

A biologically inspired method for vision-based docking of wheeled mobile robots[☆]

Emily M.P. Low^{a,*}, Ian R. Manchester^b, Andrey V. Savkin^a

^a School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney 2052, Australia

^b Department of Applied Physics and Electronics, Umeå University, SE-901 87 Umeå, Sweden

Received 20 February 2006; accepted 6 April 2007

Available online 17 May 2007

Abstract

We present a new control law for the problem of docking a wheeled robot to a target at a certain location with a desired heading. Recent research into insect navigation has inspired a solution which uses only one video camera. The control law is of the “behavioral” type in that all control actions are based on immediate visual information. Docking success under certain conditions is proved mathematically and simulation studies show the control law to be robust to camera intrinsic parameter errors. Experiments were performed for verification of the control law.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Wheeled robots; Navigation; Robust control; Computer vision

1. Introduction

It is currently very popular among roboticists to draw inspiration from the animal kingdom [1,2]. This trend is termed “biomimetics”. Robot navigation strategies thus derived, often categorized as “behavioral” or “reactive” robotics, aim at the construction of simple control strategies which use direct sensory information, rather than a structured environmental model. Such strategies demonstrate an intimate relationship between movement control and vision. The use of vision is recommended when robots must operate in a dynamic environment.

In this paper we present one such control strategy and its experimentation for the problem of positioning a wheeled robot to a target at a certain location with a certain heading, i.e. *docking*, using information provided by a video camera. The kinematics of the robot are non-holonomic, so standard techniques of visual servoing (see, e.g., [3]) cannot be directly applied. We introduce a change of variables and a camera space

regulation condition which allow solution of the problem via a relatively simple nonlinear control law.

This paper draws on previous work in precision missile guidance [4,5] that involved missile guidance with an impact-angle constraint, and was built on a combination of geometrical considerations, and recent work in robust control and filtering theory [6–10].

The remarkable ability of honeybees and other insects like them to navigate effectively using very little information is a source of inspiration for the proposed control strategy. In particular, the work of Srinivasan and his co-authors [11–13], explaining the use of optical-flow in honeybee navigation where a honeybee makes a smooth landing on a surface without the knowledge of its vertical height above the surface. Analogous to this, the control strategy we present, which is originally published by the co-authors [14], is solely based on instantaneously available visual information and requires no information on the distance to the target. Thus, it is particularly suitable for robots equipped with a video camera as their primary sensor.

From a behavioral point of view, the problem of controlling eye-head systems is a fundamental issue for completion of specific tasks [15]. In this paper, we describe and experimentally investigate a vision-based docking system [16] for controlling a wheeled mobile robot approach to a static target using a video camera. The docking system consists of

[☆] This work was supported by the Australian Research Council.

* Corresponding author. Tel.: +61 2 9385 5701; fax: +61 2 9385 5993.

E-mail addresses: may.low@student.unsw.edu.au (E.M.P. Low), Ian.Manchester@tfe.umu.se (I.R. Manchester), A.Savkin@unsw.edu.au (A.V. Savkin).

the behavior-based control law and a vision design. The vision design includes a pan video camera with a visual gaze algorithm that mimics the ability of many living insects to control their direction of gaze, enabling fixation on a specific part of an environment. As a result, it captures images more suitable for completion of a task.

Computer vision-processing techniques [17] allow wheeled robots to understand an environment. Underlying all these techniques is the need to recognize an object of interest in an environment. We use both edge and region detection techniques. Specifically, intersection of the edges of a rectangle produces corners and region information is employed in the form of optical flow. Subsequently, these visual parameters are provided to the control law which regulates the motion of the wheeled robot to the target.

The proposed vision-based robotic docking system was implemented and verified by various experiments using a wheeled robot and a pan video camera in a laboratory setting. In each experiment, the aim was to dock a wheeled robot at a certain location with a different heading. The experimental results demonstrated the effectiveness of the control law.

Docking is required in almost all applications of wheeled robots particularly when mobile wheeled robots are required to recharge their batteries for long term operation. It is envisaged that wheeled robots will play a significant role in search and rescue operations, in port automation and even in autonomous highway systems.

The rest of the paper is organized as follows. Section 2 discusses work in the literature related to the docking problem. Section 3 defines the problem statement. Section 4 introduces the control law for docking a wheeled mobile robot and presents the derivation and mathematically rigorous analysis of the control law. Section 5 includes simulation studies on the robustness of the control law. Section 6 describes the design of the vision-based docking system and reports experimental results. Our conclusions are drawn in Section 7. Lastly, an appendix includes computer vision algorithms used in this work.

2. Related work

Most studies of this problem can be roughly grouped into two approaches. One focuses on the robot's "configuration space", i.e. the relative positions and angles of the robot and target, and perhaps obstacles, in the plane. All these relations are assumed to be available to the control law, and from them it chooses some desirable path. Examples are found in [18–21] and references therein.

The method described in [18] is similar in its approach to the method presented in this paper, in that the aim is to follow to a circular path. The main differences are that, firstly, they assume a slightly simpler kinematic model (often termed the *unicycle* model), and secondly, they are able to prove exponential stabilization to the desired final location, but at the expense of a control law which is more complicated and requires more information.

The other main approach focuses on "camera space" or "visual space". It is no longer assumed that the robot has access

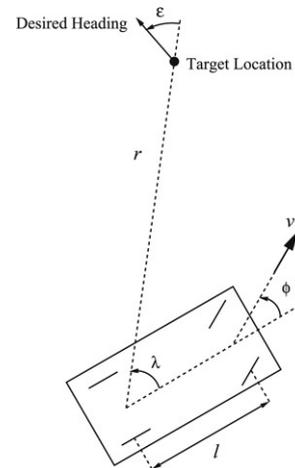


Fig. 1. System geometry.

to the full configuration, but only the image of the target (and obstacles) as the camera sees them. Typically it also knows how they ought to look if the goal is achieved. From this information a control law is assigned which drives the appearance of the target towards its goal. That is, dynamics are examined in camera space. Examples of this approach are found in the papers [22–27] and references therein.

Our paper can be seen as a blend of the two approaches. A simple camera-space condition is defined which, if kept, leads to desirable configuration-space trajectories.

3. Problem statement

Our aim is to design a control law by which a car-like vehicle may dock to a target point. The information available to the control law is consistent with the use of a video camera as the main sensor.

We now described the kinematic model of the robot, the measurements available to it, and finally give a complete definition of the problem statement.

The relative position of vehicle and target is given in polar form (see Fig. 1). The vehicle's position is an extension-less point in the plane, and is identified in a physical system with the mid-point of the rear axle. The scalar quantity r is the range between the vehicle and the target, and the angle ε is the angle between the desired heading and the line-of-sight from the car to the target. These two quantities can be thought of as polar coordinates, placing the vehicle with respect to the fixed target frame.

Two more angles are required to completely characterize the state of the system. These are the heading of the vehicle, and the angle of its steering wheels. The angle λ is the angle between the vehicle's current heading and the line-of-sight. The angle ϕ is the angle of the steering wheels with respect to the centerline of the car, and is controlled with the input u . The forward speed is controlled with the input v .

The reason for this unusual representation of the state will become clear later in the paper, when the *CNG Principle* is described, and the control law derived.

The state-space of the car–target system is then the manifold $\mathbb{R} \times \mathbb{T}^3$ of states $(r, \lambda, \varepsilon, \phi)$, where \mathbb{T} is the circle group: $\mathbb{R} \bmod 2\pi\mathbb{Z}$. The equations of motion on this manifold are given by the following differential equations. These are given for a front-wheel-drive car. To make our control law independent of the forward-velocity of the car, the dynamics are derived with respect to path length, not time.

The change of variables $ds = v \cos \phi dt$ allows us to pass from one representation to another.

Hereafter, x' denotes derivative of a variable x with respect to path length s . The dynamics of the states in this form are given below:

$$\lambda' = \frac{\sin \lambda}{r} - \frac{\tan \phi}{l}, \quad \varepsilon' = \frac{-\sin \lambda}{r}, \quad (1)$$

$$r' = -\cos \lambda, \quad \phi' = \frac{u}{v \cos \phi}. \quad (2)$$

Where l is the distance between the front wheels and rear wheels.

We now discuss the measurements available. Inspired by the elegant instinctual behavior of insects, and the practical need for controlling vehicles with simple sensors, we use a measurement model consistent with a single video camera mounted on the robot, and an optical flow algorithm.

The main restriction felt with this model is that the range to the target, r , is not directly measurable. Furthermore, in certain situations it is unobservable, or weakly observable, from the measurements we do have. For this reason we do not use this quantity in our control law.

The angular position of the dock–target in the field of view is the angle λ . The derivative of this variable is the optical flow of the image. An optical flow algorithm such as [28] can calculate this value.

The angle ε must be known, as it is not an environmental variable, but part of the problem statement. This variable is calculated by assuming the target–heading is defined as an abstract bearing, the heading of the vehicle is dead-reckoned and from this and the angle λ , ε is calculated. Details are provided later in the paper.

Further to the information from the video camera, we need some knowledge of the internal state of the vehicle. Specifically, we assume knowledge of the forward speed v , the angle of the steering wheels ϕ and the distance between the axles l .

3.1. Complete problem statement

Our complete problem statement is this. To find a control law of the form

$$u = f(l, \phi, v, \varepsilon, \lambda, \dot{\lambda}) \quad (3)$$

such that range and angle error at final time, i.e. $r(T)$ and $\varepsilon(T)$, are minimized. Corresponding to this, we make the following definition:

Definition 1. A docking manoeuvre is considered perfect if there exists some finite time T such that

$$r(T) = 0,$$

$$\lim_{t \rightarrow T} \varepsilon(t) = 0.$$

A limit is used in the above definition because if $r = 0$ the angle ε is undefined.

4. Control law

From the optical flow measurements, we can cancel the component due to the robot's rotation ($= v \sin \phi / l$), and retain only the component due to the relative motion of robot and dock–target. We denote this remaining flow \mathcal{O}_f , so:

$$\mathcal{O}_f := \dot{\lambda} + \frac{v \sin \phi}{l}. \quad (4)$$

The control input u is then chosen as:

$$e_h := \lambda - \varepsilon, \quad e_c := \frac{2\mathcal{O}_f}{v \cos \phi} - \frac{\tan \phi}{l}, \quad (5)$$

$$u := lv \cos^3 \phi (ae_c + be_h). \quad (6)$$

Here we can think of e_h as the heading error, and e_c as the curvature error, as the car describes a path toward the target.

The gains a and b should both be positive, and can be chosen with the following guidelines:

- The dynamics of the linear system $e_h'' + ae_h' + be_h = 0$ should represent suitable regulation to the desired path,
- The range $r_0 := 2/a$ should be small enough that divergence from the desired path within this region of the target is acceptable.

A discussion of the reasoning behind this control law, and the tuning guidelines, is presented over the next two sections.

4.1. Control law derivation

The method with which we arrived at the above control law is slightly different from most previous approaches. The control objective is to reach some final state, but rather than trying to derive a controller which provides some type of stability to this state, our approach has two stages.

Firstly, simple geometry allows us to pass from the terminal condition to a condition on the instantaneous configuration of the vehicle, this is what we call the *CNG Principle*. Secondly, from this instantaneous condition we derive a feedback-control law using methods similar to feedback linearization.

The following theorem forms the basis of our control law, and was proved in [4]:

Theorem 1 (Circular-Navigation-Guidance Principle). *Introduce the circle uniquely defined by the following properties: The initial and final positions of the vehicle lies on the circle; the desired final-heading vector at the target's position is a tangent to the circle.*

Suppose that a controller of the form (3) is designed such that the angles λ and ε are kept exactly equal over the full docking manoeuvre, then the vehicle's trajectory will be an arc

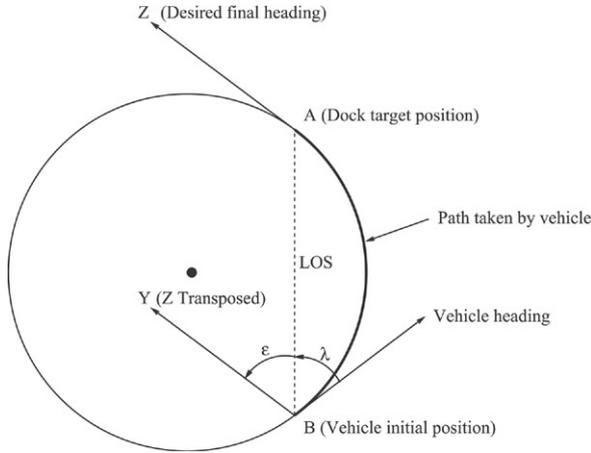


Fig. 2. Geometry for Theorem 1.

on this circle. Furthermore, this will result in a perfect docking manoeuvre, as defined in Definition 1.¹ □

This is visualized in Fig. 2, where the point A is the dock target position, and B the vehicle's initial position. BA, then, is the line-of-sight, and let AZ (equivalently BY) be the desired final-heading vector.

Note that in the case where $\lambda = \varepsilon = \pi$ and $\phi = 0$, the car is heading away from the target, and will continue to do so forever. In a sense, the car is following a circle of infinite radius: a straight line.

It is only in this case, corresponding to just a one-dimensional line in a four-dimensional manifold, where a perfect docking manoeuvre will not occur. Since this is a “thin” set, and would be simple to overcome in practice, we do not consider it further.

In order to regulate λ to be equal to ε , we consider two errors: $\lambda - \varepsilon$ and $\lambda' - \varepsilon'$. The second of these can be expanded as follows, from Eqs. (1), (2) and (4):

$$\lambda' - \varepsilon' = \frac{2 \sin \lambda}{r} - \frac{\tan \phi}{l}, \quad (7)$$

$$= \frac{2\mathcal{O}_f}{v \cos \phi} - \frac{\tan \phi}{l}, \quad (8)$$

giving us Eq. (5).

This can also be interpreted in the following way: Given any position of the car in the plane, relative to the dock target, there exists a unique circle it should follow. To follow this circle, it must have a certain instantaneous heading and curvature. There are then two errors worth considering: heading error and curvature error. e_h is obviously the heading error, and e_c is the curvature error.

This follows, since the curvature of the circle defined in Theorem 1 is given by the function $2 \sin \lambda / r$, and the instantaneous curvature of the vehicle is given by the function $\tan \phi / l$.

If both of these errors are zero, then the vehicle will follow a circular path to the dock target. We can think of these error

functions as describing a two-dimensional target sub-manifold of the four-dimensional state-space: $M := \{(r, \lambda, \varepsilon, \phi) : e_h = 0 \text{ and } e_c = 0\}$.

Viewed like this, our objective is similar to that of sliding-mode control: to regulate the system to a particular sub-manifold on which it is known to behave well.

So we have transformed the terminal-state control problem into an instantaneous-state control problem, i.e. the regulation of e_h and e_c . This is reminiscent of the way a honeybee can land on a surface by regulating certain visual cues. We now tackle this regulation problem in a way similar to input–output linearization (see, e.g., [29], Chapter 13), and analyze the resulting control law using Lyapunov theory.

Let us choose the heading error, $e_h = \lambda - \varepsilon$, as an output function, and attempt to regulate it using input–output linearization.

Differentiating e_h with respect to path-length, we obtain: $e'_h = \frac{2 \sin \lambda}{r} - \frac{\tan \phi}{l} = e_c$.

We differentiate this again, obtaining

$$e''_h = \frac{2 \cos \lambda}{r} e_c - \frac{\sec^3 \phi}{lv} u. \quad (9)$$

In this equation we note that the control appears explicitly, so a natural approach would be to introduce a fictional control input \bar{u} and set

$$u = lv \cos^3 \phi \left(-\bar{u} + \frac{2 \cos \lambda}{r} e_c \right) \quad (10)$$

rendering the dynamics from \bar{u} to e_h linear, in fact just a double integrator. However, since the range r is unknown to the controller, we cannot do this.

We then “almost feedback linearize” the system, and treat the first term in (9) like an uncertainty. The second term is canceled with the nonlinear control law: $u = lv \cos^3 \phi (ae_c + be_h)$

As given in Section 4, then we have

$$e''_h + \left(a - \frac{2 \cos \lambda}{r} \right) e'_h + be_h = 0. \quad (11)$$

If r is large, this is “almost” like the linear system $e''_h + ae'_h + be_h = 0$, and it is clear that, by choosing a and b , both the errors e_h and $e_c = e'_h$ can be made to converge in any desired fashion.

4.2. Control law analysis

Since our control law only “almost” linearized the system, we need some further analysis to understand how the system will behave.

The following simple theorem says this: if we start with zero errors, we will continue to have zero errors and achieve a perfect docking manoeuvre. Another way to put this is that if, at any time, the state $(r, \lambda, \varepsilon, \phi) \in M$ then it will stay in M .

Theorem 2. Suppose the vehicle system (1) and (2) has the desired heading and curvature, i.e. $e_h(0) = 0$ and $e_c(0) = 0$, then the vehicle will perform a perfect docking manoeuvre, as per Definition 1. □

¹ In [4] the definition of perfect intercept was slightly different. However, in the case we consider here it is equivalent to Definition 1.

Proof of Theorem 2. It is clear from the equation of the system (11) that, $e_h(t) = 0$ and $e_c(t) = 0$ at some time t , then they have been, and will be, zero for all time. This implies, then, that $\lambda = \varepsilon$ for all time, and the claim follows from Theorem 1. ■

Now suppose the state starts outside M , that is, with incorrect heading and curvature. Now we'd like to know something about convergence to the target sub-manifold. The dynamics of (11) are those of a linear system with time-varying coefficients, and can be analyzed with Lyapunov theory.

Theorem 3. Consider the function

$$V(e_h, e_c) := be_h^2 + e_c^2. \tag{12}$$

This is a positive-definite quadratic form in the heading and curvature errors, and may be considered as the distance to the target sub-manifold.

Let $[s_1, s_2]$, $s_2 > s_1$ be any path interval over which $V(e_h, e_c, s) \neq 0$ and the following inequality holds:

$$a - 2 \cos \lambda / r > 0. \tag{13}$$

Then $V(e_h(s_2), e_c(s_2)) < V(e_h(s_1), e_c(s_1))$. That is, over any interval of non-zero length, the norm of the errors strictly decreases. □

Proof of Theorem 3. For the proof of this theorem, consider the following linear parameter-varying realization of the system (11), with a state $x = [e_h e_c]^T$:

$$x' = Ax + Bw \tag{14}$$

$$z = Cx. \tag{15}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -b & 2 \cos \lambda / r - a \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [0 \quad 1].$$

Furthermore, consider the Lyapunov function $V(x) = x^T P x$ where

$$P = \begin{bmatrix} b & 0 \\ 0 & 1 \end{bmatrix}. \tag{16}$$

The derivative of this Lyapunov function with respect to distance travelled reduces to

$$V(x)' = -2e_c^2(a - 2 \cos \lambda / r). \tag{17}$$

Now, for any s in the interval $[s_1, s_2]$, it follows from $V(e_h, e_c, s) \neq 0$ that $x(s) \neq 0$. Since x is observable from e_c , it follows that

$$\int_{s_1}^{s_2} e_c(s)^2 ds > 0,$$

and since inequality (13) holds, clearly

$$\delta := 2 \int_{s_1}^{s_2} e_c(s)^2 \left(a - \frac{2 \cos \lambda(s)}{r(s)} \right) ds > 0.$$

Now,

$$V(e_h(s_2), e_c(s_2)) = V(e_h(s_1), e_c(s_1)) + \int_{s_1}^{s_2} V(e_h, e_c)' ds,$$

$$= V(e_h(s_1), e_c(s_1)) - \delta$$

$$< V(e_h(s_1), e_c(s_1)),$$

and the theorem is proved. ■

This theorem reflects the following physically meaningful problem: When the vehicle is very close to the desired target location, large gains are required to make it swing around and track the correct path.

It should be noted that if the range is measurable, either through some other sensor device, or through vision-processing techniques such as stereopsis, optical flow or image looming, this problem will still be present. Indeed, suppose the control law (10) were used, then as the range decreased the gains would become extremely large, due to the $1/r$ term. The actuator constraints on any real system would thus prevent the exact feedback linearization which is attempted.

5. Robustness

It has been mentioned in the literature that a particularly important test of a docking algorithm is the robustness of its terminal positioning precision to imperfect modeling of the kinematics and camera calibration [27,19].

The parameters chosen for the simulation were: $l = 1$ m, $v = 1$ m/s, $a = 4$, $b = 4.04$. The initial conditions were $r(0) = 7$ m, $\lambda(0) = \pi/4$ rad, $\varepsilon = \pi/4$ rad, $\phi = \pi/8$ rad.

These parameters imply that the area in which the path could begin to diverge is approximate $2/a = 0.5$ m. Note that in all simulated cases, the terminal positioning error was much smaller than this.

In all the following simulations, the control law is derived as above, as though all parameters were nominal. We then simulate a system where parameters are perturbed by some amount.

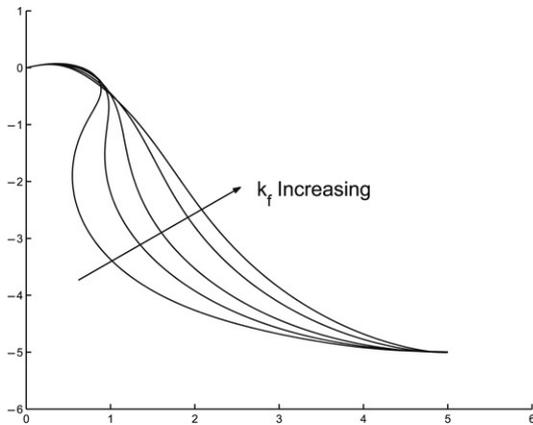
5.1. Camera calibration

Here we simulate the effect of incorrect camera calibration. We skew the measurement of λ and the optical flow in a way consistent with an incorrect assumption on the focal length of the camera. We introduce the ratio k_f as the true focal length divided by the assumed focal length.

This parameter was varied from 0.6 to 1.8. In Fig. 3 we see graphical plots of trajectories, and numerical data for the final range and final-angle error. It is clear that, although the trajectories throughout the middle stage of the docking manoeuvre vary widely, in all cases the robot docked with less than 1 cm positioning error, and less than 10° angle error.

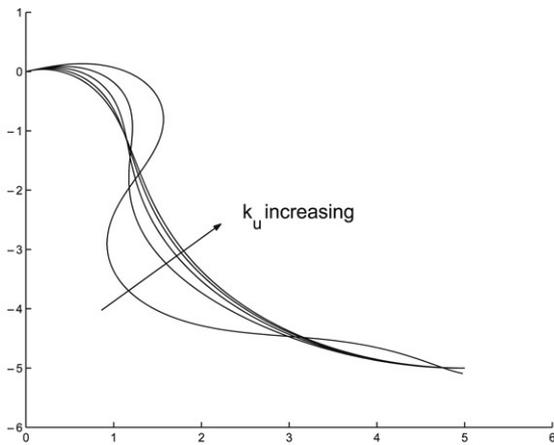
5.2. Control input gain

We now move on to consider errors in the kinematic model, specifically, in the steering wheel system. Firstly, we investigate what happens if the relationship between the control input and steering-wheel movement is not what we expect. Instead of the assumed relation $\dot{\phi} = u$, we instead simulate the system $\dot{\phi} = k_u u$, where k_u is an unknown gain term.



k_f	0.6	0.8	1	1.4	1.8
$r(T) (\times 10^{-2})$	0.44	0.04	0.06	0.23	0.36
$\varepsilon(T)$ (deg)	7.93	2.42	0.12	7.25	9.05

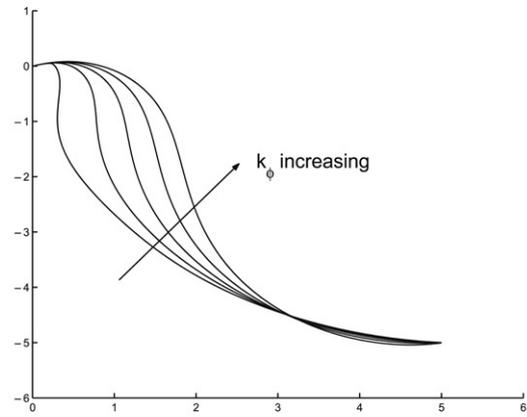
Fig. 3. The effect of incorrect camera calibration.



k_u	0.2	0.5	1	2	5
$r(T) (\times 10^{-2})$	9.22	0.04	0.06	0.07	0.02
Angle Error (deg)	12.89	1.09	0.12	0.10	0.14

Fig. 4. The effect of incorrect $u \rightarrow \dot{\phi}$ gain.

Fig. 4 depicts the trajectories and error data as k_u is varied from 0.2 to 5. It is noted that increasing k_u significantly, which means that our control input is stronger than we expect, has little effect on the performance of the vehicle. A larger effect is observed when the control input is weaker than expected, but performance is still very good.



k_ϕ	0.6	0.8	1	1.2	1.4
$r(T) (\times 10^{-2})$	0.03	0.08	0.06	0.06	0.33
Angle Error (deg)	4.33	2.49	0.12	4.37	13.12

Fig. 5. The effect of erroneous measurement of ϕ .

Reducing it significantly (to around 0.2) results in some large oscillations in the trajectory, and errors in both final position and final angle. However, reduction of k_u even to 0.5, meaning our control input is half as strong as we think, does not significantly degrade performance.

5.3. Measurement of steering-wheel angle

The control law (4)–(6) depends explicitly on our knowledge of the current steering wheel angle, ϕ . In the next set of simulations we consider what happens when this information is wrong. Suppose ϕ is read by a potentiometer which is not tuned correctly, so the resulting measurement is a fixed gain of what it should be. Hence, in the calculation of our control law we replace ϕ with $k_\phi \phi$.

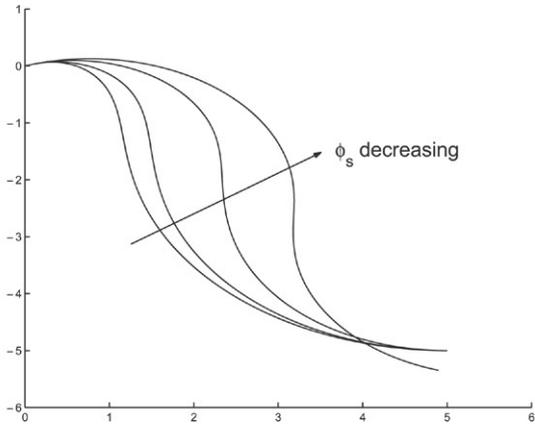
Fig. 5 shows the resulting trajectories as k_ϕ is varied from 0.6 to 1.4. Once again, it is seen that the middle stages of the trajectory are strongly affected, but the terminal errors remain quite small. We note that the terminal position was very small for all cases, but as k_ϕ got very large, the angle error did increase notably.

5.4. Steering-wheel angle saturation

In the last set of simulations, we suppose that the steering-wheel angle is restricted to be within some range of angles. This will obviously be true for many practical robotic vehicles, and essentially results in a lower bound on the turning-circle radius.

We represent this with the constraint $|\phi| \leq \phi_s$.

In Fig. 6 we depict four trajectories, and four sets of terminal error data. These are for the cases where, firstly, the steering wheel angle is not restricted, and then when it is restricted by $\phi_s = \pi/4, \pi/6,$ and $\pi/8,$ respectively. In the first three cases



ϕ_s	∞	$\pi/4$	$\pi/6$	$\pi/8$
$r(T) (\times 10^{-2})$	0.06	0.02	0.07	36.20
Angle Error (deg)	0.12	0.12	0.15	16.90

Fig. 6. The effect of saturation of the steering wheels.

the saturation has little, if any, effect on the performance. In the final case, the performance was significantly degraded, simply because the car could not turn around fast enough to get on the right path.

These simulations show that the control law derived above, which does not explicitly account for steering-wheel saturation, does handle sufficiently small levels of saturation without any degradation in performance.

5.5. Some comments on simulation results

These four sets of simulations show promising prospects for application of our control law when kinematic and camera models are subject to large errors. As has been said in the literature, an important issue in docking problems is robustness of terminal positioning, and in each simulated case above remarkable robustness was observed.

6. A vision-based docking system

In this section, we present a vision-based docking system depicted in Fig. 7 for implementing the behavior-based control law, which is strongly dependent on information from a video camera. We first describe a vision system for recognizing and maintaining a target of interest on an image plane as well as providing control information for the control law. We then discuss the experimental setup and finally report on experimental results.

Our vision-based docking system uses three reference frames as presented in Fig. 8:

- A global frame, $\langle g \rangle$ attached to the static world that is useful for inferring the final heading of the wheeled robot at dock position.
- A robot frame, $\langle r \rangle$ attached to the wheeled robot for describing the velocities of the wheeled robot.
- A camera frame, $\langle c \rangle$ attached to the camera for describing relative motion of the dock target and the camera.

6.1. Vision-based object recognition

A fundamental problem from the computer vision point of view is to design a vision system to recognize an object of interest in an environment. There are many advanced computer vision algorithms to recognize an object of interest in an environment and to achieve detection of image features such as corners or contours, see [30,31] and references therein. However, detailed considerations of these algorithms are beyond the scope of this work. The common understanding among researchers in the area of computer vision is that selection of computer vision algorithms for object recognition is highly application dependent and a compromise between a few practical factors such as robustness of algorithms, amount of useful information which can be extracted from image sequences, computational efficiency and accuracy.

Our vision system is based on the concept of the active vision paradigm [31], which encourages the use of compact and immediate representations of an environment. In other words, the vision system recognizes an environment by recognizing individual landmarks or objects rather than maintaining dense maps of an environment. The key advantage of the active vision

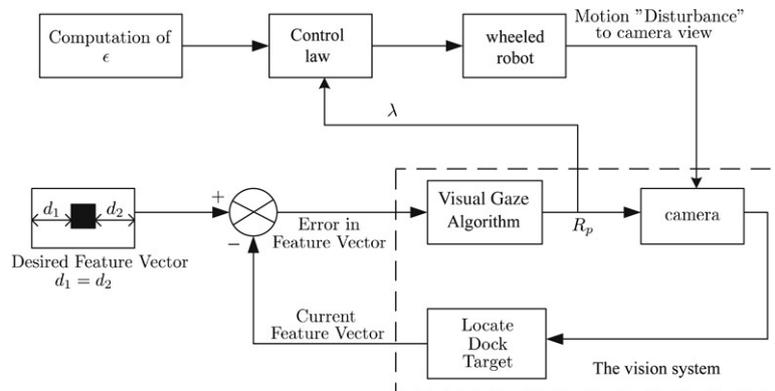


Fig. 7. A vision-based docking system.

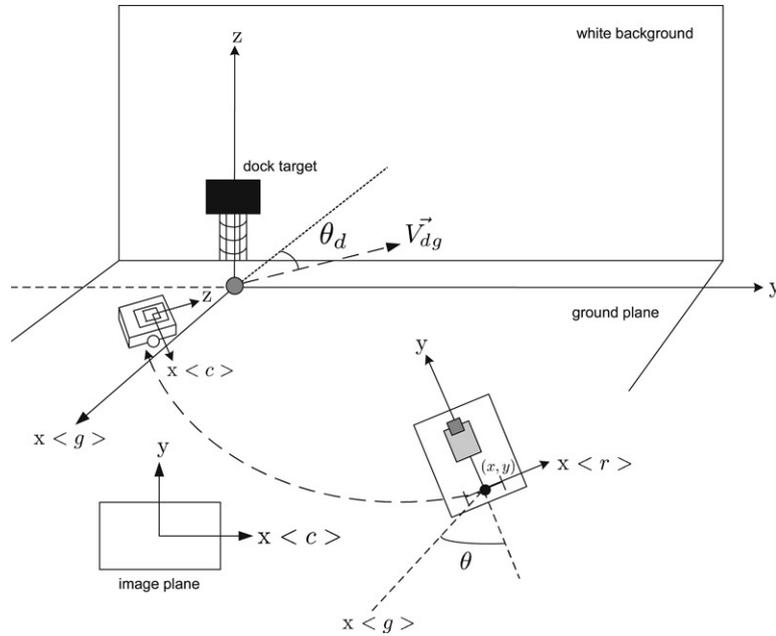


Fig. 8. Position of wheeled robot with respect to the global frame.

paradigm is the use of prior knowledge of an object in an environment to simplify selection and application of computer vision algorithms to image sequences.

To simplify the complexity of selection and computation of computer vision algorithms, we choose the object of interest as a black rectangular cardboard against white background environment. We choose the corners of the black rectangular cardboard as image features of interest. As a result, we use a corner detection algorithm [32] in computer vision to detect the image features. For purposes of completeness, we briefly describe the corner detection algorithm in [Appendix A](#).

In practice, directly applying the corner detection algorithm leads to a resultant of too many corners that are not well localized, as shown in [Fig. 9](#). We provide an algorithm for achieving good localization of corners.

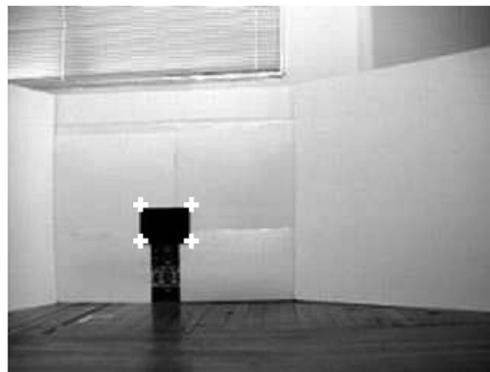
First, we separate the resultant corners into different clusters based on a Euclidean distance measure and each cluster will eventually contain a localized corner. The coordinates of the localized corner is computed by averaging the coordinates of all resultant corners in each cluster.

Second, we are interested in detecting only the four corners of the object, a black rectangular cardboard. To isolate these corners, we assume that the object is always maintained within an adaptive number of pixel rows from the center of the image plane. All localized corners outside this boundary are regarded as false corners. Then by checking that each vertical side of the object will give two localized corners having almost the same column coordinates will further isolate the right corners belonging to the object. [Fig. 9](#) shows four distinct localized corners of the object which are subsequently used as feedback information to control the angular rate of a pan video camera to maintain the object within the camera field of view at all times. Details are given in the following section.

The vision system described in this section is suitable for experimental investigation of various vision-based control strate-



(a) Poor localization of corners.



(b) Good localization of corners.

Fig. 9. Resultant corners computed by corner detection algorithm.

gies in a laboratory setting. Specifically, for experimental investigation of vision-based wheeled robot navigation problems [33] and biologically inspired decentralized control strategies, which arise in very recent research in vision-based multiple mobile robots coordination [34] and flocking [35,36].

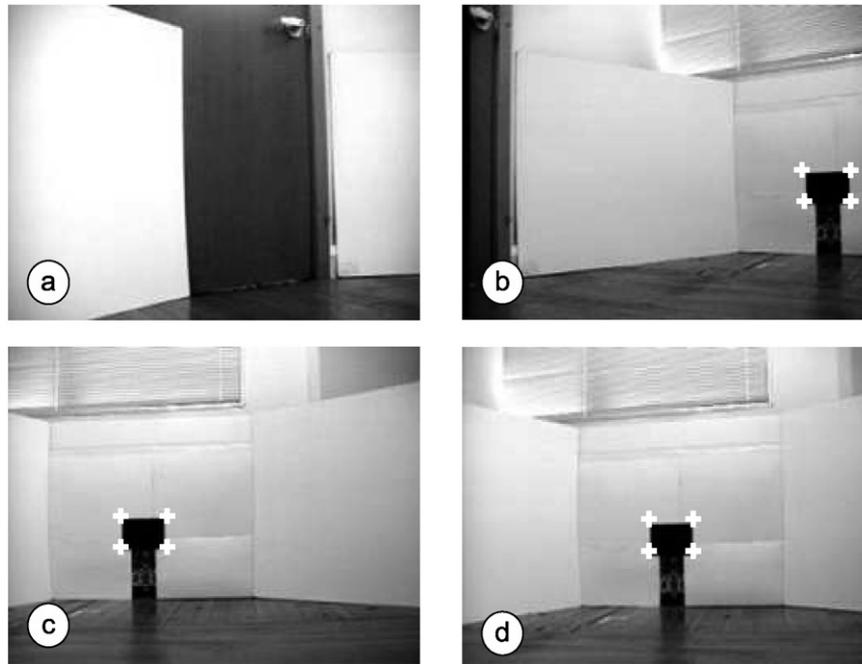


Fig. 10. Saccadic redirection of a pan camera to locate and track an object of interest on the center of image plane.

6.2. A visual gaze algorithm for a pan video camera

A video camera placed on a wheeled robot at any initial position is usually not pointing in a good direction to view relevant parts of an environment where enough visual information can be extracted for completing a task. Furthermore, as a wheeled robot moves, a video camera might lose sight of an object of interest in an environment. Now, the problem is how to ensure a video camera placed on a wheeled robot always maintains an object of interest within a limited camera field of view?

In this section, we present a visual gaze algorithm based on the concept of eye-head coordination [15]. The visual gaze algorithm illustrated in Fig. 7 permits automatic control of a pan video camera that attempts to maintain an object of interest on the center of the image plane. Consequently, continuous visibility of the object for the entire docking process is achieved.

The visual gaze algorithm exhibits the following capabilities:

- When a wheeled robot is located at an initial position and orientation, the pan camera does a 180° scan of the environment to locate the object of interest on the center of the image plane. This is referred as the initialization stage for the camera. Fig. 10 displays an image sequence for the initialization stage which is similar to the behavior of an insect moving its eyes when controlling its direction of gaze.

The initial pan angle of camera, R_{ip} is determined during the initialization stage. The angle R_{ip} is defined as the angle between the heading of the wheeled robot and the optical axis of the camera where the object is on the center of the image plane.

- A proportional controller for the pan camera is employed to maintain an object of interest on the center of image plane while the wheeled robot is moving. The control signal, $R_p(t)$ is computed by evaluating the coordinates of the localized corners on the image plane, which give measurements of two distances d_1 and d_2 from each side of the image plane, illustrated in Fig. 7, and a gain, K_{pan} , given in (18).

$$R_p(t) = R_{ip} - K_{pan}(d_1 - d_2). \quad (18)$$

- In the situation when the motion of the wheeled robot is abrupt, the visual gaze algorithm has the ability to perform a rescan of the environment to search for the object of interest and resume tracking of the object.
- It is understandable that blurred images from video camera are of little use. This problem may occur due to poor video signal transmission or abrupt motion of the robot. The visual gaze algorithm will send a command to stop the motion of the wheeled robot and resume motion once a sharp image is captured.

6.3. Calculation of control information for the control law

In the previous section, we have discussed how to develop a wheeled robot with the ability to recognize and track a dock target in an environment using a pan video camera. In this section, we provide an account of the calculation of visual parameters, namely λ , ϵ and $\dot{\lambda}$ require by the control law, (6).

The relative bearing to the target denoted as angle λ in Fig. 2 can be directly obtained from a pan video camera along with the visual gaze algorithm. The angle λ corresponds to angle $R_p(t)$ in Fig. 11.

We denote \vec{V}_d as the desired final heading vector of the wheeled robot at dock position. Fig. 11 shows that \vec{V}_d in $\langle g \rangle$ is

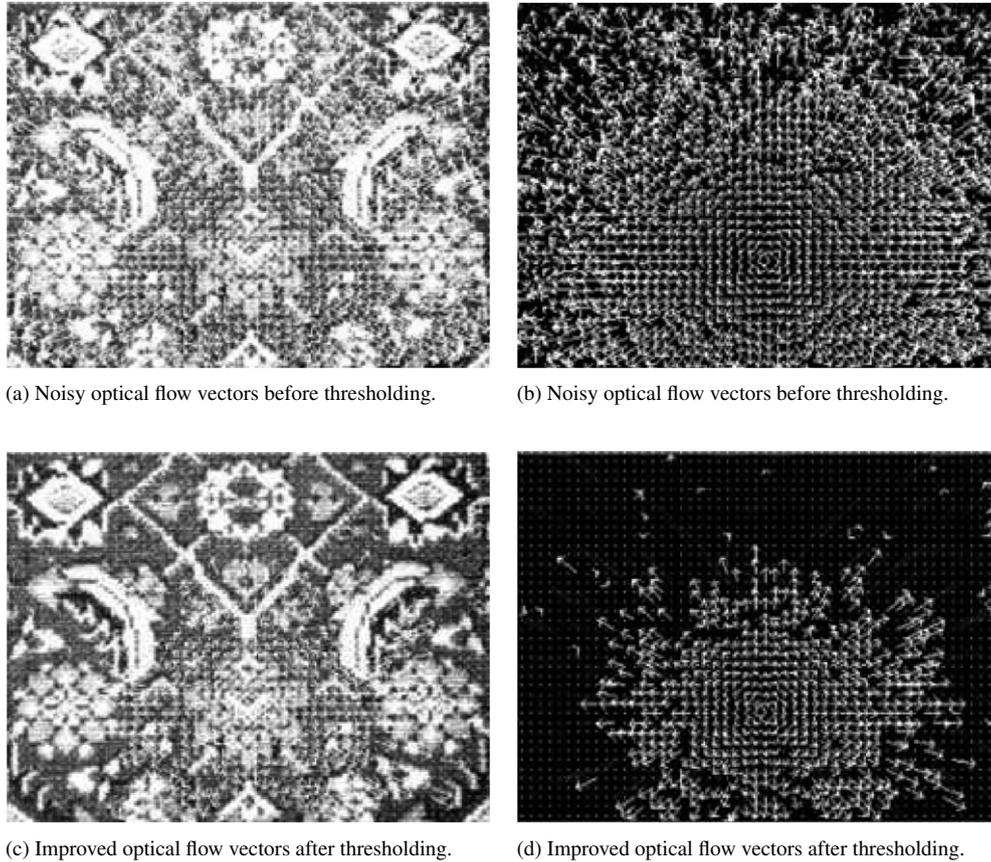


Fig. 12. Optical flow vectors over a surface patch with translational motion.

$$\omega_x = \omega_z = 0.$$

$$\begin{aligned} v_x &= \frac{T_z x}{Z} - \Omega_x, & \Omega_x &= \omega_y(1 + x^2) \\ v_y &= \frac{T_z y}{Z} - \Omega_y, & \Omega_y &= \omega_y xy. \end{aligned} \quad (24)$$

We recover $\hat{\lambda}$, which is represented as $\hat{\omega}_y$, by choosing a region, R , on the image plane where there is sufficient correctness of optical flow vectors based on least square methods in (25).

$$\hat{\omega}_y = \frac{\sum_R (v_y xy - v_x y^2)}{\sum_R y^2}. \quad (25)$$

Lastly, it is required to stop the wheeled robot in front of the dock target. The image of the dock target gets bigger as the wheeled robot moves toward it. The distance between the localized corners of the dock target on the image plane gives an indication of the actual distance between the wheeled robot relative to the dock target in (g). We use this information to stop the wheeled robot when it is approaching near the dock target.

6.4. Experimental setup

The proposed control law was experimentally verified using the vision-based docking system in Fig. 7. This section describes the experimental setup and reports the results.

The experiments were carried out on a Pioneer 3 wheeled robot from ActivMedia. The wheeled robot is equipped with a pan-tilt-zoom (PTZ) color video camera. The control algorithms operate at a calculation period of 0.5 s, computer vision algorithms and data logger were implemented in C++ with the ARIA [38] software development environment running in the Linux operating system. A resolution of 320×240 pixels was selected for image processing.

A simplified version of the control law in (6) was investigated by experiments using a unicycle wheeled robot, shown in Fig. 13. In this case, the proposed control law becomes $\omega := u = ae_c + be_h$. We are concerned with controlling the turning rate of the wheeled robot. The following parameters were used, $K_{\text{pan}} = 0.16$, $a = 0.1$ and $b = 0.32$ and the linear velocity of the wheeled robot is $v = 0.1$ m/s. The v and ω of the wheeled robot are related to the velocity of the left and right wheel of the wheeled robot by $\omega_l = v - l\omega$ and $\omega_r = v + l\omega$ respectively, where l is half the distance between the two wheels.

6.5. Experimental results

Experimental results are presented for two cases where the wheeled robot was located at initial pose of $(x, y, \theta) = (3.0 \text{ m}, 0.72 \text{ m}, \frac{\pi}{2})$ and $(x, y, \theta) = (2.6 \text{ m}, 0.82 \text{ m}, \frac{\pi}{2})$ respectively. The objective of the experiments was to dock the wheeled robot in front of the target which corresponds to a

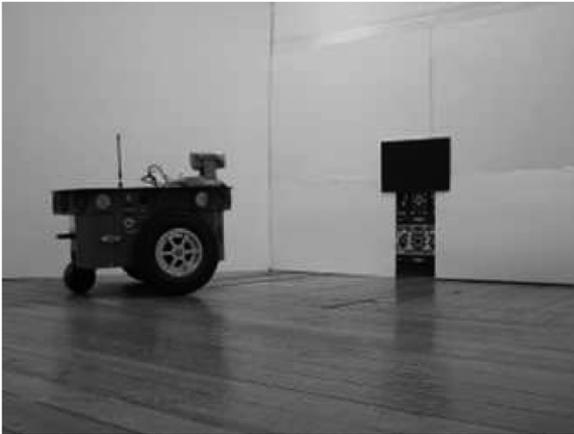


Fig. 13. Experimental setup: wheeled robot and dock target.

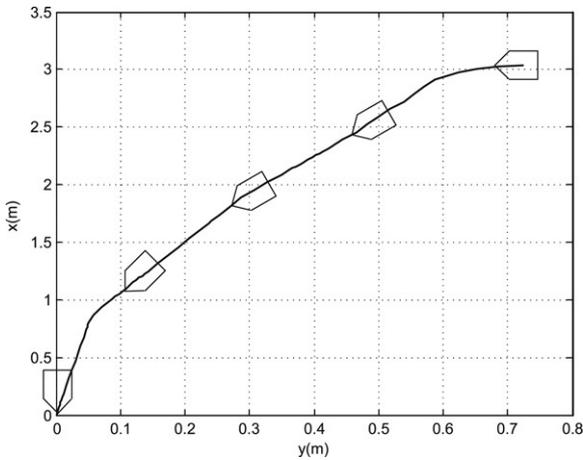


Fig. 14. Resultant trajectory of wheeled robot for $\theta_d = 0^\circ$.

position close to $(x, y) = (\sim 0 \text{ m}, \sim 0 \text{ m})$ with a desired heading of $\theta_d = 0^\circ$ and $\theta_d = 30^\circ$, respectively. Figs. 14 and 16 display the resultant trajectories of the wheeled robot based on measurements from odometry. Figs. 15 and 17–19 contain various results obtained from the experiments. These results are described as follows:

- For all experiments conducted, the final heading, θ_R of wheeled robot was within 0.2 rad.
- The time response of λ was not equal to ϵ at $t = 0$. At some point in time, λ was kept close to ϵ . This is the key condition for the *CNG Principle* in order for a wheeled robot to travel along a circular path to the final position.
- The snapshots in Figs. 18 and 19 display the docking process of the wheeled robot for the two different cases. The snapshots are described as follows. First, display in (a), the wheeled robot locates and locks target. Then, display in (b) and (c), the wheeled robot approaches the dock target based on CNG while fixing on the dock target using the visual gaze algorithm. And, display in (d), the wheeled robot's docking position $(x, y, \theta) = (\sim 0, \sim 0, \sim \theta_d)$ and its camera's view.

Lastly, experimental results also verified the *CNG Principle*. Specifically, by keeping λ equal to ϵ for the entire docking

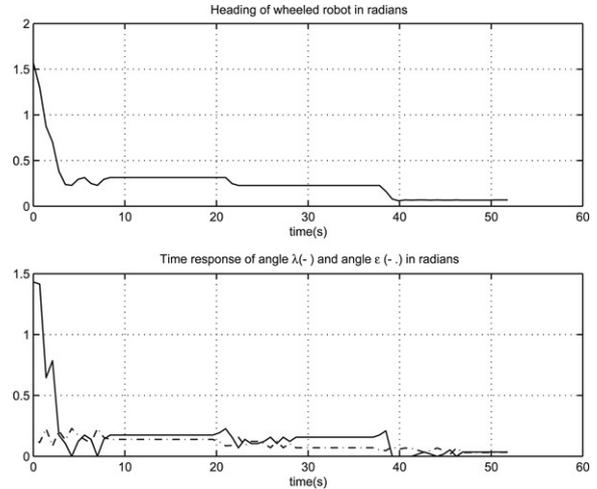


Fig. 15. Time response of angles $\theta_R, \epsilon, \lambda$ for $\theta_d = 0^\circ$.

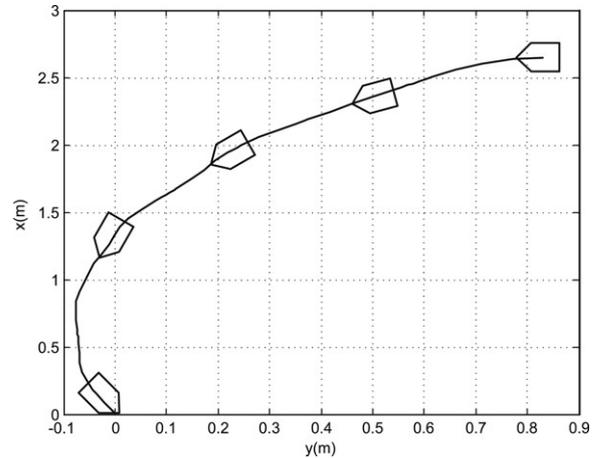


Fig. 16. Resultant trajectory of wheeled robot for $\theta_d = 30^\circ$.

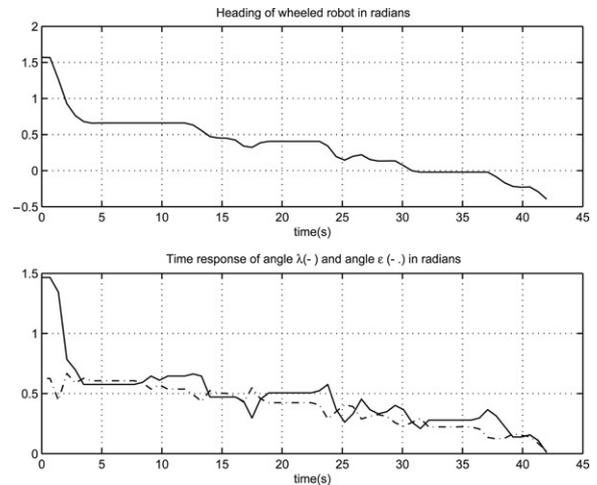


Fig. 17. Time response of angles $\theta_R, \epsilon, \lambda$ for $\theta_d = 30^\circ$.

process, the trajectories taken by the wheeled robot were circular as depicted in Figs. 14 and 16.



Fig. 18. Snapshots of the docking process for $\theta_d = 0^\circ$.

7. Conclusion

We have presented a vision-based docking system for controlling a wheeled robot to perform docking. The docking system consists of a behavior-based control law based on a navigation technique called the *CNG Principle* and a vision system design. The behavior-based control law is strongly dependent on information from a video camera. We have described a vision system design that consists of pan video camera and a visual gaze algorithm which mimics behavior of insects. Also, computer vision algorithms are presented to recognize objects of interest and to compute visual parameters required by the behavior-based control law. The vision-based docking system was experimentally investigated using a wheeled robot and a pan video camera in a laboratory setting.

Experimental results verified the applicability of the control law and the concept of the navigation technique. We believe that the vision-based docking system would have applications to manufacturing industries and autonomous highway systems.

Appendix A. Algorithm for corner detection

For each image frame, $I(x, y)$, the following is required to detect whether a given pixel (x, y) is a corner feature:

- Compute the image intensity gradient in the vertical and horizontal direction, (I_x, I_y) , using the gradient filter. The image intensity gradient is shown in Fig. A.1.
- Let W be a square region of support of $N \times N$ pixels (typically, $N = 5$). At every image point, (x, y) , compute

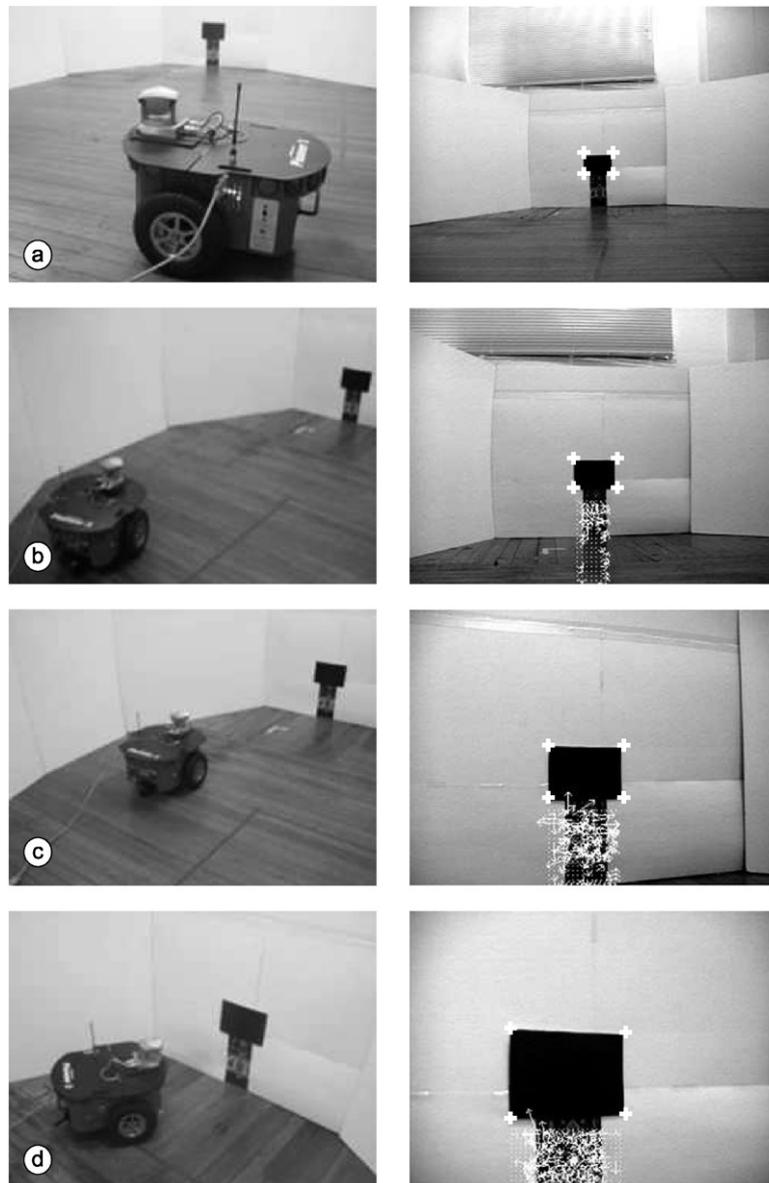


Fig. 19. Snapshots of the docking process for $\theta_d = 30^\circ$.

the matrix, (A.1), using all pixels in the window, W .

$$M = \begin{bmatrix} \sum_{p_i \in W} I_x(p_i) & \sum_{p_i \in W} I_x(p_i)I_y(p_i) \\ \sum_{p_i \in W} I_y(p_i)I_x(p_i) & \sum_{p_i \in W} I_y(p_i) \end{bmatrix}. \quad (\text{A.1})$$

This matrix characterizes the structure of the grey level in M . This is given in the eigenvalues, λ_1 and λ_2 , of M and its geometric interpretation.

If $\lambda_1 = \lambda_2 = 0$, it implies no intensity change in W . If $\lambda_1 > 0$ and $\lambda_2 = 0$, it implies strong gradient change in one direction in W .

If the smallest eigenvalue, $\lambda_1 \geq \lambda_2 > 0$ of matrix M and λ_2 is greater than a prefixed threshold, τ , then the pixel, (x, y) , is considered a corner.

Appendix B. Algorithm for optical flow

Applying Lucas and Kanade, [28], the input is a time-varying sequence of n images, E_1, E_2, \dots, E_n . Let Q be a square region of support of $N \times N$ pixels (typically, $N = 5$).

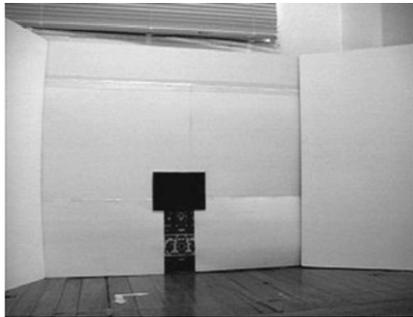
- Prefilter each image with a Gaussian filter of standard deviation, $\sigma = 1.5$ along each dimension.
- The optical flow can be estimated within Q as the constant vector, $\bar{\mathbf{v}}$, that minimizes the function in (B.1).

$$\xi[\mathbf{v}] = \sum_{p_i \in Q} [(\nabla E)^T \mathbf{v} + E_t]^2 \quad (\text{B.1})$$

where p_i is each point within the $N \times N$ patch, Q .

The solution to this least squares problem is given in (B.2).

$$\bar{\mathbf{v}} = \begin{bmatrix} \bar{v}_x \\ \bar{v}_y \end{bmatrix} = (A^T A)^{-1} A^T \mathbf{b} \quad (\text{B.2})$$



(a) Original image frame.

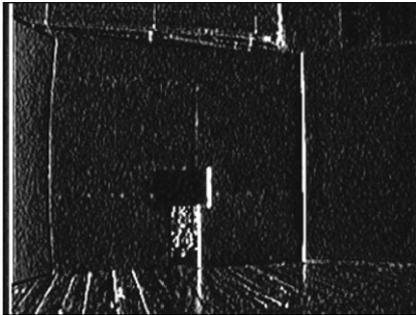
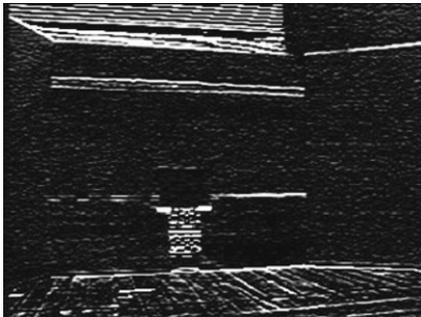
(b) Intensity gradient in the x -direction.(c) Intensity gradient in the y -direction.

Fig. A.1. Image intensity gradients.

where

$$A = \begin{bmatrix} \nabla E(p_1) \\ \nabla E(p_2) \\ \vdots \\ \nabla E(p_{N \times N}) \end{bmatrix} \quad (\text{B.3})$$

and

$$\mathbf{b} = [E_t(p_1) \quad E_t(p_2) \quad \cdots \quad E_t(p_{N \times N})]. \quad (\text{B.4})$$

References

- [1] R.C. Arkin, Behaviour-based Robotics, MIT Press, Cambridge, MA, 1998.
- [2] M.O. Franz, H.A. Mallot, Biomimetic robot navigation, *Robotics and Autonomous Systems* 30 (2000) 133–153.
- [3] S. Hutchinson, G.D. Hager, P.I. Corke, A tutorial on visual servo control, *IEEE Transactions on Robotics and Automation* 12 (5) (1996) 651–670.
- [4] I.R. Manchester, A.V. Savkin, Circular navigation guidance law for precision missile target engagements, *Journal of Guidance, Control, and Dynamics* 29 (2) (2006).
- [5] A.V. Savkin, P.N. Pathirana, F.A. Faruqi, The problem of precision missile guidance: LQR and \mathcal{H}_∞ frameworks, *IEEE Transactions on Aerospace and Electronic Systems* 39 (3) (2003) 901–910.
- [6] I.R. Petersen, V.A. Ugrinovskii, A.V. Savkin, *Robust Control Design using H^∞ Methods*, Springer-Verlag, London, 2000.
- [7] I.R. Petersen, A.V. Savkin, *Robust Kalman Filtering for Signals and Systems with Large Uncertainties*, Birkhauser, Boston, MA, 1999.
- [8] A.V. Savkin, I.R. Petersen, A connection between H-infinity control and absolute stabilizability of uncertain systems, *Systems and Control Letters* 23 (1994) 197–203.
- [9] A.V. Savkin, I.R. Petersen, Nonlinear versus linear control in the absolute stabilizability of uncertain linear systems with structured uncertainty, *IEEE Transactions on Automatic Control* 40 (1) (1995) 122–127.
- [10] A.V. Savkin, I.R. Petersen, Recursive state estimation for uncertain systems with an integral quadratic constraint, *IEEE Transactions on Automatic Control* 40 (6) (1995) 1080–1083.
- [11] M.V. Srinivasan, S.W. Zhang, J.S. Chahl, E. Barth, S. Venkatesh, How honeybees make grazing landings on flat surfaces, *Biological Cybernetics* 83 (2000) 171–183.
- [12] M.V. Srinivasan, S.W. Zhang, M. Lehrer, T.S. Collett, Honeybee navigation en route to the goal: Visual flight control and odometry, *Journal of Experimental Biology* 199 (1996) 237–244.
- [13] G.L. Barrows, J.S. Chahl, M.V. Srinivasan, Biomimetic visual sensing and flight control, *Aeronautical Journal* 107 (1069) (2003) 159–168.
- [14] I.R. Manchester, A.V. Savkin, Vision-based docking for biomimetic wheeled robots, in: *Proceedings of the 15th IFAC World Congress*, Prague, Czech Republic, July 2005.
- [15] G. Cannata, E. Grosso, On perceptual advantages of active robot vision, *Journal of Robotic Systems* 16 (3) (1999) 163–183.
- [16] E.M.P. Low, I.R. Manchester, A.V. Savkin, A method for vision-based docking of wheeled mobile robots, in: *Proceedings of the IEEE International Conference on Control Applications*, Munich, Germany, October 2006.
- [17] B.K.P. Horn, *Robot Vision*, The MIT Press, Cambridge, MA, 1986.
- [18] C. Canudis de Wit, O.J. Sørndalen, Exponential stabilization of mobile robots with nonholonomic constraints, *IEEE Transactions on Automatic Control* 37 (11) (1992) 1791–1797.
- [19] J.P. Laumond (Ed.), *Robot Motion Planning and Control*, Springer-Verlag, New York, NY, 1998.
- [20] P. Souères, J.P. Laumond, Shortest paths synthesis for a car-like robot, *IEEE Transactions on Automatic Control* 41 (5) (1996) 672–688.
- [21] A. Kelly, B. Nagy, Reactive nonholonomic trajectory generation via parametric optimal control, *International Journal of Robotics Research* 22 (7) (2003) 583–601.
- [22] J. Santos-Victor, G. Sandini, Divergent stereo in autonomous navigation: From bees to robots, *International Journal of Computer Vision* 14 (1995) 159–177.
- [23] K. Hashimoto, T. Noritsugu, Visual servoing of nonholonomic cart, in: *IEEE International Conference on Robotics and Automation*, April 1997, pp. 1719–1724.
- [24] S. Lee, M. Kim, Y. Youm, W. Chung, Control of a car-like mobile robot for parking problem, in: *IEEE International Conference on Robotics and Automation*, September 1999, pp. 1–6.
- [25] F. Conticelli, B. Allotta, P.K. Khosla, Image-based visual servoing of nonholonomic mobile robots, in: *38th IEEE Conference on Decision and Control*, December 1999, pp. 3496–3501.
- [26] H. Zhang, J.P. Ostrowski, Visual motion planning for mobile robots, *IEEE Transactions on Robotics and Automation* 18 (2) (2002) 199–208.
- [27] A. Cárdenas, B. Goodwin, S. Skaar, M. Seelinger, Vision-based control of a mobile base and on-board arm, *International Journal of Robotics Research* 22 (9) (2003) 677–698.
- [28] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: *Proceedings of DARPA Image Understanding Workshop*, 1981, pp. 121–130.
- [29] H.K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 2002.

- [30] C. Tomasi, T. Kanade, Detection and tracking of point features, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, April, 1991.
- [31] A. Blake, M. Isard, Active Contours, Springer-Verlag, 1998.
- [32] Y. Ma, S. Soatto, J. Kosecka, S.S. Sastry, An Invitation to 3-D Vision, Springer-Verlag, New York, NY, 2004.
- [33] I.R. Manchester, E.M.P. Low, A.V. Savkin, Vision-based interception of a moving target by a mobile robot, in: Proceedings of the IEEE International Conference on Control Applications, Singapore, October 2007.
- [34] N. Moshtagh, A. Jadbabaie, K. Daniilidis, Vision-based control laws for distributed flocking of nonholonomic agents, in: Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, May 2006.
- [35] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Transactions on Automatic Control 48 (6) (2003) 998–1001.
- [36] A.V. Savkin, Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model, IEEE Transactions on Automatic Control 49 (6) (2004) 981–982.
- [37] A. Dev, B.J.A. Krose, F.C.A. Groen, Confidence measure for image motion estimation, in: Proceedings 1997 RWC Symposium, RWC Technical Report TR - 96001, 1997, pp. 199–206.
- [38] ActivMedia, <http://www.activmedia.com/>.



Emily Low was born in 1979 in Singapore. She is currently a Ph.D. candidate in Electrical Engineering at the University of New South Wales, Sydney, Australia. She received her B.E. (Hons 1) degree in 2003 from Nanyang Technological University, Singapore. From 1999 to 2000, she worked as technologist on navigation technology in the Defence Science Organization, Singapore. From 2003 to 2004, she worked as software engineer on smart card technology in Gemplus Technologies Asia. Her current research interests include vision-based navigation and control of mobile robots.



Ian R. Manchester was born in Sydney, Australia, in 1979. He completed the B.E. (Hons 1) degree in 2001 and the Ph.D. degree in 2005, both in Electrical Engineering at the University of New South Wales, Sydney, Australia. Since 2005 he has held a post-doctoral position in the control systems group at the Department of Applied Physics and Electronics, Ume University, Ume, Sweden. In 2006 he spent a month as a visiting researcher at the Intelligent Systems Research Centre, Sungkyunkwan University, Suwon, South Korea. His current research interests include vision-based guidance and robotics, control of underactuated and non-holonomic systems, and modelling and identification of the human cerebrospinal fluid system. He has published several journal and conference articles on these topics.



Andrey V. Savkin was born in 1965 in Norilsk, USSR. He received the M.S. degree (1987) and the Ph.D. degree (1991) from The Leningrad University, USSR. From 1987 to 1992, he worked in the All-Union Television Research Institute, Leningrad. From 1992 to 1994, he held a postdoctoral position in the Department of Electrical Engineering, Australian Defence Force Academy, Canberra. From 1994 to 1996, he was a Research Fellow with the Department of Electrical and Electronic Engineering and the Cooperative Research Center for Sensor Signal and Information Processing at the University of Melbourne, Australia. Since 1996, he has been a Senior Lecturer, and then an Associate Professor with the Department of Electrical and Electronic Engineering at the University of Western Australia, Perth. Since 2000, he has been a Professor with the School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney.

His current research interests include robust control and filtering, hybrid dynamical systems, missile guidance, networked control systems, computer-integrated manufacturing, control of mobile robots, computer vision, and application of control and signal processing to biomedical engineering and medicine. He has published four books and numerous journal and conference papers on these topics and served as an Associate Editor for several international journals and conferences.