

DEPARTAMENTO DE INFORMÁTICA E INGENIERÍA DE SISTEMAS

CENTRO POLITÉCNICO SUPERIOR

UNIVERSIDAD DE ZARAGOZA

**CONTROL DE ROBOTS MÓVILES MEDIANTE VISIÓN  
OMNIDIRECCIONAL UTILIZANDO LA GEOMETRÍA  
DE TRES VISTAS**

TRABAJO FIN DE MÁSTER DEL  
MÁSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA

PROGRAMA OFICIAL DE POSGRADO EN INGENIERÍA  
INFORMÁTICA

AUTOR: MIGUEL ARANDA CALLEJA

DIRECTORES: CARLOS SAGÜÉS BLÁZQUIZ Y GONZALO LÓPEZ  
NICOLÁS

CURSO ACADÉMICO 2009/10

Septiembre 2010

# Control de robots móviles mediante visión omnidireccional utilizando la geometría de tres vistas

## Resumen

Este trabajo trata acerca del control visual de robot móviles. Dentro de este campo tan amplio de investigación existen dos elementos a los que prestaremos especial atención: la visión omnidireccional y los modelos geométricos multi-vista. Las cámaras omnidireccionales proporcionan información angular muy precisa, aunque presentan un grado de distorsión significativo en dirección radial. Su cualidad de poseer un amplio campo de visión hace que dichas cámaras sean apropiadas para tareas de navegación robótica. Por otro lado, el uso de los modelos geométricos que relacionan distintas vistas de una escena permite rechazar emparejamientos erróneos de características visuales entre imágenes, y de este modo robustecer el proceso de control mediante visión.

Nuestro trabajo presenta dos técnicas de control visual para ser usadas por un robot moviéndose en el plano del suelo. En primer lugar, proponemos un nuevo método para homing visual, que emplea la información dada por un conjunto de imágenes de referencia adquiridas previamente en el entorno, y las imágenes que toma el robot a lo largo de su movimiento. Con el objeto de sacar partido de las cualidades de la visión omnidireccional, nuestro método de homing es puramente angular, y no emplea información alguna sobre distancia. Esta característica, unida al hecho de que el movimiento se realiza en un plano, motiva el empleo del modelo geométrico dado por el tensor trifocal 1D. En particular, las restricciones geométricas impuestas por dicho tensor, que puede ser calculado a partir de correspondencias de puntos entre tres imágenes, mejoran la robustez del control en presencia de errores de emparejamiento. El interés de nuestra propuesta reside en que el método de control empleado calcula las velocidades del robot a partir de información únicamente angular, siendo ésta muy precisa en las cámaras omnidireccionales. Además, presentamos un procedimiento que calcula las relaciones angulares entre las vistas disponibles de manera indirecta, sin necesidad de que haya información visual compartida entre todas ellas.

La técnica descrita se puede clasificar como basada en imagen (image-based), dado que no precisa estimar la localización ni utiliza información 3D. El robot converge a la posición objetivo sin conocer la información métrica sobre la trayectoria seguida. Para algunas aplicaciones, como la evitación de obstáculos, puede ser necesario disponer de mayor información sobre el movimiento 3D realizado. Con esta idea en mente, presentamos un nuevo método de control visual basado en entradas sinusoidales. Las sinusoides son funciones con propiedades matemáticas bien conocidas y de variación suave, lo cual las hace adecuadas para su empleo en maniobras de aparcamiento de vehículos. A partir de las velocidades de variación sinusoidal que definimos en nuestro diseño, obtenemos las expresiones analíticas de la evolución de las variables de estado del robot. Además, basándonos en dichas expresiones, proponemos un método de control mediante realimentación del estado. La estimación del estado del robot se obtiene a partir del tensor trifocal 1D calculado entre la vista objetivo, la vista inicial y la vista actual del robot. Mediante este control sinusoidal, el robot queda alineado con la posición objetivo. En un segundo paso, efectuamos la corrección de la profundidad mediante una ley de control definida directamente en términos del tensor trifocal 1D. El funcionamiento de los dos controladores propuestos en el trabajo se ilustra mediante simulaciones, y con el objeto de respaldar su viabilidad se presentan análisis de estabilidad y resultados de simulaciones y de experimentos con imágenes reales.

# Omnidirectional Visual Control of Mobile Robots using Three-View Geometry

## Abstract

This work investigates the topic of visual control of mobile robots. Two elements of particular interest to us within this vast field are the use of omnidirectional vision and multiple-view geometric models. Omnidirectional imaging provides very accurate angular information, although it is affected by significant distortion in the radial direction, and a wide field of view, which is advantageous for robot navigational tasks. Additionally, the geometric models relating different views of the environment allow to reject image feature-matching errors and make the visual control process more robust.

We present two visual control techniques to be used on a robot moving on the ground plane. First, we propose a new method for visual homing. The technique employs a reference set of images of the environment previously acquired at different locations and the images taken by the robot during its motion. In order to take advantage of the qualities of omnidirectional vision, we define a purely angle-based approach, without requiring any distance information. This approach, taking the planar motion constraint into account, motivates the use of the 1D trifocal tensor. The additional geometric constraints enforced by the tensor, which can be computed from point correspondences between three views, improve the robustness of the method in the presence of mismatches. The interest of this proposal is that the designed control scheme computes the robot velocities only from angular information, being this very precise information. In addition, we present a procedure that computes the angular relations between all the views even if they are not directly related by feature matches.

This approach can be classified as image based, given that neither pose estimation nor 3D information are needed. Then, the robot is guaranteed to converge to the target location without knowing the metric information about the trajectory followed. In some applications, such as obstacle avoidance, more information about the 3D motion can be needed. Taking this idea into account, we also present an omnidirectional visual control method based on sinusoidal inputs. We were motivated by the well-known mathematical properties of sinusoids and their smoothness, which makes them particularly suited for parking manoeuvres of car-like vehicles. We derive analytical expressions for the evolutions of the robot's state variables under the sinusoidal-varying velocities defined in our design, and we propose a state feedback control law based on those expressions. We describe a way to estimate the state of the robot (i.e. its position and orientation) from the 1D trifocal tensor computed between the goal, current and initial views. This sinusoidal-based control aligns the robot with the target; the depth correction is carried out in a second step, in which the control law is defined directly in terms of the elements of the 1D trifocal tensor. The performance of the controllers is illustrated through simulations, and the feasibility of the proposed approaches is supported by the stability analysis and the results from simulations and experiments with real images.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Omnidirectional visual homing</b>	<b>6</b>
2.1	Angles from the 1D trifocal tensor . . . . .	7
2.1.1	1D trifocal tensor computation and epipole extraction . . . . .	7
2.1.2	Ambiguity resolution . . . . .	8
2.2	Reference-set angles computation . . . . .	9
2.2.1	Complete solution of four-view sets . . . . .	10
2.3	Homing strategy . . . . .	12
2.3.1	Control law . . . . .	13
2.3.2	Stability Analysis . . . . .	14
<b>3</b>	<b>Sinusoidal input-based visual control</b>	<b>15</b>
3.1	System model . . . . .	15
3.2	Sinusoidal input-based control scheme . . . . .	16
3.2.1	Open-loop state evolution . . . . .	16
3.2.2	Closed-loop control . . . . .	19
3.3	1D trifocal tensor-based depth correction . . . . .	20
3.4	State estimation through the 1D trifocal tensor . . . . .	21
3.5	Stability analysis . . . . .	22
3.5.1	First step . . . . .	22
3.5.2	Second step . . . . .	23
<b>4</b>	<b>Conclusion</b>	<b>24</b>

# 1 Introduction

We will start off by providing some background and highlighting relevant related research that motivated and inspired the realization of this work, part of which has been published in [1].

Vision sensors have been widely used for robot navigational tasks [2] due to the high amount of information that can be extracted from them. Omnidirectional cameras in particular are being increasingly employed in navigation, aiming to take advantage of their wide field of view. In contrast to what occurs with their radial information, which is strongly affected by distortion, the angular information provided by these cameras is rich and precise.

Homing -i.e. the ability to return to a previously known location after having been moved away from it- is a fundamental task for a mobile robot. Vision-based homing is often inspired by the mechanisms that certain animal species, such as insects [3,4], utilize to return to their known home location. Angle-based homing methods using omnidirectional vision have been proposed, being [5] an early work and [6,7] examples of more recent contributions. These are purely feature-based approaches where the angles of landmarks in the images are used to generate a homing vector. A way to increase the robustness to feature matching errors of visual homing (and of vision-based robot control techniques in general) is through the use of the geometric models that relate multiple views of a scene. In this respect, visual control methods have been presented using the epipolar geometry, which expresses the relations between two views [8], and the trifocal tensor, which encapsulates the three-view geometric constraints [9]. In particular, robot navigation on the ground plane lends itself to the use of the 1D trifocal tensor, the unique matching constraint between 1D views of a 2D scene [10]. The 1D trifocal tensor can be computed linearly from point correspondences, and employed to perform 2D projective reconstruction. It has been used for 2D robotic localization tasks [11, 12]. The geometric information encapsulated in the 1D trifocal tensor can also be used directly for control applications [13], or as a tool to estimate the robot's state or other parameters of interest.

Some basic goals in the field of control of mobile robots are to generate smooth, flexible, analyzable trajectories. Sinusoidal functions are good candidates to fulfill these objectives, owing to their well-known mathematical properties and characteristic smoothness. Sinusoids have been employed for steering a class of nonholonomic systems, specifically those which can be expressed in chained form [14]. Trajectories of sinusoidal nature have also been shown to be particularly suited for parallel parking manoeuvres [15]. The use of sinusoids in these approaches lies within the field of motion planning, i.e. they are open-loop techniques where no feedback control is employed. In [16] some results are presented on the stabilization of systems in chained form with sinusoids.

The contents of this work are organized as follows: Section 2 introduces an angle-based omnidirectional visual homing method employing the 1D trifocal tensor. In section 2.1 we discuss the calculation of the angles between views from the estimation of the 1D trifocal tensor. We also propose a method for the resolution of the ambiguities in the estimation of the angles. Section 2.2 presents the procedure for the computation of all the angular information needed for the homing task. In section 2.3 the designed control strategy is described.

Throughout section 3 we present a visual method for controlling a mobile robot to a target pose using sinusoids as velocity inputs. Section 3.1 describes how we model the system to be controlled. Section 3.2 discusses the first step of our control approach, which is when the sinusoidal inputs are used. In section 3.2.1 we derive the open-loop robot trajectories associated with the sinusoidal velocities, while

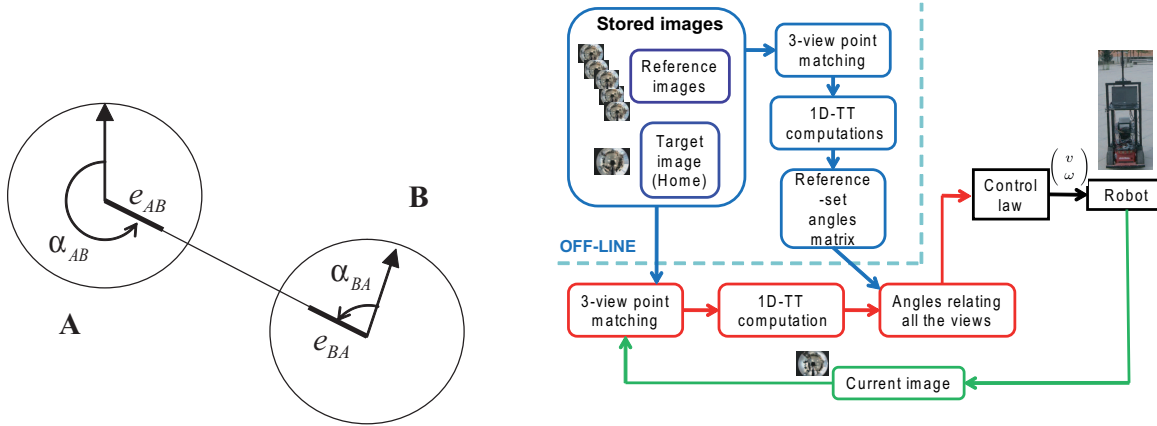


Figure 1: Overview of the visual homing control loop (right). Nomenclature and conventions used throughout the article (left).  $e_{AB}$  is the epipole (i.e. the projection of the optical center) of view B in view A.  $\alpha_{AB}$  is the angle or direction of that epipole, i.e. its angular polar coordinate in view A. The reference axis for the measurement of the angles is the robot’s axis of orientation (the vertical axis of the images). The angles are measured counterclockwise and are defined between  $-\pi$  and  $\pi$ .

section 3.2.2 presents a state-feedback control method based on these trajectories. The second step of the control method is discussed in section 3.3. In section 3.4 we describe how the state estimation is obtained from the 1D trifocal tensor. The stability of the two-step control is analyzed in section 3.5.

Finally, the conclusion is given in section 4. The experimental results from the methods presented in this work can be found in the Appendix.

## 2 Omnidirectional visual homing

We propose a homing method that makes use of the angular information between omnidirectional views extracted by means of the 1D trifocal tensor. The three-view geometric constraints enforced by this tensor on the point correspondences make the calculations more robust to outliers when compared to feature-based methods. Our approach employs only the visual information provided by omnidirectional images to obtain the angles between the current position and a set of previously acquired reference images taken at different locations, any of which can be selected as the home (or goal) position. A control law based on the estimated angles is used to guide the robot to the target. Some nomenclature and conventions used throughout this document are illustrated in Fig. 1, which also provides an overview of the employed control loop.

## 2.1 Angles from the 1D trifocal tensor

The trifocal tensor is the mathematical entity that encapsulates all the geometric relations between three views that are independent of scene structure. In particular, the 1D trifocal tensor relates three 1D views on a plane, and presents some interesting properties; namely, it is the only matching constraint between 1D views, it can be estimated linearly from a minimum of seven three-view point matches (or five, if the calibration of the cameras is known [17]), and 2D projective reconstruction can be obtained from it.

### 2.1.1 1D trifocal tensor computation and epipole extraction

The projections of a given point in three 1D views (which we will refer to as  $A$ ,  $B$  and  $C$ ) on a plane are related by the following trilinear constraint [10]:

$$\sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 T_{ijk} u_i^A u_j^B u_k^C = 0, \quad (1)$$

where  $T_{ijk}$  are the elements of the  $2 \times 2 \times 2$  *1D trifocal tensor*,  $\mathbf{T}$ , between the views, and  $\mathbf{u}^A$ ,  $\mathbf{u}^B$  and  $\mathbf{u}^C$  are the homogeneous coordinates of the projections of the point in each view.  $\mathbf{T}$  is defined up to a scale factor and therefore can be calculated, in the uncalibrated case, from a minimum set of seven point correspondences across the views.

The process we follow to estimate  $\mathbf{T}$  starts by detecting relevant features in three omnidirectional images, e.g. by means of the SIFT keypoint extractor [18], and finding matches between them. The angles ( $\alpha$ ) of the matched image points, measured counterclockwise from the vertical axis, are converted to a 1D projective formulation, with the corresponding homogeneous 1D coordinates being  $(\sin \alpha, \cos \alpha)^T$ . In this mapping, the distinction between angles differing by  $\pi$  is lost.

Each of the point matches in 1D projective coordinates gives rise to an equation of the form of (1). If more than seven correspondences are available, we find a least squares solution to the resulting system of linear equations through SVD. In this process, a robust estimation method (RANSAC) is employed in order to reject wrong matches.

After  $\mathbf{T}$  has been estimated, the epipoles are extracted from it using a procedure presented in [19] that we briefly describe next. A 1D homography is a mapping between projected points in two lines (two of the 1D views, in our case) induced by another line. From the coefficients of the trifocal tensor, we can directly extract what are known as the *intrinsic homographies*; for example, the two intrinsic homographies from  $A$  to  $B$ ,  $\mathbf{K}_{AB}$  and  $\mathbf{L}_{AB}$ , are obtained by substituting the lines defined by  $\mathbf{u}^C = (1, 0)^T$  and  $\mathbf{u}^C = (0, 1)^T$  in (1), yielding

$$\mathbf{K}_{AB} = \begin{bmatrix} -T_{211} & -T_{221} \\ T_{111} & T_{121} \end{bmatrix}, \quad \mathbf{L}_{AB} = \begin{bmatrix} -T_{212} & -T_{222} \\ T_{112} & T_{122} \end{bmatrix}. \quad (2)$$

Now,  $\mathbf{H}_A = \mathbf{K}_{AB} \mathbf{L}_{AB}^{-1}$  is a homography from  $A$  to itself; by definition, the epipoles are the only points that are mapped to themselves by such a homography, i.e.:  $\mathbf{e}_{AB} = \mathbf{H}_A \mathbf{e}_{AB}$  and  $\mathbf{e}_{AC} = \mathbf{H}_A \mathbf{e}_{AC}$ . Therefore we can calculate them as the eigenvectors of matrix  $\mathbf{H}_A$ ; it is important to note, though, that with this method we do not know which of the other two views ( $B$  or  $C$ ) each of the two recovered epipoles corresponds to. By mapping this pair of epipoles to the other views through the intrinsic homographies, we finally obtain the six epipoles of the set of three 1D views.

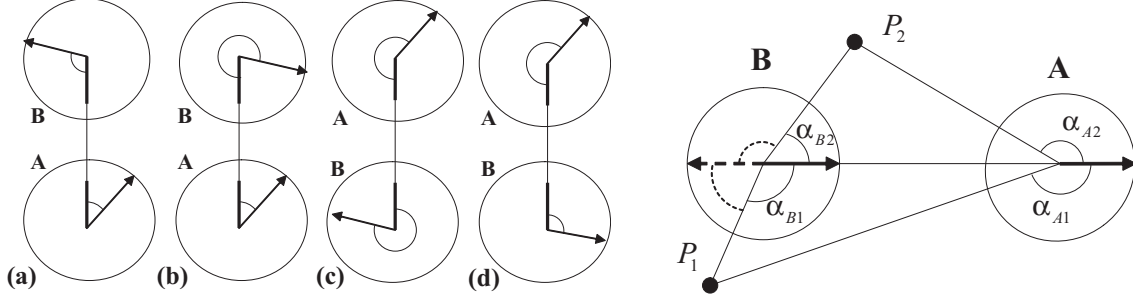


Figure 2: Four possible 2D reconstructions from the epipoles between two views extracted from the 1D trifocal tensor (left). The relations between the angles of the projections of matched points, such as  $P_1$  and  $P_2$ , in two aligned views are used to resolve the 2D reconstruction ambiguities (right).

### 2.1.2 Ambiguity resolution

There are three ambiguities that need to be resolved in order to determine the correct values of the angles of the 2D epipoles from the values of the epipoles extracted using the 1D trifocal tensor.

Firstly, as mentioned in section 2.1.1, an epipole in a given view recovered from the 1D trifocal tensor may be assigned to any of the two other views. This results in two possible solutions in the assignment of the set of six epipoles between the three views. As shown in [10, 17], both solutions give completely self-consistent 2D projective reconstructions, regardless of the number of point matches between the views. This fundamental ambiguity in the 2D reconstruction from three 1D views can only be resolved through the use of a fourth view, as noted in [11]. The method we employ to resolve the ambiguity operates in the following way: having a group of four views (which we can call A, B, C and D), we calculate two different trifocal tensors between them; for instance, the tensor relating A, B and C, and the tensor between B, C, and D. Since the epipoles between B and C must be identical in the two estimations, by detecting the common (or, in a real situation, the closest) epipoles in these two views we can disambiguate the assignment of the complete set of epipoles.

The origin of the two other ambiguities lies in the fact that in the mapping of 2D image points to 1D projective coordinates, the distinction between bearings differing by  $\pi$  is lost. The angle of a recovered 1D epipole  $(e_1, e_2)^T$  is obtained as  $\arctan(e_1/e_2)$  in 2D. As a consequence, from the 1D epipole we can extract two different angles in a 2D view, separated by  $\pi$  radians. There are, therefore, four possible solutions for the values of the epipoles between two given views A and B, which may be interpreted as emerging from two combined reconstruction ambiguities; namely, an ambiguity in the direction of the translation vector from view A to view B, which accounts for the difference between solutions (a) and (c) in Fig. 2, and an ambiguity of  $\pi$  radians in the orientation of view B, illustrated, for example, by solutions (a) and (b) in the same figure.

This double ambiguity for a set of two views might be resolved through point reconstruction, but instead we use a simple method employing only the angles of matched image points. We first choose one of the two possible values for each angle. We name these selected angles  $\alpha_{AB}^s$  and  $\alpha_{BA}^s$ . Although these two choices are arbitrary, let us suppose, for simplicity, that we have picked the angles so that both of them are between 0 and  $\pi$  radians. Our procedure is based on the alignment of the two views, which is achieved through the rotation of the image points by the selected angles. This is done simply



Table 1: Disambiguation of the angles of the epipoles in two views

B in front of A	B reversed (*)	$\alpha_{AB}$	$\alpha_{BA}$	Case in Fig. 2
1	1	$\alpha_{AB}^s$	$\alpha_{BA}^s$	(a)
1	0	$\alpha_{AB}^s$	$\alpha_{BA}^s + \pi$	(b)
0	1	$\alpha_{AB}^s + \pi$	$\alpha_{BA}^s + \pi$	(c)
0	0	$\alpha_{AB}^s + \pi$	$\alpha_{BA}^s$	(d)

(\*)After alignment of the images using the selected angles,  $\alpha_{AB}^s$  and  $\alpha_{BA}^s$ .

by subtracting the selected angle from the angular coordinate of every matched image point. Once the images have been aligned, if the two cameras are pointing in the same direction, the two projections of any given point in them will lie on the same side with respect to the camera’s axis of orientation, whereas if the cameras are pointing in opposite directions, the projections will lie on opposite sides of the axis (Fig. 2). We use a voting procedure integrating the individual results of this test for all the matched points to determine whether aligning the two cameras using the selected angles leaves them pointing in the same direction or reversed with respect to one another.

Next, we want to obtain the sign of the scale of the translation from view A to view B, i.e. establish if B is in front of or behind A. Having the two images aligned and pointing in the same direction (rotating view B by an additional  $\pi$  radians if required) we use the fact that the projections of points in the camera that is in front will give larger angles, in absolute value, than the projected points in the camera that is situated behind. This is again illustrated in Fig. 2. For every matched point, we subtract the absolute values of the angles of its projections in views A and B. We square these results (keeping their sign), in order to give the more discriminant points a greater weight, and then add them up. If the sum is positive, A is estimated as being behind B; otherwise, A is in front of B. This result directly gives us the angle of the epipole in view A, and its combination with the outcome of the orientation test determines the value of the angle in view B, as shown in Table 1. Additionally, for every group of three views the joint consistency of the three results of this two-view disambiguation procedure is checked.

## 2.2 Reference-set angles computation

The initial stage of our method involves the calculation of the angular relations between the images on the reference set. This processing can be done off-line and therefore its time consumption is not a critical issue. The aim is to build and store a matrix containing the angles of the epipoles between every pair of reference views, for their use during homing.

We will name that matrix  $\mathbf{M}$ , with  $M(i, j) = \alpha_{ij}$  being the angle of the epipole of view  $j$  in view  $i$ . All the diagonal elements of  $\mathbf{M}$  are defined as zero and the size of the matrix is  $(n \times n)$ , where  $n$  is the number of reference views.

Relevant features are extracted and matched between every pair of images on the reference set, and the resulting point correspondences are stored. We then start an estimation procedure that operates as follows:

- A set of four images (which we can call A, B, C and D) taken in two groups of three (e.g. A-B-C and B-C-D) are processed in each step. For each trio of images we obtain three-view point correspondences by taking the common two-view matches between them. From a minimum number of seven point



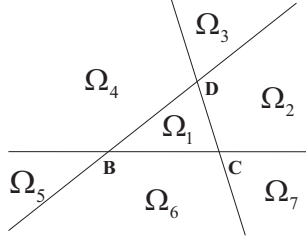


Figure 4: Seven regions where point A can be located.

geometric setting of the type shown in Fig. 3, where two triangles are known between the locations of four views, comes up in our method every time we estimate two trifocal tensors from a four-view set. This section describes the method employed to calculate the two unknown angles in this configuration.

We use the notation  $\widehat{ABC}$  to refer to the angular size ( $> 0$ ) of the angles in a triangle. Without loss of generality, we can formulate the problem in the following terms: all the angles from every view to the others in the set are known except the angles of the epipoles between views A and D. Therefore all the angles in the four triangles formed by the set of four views are known, except the ones including both vertices A and D (represented with dashed lines in Fig. 3). Our objective is to calculate the angles  $\alpha_{AD}$  and  $\alpha_{DA}$  of the epipoles  $e_{AD}$  and  $e_{DA}$ , which can be directly obtained from the knowledge of the angles of the triangles at those vertices. We start by applying the law of sines on the set of four triangles (ABC, ABD, ACD and BCD), which finally yields the following expression

$$\frac{\sin \widehat{ABD}}{\sin \widehat{ACD}} = K_A, \quad (3)$$

where  $K_A$  is a known value given by

$$K_A = \frac{\sin \widehat{CBD} \cdot \sin \widehat{BAD}}{\sin \widehat{BCD} \cdot \sin \widehat{CAD}}. \quad (4)$$

Using the intrinsic relationship between the three angles at vertex A and applying trigonometric identities, we can calculate the individual values of the angles in (3). We must, however, take into account the fact that the location of A with respect to the other three vertices changes the geometry of the set and, consequently, the relation between the angles at the aforementioned vertex. Therefore, we need to divide the plane into seven regions, as shown in Fig. 4, to account for these differences. It turns out that the expression that gives the angle  $\widehat{ABD}$  has the same form in all cases (i.e. for all regions), but the signs of two of its terms, denoted as  $sign_1$  and  $sign_2$ , are dependent on the region where A lies

$$\widehat{ABD} = \arctan \frac{sign_1 \cdot K_A \sin(\widehat{ABC})}{1 + sign_2 \cdot K_A \cos(\widehat{ABC})}. \quad (5)$$

We can easily determine the region in which A is located using the known angles of the epipoles in views B and C, and choose the appropriate values of  $sign_1$  and  $sign_2$  as shown in Table 2.

Table 2: Values of signs for the different regions in which A may lie

Region of vertex A	Relation between angles at vertex A	$sign_1$	$sign_2$
$\Omega_1$	$\widehat{ACD} = 2\pi - \widehat{ABD} - \widehat{ABC}$	-1	1
$\Omega_2, \Omega_5$	$\widehat{ACD} = \widehat{ABD} + \widehat{ABC}$	1	-1
$\Omega_3, \Omega_6$	$\widehat{ACD} = \widehat{ABC} - \widehat{ABD}$	1	1
$\Omega_4, \Omega_7$	$\widehat{ACD} = \widehat{ABD} - \widehat{ABC}$	-1	-1

The angle of the epipole of view D in view A is finally obtained as follows

$$\alpha_{AD} = \begin{cases} \alpha_{AB} + \widehat{ABD}, & \text{if } 0 \leq \alpha_{BA} - \alpha_{BD} < \pi \\ \alpha_{AB} - \widehat{ABD}, & \text{if } 0 > \alpha_{BA} - \alpha_{BD} \geq -\pi \end{cases}. \quad (6)$$

The angle of the epipole in view D of view A ( $\alpha_{DA}$ ) can be calculated through a completely analogous process, simply interchanging the roles of vertices A and D. The results are validated using geometric consistency checks. By employing the procedure we have just presented, we can calculate the two unknown angles and thus obtain the complete set of angles between the four views. In addition, this method is useful for two other purposes within our homing technique:

- In the initial stage, detailed in section 2.2, this method allows to fill in the missing elements in the matrix of epipole angles, corresponding to pairs of views that could not be linked directly due to the impossibility to find a sufficiently large set of three-view matches between them.
- During homing, it enables us to obtain all the angles needed to generate the motion commands employing a minimum number of three views; we only need to compute the trifocal tensor between the current image taken by the robot and two of the reference images, which reduces the cost of the algorithm.

### 2.3 Homing strategy

In this section we describe the strategy designed in order for the mobile robot to perform homing. We assume the robot moves on the ground plane and has nonholonomic motion constraints. The homing method is based solely on the computation of the angles between the locations in which a series of omnidirectional images of the environment were obtained. This group of snapshots consists of the image taken by the robot from its current position and a set of previously acquired reference images, which includes an image obtained at the desired target location. The angles between the views on the reference set have been previously computed and stored, as described in section 2.2. Therefore only the angles between the robot and the reference views must be worked out during homing.

In every step of the robot's motion, the camera takes an omnidirectional image, from which key points are extracted. When sufficient point matches are found between the current and two of the reference images, the 1D trifocal tensor is calculated as detailed in section 2.1.1. From the tensor, aided by the knowledge of the angles on the reference set, we can extract the angles between the current and the two other views. Finally, with the method explained in section 2.2.1 all the angles of the epipoles in all the views can be computed.

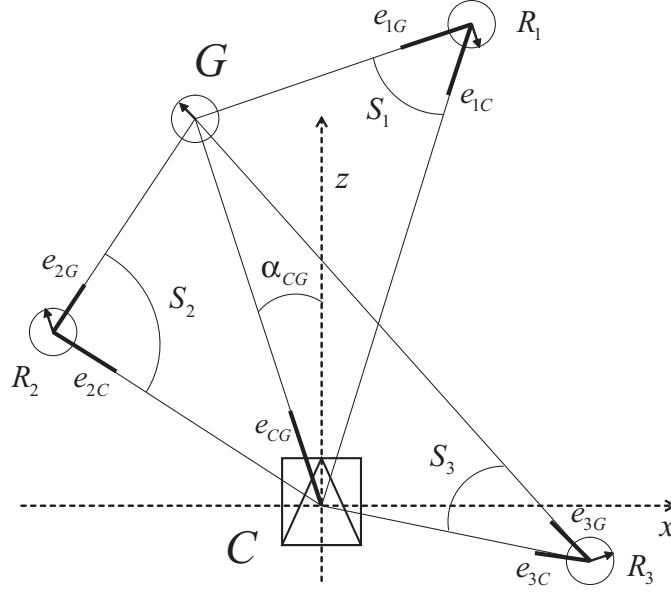


Figure 5: Elements involved and angles employed in the homing strategy.  $C$  is the robot's current localization, at the coordinate origin  $(0, 0, 0)$ .  $G$  is the goal location.  $R_i$  are reference views. Three of the  $n$  views on the reference set are depicted as example.

### 2.3.1 Control law

For every reference view  $R_i(x_i, z_i, \varphi_i)$  (where  $x_i$ ,  $z_i$  and  $\varphi_i$  define its position and orientation in the ground plane), the difference between the angles of its epipoles with respect to the current and goal locations defines an angular sector of size  $S_i = |\alpha_{iC} - \alpha_{iG}|$ , as illustrated in Fig. 5. We use the average value of the angular sizes of these sectors to set the linear velocity at which the robot will move toward the target position

$$v = k_v \text{sign}(\cos \alpha_{CG}) \cdot \frac{1}{n} \sum_{i=1}^n S_i, \quad (7)$$

where  $k_v > 0$  is a control gain. When the target is behind the robot,  $\text{sign}(\cos \alpha_{CG})$  will be negative, therefore generating backward motion. As the robot moves closer to the goal, the mean size of the angular sectors seen from the reference positions will become smaller; thus, the robot's linear velocity will gradually decrease and eventually become zero when the target is reached.

The direction in which the robot travels is determined by the angle at which the goal position is seen from the current location, i.e. the angle  $\alpha_{CG}$  of the epipole  $e_{CG}$ . The angular velocity of the control law is given by

$$\omega = k_\omega (\alpha_{CG} - \alpha_{CG}^d), \quad (8)$$

$$\alpha_{CG}^d = \begin{cases} 0 & \text{if } |\alpha_{CG}| \leq \frac{\pi}{2} \\ \pi & \text{if } |\alpha_{CG}| > \frac{\pi}{2} \end{cases}, \quad (9)$$

where  $k_\omega > 0$  is a control gain. From a minimum number of four reference views, one of which would be

the view from the target location, the robot will navigate to the home position. Note that the orientation in which the robot reaches the target position is not controlled, since, by definition, the purpose of the homing task is getting to the goal location.

### 2.3.2 Stability Analysis

In the following, the stability of the control scheme is analyzed by means of *Lyapunov's Direct Method* [20]. We define the candidate Lyapunov function as

$$V(\mathbf{x}, t) = \frac{\rho^2}{2} + \frac{(\alpha_{CG} - \alpha_{CG}^d)^2}{2} \quad (10)$$

where  $\rho$  is the distance between the current and goal positions, and  $\mathbf{x}$  is the state of the system, determined by  $\rho$  and  $\alpha_{CG}$ .

Next, we show that the candidate function (10) is a Lyapunov function when using the proposed control law. We need to prove that  $V$  is positive definite,  $\dot{V}$  is negative definite and  $V$  is radially unbounded. The function  $V$  is positive definite, given that  $V > 0$  for all  $\mathbf{x} \neq \mathbf{0}$  and  $V(\mathbf{0}) = 0$ . It is straightforward that  $V(\mathbf{x})$  is radially unbounded, given that  $V(\mathbf{x}) \rightarrow \infty$  as  $\|\mathbf{x}\| \rightarrow \infty$ . Next, we prove that the derivative  $\dot{V}(\mathbf{x})$  is negative definite. The Lyapunov candidate function derivative is

$$\dot{V} = \rho \dot{\rho} + (\alpha_{CG} - \alpha_{CG}^d) \dot{\alpha}_{CG} . \quad (11)$$

The dynamics of the system as a function of the input velocities are given, using the derivatives in polar coordinates with the origin at the goal, by  $\dot{\rho} = -v \cos(\alpha_{CG})$  and  $\dot{\alpha}_{CG} = -\omega + v \sin(\alpha_{CG})/\rho$ . Using the control velocities (7), (8) we obtain

$$\begin{aligned} \dot{V} = & -k_v \rho \operatorname{sign}(\cos \alpha_{CG}) \cos(\alpha_{CG}) \frac{1}{n} \sum_{i=1}^n S_i \\ & -k_w (\alpha_{CG} - \alpha_{CG}^d)^2 - (\alpha_{CG} - \alpha_{CG}^d) \\ & \cdot \sin(\alpha_{CG}) \frac{k_v}{\rho} \operatorname{sign}(\cos \alpha_{CG}) \cdot \frac{1}{n} \sum_{i=1}^n S_i . \end{aligned} \quad (12)$$

By definition  $\rho \geq 0$  and  $S_i \geq 0$ . It is straightforward that the first two terms of (12) are negative definite, but the last term can be positive. The interpretation is that the convergence speed provided by the angular velocity has to be higher than the linear velocity. Otherwise, the angular error is not corrected fast enough and the robot will move following spirals around the goal. However, the stability can be guaranteed if the control gains are selected properly. It is guaranteed that  $\dot{V} < 0$  if the next inequality holds

$$|k_w \cdot (\alpha_{CG} - \alpha_{CG}^d)| > |\sin(\alpha_{CG}) \frac{k_v}{\rho} \cdot \frac{1}{n} \sum_{i=1}^n S_i| . \quad (13)$$

The terms of the right expression are bounded except  $1/\rho$ . However, note that  $\sum_{i=1}^n S_i/(n\rho)$  is indeed bounded. We can express  $S_i$  as a nonlinear function of  $\rho$ :  $\sin(S_i) = f(\rho, \overline{R_i G}, \alpha_{R_i G})$ ; when  $\rho \rightarrow 0$  we can approximate  $\sin(S_i) \approx S_i$  and then  $S_i = \rho f(\overline{R_i G}, \alpha_{R_i G})$ . Therefore, the right part in (13) is bounded. Then, taking into account the initial value  $\alpha_{CG}(0)$ , we can always define  $k_w$  in such a way that  $\dot{V} < 0$  and the system is globally asymptotically stable.

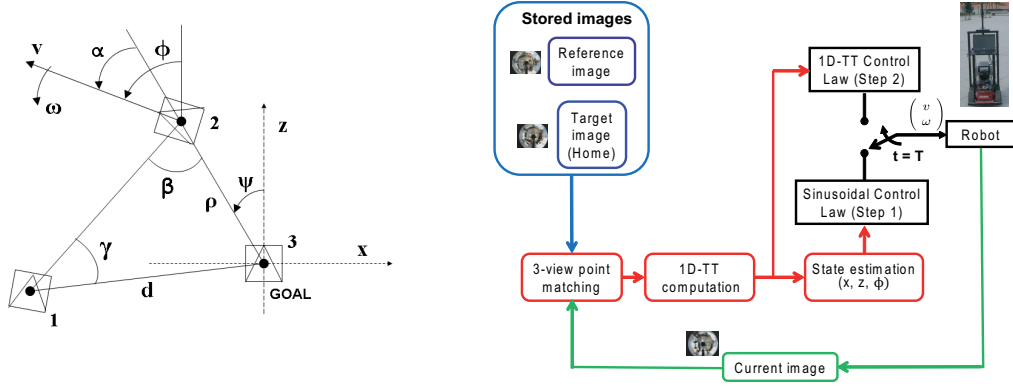


Figure 6: Overview of the sinusoidal-based visual control loop (right). Definition of the elements of our system and the geometric variables employed (left). '3' identifies the goal view, which serves as the global coordinate system. '2' is the robot's current location, and '1' is the location of the additional reference view.

### 3 Sinusoidal input-based visual control

We present a control method based on sinusoidal inputs for a nonholonomic mobile robot. We define our particular choice of sinusoidal velocities and derive the state trajectories from them through integration. Using the open-loop expressions of the state variables, we design a state-feedback control law. The state of the system is estimated from the 1D trifocal tensor between the current view, the initial view and the target view. The method has two steps: the sinusoidal-based control part is followed by a second step consisting in a straight-line trajectory to carry out depth correction.

#### 3.1 System model

A nonholonomic robot moving on the ground plane constitutes a dynamical system on which we are going to perform a control task. 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.  $\mathbf{x}_3 = (0, 0, 0)$ . The nonholonomic differential kinematics of the vehicle expressed in state space form as a function of the translation and rotation velocities of the robot  $(v, \omega)$  is as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega. \quad (14)$$

Since the primary information we will extract from the system will be of angular nature, it will be also useful to express the system's state and its kinematics in polar coordinates  $(\rho, \alpha, \phi)$  as illustrated in Fig. 6. 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:

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos \alpha \\ -\frac{1}{\rho} \sin \alpha \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \omega. \quad (15)$$

### 3.2 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. Our proposed open-loop velocities follow a sinusoidal time variation expressed by:

$$v = a \sin(\Omega t) \quad (16)$$

$$\omega = A \cdot b \sin(2\Omega t). \quad (17)$$

We will assume throughout that the angular frequency of the sinusoid ( $\Omega$ ) is set to a constant value. This is a design parameter whose value will be determined by the time lapse in which we want the first step of the control to be carried out (given by one half-period of the sinusoid, i.e.  $T/2 = \pi/\Omega$ ).  $A = -\text{sign}(v)$  is a parameter used to set the appropriate sign of the angular velocity wave so that the robot trajectories defined in our approach are obtained. We will only discuss the case where the motion ends on  $t = T/2$ , and therefore  $A$  will be equal to -1. For their part,  $a$  and  $b$  are real values that set the amplitudes of the sinusoidal velocity waves. In an open-loop scenario,  $a$  and  $b$  are defined as constants, but when a feedback control strategy is used, as described in section 3.2.2, these two parameters will be our control variables. Without loss of generality, we 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 ( $t_0 \neq 0$ ) of the sinusoidal velocity waves, which will now run from  $t_0$  to  $T/2$ . This will be described in more detail in section 3.2.2.

When the velocities in (16) and (17) are used, the resulting motion will be smooth, due to both the choice of a sinusoidal time variation and the fact that the rotational velocity is zero at the points where the linear velocity is either maximum or zero. For the sake of smoothness, safety and feasibility of the motion commands, higher rotational speeds are associated to intermediate linear velocity values in our approach. We believe these properties make the proposed open-loop velocities appropriate for their use on a vehicle with nonholonomic motion constraints.

#### 3.2.1 Open-loop state evolution

In this section we will obtain the analytical expressions for the evolutions of the state variables in an open-loop scenario under the sinusoidal-varying velocity commands proposed in section 3.2.

By integrating the sinusoidal velocities given in (16) and (17) over time, we can derive the equations for the open-loop evolution of the three state variables. We consider the robot's state at  $t = 0$  to be  $(x_0, z_0, 0)^T$ . We will first obtain the time variation of the orientation component:



$$\phi(t) = \phi_0 + \int_0^t \dot{\phi} d\tau = \int_0^t \omega d\tau = \int_0^t A \cdot b \sin(2\Omega\tau) d\tau = \frac{Ab}{2\Omega} (1 - \cos 2\Omega t). \quad (18)$$

Once we know how  $\phi$  evolves with time, we can substitute this result in (14) to obtain the time variations for the two other state variables. We will first determine the evolution of  $x(t)$ , for which we will use the Taylor series representation of  $\sin \phi$ :

$$\begin{aligned} \dot{x}(t) &= -v(t) \sin \phi(t) = -a \sin(\Omega t) \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \phi^{2n+1} = \\ &= -a \sin(\Omega t) \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \left( \frac{Ab}{2\Omega} \right)^{2n+1} (1 - \cos 2\Omega t)^{2n+1} = \\ &= -a \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n+1}}{(2n+1)!} \left( \frac{b}{\Omega} \right)^{2n+1} \sin^{4n+3}(\Omega t), \end{aligned} \quad (19)$$

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

$$x(t) = x_0 + \int_0^t \dot{x}(\tau) d\tau = x_0 + \int_0^t -a \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n+1}}{(2n+1)!} \left( \frac{b}{\Omega} \right)^{2n+1} \sin^{4n+3}(\Omega \tau) d\tau. \quad (20)$$

The integral of the sine function in (20) can be expressed as:

$$\int \sin^{4n+3}(\Omega t) dt = -\frac{\cos \Omega t}{\Omega} \cdot {}_2F_1(1/2, -2n-1; 3/2; \cos^2 \Omega t),$$

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

$${}_2F_1(\alpha, \beta; \gamma; \chi) = \sum_{k=0}^{\infty} \frac{(\alpha)_k (\beta)_k}{(\gamma)_k} \frac{\chi^k}{k!}$$

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

$$x(t) = x_0 + \left[ \frac{a \cos(\Omega t)}{\Omega} \cdot \Gamma(b, t, 1) \right]_0^t, \quad (21)$$

where we define:

$$\Gamma(b, t, m) = \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n+m}}{(2n+m)!} \left( \frac{b}{\Omega} \right)^{2n+m} \cdot {}_2F_1(1/2, -2n-m; 3/2; \cos^2 \Omega t). \quad (22)$$

Thus,  $x$  can be exactly determined at any time through the sum of series of infinite terms. Note, however, that the index of the sums,  $n$ , is the index of the Taylor series representation of  $\sin \phi$ . The values of  $\phi$  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 for  $x(t)$ .

The time evolution of the state variable  $z$  can be worked out in an analogous way, through the integration of the corresponding expression in (14). This time we use the Taylor series expansion of  $\cos \phi$ .

$$\begin{aligned} \dot{z}(t) = v \cos \phi &= a \sin(\Omega t) \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \phi^{2n} = a \sin(\Omega t) \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \left( \frac{Ab}{2\Omega} \right)^{2n} (1 - \cos 2\Omega t)^{2n} = \\ &= a \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n}}{(2n)!} \left( \frac{b}{\Omega} \right)^{2n} \sin^{4n+1}(\Omega t), \end{aligned} \quad (23)$$

$$z(t) = z_0 + \int_0^t \dot{z} d\tau = z_0 + \int_0^t a \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n}}{(2n)!} \left( \frac{b}{\Omega} \right)^{2n} \sin^{4n+1}(\Omega \tau) d\tau. \quad (24)$$

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

$$\int \sin^{4n+1}(\Omega t) dt = -\frac{\cos \Omega t}{\Omega} \cdot {}_2F_1(1/2, -2n; 3/2; \cos^2 \Omega t), \quad (25)$$

and  $z(t)$  has the following open-loop expression:

$$z(t) = z_0 - \left[ \frac{a \cos(\Omega t)}{\Omega} \cdot \Gamma(b, t, 0) \right]_0^t. \quad (26)$$

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, in an open loop scenario, for the state variables  $x$  and  $\phi$  to go to zero at  $t = T/2$ .

The  $z$  state variable is not controlled in the first step of our control scheme, as will be discussed in section 3.2.2. It can therefore 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 value of the orientation component,  $\phi_{max}$ , that the robot will achieve during its motion. This can give us a good sense of what the trajectory will be like (see Fig. 7).

Once we have chosen a value of  $\phi_{max}$ ,  $b$  is readily obtained, since from (18) we can see that  $b_{OL} = \Omega \cdot \phi_{max}$ . The functional variation of  $\omega$  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 value of  $b$ , 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 (21), which yields:

$$a_{OL} = \frac{x_0 \cdot \Omega}{2 \cdot \Gamma(b_{OL}, 0, 1)}. \quad (27)$$

### 3.2.2 Closed-loop control

Since the system will be subject to disturbances (e.g. measurement noise, motion drift and model errors), in order to perform robust control we need to use a closed-loop control strategy based on state feedback. We will carry out the control task by adjusting the amplitudes of the sinusoidal velocity waves we are using as inputs to the system. During closed-loop operation, in every step  $i$  of the motion (i.e. at every given time  $t_i$ ), the state of the robot will be estimated and the amplitudes ( $a$  and  $b$ ) of the sinusoids required in order to steer both  $x$  and  $\phi$  to zero at  $t = T/2$  will be computed.

We would like to control the robot to its target state, i.e. make  $(x, z, \phi) = (0, 0, 0)$  at the end of the motion. However, it is well known that the nonholonomic system of a wheeled unicycle mobile robot modeled in a Cartesian state space representation considered in this work cannot be stabilized to a given configuration by means of smooth state-feedback control [21].

In our case, it turns out that with the two degrees of freedom we have in the feedback 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. Therefore we can make  $x$  and  $\phi$  go to zero in  $t = T/2$ , but we cannot do the same with  $z$ . This is why we need to perform a second control step, which we will describe in section 3.3, in order to correct  $z$  to zero.

Our closed-loop control strategy is based on adjusting the amplitudes of the sinusoidal velocity inputs in such a way that they respond to the disturbances in the system. If the state of the system at any given time  $t$ , such that  $0 < t < T/2$ , is  $(x(t), z(t), \phi(t))$ , the amplitudes,  $a$  and  $b$ , of the inputs at that time are computed by enforcing the constraints that both  $x$  and  $\phi$  must become 0 in  $t = T/2$ . For this purpose we use the previously obtained equations (18) and (21), finally obtaining the following results:

$$b(t) = \frac{-\phi(t) \cdot \Omega}{\sin^2(\Omega t)} \quad (28)$$

$$a(t) = \frac{x(t) \cdot \Omega}{\cos(\Omega t) \cdot \Gamma(b(t), t, 1) + \Gamma(b(t), 0, 1)}. \quad (29)$$

Expressions (28) and (29) are valid for  $0 < t < T/2$ . The values of  $a$  and  $b$  at  $t = 0$  can be obtained from the open loop expressions given in section 3.2.1 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 (28) and (29) 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 ( $a_{max}$  and  $b_{max}$ ) which can be a function of the open loop amplitudes, and we can also limit their variation between two consecutive steps  $t$  and  $t + \Delta t$ , e.g. enforcing  $|a(t + \Delta t)| \leq K \cdot |a(t)|$  and  $|b(t + \Delta t)| \leq K \cdot |b(t)|$ , with  $K > 1$ .

Note that, as already mentioned in section 3.2, our control method can be used for any arbitrary initial orientation of the robot, i.e.  $\phi_{start} \neq 0$ . Indeed, if we set a maximum desired value of  $\phi$  along the trajectory,  $\phi_{max}$ , and consider  $b$  as computed using the open-loop expression given in section 3.2.1 (i.e.  $b = \Omega \cdot \phi_{max}$ ), then the starting time  $t_0$  of the input sinusoids needed for this case can be obtained as:

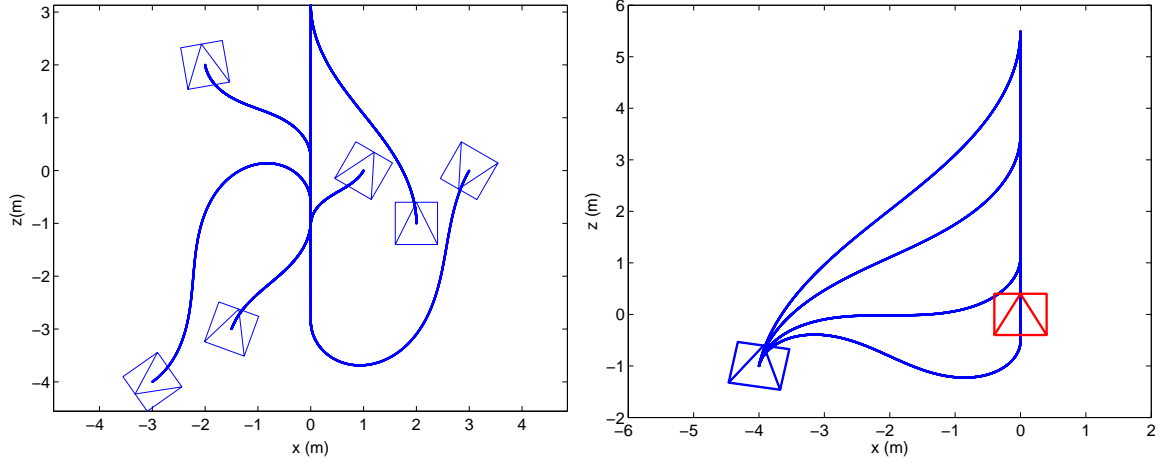


Figure 7: Left: Example robot trajectories obtained with the sinusoidal input-based control law. This law aligns the vehicle with the target pose (which is  $(0, 0, 0^\circ)$ ). In a second control step, the depth error is corrected following a straight-line path. Right: Control paths from starting position  $(-4, -1, -10^\circ)$  with  $|\phi_{max}| = 45^\circ, 60^\circ, 90^\circ$  and  $120^\circ$  (top to bottom curves).

$$t_0 = \frac{T}{2\pi} \arcsin \sqrt{\frac{\phi_{start}}{\phi_{max}}}. \quad (30)$$

$\phi_{max}$  must be chosen to have the same sign as  $\phi_{start}$  and a greater absolute value. This ensures that the computed  $t_0$  will be in the correct range, i.e.  $0 < t_0 < T/2$ . The sinusoidal inputs will now run from  $t_0$  to  $T/2$  to leave the robot aligned with the target along the  $z$  axis. Note that our method can handle any value of  $\phi_{start}$ , even if the robot is initially reversed with respect to the desired final pose, as long as  $\phi_{max}$  is chosen to have an appropriate value. Some example trajectories obtained using the sinusoidal input-based control law are illustrated in Fig. 7. Varying choices of the design parameter  $\phi_{max}$  result in paths of different characteristics, as can be seen in the right part of the figure.

### 3.3 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  $(0, z_2, 0)^1$ . The correction of the depth coordinate is performed in the second step of the control. Since in this particular situation the 1D trifocal tensor elements provide all the information necessary for the control task, we will use them directly in the feedback loop, without estimating the state of the robot explicitly. Indeed, the trifocal tensor elements when the state is  $(0, z_2, 0)$  are as follows:

<sup>1</sup>In this section, subindexes are used to identify the cameras, being  $(x_2, z_2, \phi_2)$  the current location and  $(x_1, z_1, \phi_1)$  the location of the fixed reference view.  $(x_3, z_3, \phi_3) = (0, 0, 0)$  is the location of the target view as defined in Fig. 6.

$$\mathbf{T}_1 = \begin{bmatrix} -z_2 \sin \phi_1 & -t_{z1} - z_2 \cos \phi_1 \\ t_{z1} & 0 \end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix} z_2 \cos \phi_1 & t_{x1} + z_2 \sin \phi_1 \\ -t_{x1} & 0 \end{bmatrix}, \quad (31)$$

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 [13].

We want to make the linear velocity of the robot proportional to the distance to the target. With this objective in mind, we can initially define it in the following way:

$$v = s_v \cdot \sqrt{T_{111}^2 + T_{211}^2} = \sqrt{z_2^2 \sin^2 \phi_1 + z_2^2 \cos^2 \phi_1} = |z_2|. \quad (32)$$

In order for the robot to move towards the target,  $s_v$ , which denotes the sign of  $v$ , must satisfy:  $s_v = -\text{sign}(z_2)$ . The sign of  $z_2$  can be readily inferred from the signs of  $T_{111}$  or  $T_{211}$ , since  $\phi_1$  is a known fixed angle. We also need to take into account the fact that the computed 1D trifocal tensor 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:  $T_N = \sqrt{T_{121}^2 + T_{221}^2} = \sqrt{t_{z1}^2 + t_{x1}^2} = \sqrt{x_1^2 + z_1^2}$ . The linear velocity of the second step of our control, based on the normalized 1D trifocal tensor elements, is then as follows:

$$v = k_v \cdot s_v \cdot \sqrt{\frac{T_{111}^2 + T_{211}^2}{T_{121}^2 + T_{221}^2}}, \quad (33)$$

where  $k_v$  is a positive control gain. As for the angular velocity, it is chosen in such a way that it performs a proportional correction of the orientation error of the robot, i.e.:

$$w = -k_w \cdot \phi_2, \quad (34)$$

where  $k_w$  is a positive control gain.  $\phi_2$  is computed in the same way (described in section 3.4) as in the first step of the control, .

### 3.4 State estimation through the 1D trifocal tensor

In order to perform the feedback control defined in section 3.2.2, we need to estimate the state of the robot in every step of its motion. We will only use omnidirectional visual information for this purpose. In section 2.1 we described the way in which the relative angles between three views can be extracted from the 1D trifocal tensor, using the epipoles. We will compute these angles from a set of three views; namely, the target view, the current view and a third view which can be given by the start image. A priori information can be used to disambiguate between alternative solutions. Once we know the angles between the views, it is straightforward to work out the angular coordinates  $\alpha$  and  $\phi$  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. 6).  $d$  is related to the state variable  $\rho$  and the angular sectors  $\gamma$  and  $\beta$  through the law of sines:

$$\rho = d \cdot \sin(\gamma) / \sin(\beta). \quad (35)$$

This expression can be used if  $\beta > 0$ , 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 using (35) and the dynamics of the system in (15). The following expression is obtained:

$$d = \frac{\sin \beta}{\sin \gamma} \cdot \frac{\dot{\psi}}{v \sin \alpha}. \quad (36)$$

We can estimate the derivative of  $\psi$  as:  $\hat{\dot{\psi}} = (\psi(t + \Delta t) - \psi(t)) / \Delta t$ . 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  $\beta > 0$  and  $\gamma > 0$ . 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. We must also select this prior motion so that during the subsequent control stage the robot will not cross the line between the initial and goal locations, in order for (35) to be usable.

During the initial motion we can obtain a number of estimates of  $d$  using (36) and compute their average, until we achieve a sufficiently stable value. At that point, the control phase can start.  $\rho$  is computed during control using (35), and  $x$ , the position variable we use in the feedback control loop, is obtained as  $x = -\rho \sin \psi$ .

### 3.5 Stability analysis

Next we will analyze the stability of both steps of the control method.

#### 3.5.1 First step

Under the sinusoidal input-based control law defined in the first step of our method, the system is non-autonomous and finite-time, running from  $t = t_0$  to  $t = T/2$ . Classical stability analysis primarily studies the asymptotic behavior of systems, and therefore its results and conclusions are not directly applicable in our case [22]. However, we can still use Lyapunov analysis [23] to illustrate the stability properties of the system under our control law.

We define the following Lyapunov-candidate function:  $V = \frac{1}{2}(Px^2 + \phi^2)$ , where  $P$  is a positive real number. This is a positive definite and radially unbounded function, for analogous reasons as those stated for the function used in section 2.3.2. Under the closed-loop control law of the first step of our method, the derivative of this function is as follows:

$$\dot{V} = x\dot{x} + \phi\dot{\phi} = -P \cdot x^2 \cdot \Omega \cdot F(\phi, t) + 2\phi^2 \Omega \cot(\Omega t), \quad (37)$$

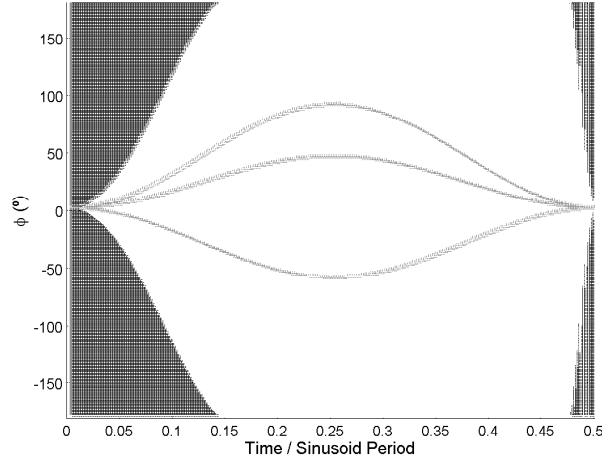


Figure 8: Graphical stability analysis from equation (38). The regions of stability are displayed in white. The curves represent three sample trajectories of the variable  $\phi$ , which are contained in the stability area.

where  $F(\phi, t) = \sin \Omega t \cdot \sin \phi / [\cos(\Omega t) \cdot \Gamma(b(t), t, 1) + \Gamma(b(t), 0, 1)]$ . The stability requirement is that in order for  $V$  to be a Lyapunov function,  $\dot{V}$  must be negative during the system's operating time interval ( $0 < t < T/2$ ). The condition  $\dot{V} < 0$  finally yields the following inequality:

$$Px^2 > \frac{2\phi^2 \cot(\Omega t)}{F(\phi, t)}. \quad (38)$$

Since  $x \neq 0$ , when the right side of (38) has a positive bound, we can always select a suitable value of  $P$  so that the inequality holds. Figure 8 shows the regions where the right side of (38) has a positive bound, along with the variation of  $\phi$  in three typical system trajectories, with  $|\phi_{max}| = 45^\circ, 60^\circ$  and  $90^\circ$ . Unless strong disturbances are present, the system will be stable, since the trajectories sit inside the stability regions.

### 3.5.2 Second step

For the stability analysis of the second step of our control (in which  $x = 0$ ) we define the following positive definite, radially unbounded Lyapunov-candidate function:  $V = \frac{1}{2}(z^2 + \phi^2)$ . Its closed-loop derivative is as follows:

$$\dot{V} = z\dot{z} + \phi\dot{\phi} = z \cdot v \cos \phi + \phi \cdot w = z \cdot \frac{-z}{\sqrt{x_1^2 + z_1^2}} \cos \phi - k_w \cdot \phi^2. \quad (39)$$

It is straightforward to see that (since  $\phi$  has small values and therefore  $\cos \phi$  is positive) the two terms of equation (39) are negative, and consequently the system under the control law of the second step is asymptotically stable.

## 4 Conclusion

We have presented a method for omnidirectional visual homing to be used on a robot moving on the ground plane. The visual information is provided by a set of omnidirectional reference images, and the 1D trifocal tensor is the tool used to estimate the angular information, being this information very precise. We have presented a method to compute all the angular relations between the views even if there is no direct matching information between them. The designed control law employs these angular relations to guide the robot to the target location. The stability of this control law has been analyzed and the experimental evaluation shows the feasibility of the proposed technique. The computational cost of the proposed method is low, and the speed of its implementation is mainly limited by the time consumption of the feature extraction process. The method can be directly applied in settings where stored image databases are available. In addition, it can be robust to changes in the environment, as long as sufficient features between images can be matched.

We have also proposed a method for sinusoidal input-based visual control of a mobile robot. From the definition of the desired sinusoidal-varying velocities, we have derived analytical expressions for the evolutions of the state variables of the robot. In addition, we have presented a state feedback control law based on these expressions and a method to estimate the robot's state from the 1D trifocal tensor. The control technique we have proposed is two-step. First, the sinusoidal-based part of the control aligns the robot with the target; then, the depth coordinate is corrected, employing the 1D trifocal tensor elements directly. The stability of both control steps has been analyzed. Simulations have shown the smoothness of the motions generated with this approach as well as its potential to withstand state noise and motion drift.



## References

- [1] 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.
- [2] 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.
- [3] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner, “A mobile robot employing insect strategies for navigation,” *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 39–64, 2000.
- [4] K. Weber, S. Venkatesh, and M. V. Srinivasany, “Insect-inspired robotic homing,” *Adaptive Behavior*, vol. 7, pp. 65–97, 1998.
- [5] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. M. Riseman, “Image-based homing,” *Control Systems Magazine, IEEE*, vol. 12, no. 1, pp. 38–45, Feb. 1992.
- [6] A. A. Argyros, K. E. Bekris, and S. C. Orphanoudakis, “Robot homing based on corner tracking in a sequence of panoramic images,” in *Computer Vision and Pattern Recognition Conf.*, 2001, pp. 3–10.
- [7] K. E. Bekris, A. A. Argyros, and L. E. Kavraki, “Angle-based methods for mobile robot navigation: Reaching the entire plane,” in *Int. Conference on Robotics and Automation*, 2004, pp. 2373–2378.
- [8] 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.
- [9] 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.
- [10] L. Quan, “Two-way ambiguity in 2D projective reconstruction from three uncalibrated 1D images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 212–216, 2001.
- [11] F. Dellaert and A. W. Stroupe, “Linear 2D localization and mapping for single and multiple robot scenarios,” in *International Conference on Robotics and Automation*. IEEE, 2002, pp. 688–694.
- [12] J. J. Guerrero, A. C. Murillo, and C. Sagüés, “Localization and matching using the planar trifocal tensor with bearing-only data,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 494–501, 2008.
- [13] G. López-Nicolás, “Visual control of mobile robots through multiple view geometry,” Ph.D. dissertation, DIIS, University of Zaragoza, 2008.
- [14] R. Murray and S. S. Sastry, “Nonholonomic motion planning: Steering using sinusoids,” *IEEE Transactions on Automatic Control*, vol. 38, pp. 700–716, 1993.
- [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, pp. 802–819, 1993.

- [16] A. Teel, R. Murray, and G. Walsh, “Nonholonomic control systems: From steering to stabilization with sinusoids,” in *Proc. 31st IEEE Conference on Decision and Control*, 1992, pp. 1603–1609.
- [17] K. Åström and M. Oskarsson, “Solutions and ambiguities of the structure and motion problem for 1D retinal vision,” *Journal of Mathematical Imaging and Vision*, vol. 12, no. 2, pp. 121–135, 2000.
- [18] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [19] A. Shashua and M. Werman, “Trilinearity of three perspective views and its associated tensor,” in *International Conference on Computer Vision*, 1995, pp. 920–925.
- [20] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2001.
- [21] C. Samson and K. Ait-Abderrahim, “Feedback control of a nonholonomic wheeled cart in cartesian space,” in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 1136–1141.
- [22] L. Weiss and E. Infante, “On the stability of systems defined over a finite time interval,” in *Proc. of the National Academy of Sciences*, vol. 40, 1965, pp. 440–448.
- [23] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.