model of the system to be controlled. Section III discusses the first step of our control approach, which is based on the sinusoidal inputs. The second step of the control method is discussed in section IV. In section V we describe how the state estimation is obtained from the 1D trifocal tensor. The stability of the control scheme is analyzed in section VI. Section VII presents the results of the experimental evaluation of the method. A discussion of the method is presented in Section VIII and finally, the conclusion of the paper is given in section IX.

II. System Model

A nonholonomic robot moving on the ground plane constitutes the dynamical system to be controlled. The state of this system is defined by the robot's localization, given by $\mathbf{x} = (x, z, \phi)^T$. The origin of the coordinate system is the goal location, given by the localization at which the target image was obtained, i.e. $\mathbf{x_3} = (0, 0, 0)^T$. The nonholonomic differential kinematics of the vehicle expressed in state space form as a function of the translation and rotation velocities of the robot (v, ω) is as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega. \tag{1}$$

Since the primary information we will extract from the system through the 1D trifocal tensor is of angular nature, it is also useful to express the system's state and its kinematics in polar coordinates $(\rho, \alpha, \phi)^T$ as illustrated in Fig. 2. The lateral and depth coordinates are related to the polar ones through $x = -\rho \sin\psi$ and $z = \rho\cos\psi$, while the alignment error is defined as: $\alpha = \phi - \psi$. The kinematics in polar coordinates are:

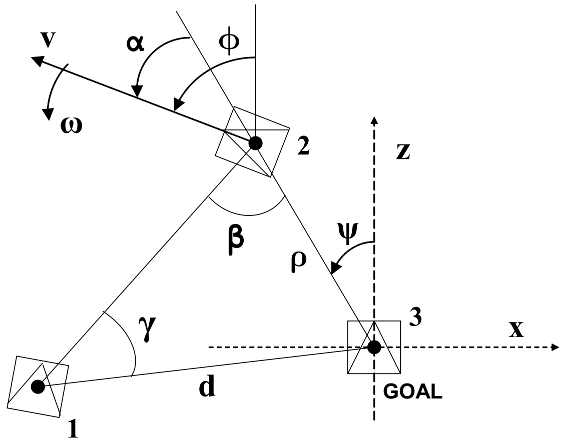

Fig. 2. Definition of the elements of our system and the geometric variables employed. '**3**' identifies the goal view, which serves as the global coordinate system. '**2**' is the robot's current location, and '**1**' is the location of the initial view. γ and β are angular sectors (≥ 0) and d is the distance between the initial and target locations.

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos\alpha \\ -\frac{1}{\rho}\sin\alpha \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \omega. \tag{2}$$

III. Sinusoidal input-based control scheme

Our approach for the first step of the control strategy is based on defining the desired trajectories of the state variables in a sinusoidal-varying velocity framework. The velocities we propose follow a sinusoidal time variation expressed by:

$$v = a\sin(\Omega t) \tag{3}$$

$$\omega = b\sin(2\Omega t). \tag{4}$$

We assume throughout that the angular frequency of the sinusoid (Ω) is set to a constant value. This is a design parameter whose value will be determined by the time interval in which we want the first step of the control to be carried out. This interval is equal to one half-period of the sinusoid, i.e. $T/2 = \pi/\Omega$. The parameters a and b are real values that set the amplitudes of the sinusoidal velocity waves. We first consider the robot's state at $t = 0$ to be $(x_0, z_0, 0)^T$. If the initial orientation of the robot is nonzero, we will compute the starting time ($t_s \neq 0$) of the sinusoidal velocity waves, which will now run from t_s to $T/2$. This will be described in section III-B.

The velocities defined in (3) and (4) have been designed in such a way that the rotational velocity is zero at the points where the linear velocity is either maximum (in absolute value) or zero. Besides, the greatest rotational speeds are associated in our method to intermediate linear velocity values. This results in a behavior which is convenient for a vehicle with nonholonomic motion constraints. The generated motion fulfills criteria of smoothness, safety, and feasibility, and is appropriate for this kind of vehicles.

A. Evolution of the system

In this section, we obtain the analytical expressions for the evolutions of the state variables in a motion planning scenario under the sinusoidal-varying velocity commands (3) and (4). By integrating these sinusoidal inputs over time, we can derive the equations for the evolution of the three state variables. We will first obtain the time variation of the orientation component:

$$\begin{aligned} \phi(t) &= \phi_0 + \int_0^t \dot{\phi} d\tau = \int_0^t \omega d\tau \\ &= \int_0^t b\sin(2\Omega\tau)d\tau = \frac{b}{2\Omega}(1 - \cos 2\Omega t), \end{aligned} \tag{5}$$

considering $\phi_0 = 0$. Once we know how ϕ evolves with time, we can substitute this result in (1) to obtain the time variations for the two other state variables. We will first determine the evolution of $x(t)$. For this purpose we will use the Taylor series representation of $\sin\phi$:

$$\begin{aligned} \dot{x}(t) &= -v\sin\phi = -a\sin(\Omega t) \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}\phi^{2n+1} \\ &= -a\sin(\Omega t)\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}\left(\frac{b}{2\Omega}\right)^{2n+1}(1-\cos 2\Omega t)^{2n+1} \\ &= -a\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}\left(\frac{b}{\Omega}\right)^{2n+1}\sin^{4n+3}(\Omega t), \end{aligned} \tag{6}$$

where the identity: $1 - \cos 2\Omega t = 2\sin^2 \Omega t$ has been used. We can now obtain the evolution of the state variable x in the time interval $0 \leq t \leq T/2$ through integration:

$$\begin{aligned} x(t) &= x_0 + \int_0^t \dot{x}(\tau)d\tau \\ &= x_0 - \int_0^t a\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}\left(\frac{b}{\Omega}\right)^{2n+1}\sin^{4n+3}(\Omega\tau)d\tau. \end{aligned} \tag{7}$$

The integral of the sine function in (7) can be expressed as:

$$\begin{aligned} &\int_0^t \sin^{4n+3}(\Omega\tau)d\tau = \\ &\left[-\frac{\cos\Omega t}{\Omega} \cdot {}_2F_1\left(1/2, -2n-1; 3/2; \cos^2\Omega t\right)\right]_0^t, \end{aligned} \tag{8}$$

where ${}_2F_1$ is the Gaussian hypergeometric function:

$${}_2F_1(p,q;r;s) = \sum_{k=0}^{\infty} \frac{(p)_k(q)_k}{(r)_k}\frac{s^k}{k!},$$

with $(p)_k = p(p+1)(p+2)\cdots(p+k-1)$ and $(p)_0 = 1$. It can be easily seen that when q is a negative integer (which is indeed the case for us, since $q = -2n-1$) the series has only $|q+1|$ nonzero terms, i.e. $k = 0, ..., q$. We finally get to the following expression for $x(t)$:

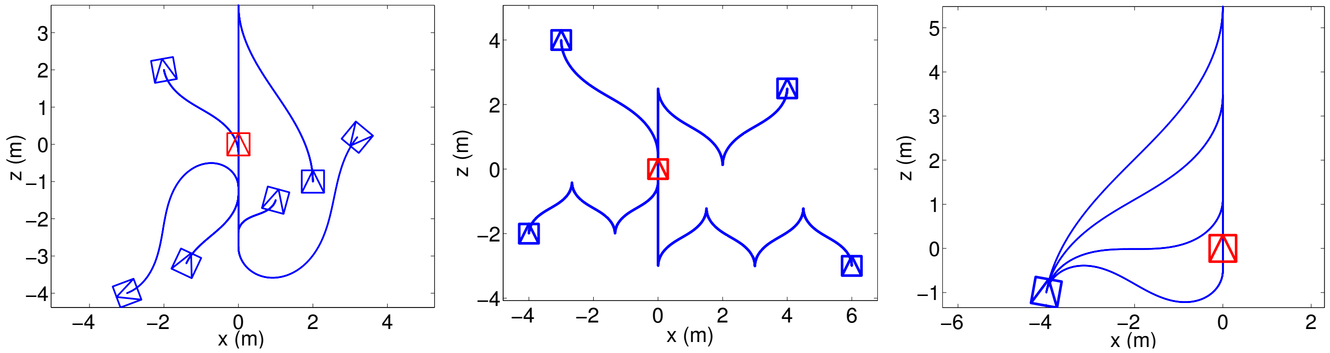

Fig. 3. Left: Example robot trajectories obtained with the sinusoidal input-based control law. This law aligns the vehicle with the target pose $(0,0,0°)^T$. In a second control step, the depth error is corrected following a straight-line path. Center: Four trajectories generated by extending the proposed method in order for the motions to span several half-periods of the sinusoid. Right: Control paths from starting position $(-4,-1,-10°)^T$ with $\phi_m = 45°$, 60°, 90° and 120° (top to bottom curves).

$$x(t) = x_0 + \left[\frac{a\cos(\Omega t)}{\Omega}\cdot\Gamma(b,t,1)\right]_0^t, \tag{9}$$

where we define:

$$\Gamma(b,t,m) = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+m)!}\left(\frac{b}{\Omega}\right)^{2n+m} \cdot {}_2F_1\left(1/2,-2n-m;3/2;\cos^2\Omega t\right). \tag{10}$$

Thus, x can be exactly determined at any time through the sums of series of infinite terms. Note, however, that the index of these sums, n, is the index of the Taylor series representation of $\sin\phi$. The values of ϕ will be, at most, in the range $[-\pi,\pi]$ (usually, the actual range will be considerably smaller); therefore, taking only a small number of terms in the sums will suffice to ensure that an accurate solution is obtained. Indeed, it can be shown that the error in the computation of $x(t)$ becomes negligible by taking only three terms $(n = 0, 1, 2)$ in the sum of (10).

The time evolution of the state variable z can be worked out in an analogous way, through the integration of the corresponding expression in (1). This time we use the Taylor series expansion of $\cos\phi$.

$$\begin{aligned}\dot{z}(t) &= v\cos\phi = a\sin(\Omega t)\cdot\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}\phi^{2n}\\ &= a\sin(\Omega t)\cdot\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}\left(\frac{b}{2\Omega}\right)^{2n}(1-\cos 2\Omega t)^{2n}\\ &= a\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}\left(\frac{b}{\Omega}\right)^{2n}\sin^{4n+1}(\Omega t),\end{aligned} \tag{11}$$

and then:

$$\begin{aligned}z(t) &= z_0 + \int_0^t \dot{z}\,d\tau\\ &= z_0 + \int_0^t a\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}\left(\frac{b}{\Omega}\right)^{2n}\sin^{4n+1}(\Omega\tau)d\tau.\end{aligned} \tag{12}$$

The integral of this sine function raised to a power depending linearly on n can be expressed through a hypergeometric function:

$$\int_0^t \sin^{4n+1}(\Omega\tau)d\tau = \left[-\frac{\cos\Omega t}{\Omega}\cdot {}_2F_1\left(1/2,-2n;3/2;\cos^2\Omega t\right)\right]_0^t, \tag{13}$$

and finally, $z(t)$ has the following expression:

$$z(t) = z_0 - \left[\frac{a\cos(\Omega t)}{\Omega}\cdot\Gamma(b,t,0)\right]_0^t, \tag{14}$$

with Γ as defined in (10).

Thus, we have obtained the analytical expressions for the trajectories of the three state variables. Next, we will work out the values of a and b required for the state variables x and ϕ to converge to zero at $t = T/2$.

The stabilization of a nonholonomic system to a given configuration is known to be a difficult problem. The vehicle considered in this work (a wheeled unicycle mobile robot modeled in a Cartesian space representation) cannot be stabilized through smooth state-feedback control [39]. Considering the particular sinusoidal inputs we have chosen, it turns out that with the two degrees of freedom we have in the control loop (namely, the values a and b of the amplitudes of the sinusoidal velocities) we will only be able to control two of the robot's state variables simultaneously. Thus, we can make x and ϕ go to zero in $t = T/2$, but not z. Therefore, this latter variable is not controlled in the first step of our control scheme. The kinematics of the reduced system can then be expressed as follows:

$$\begin{cases}\dot{x} = -v\sin\phi\\ \dot{\phi} = \omega.\end{cases} \tag{15}$$

Thus, z can have any arbitrary final value, and this degree of freedom allows us to choose one among the infinite possible trajectories between the robot's initial and final configurations. A convenient way to do so is by setting the maximum absolute value of the orientation component,

$\phi_m > 0$, that the robot will achieve during its motion. This choice can give us a good sense of what the trajectory will be like (see the right part of Fig. 3). Once we have selected a value of ϕ_m, the value of b can be readily obtained by using (5) and choosing its sign correctly, as expressed by the following equation:

$$b = \lambda \cdot \Omega \cdot \phi_m, \tag{16}$$

where $\lambda = 1$ or $\lambda = -1$ depending on the initial configuration of the robot. Note that λ is used to set the appropriate sign of b for the desired motion. The way in which this parameter is selected will be detailed later in the paper. The functional variation of ω and the assumption that $\phi_0 = 0$ ensure that the final orientation will be $\phi(T/2) = 0$ regardless of the value of b. For a given b, i.e. for a given rotational velocity variation, we can see that there is only one value of a that will steer x from its initial value x_0 to 0 at the end of the motion interval. We can determine this value by enforcing the constraint: $x(t = T/2) = 0$ in equation (9), which yields:

$$a = \frac{x_0 \cdot \Omega}{2 \cdot \Gamma(b, 0, 1)}. \tag{17}$$

B. Feedback estimation of control parameters

The previous development assumes that no perturbations are present. Since in practice the system will be subjected to disturbances (e.g. measurement noise, motion drift and model errors), we propose to re-estimate the values of a and b online. During operation, at every given time t the state of the robot is estimated, and the amplitudes of the sinusoidal inputs are computed by enforcing the constraints that both x and ϕ must become 0 on $t = T/2$. For this purpose we use the previously obtained equations (5) and (9), which lead to the following results:

$$b(t) = \frac{\phi(t) \cdot \Omega}{\sin^2(\Omega t)} \tag{18}$$

$$a(t) = \frac{x(t) \cdot \Omega}{\cos(\Omega t) \cdot \Gamma(b(t), t, 1) + \Gamma(b(t), 0, 1)}. \tag{19}$$

Expressions (18) and (19) are valid for $0 < t < T/2$. The values of a and b at $t = 0$ can be obtained from the expressions given in section III-A for the two parameters. In addition, both a and b must be set to 0 at $t = T/2$. This way, the singularities in the expressions (18) and (19) are avoided. Still, we need to ensure that the velocity values will remain within reasonable limits; therefore, we will have to bound them by setting maximum values which can be a function of the initial amplitudes, i.e. $a_{max} = K_a \cdot a(0)$ and $b_{max} = K_b \cdot b(0)$, where $K_a > 1$, $K_b > 1$.

Note that, as already mentioned in section III, our control method can be used for any arbitrary initial orientation of the robot. For a given starting value $\phi_s \neq 0$, we set $\lambda = sign(\phi_s)$ in (16) in order to define the waveform properly. In the case $\phi_s = 0$, the sign of λ can be chosen arbitrarily. The starting time t_s of the input sinusoids can be obtained as:

$$t_s = \frac{T}{2\pi} \arcsin\sqrt{\frac{\phi_s}{\lambda \phi_m}}, \tag{20}$$

where the maximum absolute value of the orientation, ϕ_m, must be selected so that $\phi_m > |\phi_s|$ to ensure the computed t_s is in the correct range, $0 < t_s < T/2$. The sinusoidal inputs will now run from t_s to $T/2$ to leave the robot aligned with the target along the z axis. Since there are two possible solutions of (20) within the interval $(0, T/2)$, we choose the value of t_s that generates a more suitable trajectory (for example, the shorter one) considering the initial distance to the target axis (x_s) and the duration of the motion ($T/2 - t_s$). Note that our method can handle any starting value ϕ_s, even if the robot is initially reversed with respect to the desired final pose. Some example trajectories obtained using the sinusoidal input-based control law are shown in Fig. 3. The flexibility of the method is illustrated in the right part of the figure, where varying choices of the design parameter ϕ_m result in control paths of different characteristics. In addition, although the control method we present runs for one half-period of the sinusoidal wave, it is straightforward to extend it to generate trajectories spanning several half-periods. Just for illustration purposes, examples of this are shown in the center plot of Fig. 3.

IV. 1D TRIFOCAL TENSOR-BASED DEPTH CORRECTION

The first step of our control scheme corrects both the lateral position and the orientation of the robot. Thus, at the end of this stage the robot's state is $(0, z_2, 0)^T$. The correction of the depth coordinate is performed in the second step of the control employing the 1D trifocal tensor elements directly. In this section, subindexes are used to identify the cameras, being $(x_2, z_2, \phi_2)^T$ the current location and $(x_1, z_1, \phi_1)^T$ the location of the fixed reference view. Without loss of generality, $(x_3, z_3, \phi_3)^T = (0, 0, 0)^T$ is the location of the target view as defined in Fig. 2. Since in this particular situation the 1D trifocal tensor elements provide all the information necessary for the control task, we use them directly in the feedback loop, without estimating the state of the robot explicitly. In particular, the trifocal tensor elements when the state is $(0, z_2, 0)^T$ are as follows:

$$\mathbf{T_1} = \begin{bmatrix} -z_2 \sin\phi_1 & -t_{z1} - z_2 \cos\phi_1 \\ t_{z1} & 0 \end{bmatrix},$$
$$\mathbf{T_2} = \begin{bmatrix} z_2 \cos\phi_1 & t_{x1} + z_2 \sin\phi_1 \\ -t_{x1} & 0 \end{bmatrix}, \tag{21}$$

where the tensor has been broken down in matrices $\mathbf{T_1}$ (representing elements T_{1jk}) and $\mathbf{T_2}$ (representing elements T_{2jk}); and $t_{x1} = -x_1 \cos\phi_1 - z_1 \sin\phi_1$, $t_{z1} = x_1 \sin\phi_1 - z_1 \cos\phi_1$ express location relative to the first camera's local coordinate system [37].

From the information provided by the 1D trifocal tensor entries, we can obtain the distance to the target location, as follows:

$$\sqrt{{T_{111}}^2+{T_{211}}^2}=\sqrt{{z_2}^2{\sin\phi_1}^2+{z_2}^2{\cos\phi_1}^2}=|z_2|. \tag{22}$$

In order for the robot to move towards the target, the sign of the linear velocity v must be opposite to the sign of z_2, which can be obtained using the angular coordinate ψ: $sign(v)=-sign(cos\psi_2)$. We also need to take into account the fact that the 1D trifocal tensor computed from point correspondences is obtained up to scale. Therefore, if its elements are to be used for control tasks, they need to be normalized to a fixed scale. We achieve this by using the following normalization factor: $\sqrt{{T_{121}}^2+{T_{221}}^2}=\sqrt{{t_{z1}}^2+{t_{x1}}^2}=\sqrt{{x_1}^2+{z_1}^2}$. The linear velocity of the second step of our control, based on the normalized 1D trifocal tensor elements, is then as follows:

$$v=-sign(cos\psi_2)\cdot k_v\sqrt{\frac{{T_{111}}^2+{T_{211}}^2}{{T_{121}}^2+{T_{221}}^2}}\cdot f_r(t), \tag{23}$$

where k_v is a positive control gain. The motivation of the term $f_r(t)$ is to avoid the velocity jump at the start of the second control step due to the resultant linear velocity evolution of the robot proportional to the distance to the target, which is not desirable. For this reason, we use a weight function to make the linear velocity vary smoothly. In keeping with the sinusoidal-input framework that is the basis of our paper, we use for this purpose a sinusoidal function $f_r(t)$ defined as follows:

$$f_r(t)=\begin{cases}\sin(\pi t/(2t_r)) & , \quad t\le t_r\\ 1 & , \quad t>t_r.\end{cases} \tag{24}$$

Notice that $f_r(t)$ is simply a wave that starts as a sine running for a quarter of a period (i.e. a function rising smoothly from an initial value of zero to a value of one at $t=t_r$) and that maintains its value (i.e. stays at one) thereafter. Thus, this weighting function only operates at the beginning of the second control step. Although the needed motion is straight along the z axis, in order to compensate for noise or drift, we perform orientation control using the following angular velocity:

$$w=-k_w\cdot\phi_2, \tag{25}$$

where k_w is a positive control gain.

V. State estimation through the 1D trifocal tensor

In order to perform the feedback control defined in section III, we need to estimate the state of the robot along its motion. We will only use omnidirectional visual information for this purpose. The reader is referred to our previous work [38] where we provide a thorough description of the procedure through which the relative angles between the locations of three views can be extracted from the 1D trifocal tensor. The tensor is computed linearly from points matched across the three views, which in our case are the target view, the current view and a reference view (which can be the initial image). Once we know the angles between the views, it is straightforward to work out the angular coordinates α and ϕ of the robot's state representation in polar form.

However, since the trifocal tensor is defined up to an unknown scale, distance estimations cannot be directly extracted from it. In order to compute distance parameters, we will initially need to use the derivatives of the known angles. These are rather noisy estimates; in order to avoid using them during control, we will take advantage of the fact that we have three views forming a triangle and a fixed reference distance between two of them.

We define d as the fixed distance between the initial (or reference, in a more general sense) position and the target position (see Fig. 2). d is related to the state variable ρ and the angular sectors γ and β through the law of sines:

$$\rho=d\cdot\sin(\gamma)/\sin(\beta). \tag{26}$$

This expression can be used if $\beta\neq 0$, which will be true as long as the robot does not cross the line joining the reference and goal locations during its motion. Before the control task is started, the robot performs an initial motion whose objective is to compute d. This is done by using (26) and (2), leading to the following expression:

$$d=\frac{\sin\beta}{\sin\gamma}\cdot\frac{\dot\psi}{v\sin\alpha}. \tag{27}$$

We can estimate the derivative of ψ as: $\widehat{\dot\psi}=(\psi(t+\Delta t)-\psi(t))/\Delta t$. The initial motion executed prior to the control task must be such that the robot does not cross the line joining the reference and goal positions, thus ensuring $\beta\neq 0$ and $\gamma\neq 0$. In addition, the robot must not move in the direction towards the target, in order to ensure $\dot\psi\neq 0$ and $\sin\alpha\neq 0$. It is straightforward to generate a motion fulfilling these conditions, since from the computation of the trifocal tensor, we know the angles between the three locations at all times.

During the initial motion we can obtain a number of estimates of d using (27) and compute their average, until we achieve a sufficiently stable value. Then, during the control phase, ρ is computed using (26), and x, the position variable we use in the feedback control loop, is obtained as $x=-\rho\sin\psi$.

VI. Stability analysis

The stability of the system under the proposed control method is analyzed in the following.

Proposition 1: The sinusoidal input-based control law (3), (4) with (18) and (19) achieves global asymptotic stabilization of the system (15).

Proof:

We will use Lyapunov analysis [40] to assess the stability properties of the system when the sinusoidal input-based control law is used.

We define the following positive definite, radially unbounded Lyapunov-candidate function: $V=\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$, where

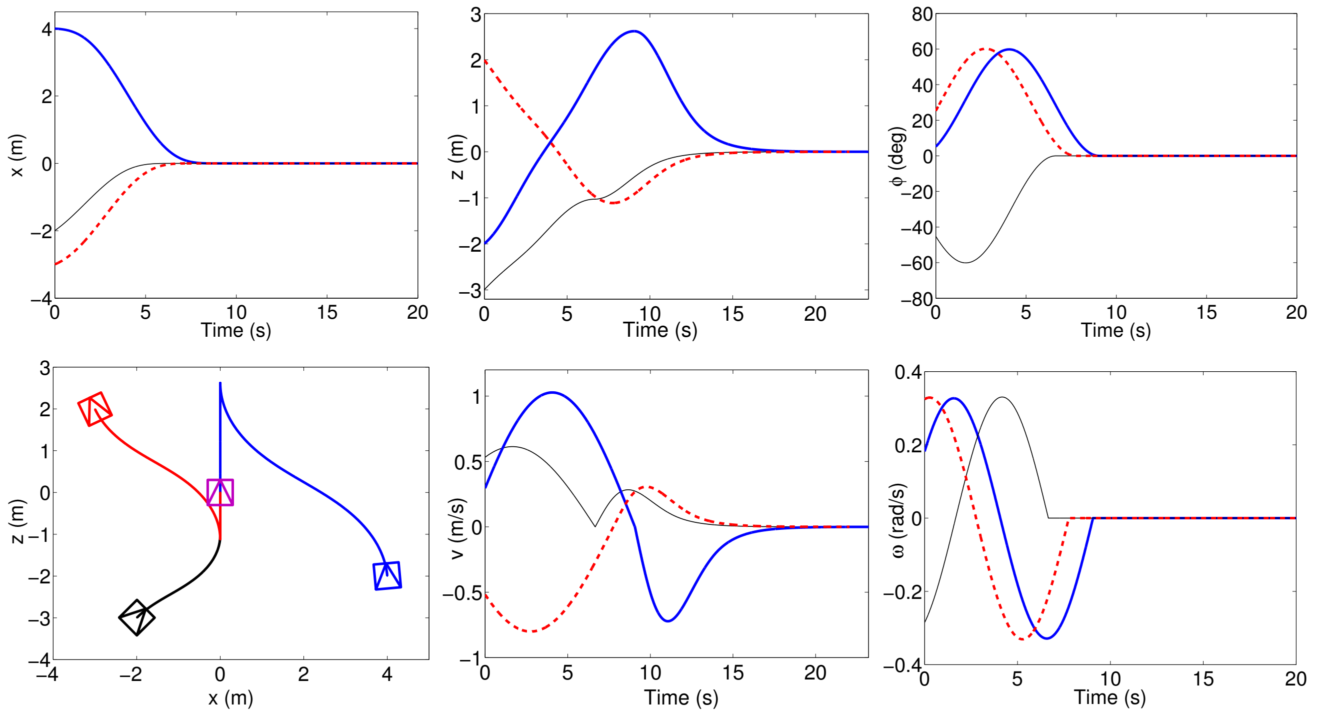

Fig. 4. Three sample robot trajectories for the sinusoidal input-based control, from starting locations $(4, -2, 5°)^T$, $(-3, 2, 25°)^T$ and $(-2, -3, -45°)^T$. The evolutions of the state variables x (left), z (center) and ϕ (right) are displayed in the top row. The second row shows the robot paths (left) for each trajectory, the linear velocity (center) and the angular velocity (right).

$\mathbf{x} = (x, \phi)^T$ is the reduced state vector of the system and $\mathbf{P}$ is the following positive definite matrix:

$$\mathbf{P} = \begin{bmatrix} P_x & 0 \\ 0 & 1 \end{bmatrix}, \tag{28}$$

with $P_x > 0$. The derivative of V is as follows: $\dot{V} = \frac{1}{2}(\dot{\mathbf{x}}^T\mathbf{P}\mathbf{x} + \mathbf{x}^T\mathbf{P}\dot{\mathbf{x}}) = P_x x\dot{x} + \phi\dot{\phi}$. Under the feedback control law of the first step of our method, based on input sinusoids of amplitudes computed using (18) and (19), we have:

$$\dot{V} = -P_x \cdot x^2 \cdot \Omega \cdot F(\phi, t) + 2\phi^2\Omega\cot(\Omega t), \tag{29}$$

where:

$$F(\phi, t) = \frac{\sin\Omega t \cdot \sin\phi}{\cos(\Omega t)\cdot\Gamma(b(t), t, 1) + \Gamma(b(t), 0, 1)}. \tag{30}$$

The asymptotic stability condition, $\dot{V} < 0$, is assured when:

$$P_x > \frac{2\phi^2\cot(\Omega t)}{x^2 F(\phi, t)}. \tag{31}$$

If no perturbations are present, the values of $a(t)$ and $b(t)$ obtained using (18) and (19) will remain constant throughout the operation period. Taking this into account, we have that ϕ varies according to (5), and x follows the evolution defined in (9), which is equivalent to:

$$x(t) = \frac{x_0}{2\Gamma(b, 0, 1)} \cdot (\Gamma(b, 0, 1) + \cos\Omega t \cdot \Gamma(b, t, 1)). \tag{32}$$

Substitution of $F(\phi, t)$, x and ϕ in (31) finally yields the following expression:

$$P_x > \frac{8b^2\Gamma^2(b, 0, 1)\sin^2\Omega t\cos\Omega t}{x_0^2\Omega^2\sin(\frac{b}{\Omega}\sin^2\Omega t)(\Gamma(b, 0, 1) + \cos\Omega t \cdot \Gamma(b, t, 1))}. \tag{33}$$

It can be demonstrated that the right part of (33) has a global maximum in the time interval $0 < t < T/2$ when $t \to 0$. In particular, we have that:

$$\begin{aligned} &\lim_{t\to 0} \frac{8b^2\Gamma^2(b, 0, 1)\sin^2\Omega t\cos\Omega t}{x_0^2\Omega^2\sin(\frac{b}{\Omega}\sin^2\Omega t)(\Gamma(b, 0, 1) + \cos\Omega t \cdot \Gamma(b, t, 1))} \\ &= \lim_{t\to 0} \frac{\sin^2\Omega t}{\sin(\frac{b}{\Omega}\sin^2\Omega t)} \lim_{t\to 0} \frac{8b^2\Gamma^2(b, 0, 1)/x_0^2\Omega^2}{\Gamma(b, 0, 1) + \cos\Omega t \cdot \Gamma(b, t, 1)} \\ &= \frac{4b\Gamma(b, 0, 1)}{x_0^2\Omega}. \end{aligned} \tag{34}$$

Therefore, there exists an upper bound to the right part of (33), and P_x can be defined as required. Thus, the system under the proposed sinusoidal control law is asymptotically stable.

$\blacksquare$

Proposition 2: The 1D trifocal tensor-based control law (23) achieves global stabilization of the system, while the rotational velocity (25) achieves lateral drift compensation assuming that proposition 1 is accomplished.

Proof:

In order to analyze the stability of the second step of our method, we define the following positive definite, radially unbounded Lyapunov-candidate function: $V = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$, where $\mathbf{x} = (z, \phi)^T$ is the reduced state vector of the system and:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{35}$$

The derivative of V is as follows:

$$\begin{aligned} \dot{V} &= zv\cos\phi + \phi w \\ &= k_v z \frac{-z}{\sqrt{{x_1}^2 + {z_1}^2}} f_r(t)\cos\phi - k_w \phi^2. \end{aligned} \tag{36}$$

Note that the angular value ϕ is maintained at zero with finite time convergence by the control action (25), keeping the alignment toward the goal. Under the assumption that proposition 1 is verified, it is straightforward to see that the two terms of equation (36) are negative (since ϕ has small values and therefore $\cos\phi$ is positive, and $f_r(t)$ is also positive). Consequently, the system under the control law of the second step is asymptotically stable.

■

VII. Experimental results

Several simulations and experiments with a real robot have been carried out to illustrate the performance of our approach.

A. Simulations

Next, we present some simulation results for our proposed visual control method. From the points projected and matched between three virtual cameras, the trifocal tensor is computed and the relative angles between the views are estimated. The state variables of the system are subsequently obtained from this information and used in the feedback control. Figure 4 displays three sample trajectories, along with the velocities used. The maximum absolute value of the orientation (ϕ_m) was set to 60°. Note that the generated trajectories of the control are remarkably smooth. The robot is left aligned with the target at the end of this phase, while in the second control stage the depth is corrected following a straight-line path.

The effects of adding Gaussian noise to the angles of the projection of points in each view are illustrated in Fig. 5. The final position and orientation errors in the presence of noise of both steps of the control method are shown. The first step appears to be more sensitive to noise. Still, in both cases the final errors have small values.

Simulations with motion drift are illustrated in Fig. 6. The added drift is proportional to the linear and angular velocities of the robot. It can be seen that the control method is capable of compensating the drift so that the system reaches the desired state at the end of the control period. In order to illustrate this effect, only the sinusoidal part of the control (i.e. the first step) is shown. Note that the magnitude of the added drift is considerably high, so as to illustrate the potential of the approach to overcome it.

In order to evaluate the characteristics of our approach, we provide next a comparison with previous control methods using sinusoids. We illustrate this discussion with a simulation example, displayed in Fig. 7. The method by Murray et al. [8] employs multiple steps, and in its sinusoidal-motion part generates a path that looks similar to ours. However, the maximum rotations in absolute value occur at points where the linear velocity is zero. This is not a desirable behavior from the viewpoint of feasibility, smoothness and safety. In contrast, our method generates the rotations of maximum magnitude at intermediate linear velocities. The method due to Tilbury et al. [15] performs the control in one single step. Its resulting trajectories are hard to predict, since the specific values of the parameters can change the input waveforms and required manoeuvres considerably. As a consequence, the motion can be unsatisfactory for a nonholonomic vehicle and inefficient. Lastly, the method proposed by Teel et al. [16] is a feedback control approach which achieves asymptotic convergence to the goal configuration. The trajectories generated by this approach are not very efficient, and a high number of manoeuvres are required. Its convergence is slow, although an alternative method by M'Closkey et al. [17] generates trajectories of a similar type, but achieving exponential convergence.

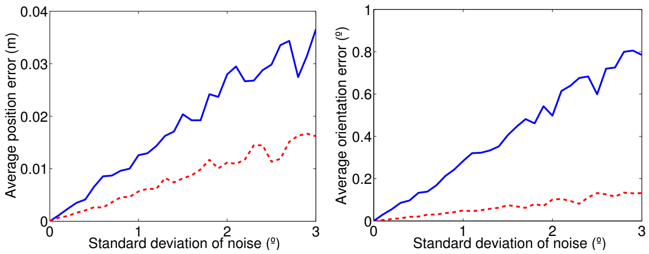

Fig. 5. Results from simulations with added Gaussian noise. Solid lines correspond to the first step of the control; dashed lines correspond to the second. The average position (left) and orientation (right) errors at the end of each step are displayed.

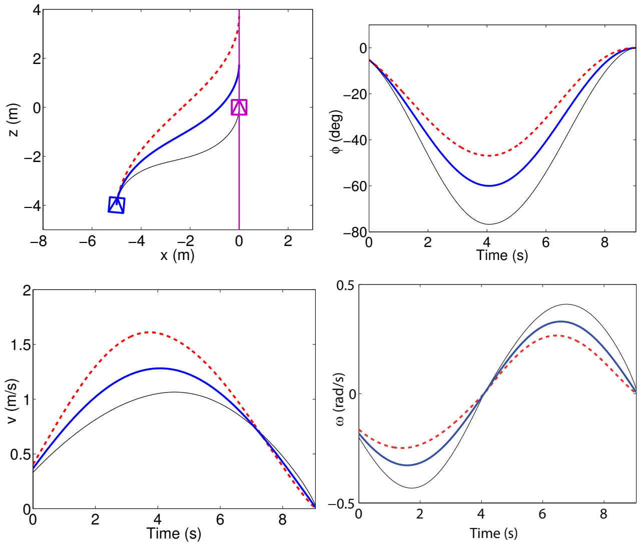

Fig. 6. Simulation results with motion drift in the sinusoidal input-based part of the control, from the initial location $(-5, -4, -5°)^T$. A driftless simulation is shown in a thick solid line. The results obtained adding +10% and -10% drift to both the linear and angular velocities of the robot are displayed with a dashed line and a thin solid line, respectively.

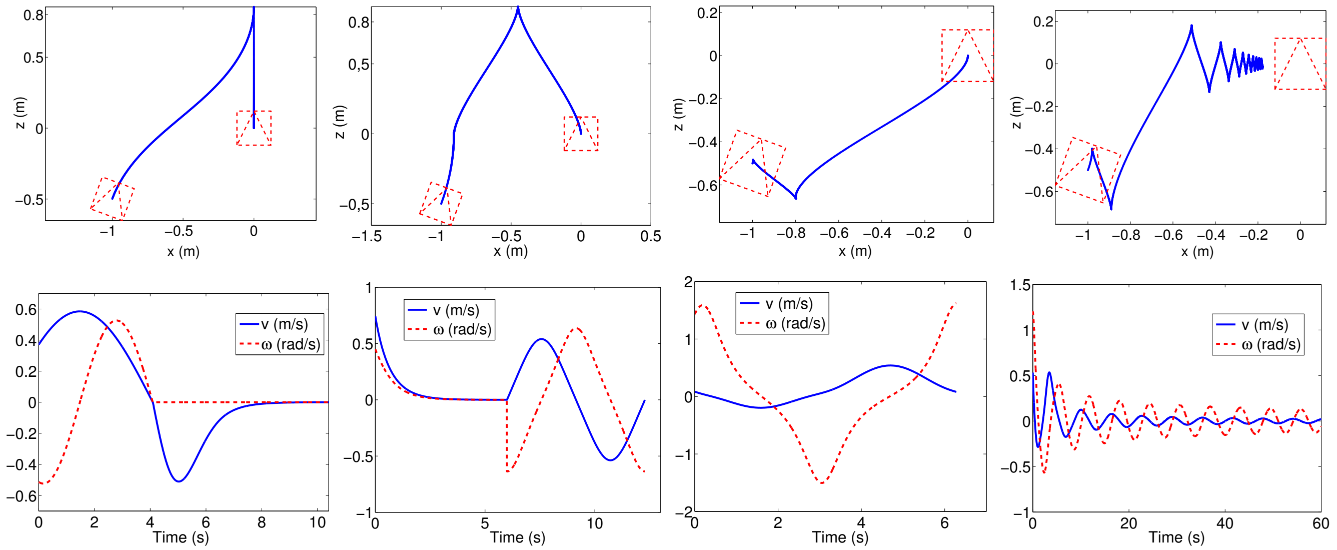

Fig. 7. Comparison of sinusoidal-input based control methods. From left to right, the four methods are: the approach presented in this paper, the method by Murray et al. [8], the method by Tilbury et al. [15], and the method due to Teel et al. [16]. The top row shows the control paths from starting location $(-1, -0.5, -20°)^T$ for the four approaches. The bottom row displays the velocities associated to the paths on top.

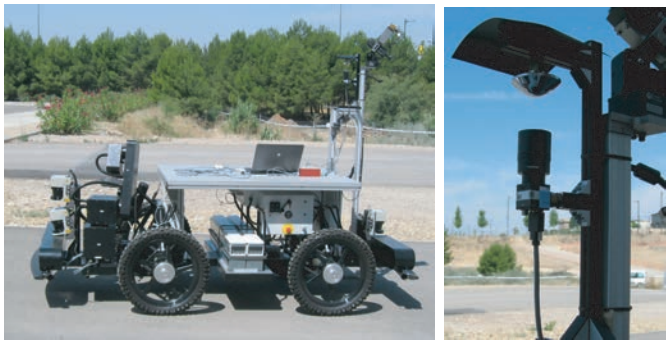

Fig. 8. Mobile robot and omnidirectional camera used in the experiments.

Contrary to our method, the velocities applied on the vehicle in the three other approaches are not actually sinusoids, due to their use of a chained form system representation. Still, the generated inputs are usually very close to sinusoidal functions. The other methods cannot be configured to generate different trajectories with desired characteristics, which is a feature of our approach. In addition, our technique requires fewer manoeuvres (only one) than the other methods in the comparison. Overall, we believe that our approach has the advantages of providing a greater knowledge of the trajectory and a motion strategy that is flexible and particularly suited for nonholonomic robots.

B. *Experiments on a real robot*

We tested our control approach on a mobile robotic platform from Robosoft, shown in Fig. 8. This four-wheeled robot is equipped with various vision, laser and GPS sensors, and also features a manipulator arm. Its maximum linear speed is 2.5 m/s, and its maximum steering angle is 0.4 radians. The front and rear wheels of this robot can be steered independently, which allows different steering modes. For our experiments, we used it as a standard car-like vehicle with rear-wheel driving. The vision system we employed was an omnidirectional camera (shown in Fig. 8) mounted on top of the robot. It consists of a Point Grey camera and a Neovision H3S mirror. The size of the acquired images was 1024 × 768. An external laptop computer on board was used to capture the images at a constant rate (7.5 frames per second) and issue the velocity commands to the robot. The experiments were performed outdoors, next to a parking lot in our university campus.

In order to test the method with real data, we generated several trajectories and computed their associated sinusoidal velocities according to our control approach. At the end of each of these trajectories, we planned a straight line motion that would correspond to the second step of our method. Since our approach was developed for a unicycle robot, we transformed our angular velocity commands to the equivalent steering angle commands for a car-like robot. This conversion did not cause any issues, due to our control method's properties of smoothness, feasibility and flexibility.

We describe next the results from one of these trajectories. The sinusoidal part of it lasted 30 seconds, while the straight-line motion lasted approximately 10 seconds. Figure 10 shows a sequence of images acquired in this trajectory and pictures of the robot at the instants when the images were captured, in order to illustrate its motion. Further illustration of the trajectory followed by the robot during the control execution and the captured image sequence can be found in the provided **video attachment.** We used the SIFT keypoint extractor [41] to find matches between sets of three images: the current image along the trajectory, the goal image, and a reference image (e.g. the initial one). From every set of three view correspondences, we computed the 1D trifocal tensor using the RANSAC robust estimation algorithm. This procedure filters out wrong feature matches,

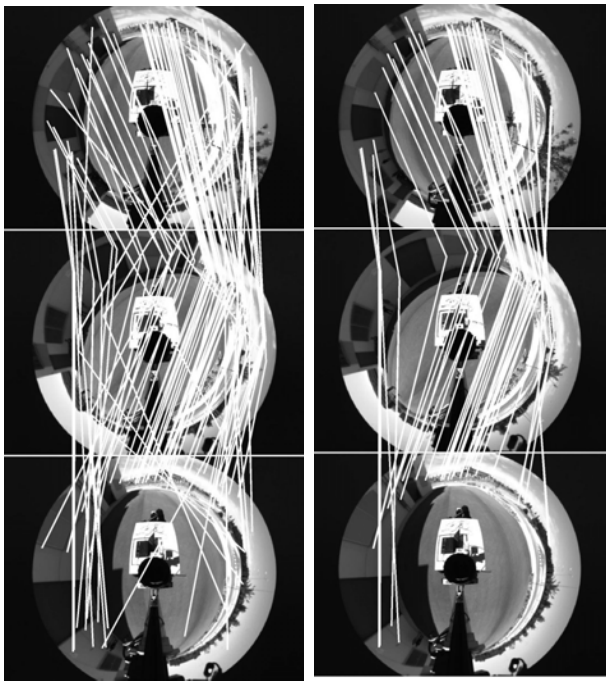

Fig. 9. Example of a trio of images with their putative SIFT correspondences joined by lines (left). Feature matches remaining after the robust computation of the 1D trifocal tensor (right).

thereby enhancing the robustness of the control performance. Figure 9 shows an example of the outlier rejection achieved in this experiment through the use of the 1D trifocal tensor model.

We then used the 1D trifocal tensor to compute the state of the robot, following the method explained in section V. Figure 11 shows the estimated $x - z$ path superimposed on the robot's odometry. Wrong estimations occurring at some iterations (due to the lack of a sufficient amount of correct three-view matches) were robustly detected and discarded. This was achieved by imposing geometric consistency constraints on the angles obtained in each iteration between the three views (i.e. by checking that the three views form a triangle). We refer to our previous work [38] for more details on how this is carried out. This procedure, together with the rejection of wrong feature matches achieved by using the trifocal tensor model, makes it possible to obtain robust state estimations. As can be seen, the results of the computation of the position follow closely the trajectory estimated through the odometry.

The computations of the control inputs associated to our approach for both parts of the trajectory are also illustrated in Fig. 11. Again, wrong estimations along the complete path were automatically detected and rejected. For the sinusoidal-based part of the motion, we computed the associated signals at every time instant using the feedback estimation of the control parameters described in section III-B. The variations of the linear velocity were smoothed out in the final part of the sinusoidal control step through a weighted average filter. We used a coarse initial estimation (which is refined along the motion) of the distance d defined in section V, for simplicity. As can be seen, the velocities are close to the theoretically expected waveforms.

For the second part of the motion, i.e. the straight-line path, we computed the controls associated to our method directly from the elements of the 1D trifocal tensor, as described in section IV. The angular velocity would ideally be expected to stay at zero and, since the robot moved at constant speed in this segment and the image acquisition rate was also constant, the translational velocity v would ideally be expected to decay linearly. The last sections of the v and ω plots in Fig. 11 show that the results of the computations for the second part of the control method were correct. Notice that, for simplicity reasons, we did not employ the sinusoidal weight function of section IV in these computations, since the vehicle was not subjected to a velocity jump of relevant magnitude at the beginning of the second control step.

Figure 12 shows the representation of the evolution of the absolute control error in this experiment for each of the three state variables, in order to illustrate the performance of the control method and its convergence. It also displays the path actually followed by the vehicle, as measured by the robot's odometry, along with the ground truth given by the desired path and final location. The final absolute pose errors of the experiment were: 0.14 m. in x (2.3 % error), 0.26 m. in z (2 % error) and 2.4 deg. in ϕ (4.7 % error). Thus, the accuracy of the control method in this experiment was satisfactory, as the errors were maintained in relatively small values.

VIII. Discussion

In this section we discuss a number of practical aspects related to the proposed control method, as well as some of its properties and possible applications.

A. Practical considerations

The proposed control approach can be readily used with any sensor that provides bearing measurements (e.g. a laser scanner, a conventional camera, or an omnidirectional one). Employing an omnidirectional vision sensor provides several benefits: thanks to their wide field of view, these sensors provide a high amount of visual information which, in addition, is very precise in the angular coordinate. Having the complete view surrounding the vehicle allows our technique to be more robust and flexible (in terms of the admissible paths and starting poses) than it would be if a conventional camera was used.

Although our control method was derived considering the unicycle kinematic model, we believe that it can be easily extended to more general classes of nonholonomic vehicles, due to the fact that the motions are designed to avoid sharp changes. This is illustrated by the conversion from a unicycle to a car-like kinematic model carried out in the experiment with the real robot. This transformation was done directly, computing the value of the steering angle (s) for the rear-wheel driving car-like vehicle as: $s = \arctan(\omega \cdot L/v)$, where ω is the angular velocity, v is the linear velocity and L is the distance between the front and rear wheels of the vehicle. The two representations are equivalent for a car-like vehicle

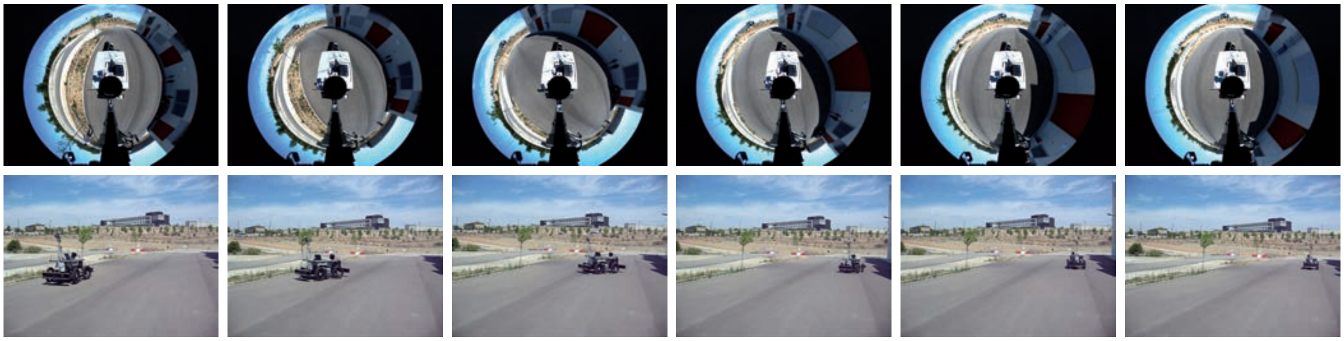

Fig. 10. Sequence of images acquired by the robot in the trajectory used in the experiments. The top row shows the images, the bottom row shows the location of the robot when the corresponding image was captured.

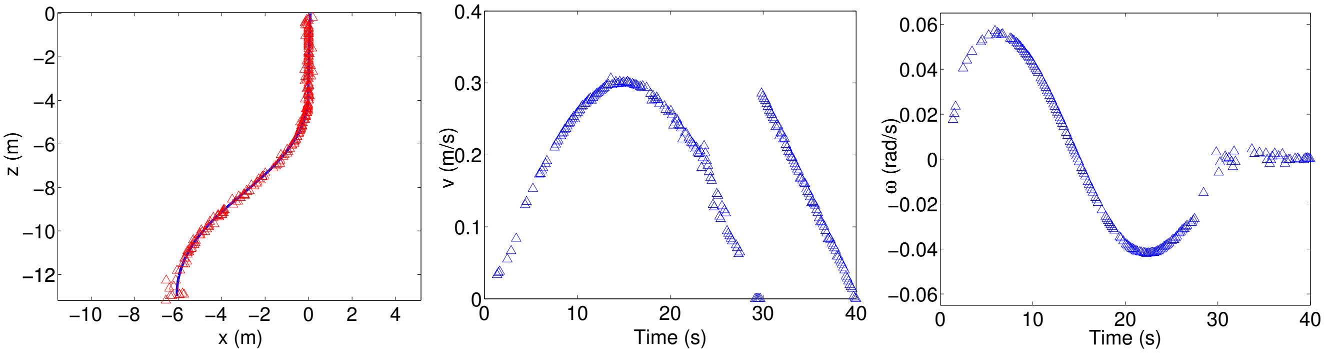

Fig. 11. Results from the experiment with the real robot. Estimation of the position (marked with triangles) along the trajectory superimposed on the odometry, which is shown in a solid line (left). Computed linear velocity (center) and angular velocity (right).

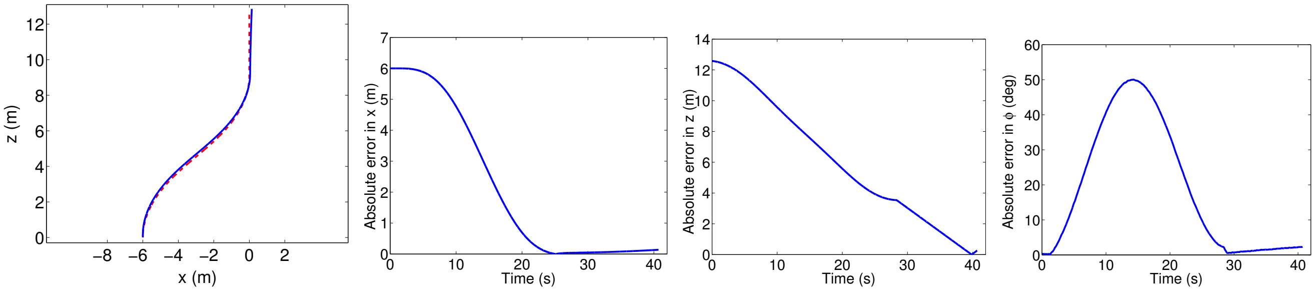

Fig. 12. Results from the experiment with the real robot. From left to right: representation of the robot's trajectory measured by the odometry (solid line) and the desired trajectory (dashed line); evolution of the absolute control error in the state variables x, z and ϕ.

[9], [29]. The fact that the velocities of our approach vary smoothly, means that the corresponding steering angle s also varies smoothly, which facilitates the method's successful implementation on the real car-like robot.

As can be seen in the omnidirectional images captured in the experiment with the real robot, the camera was not situated in the geometrical center of the vehicle. This practical issue has an effect on the trajectories performed by the robot, which turn out slightly different from the theoretical ones. Still, the convergence behavior of the control is not affected. It can be shown that, as long as the camera center is situated in the longitudinal axis of the vehicle, the equilibrium conditions of both the first and second steps of the control are achieved equivalently either considering a camera-centered frame or a vehicle-centered frame. We implemented our control method in the experiment with the real robot assuming the camera was approximately in the longitudinal axis of the vehicle. The experimental results confirm that the error associated to this assumption was insignificant in practice.

As is common practice in control, we provide a proof of the stability of the system when no perturbations are present (section VI), assuming that in practice these unknown disturbances would be bounded and within moderate limits. The behavior of the system with non-ideal conditions is illustrated in the simulations presented in section VII-A. They show that our method can be robust with respect to measurement noise and motion drift. In addition, the experiment on the real robot in section VII-B illustrates the feasibility of the control approach in a real scenario.

In the second step of our method (the depth correction phase) we assume that the orientation control corrects the angular error with finite-time convergence, as stated in the proof of proposition 2. Assuming as well that proposition 1 is accomplished (i.e. there is no lateral error at the start of the second step) and given that the orientation is maintained at zero with our controller, the lateral error will also stay at zero. In a practical situation, this means that the magnitude of the lateral drift generated by our method is expected to be small, bounded and not relevant.

B. Method properties and possible uses

By looking at the characteristics of its velocity inputs and of the trajectories it generates, we can draw some qualitative conclusions regarding the properties of our control approach. In general, it is clear that avoiding sharp changes in the motion of a vehicle is a desirable property. Sinusoidal waves are very often found in nature, and they are characterized by their avoidance of sudden changes. When the velocity commands passed to a mobile robot are pure sinusoids, it is ensured that the transitions will not be sharp. This is the reason why we chose to use sinusoidal inputs. Furthermore, the coupled evolution of the linear and angular velocities is also designed to generate a smooth vehicle motion. Our method provides great knowledge and control on the trajectory. In addition, it is flexible, since motions of different shapes and curvatures are possible by selecting the parameters of the input signals appropriately. Thus, it can be designed to fit different criteria (e.g. minimize the length of the path or the required maximum rotation) depending on the situation.

In addition, based on the shape of the trajectories generated by our approach, we claim that it is well suited for such tasks as obstacle avoidance and parking manoeuvres. The robot paths are clearly reminiscent of a typical parking manoeuvre, and the combination of a curved path in the first control step and a straight one in the second is suitable for the task of reaching a goal while avoiding an obstacle placed between the initial and target locations. In addition, the knowledge of the trajectories and the flexibility provided by our method make it possible to design a trajectory adapted to the particular conditions of the obstacle avoidance or parking scenario. If the trajectory has to be modified over time, it is indeed possible to reset the control execution and start a new, replanned trajectory at any time, since our method can work for any starting configuration, as discussed in section III. Our technique could be readily used to avoid static obstacles. However, if general dynamic obstacles were considered, an additional obstacle avoidance method would have to be used along with our control approach.

IX. Conclusion

We have proposed a sinusoidal input-based method to perform visual control of a mobile robot. From the definition of the desired sinusoidal-varying velocities, we have derived analytical expressions for the evolution of the system and proposed a new control law based on these expressions. We have also presented a method to estimate the robot's state using the geometric information encapsulated in the 1D trifocal tensor. The stability of the system under the proposed control has been analyzed. We have shown, both theoretically and through experiments, that our approach generates smooth and flexible trajectories suited for a vehicle with nonholonomic constraints. Considering the characteristics of these trajectories, we believe that our technique can be adequate for parking manoeuvres and obstacle avoidance uses.

References

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[2] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[3] ——, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[4] S. Cho and J. Lee, "Localization of a high-speed mobile robot using global features," *Robotica*, vol. 29, no. 5, pp. 757–765, 2011.

[5] S. Baker and S. K. Nayar, "A theory of single-viewpoint catadioptric image formation," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 175–196, 1999.

[6] M. A. Salichs and L. Moreno, "Navigation of mobile robots: open questions," *Robotica*, vol. 18, no. 3, pp. 227–234, 2000.

[7] M. Defoort, J. Palos, A. Kokosy, T. Floquet, and W. Perruquetti, "Performance-based reactive navigation for non-holonomic mobile robots," *Robotica*, vol. 27, no. 2, pp. 281–290, 2009.

[8] R. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 700–716, 1993.

[9] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion, planning and control*, ser. Lecture Notes in Control and Information Sciences 229, J.-P. Laumond, Ed. Springer, 1998.

[10] P. Salaris, D. Fontanelli, L. Pallottino, and A. Bicchi, "Shortest paths for a robot with nonholonomic and field-of-view constraints," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 269 – 281, 2010.

[11] X. Zhang, Y. Fang, and X. Liu, "Motion-estimation-based visual servoing of nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1167 – 1175, 2011.

[12] P. Coelho and U. Nunes, "Path-following control of mobile robots in presence of uncertainties." *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 252–261, 2005.

[13] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, no. 07, pp. 1037–1048, 2011.

[14] A. Rosales, G. Scaglia, V. A. Mut, and F. di Sciascio, "Trajectory tracking of mobile robots in dynamic environments - a linear algebra approach." *Robotica*, vol. 27, no. 7, pp. 981–997, 2009.

[15] D. Tilbury, R. Murray, and S. Sastry, "Trajectory generation for the n-trailer problem using Goursat normal form," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 802–819, 1993.

[16] A. Teel, R. Murray, and G. Walsh, "Nonholonomic control systems: From steering to stabilization with sinusoids," in *IEEE Conference on Decision and Control*, 1992, pp. 1603–1609.

[17] R. T. M'Closkey and R. M. Murray, "Exponential stabilization of driftless nonlinear control systems using homogeneous feedback," *IEEE Trans. on Automatic Control*, vol. 42, no. 5, pp. 614–628, 1995.

[18] C. Sagues and J. J. Guerrero, "Motion and structure for vision-based navigation," *Robotica*, vol. 17, no. 4, pp. 355–364, 1999.

[19] J. Chen, W. Dixon, M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 407–416, April 2006.

[20] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira, "Vision-based control for car platooning using homography decomposition," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 2161–2166.

[21] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 2008.

[22] A. A. Argyros, K. E. Bekris, S. C. Orphanoudakis, and L. E. Kavraki, "Robot homing by exploiting panoramic vision," *Autonomous Robots*, vol. 19, no. 1, pp. 7–25, 2005.

[23] R. Möller, A. Vardy, S. Kreft, and S. Ruwisch, “Visual homing in environments with anisotropic landmark distribution,” *Autonomous Robots*, vol. 23, no. 3, pp. 231–245, 2007.

[24] W. Stürzl and H. A. Mallot, “Efficient visual homing based on Fourier transformed panoramic images.” *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 300–313, 2006.

[25] R. Möller, M. Krzykawski, and L. Gerstmayr, “Three 2D-warping schemes for visual robot navigation,” *Autonomous Robots*, vol. 29, no. 3-4, pp. 253–291, 2010.

[26] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, “Scale-only visual homing from an omnidirectional camera,” in *IEEE International Conference on Robotics and Automation*, 2012, pp. 3944 –3949.

[27] S.-E. Yu and D. Kim, “Landmark vectors with quantized distance information for homing navigation,” *Adaptive Behavior*, vol. 19, no. 2, pp. 121–141, 2011.

[28] Y. Fu, T.-R. Hsiang, and S.-L. Chung, “Multi-waypoint visual homing in piecewise linear trajectory,” *Robotica, Available on CJO*, 2012, doi=10.1017/S0263574712000434.

[29] K. Usher, P. R. Ridley, and P. I. Corke, “Visual servoing of a car-like vehicle - an application of omnidirectional vision,” in *IEEE International Conference on Robotics and Automation*, 2003, pp. 4288–4293.

[30] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[31] R. Basri, E. Rivlin, and I. Shimshoni, “Visual homing: Surfing on the epipoles,” *International Journal of Computer Vision*, vol. 33, no. 2, pp. 117–137, 1999.

[32] G. López-Nicolás, C. Sagüés, J. J. Guerrero, D. Kragic, and P. Jensfelt, “Switching visual control based on epipoles for mobile robots,” *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 592–603, 2008.

[33] G. López-Nicolás, J. J. Guerrero, and C. Sagüés, “Visual control through the trifocal tensor for nonholonomic robots,” *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 216 – 226, 2010.

[34] G. López-Nicolás, H. M. Becerra, M. Aranda, and C. Sagüés, “Visual navigation by means of three view geometry,” in *Robot 2011 Workshop, Sevilla, Spain*, 2011.

[35] A. Shademan and M. Jägersand, “Three-view uncalibrated visual servoing,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 6234 –6239.

[36] O. Faugeras, L. Quan, and P. Sturm, “Self-calibration of a 1D projective camera and its application to the self-calibration of a 2D projective camera,” *IEEE Transac. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1179–1185, 2000.

[37] H. M. Becerra, G. López-Nicolás, and C. Sagüés, “Omnidirectional visual control of mobile robots based on the 1D trifocal tensor,” *Robotics and Autonomous Systems*, vol. 58, no. 6, pp. 796–808, 2010.

[38] M. Aranda, G. López-Nicolás, and C. Sagüés, “Omnidirectional visual homing using the 1D trifocal tensor,” in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2444–2450.

[39] C. Samson and K. Ait-Abderrahim, “Feedback control of a non-holonomic wheeled cart in cartesian space,” in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1136–1141.

[40] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991, pp. 57–68.

[41] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.