Journal of Intelligent & Robotic Systems (2019) 95:731–743
https://doi.org/10.1007/s10846-018-0896-3

Nonlinear Model Predictive Visual Path Following Control to Autonomous Mobile Robots

Tiago T. Ribeiro[1] · **André G. S. Conceição**[1]

Received: 29 August 2017 / Accepted: 21 June 2018 / Published online: 17 July 2018

Abstract

This paper proposes a novel approach to the visual path following problem based on Nonlinear Model Predictive Control. Simplified visual features are extracted from the path to be followed. Then, aiming to calculate the control actions directly from the image plane, a regulatory model is obtained in the optimal control problem scope. For this purpose a *Serret-Frenet* system is placed in the center of camera's field of view and the optimal control actions generate velocity references to an inner loop embedded in the robot. Stability issues are handled through a classical method and a new approach based on constraints relaxation is proposed in order to guarantee feasibility. Experimental results with a nonholonomic platform illustrate the performance of the proposed control scheme.

Keywords Path-following · Visual control · Nonlinear model predictive control · Autonomous robots

1 Introduction

Recently, in order to get solutions for high precision autonomous robotic navigation, great efforts have been devoted to the development of control strategies based on computer vision [1–3], with special emphasis on the visual servoing based solutions.

Two of the main techniques for visual servoing are the Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS), with the latter being more efficient from computational point of view due to the absence of algorithms for estimating the pose [4]. In both approaches, an important task is to position an end effector to capture a reference image by a camera attached to the robot (for a complete review, see [5] - Chapter 15).

To improve the navigation capabilities of actual applications, these techniques are adopted to cope with multiple reference images [6–8], intrinsic characteristic of applications such as continuous path following based on computer vision. In which case, traditional visual servoing approaches become computationally inefficient due to the need to calculate the inverse of iteration matrix for a large number of visual features.

In the case of visual path following of mobile robots, the objective is to minimize errors obtained from simplified visual features, with the control law calculated directly from the image plane [9–11]. Such requirement of simplicity arises from the needs of maintaining low computational complexity, especially for fast and non-linear dynamic systems.

In many solutions, such features are used to set a lateral displacement and orientation errors relating to a horizon in front of the camera [12]. Such approaches emphasize the need for a concise characterization of the paths to be followed, i.e. constraints on the path curvature, initial configuration and navigation speed. This was dealt with in the past by: proposing control schemes that switch between controllers of image row or column [13], handling obstacle avoiding through a dynamic window approach [14] and establishing an error metric based on the square of the distance between two samples of image [15]. However, the physical limits imposed by the visual system have

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10846-018-0896-3) contains supplementary material, which is available to authorized users.

✉ Tiago T. Ribeiro
tiagotr@ufba.br

André G. S. Conceição
andre.gustavo@ufba.br

[1] LaR - Robotics Laboratory, Department of Electrical Engineering, Federal University of Bahia, Salvador, Bahia, Brazil

not been handled explicitly by neither of the previous works.

An efficient alternative to include these aspects in the calculation of control actions is based on the use of Nonlinear Model Predictive Control (NMPC) since it is an optimal control strategy which handles the constraints of the problem and adapts well to time variant systems [16–19]. To obtain acceptable performance metrics, the practical implementation of NMPC controllers requires a detailed analysis of stability and feasibility requirements. In particular, for the vision based control, where there are constraints in field of view and frame rate, control strategies with such guarantees and efficient in the computational scope should be developed.

A visual MPC approach for steering a nonholonomic mobile robot is proposed in [20]. This linear version of predictive control copes with stability in kinematic and dynamic scope. Nevertheless, the effectiveness of NMPC controllers is lost when dealing with a continous path following problem [21]. Consequently, aspects related to feasibility and computational complexity are not considered.

The control scheme proposed in this study is based on constrained NMPC to follow visual paths considering aspects of stability and feasibility directly from the image plane. Such an approach generates an error model from simplified visual characteristics acquired through a *Serret-Frenet* coordinate system [22] brought into the center of the camera's field of view. Terminal cost and constraints are used to guarantee stability and a constraints relaxation technique is designed to ensure the recursive feasibility. With these solutions and under normal operating conditions, it is possible to ensure that the path remains in the field of view, allowing robots to follow paths with variable curvatures and different navigation speeds, improving the autonomy of the robotic system.

Experimental results using a wheeled mobile robot (WMR) and a webcam (both commercial) with an arbitrary reference path show the advantage of the proposed strategy compared to traditional approaches. Generally, such approaches do not take into account limitations on camera's field of view and frame rate nor the specification of computational cost of the control actions.

It is worth emphasizing that the ideas proposed herein are applicable on other robotic platforms, provided that their kinematic models are known.

The paper is organized as follows. Section 2 states the visual path following problem. Section 3 presents the proposed control strategy and how it deals with stability and feasibility issues. In Section 4, experimental results are presented and discussed. Finally, the conclusions are drawn in Section 5.

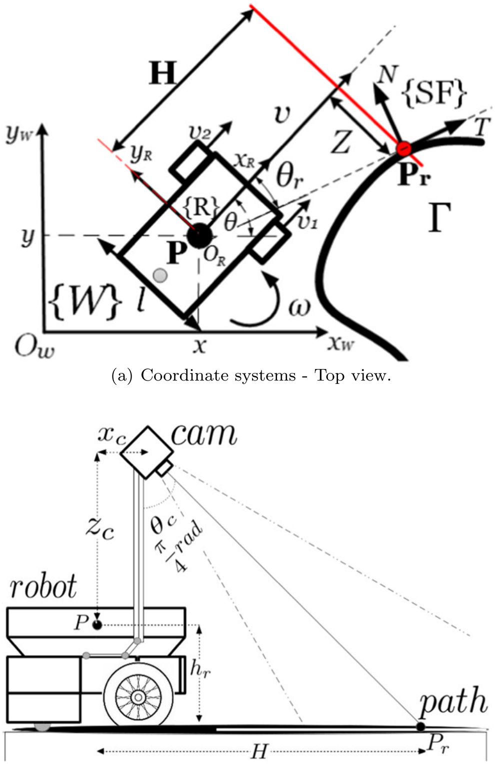
2 Problem Statement

This section is dedicated to the formulation of the path following problem using only visual information. In order to solve the problems related to visual system limitations and its implications on stability and feasibility, the model should allow the immediate application of NMPC controllers, topic of the next section.

Aiming to use simplified visual features, with the requirement of low levels of computational complexity, adaptations are made on the vision-based model proposed in [23] for aplication of classical control methods. In this case, from a desired navigation speed v, the orientation of the robot is controlled aiming to maintain the path to be followed centered in the camera's field of view. Such visual path following approach is particularly applicable in situations with the following conditions:

1. The visual system is positioned in front of the path to be followed;
2. The camera field of view is limited;
3. Image distortions are neglected;
4. The robot kinematic model must be known.

Figure 1a presents the coordinate systems and the parameters used in the modelling for the case of robots



(a) Coordinate systems - Top view.

(b) Robot and Path - Lateral view.

Fig. 1 Visual path following problem representation

with differential drive. Specifically, the world frame $\{W\}$ with unity vectors $[x_W, y_W]$, the robot frame $\{R\}$ with unity vectors $[x_R, y_R]$ and the Serret-Frenet system $\{SF\}$ with unity vectors $[T, N]$ are defined. Using this nomenclature, the kinematic model relative to $\{W\}$ is given as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{1}$$

The forward speed v and the rotational speed ω are related to the wheels speeds v_1 and v_2 as follows:

$$v = \frac{v_1 + v_2}{2}, \tag{2}$$

$$\omega = \frac{v_1 - v_2}{l}, \tag{3}$$

with l being the distance between the wheels.

Additionaly a *horizon* is defined as a straight line on the ground parallel to the front axis of the robot at a distance H of the center of mass (point P relative to $\{W\}$) as shown in Fig. 1b. In this paper H is constant, depending only of the camera pose relative to center of mass and the robot height. The point P_r is considered as the intersection point between the horizon and the path to be followed, Γ. This point is unique if the curvature of the path is continuous and double otherwise. In this case P_r will be the point closest to the longitudinal axis of the robot. Z is defined as the distance between P_r and the robot's longitudinal axis. A Serret-Frenet system $\{SF\}$ is used to characterize the movement of a virtual vehicle positioned in P_r.

Both H and Z are features whose parameters can easily be extracted from the images, since the horizon corresponds to an array of pixels at a distance $d(H)$ of the main point and Z is proportional to the number of pixels between the center of horizon and P_r. Figure 2 presents an example of image with these features. Considering H as a constant, to obtain Z directly from the image, there is a single coefficient of proportionality obtained in the calibration phase of the visual system.

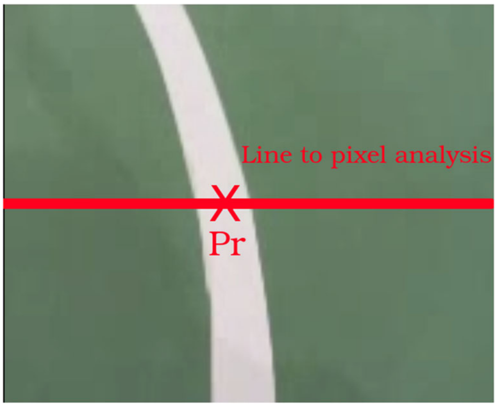


Fig. 2 Image of the path section to be followed

The point P_r (see Fig. 1a) is given as follows:

$$P_r(s(t)) = P(t) + H\mathbf{x_R}(\theta(t)) - Z(t)\mathbf{y_R}(\theta(t)). \tag{4}$$

The time derivative of this expression, omitting the temporal dependencies, is given as:

$$\dot{s}\mathbf{T}(s) = \dot{x}\mathbf{x_R} + H\dot{\theta}\mathbf{y_R} + Z\dot{\theta}\mathbf{x_R} - \dot{Z}\mathbf{y_R}. \tag{5}$$

with s being the curvilinear abscissa along the path.

The relationship between the robot coordinate system $\{R\}$ and the Serret-Frenet system $\{SF\}$ is given as follows:

$$\begin{bmatrix} \mathbf{x_R} \\ \mathbf{y_R} \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{N} \end{bmatrix} \tag{6}$$

Transfoming Eq. 5 into Serret-Frenet system and substituting Eq. 1 we get:

$$\dot{Z} = \omega H + (\omega Z + v)\tan\theta_r; \tag{7}$$

$$\dot{s} = \frac{v + \omega Z}{\cos\theta_r}. \tag{8}$$

Since $\dot{\theta}_r = \omega - \dot{s}c(s)$, where $c(s)$ is the path curvature, we get:

$$\dot{Z} = \omega H + (\omega Z + v)\tan(\theta_r); \tag{9}$$

$$\dot{\theta}_r = \omega - c(s)\frac{(v + \omega Z)}{\cos\theta_r}. \tag{10}$$

To explicit the geometric and temporal behavior of the path to be followed the time derivative of c is obtained by multiplying both sides of Eq. 8 by dc. Thus, to follow a particular path, the following condition must be met:

$$\dot{c} = \frac{dc}{ds}\frac{(v + \omega Z)}{\cos\theta_r}. \tag{11}$$

Considering $\dot{Z} = \dot{\theta}_r = 0$, ω is given as follows:

$$\omega = \frac{cv}{\cos\theta_r + cZ}; \tag{12}$$

$$\omega = \frac{v\tan\theta_r}{Z\tan\theta_r + H}. \tag{13}$$

Equating Eqs. 12 to 13, we get:

$$\theta_r = \sin^{-1}(Hc). \tag{14}$$

Thus, it is possible to verify that an equilibrium condition can be reached only when adjusting θ_r.

Aiming to formulate the visual path following problem as an optimal control problem, the system's control input is defined as $\mathbf{u}_e = \dot{\theta}_r$. With the state vector $\dot{\mathbf{x}}_e = [Z\ \theta_r]^T$, the visual path following error model is given as follows:

$$\dot{\mathbf{x}}_e = f(\mathbf{x}_e, \mathbf{u}_e) = \begin{bmatrix} \dot{Z} \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \omega H + (\omega Z + v)\tan(\theta_r) \\ \mathbf{u}_e \end{bmatrix}. \tag{15}$$

Considering the outputs as the states themselves, the problem of controlling the movements of the robot, based on visual information, can be summarized as follows:

Find ω, *such that* Z *and* θ_r *are feasible.*

Particularly important step for the success of implementing this method is the calculating of the visual path curvature $c(s)$. As Z and θ_r are obtained as rough estimates, without strict requeriments in the visual system's calibration, any error in these measurements will propagate through the $c(s)$ calculation thus hindering the method's accuracy.

In this work, this curvature is obtained with reasonable accuracy through the use of 2 points near to the main point and calculating the radius of the circle passing through these three points. Figure 3 illustrates the procedure.

It's worth pointing out that accuracy of the movements of the robots along the visual path will be of great relevance for the calculus of the visual curvatures. The experimental platform used in this study enabled a satisfatory aplication of this method.

3 Control Approaches

The model described by Eq. 15 is nonlinear and has constraints on inputs and states, justifying the use of computationally efficient optimal control strategies. Approaches based on Model Predictive Control meet some of these requirements due their performance with multivariable, time variant and constrained problems. These controllers have good inherent characteristics of robustness and adapt well to disturbances, non-linearities and modelling errors due to the receding horizon scheme.

An alternative that has widely been used in several branches of the automatic control is the *NMPC* approach, which currently meets the mentioned requirements without a prohibitive computational cost. In the *NMPC* control, the system's output is predicted based on its current states and models. A control profile is obtained in open loop by a numeric optimization, applying only the first control signal to the system. At the next sampling time, a new profile is obtained considering the most recent information.

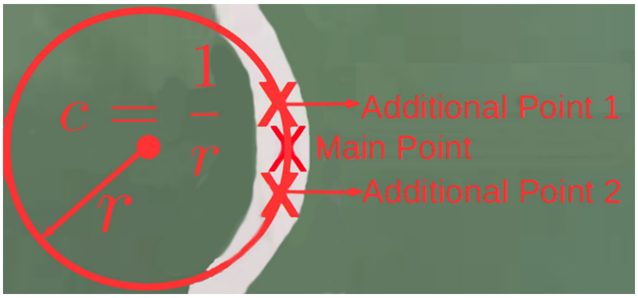


Fig. 3 Procedure to curvature estimation

3.1 NMPC Addressing Stability

Despite of the several advantages of NMPC, there are limitations regarding stability due to the use of a finite control horizon. Stable control can only be guaranteed if the control horizon is large enough. This drastically increases the demand for computational power. Thus, for a practical viability in many systems (especially the nonlinear ones) a small (i.e. stability non-guaranteeing) horizon is necessary.

The problem of stability with a small control horizon is initially addressed through the inclusion of a terminal penalty term ($V(\cdot)$) and a terminal region for the constraints (Ω) to the nominal *NMPC* problem as proposed in [24]. The *NMPC* scheme with stability guarantee takes the following form:

$$J_{\min} = \min_{\mathbf{u}_e} \int_t^{t+T_p} F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau))d\tau + V(\mathbf{x}_e(t+T_p)), \tag{16}$$

$$\text{subject to: } \dot{\mathbf{x}}_e(\tau) = f(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)), \tag{17}$$

$$\mathbf{u}_e(\tau) \in \mathcal{U}, \forall\, \tau \in [t, t+T_c], \tag{18}$$

$$\mathbf{x}_e(\tau) \in \mathcal{X}, \forall\, \tau \in [t, t+T_p], \tag{19}$$

$$\mathbf{x}_e(t+T_p) \in \Omega, \tag{20}$$

with the stage cost F given by:

$$F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) = \mathbf{x}_e^T \mathbf{Q} \mathbf{x}_e + \mathbf{u}_e^T \mathbf{R} \mathbf{u}_e, \tag{21}$$

where:

T_p is the prediction horizon;
T_c is the control horizon; With $T_c \leq T_p$;
$\mathcal{U}$ is the set of feasible inputs;
$\mathcal{X}$ is the set of feasible states and
$\mathbf{Q}$, $\mathbf{R}$ are positive defined matrices that weight the deviations into the required values.

The following Lyapunov function is chosen as a penalty term:

$$V(\mathbf{x}_e(t+T_p)) = \frac{1}{2}\mathbf{x}_e(t+T_p)^T \mathbf{P} \mathbf{x}_e(t+T_p), \tag{22}$$

with $\mathbf{x_e}(t+T_p)) = [Z_T, \theta_{r_T}]$ being the terminal state and $\mathbf{P}$ being positive definite matrix.

Considering a terminal control action $\mathbf{u}_{e_T}$, the following condition must be satisfied:

$$\dot{V}(\mathbf{x}_e(t)) + F(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \leq 0. \tag{23}$$

For a state feedback terminal control action, $\mathbf{u}_{e_T} = -\alpha\theta_{r_T}$, with $\alpha > 0$ and the following formats for weighting matrices:

$$\mathbf{Q} = diag(q_{11}, q_{22}); \tag{24}$$

$$\mathbf{R} = r_{11}; \tag{25}$$

$$\mathbf{P} = diag(p_{11}, p_{22}), \tag{26}$$

from Eq. 15 in Eq. 22 and the result in Eq. 23 we have the following development to meet the stability condition:

$$
\begin{aligned}
&p_{11} Z_T \dot{Z}_T + p_{22}\theta_{r_T}\dot{\theta}_{r_T} + F(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \\
&\quad = p_{11} z_T(\omega H + (\omega Z_T + v)\tan\theta_{r_T}) \\
&\qquad + p_{22}\theta_{r_T} u_{e_T} + F(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \\
&\quad = p_{11} Z_T(\omega H + (\omega Z_T + v)\tan\theta_{r_T}) \\
&\qquad + p_{22}\theta_{r_T} u_{e_T} + q_{11} Z_T^2 + q_{22}\theta_{r_T}^2 + r_{11} u_{e_T}^2 \\
&\quad = p_{11} Z_T(\omega H + (\omega Z_T + v)\tan\theta_{r_T}) \\
&\qquad - \alpha p_{22}\theta_{r_T}^2 + q_{11} Z_T^2 + q_{22}\theta_{r_T}^2 + r_{11}\alpha^2\theta_{r_T}^2 \\
&\quad = q_{11} Z_T^2 + (-\alpha p_{22} + q_{22} + r_{11}\alpha^2)\theta_{r_T}^2 \\
&\qquad + p_{11} Z_T(\omega H + (\omega Z_T + v)\tan\theta_{r_T}).
\end{aligned}
$$

To guarantee stability at terminal stage, the sum of the coefficients of the quadratic terms must be negative. Thus the weighting matrices can be obtained through the following condition:

$$\alpha p_{22} - q_{22} - r_{11}\alpha^2 \geq q_{11}. \tag{27}$$

The decrease in the cost function at terminal stage will be completed with the negativeness of the non quadratic term and the relative magnitude of terminal states. With these considerations, the terminal region of states - Ω is defined as follows:

$$|\theta_{r_T}| \geq |Z_T|; \tag{28}$$

$$p_{11} Z_T(\omega H + (\omega Z_T + v)\tan\theta_{r_T}) \leq 0. \tag{29}$$

This region is bounded by terminal control inputs as follows:

$$\omega = \frac{cv - \alpha\theta_{r_T}\cos\theta_{r_T}}{\cos\theta_{r_T} - cZ_T}, \tag{30}$$

with ω constrained by robot's physical limits as follows:

$$\omega_{\min} \leq \omega \leq \omega_{\max}. \tag{31}$$

3.2 NMPC Addressing Feasibility

In search of solutions with low requirements of computational complexity, this section deals with the following nominal NMPC approach:

$$J_{\min} = \min_{\mathbf{u}_e} \int_t^{t+T_p} F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) d\tau, \tag{32}$$

$$\text{subject to:} \quad \dot{\mathbf{x}}_e(\tau) = f(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)), \tag{33}$$

$$\mathbf{u}_e(\tau) \in \mathcal{U}, \forall\, \tau \in [t, t + T_c], \tag{34}$$

$$\mathbf{x}_e(\tau) \in \mathcal{X}, \forall\, \tau \in [t, t + T_p]. \tag{35}$$

To guarantee recursive feasibility, the sets $\mathcal{U}$ and $\mathcal{X}$ should be well defined at all time instants, taking into account aspects directly related to the visual system.

The optimization problem will be initially feasible if for a given initial state $\mathbf{x_0}$ the set of admissible inputs $\mathcal{U}(\mathbf{x_0})$ is not empty. The feasibility will be maintained over the control horizon when the following conditions are met for every $t_k = 0, 1, \ldots, T_c - 1$:

$$u(t_k) \in \mathcal{U}(\mathbf{x_u}(t_k, \mathbf{x_0})), \tag{36}$$

$$\mathbf{x_u}(t_{k+1}, \mathbf{x_0}) \in \mathcal{X}, \tag{37}$$

with:

$u(t_k)$ being the actual control sequence and $\mathbf{x_u}(t_k)$ and $\mathbf{x_0}$ the solutions of Eq. 17 at instants t_k and t_0, respectively.

Since only feasible control problems generate acceptable solutions, the points $\mathbf{x_0} \in \mathcal{X}$ satisfying $\mathcal{U}(\mathbf{x_0}) \neq \emptyset$ at every instant, are exactly the points for which the implicit NMPC control law μ is well defined.

Formally, the following lemma, introduced in [25, Chapter 8], sets the property of the recursive feasibility:

Lemma 1 *Let $\mathcal{A} \subseteq \mathcal{X}$ be recursively feasible for the nominal NMPC problem described by* Eqs. 32 *to* 35 *with prediction horizon $T_p \in \mathbb{N}$. Then for each $\mathbf{x} \in \mathcal{A}$ the closed-loop solution $\mathbf{x}_{\mu_{T_p}}(t_k, x)$ is well defined for all $t_k \in \mathbb{N}_0$ and satisfies $\mathbf{x}_{\mu_N}(t_k, x) \in \mathcal{A}$ and thus also $\mathbf{x}_{\mu_{T_p}}(t_k, x) \in \mathcal{X}$ for all $t_k \in \mathbb{N}_0$.*

According to the same authors the proof is obtained by straightforward induction using $\mathbf{x}_{\mu_N}(t_k, x)$ and some $\mathcal{A}$ forward invariant for μ_i, i.e., if $f(x, \mu_i) \in \mathcal{A}$ holds for all $x \in \mathcal{A}$. In this paper $\mathcal{A}$ was determined based on practical implamentation aspects, by adding a slacking policy. In such scenario, defining the bounds to $\mathcal{U}_i$ and $\mathcal{X}_i$ based on physical elements (actuation velocities and regulation errors), the initial and recursive feasibility will be limited to small regions of attraction, which implies major limitations for the practical visual path following control. If for some reason the reference path leaves the camera's field of view, the feasibility will be lost holding the control actions to the limits specified in $\mathcal{U}$ causing instability.

The approach proposed herein is to use the structure of the problem to guarantee feasibility. Specifically, since the control input defined in Eq. 15 only exist in the mathematical domain, it is possible to remove any physical sense in the definition of $\mathcal{U}$, i.e. high values are allowed and later are reduced to maximal real values in the experimental platform.

The bounds to the states constraints ($\mathcal{A}$-bounders) are defined as follows:

$$\mathbf{x^{bound}} = \mathbf{x_e^{max}} + \mathbf{x_{tol}}; \tag{38}$$

$$\mathbf{x^{bound}} \in \mathcal{X}, \tag{39}$$

with $\mathbf{x_e^{max}}$ being the vector of the maximum state errors along the prediction horizon and $\mathbf{x_{tol}}$ being the vector of constant tolerances for each state.

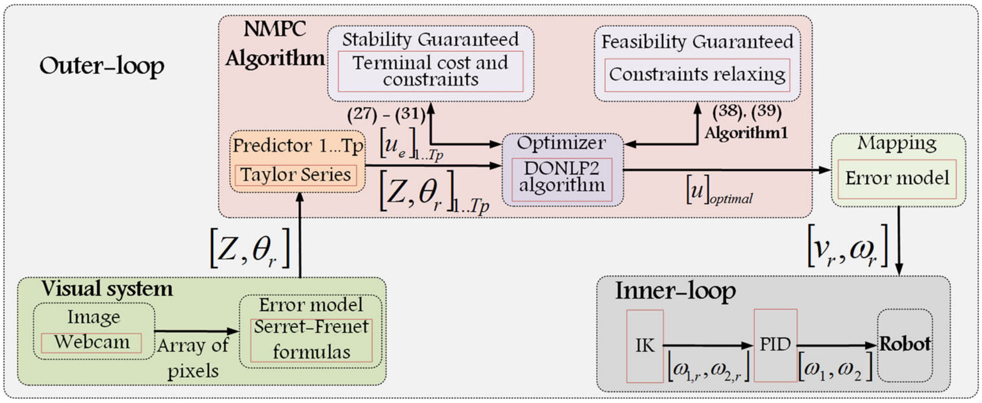


Fig. 4 Robot control structure

Algorithm 1 sinthetizes the proposed scheme to guarantee feasibility.[1]

```
Algorithm 1: Guarantee of feasibility
Data: $\mathbf{x} = [x_0, x_1, \ldots, x_{T_p}]^T$,
      $\mathbf{u} = [u_0, u_1, \ldots, u_{T_c}]^T$, $\mathbf{x}^{tol}$
Result: $\mathcal{U}$ e $\mathcal{X}$
begin
    $\{k \in \mathbb{Z}_+ \mid k > 100\}$
    $\overline{\lim}\,\mathcal{U} = k \sum_{i=0}^{i=T_c} |u_i|$
    $\overline{\lim}\ \mathcal{X} = \max(\mathbf{x}) + \mathbf{x}^{tol}$
end
return $\mathcal{U}$ and $\mathcal{X}$
```

With this formulation suboptimality as proposed in [26] can also be used to guarantee stability, since in such approach feasibility implies stability.

This handling of state constraints (Eqs. 38 and 39) is somewhat similar to soft-constraint approach [27], but without any explicit penalization in the cost function, keeping low levels of computational complexity.

3.3 Control Scheme Implementation

The proposed control scheme is based on the cascade structure, shown in Fig. 4. The inner-loop is intended to control the robot dynamics by tracking the reference speeds given by the outer-loop controller. The outer-loop physical control efforts (v, ω) are obtained by applying the optimal control input, $u(t_k)_{\text{optimal}}$, obtained from the solution of NMPC algorithms, to Eq. 10. More specifically, ω is given as follows:

$$\omega = \frac{\mathbf{u}(t_k)_{\text{optimal}} \cos\theta_r + cv}{\cos\theta_r - cZ}. \tag{40}$$

It is important to note that when addresing feasibility, these variables could have high values, thus ω is saturated to practicable levels as follows:

$$sat(\omega) = \begin{cases} \omega_{\min} & \omega \le \omega_{\min}, \\ \omega & \omega_{\min} < \omega < \omega_{\max}, \\ \omega_{\max} & \omega \ge \omega_{\max} \end{cases} . \tag{41}$$

These velocities are converted into robot rotational wheels' velocities ($\omega_{1,r}, \omega_{2,r}$), through the inverse kinematics block (IK), as follows:

$$\begin{bmatrix} \omega_{1,r} \\ \omega_{2,r} \end{bmatrix} = \frac{1}{r_w} \begin{bmatrix} 1 & \frac{l}{2} \\ 1 & -\frac{l}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{42}$$

where r_w is the wheel radius. These reference velocities are sent to the inner-loop that implements classical PID controllers.

The approach to guarantee stability and feasibility allows positioning the robots at distant points of the paths and navigating in realistic environments with practical velocities. Furthermore, since high valued errors can be regulated, it is also possible to deal with time-varying paths, being a further improvement with respect to approach originally proposed for visual path following.

4 Experimental Results

In these experiments, a webcam *Microsoft® LifeCam HD-3000* [28] was installed on a robot with a differential drive built with *Lego® Mindstorms® EV3* platform [29]. The webcam's offset pose (relative to {R}) was $x_c = 0.05$ m; $y_c = 0.05$ m; $z_c = 0.2$ m; $\theta_c = \frac{\pi}{4}$ rad. Figure 5a shows the mobile robot. An *Arduino Uno* platform implemented the inner-loop and communicated with *EV3* hardware through *I^2C* protocol. The robot communicated through *UART* protocol, under USB standard, with a personal computer that implemented the outer-loop and had the following features: *Intel® Core^{TM} i5 CPU M 430 @ 2.27 GHz X 4; 5.5 GB RAM; 64 bits.*

[1] $\overline{\lim}\ \mathcal{S}$ denotes the upper limit of the set $\mathcal{S}$.

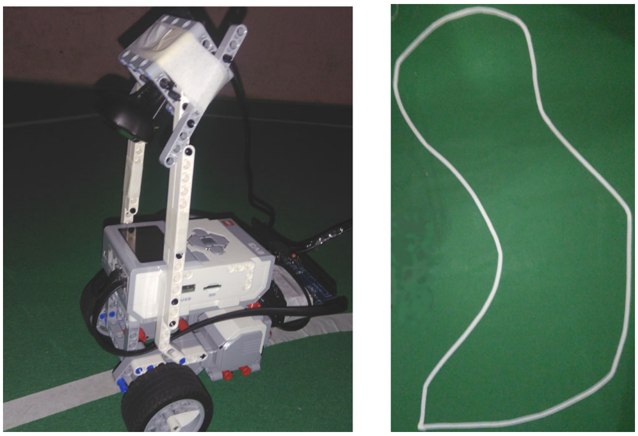

(a) Mobile robot. (b) Arbitrary path.

Fig. 5 Experimental platform

The visual system was calibrated considering the following limiting values to state errors (limits of camera's field of view): $Z = \pm 0.1$ m; $\Theta_r = \pm 0.8$ rad. The optimization problem was solved through the free package DONLP2 [30], a general purpose nonlinear optimizer for continuous variables that implements a sequential quadratic programming method with an active set technique.

The performance of the proposed methods was evaluated in a situation where the curvature function was variable, specifically for the arbitrary path illustrated in Fig. 5b. As can be seen, the variable curvature profile tends to demand more of the control actions in many instants.

Considering the characteristics of the mobile robot, visual system and the path to be followed, the first results were acquired for $v = 0.05$ m/s. It can be observed that for this speed, the path was tracked with a total navigation time of 110 s, generating 550 samples. The nominal NMPC weighting matrices were:

$$\mathbf{Q_i} = diag(10;\, 1), \tag{43}$$

$$\mathbf{R_i} = diag(0.2). \tag{44}$$

The initial bounds of the constraints were defined as follows:

$|u_e| \leq 0.2$ rad/s; $|Z| \leq 0.05$ m; $|\theta_r| \leq 0.3$ rad.

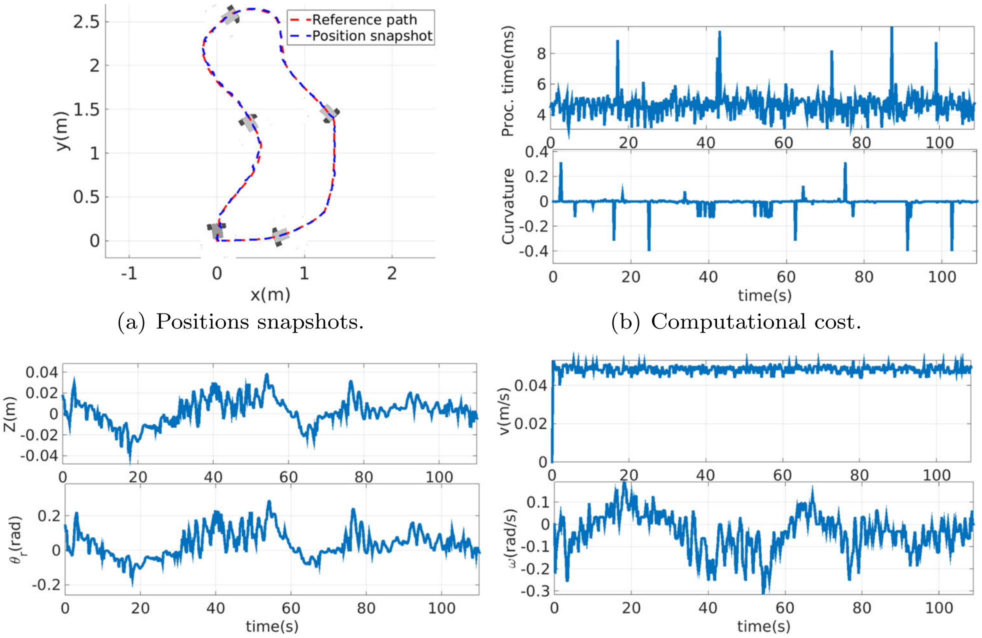
For a sampling time $T_s = 0.2$ s and defining $T_p = T_c = 3$ samples (due to the nonlinearities exposed), the controllers designed were classified as follows:

- Nominal Case (NC), without addressing stability nor feasibility;
- Stable Case (SC), stability is guaranteed by adding terminal cost and constraints;
- Feasible Case (FC), feasibility is guaranteed through constraints relaxation.

4.1 Nominal Case

The results of the nominal controller can be seen in Fig. 6. The computational cost, tracking errors, and the control efforts are shown in Fig. 6b, c and d, respectively. As can be seen, the visual path was closely followed. Figure 6b shows some peaks in processing time that are related to the low accuracy of curvature data processed on line. It is noteworthy (see Fig. 6c) that, in despite of the constraints have been satisfied, the states reached values close to the platform's limits thus reducing the regulation capability.

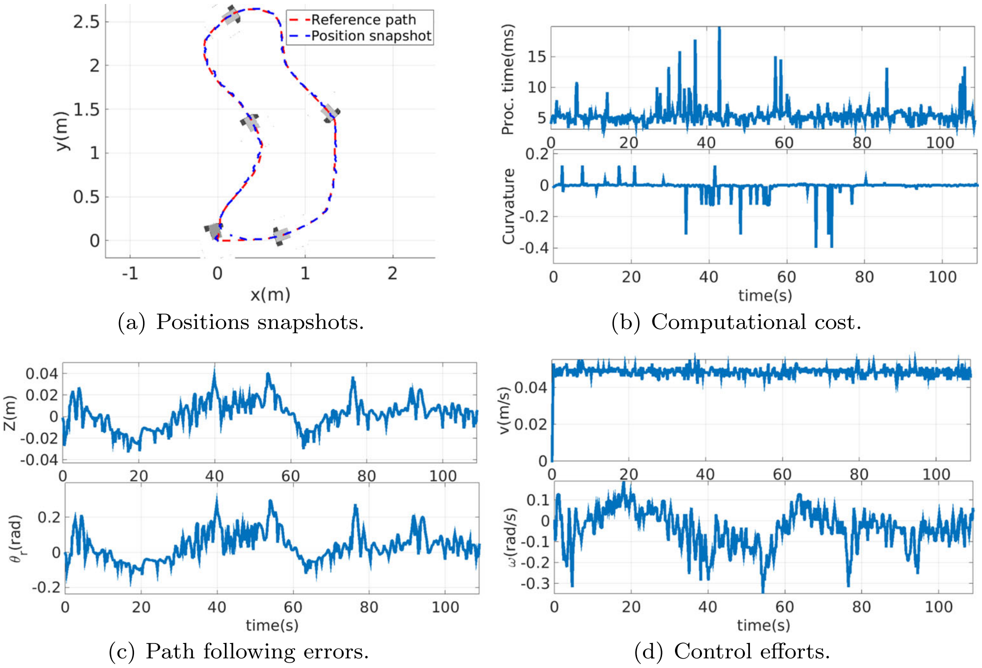
Fig. 6 Visual path following - arbitrary path - nominal NMPC



(a) Positions snapshots. (b) Computational cost.

(c) Path following errors. (d) Control efforts.

Fig. 7 Visual path following - arbitrary path - NMPC with stability guarantee



(a) Positions snapshots.

(b) Computational cost.

(c) Path following errors.

(d) Control efforts.

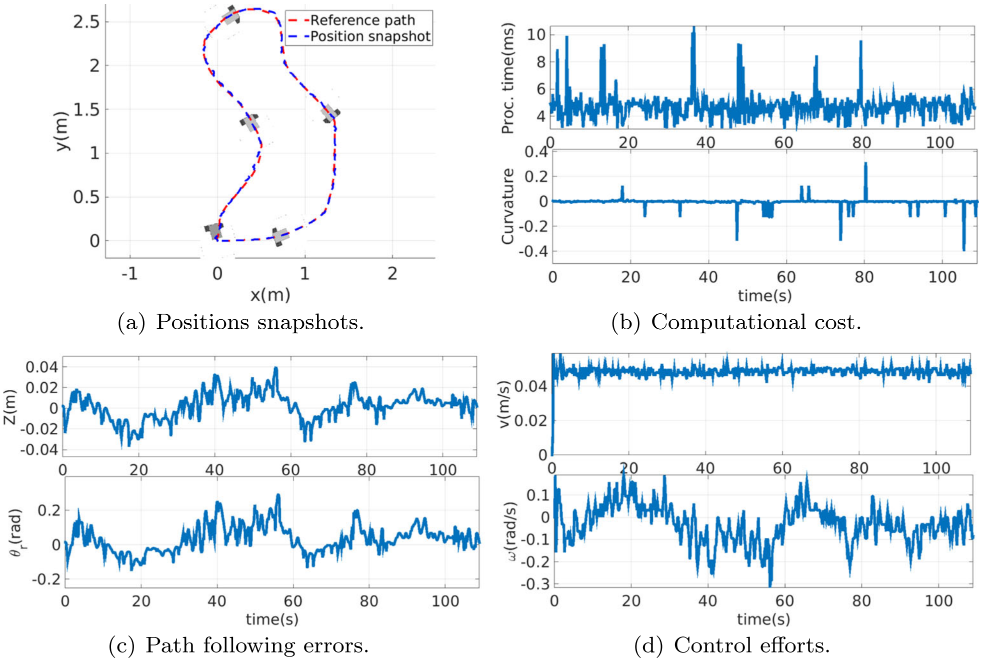
Figure 6d shows that the control efforts remained below the experimental platform's possible maximum values.

4.2 Stable Case

The results for the stable case, where the weighting matrices $\mathbf{Q_i}$ (43) and $\mathbf{R_i}$ (44) were maintained with the same weight, $P = diag(100; 100)$ and $\alpha = 2$ to satisfy the criteria presented in Eq. 27, are shown in Fig. 7.

As can be seen, results similar to the previous experiment were obtained, with a notable variation in the processing time with larger and more frequent peaks (see Fig. 7b). This is an expected result since the curvature captured is less precise due to computation lags.

Fig. 8 Visual path following - arbitrary path - NMPC with guarantee feasibility



(a) Positions snapshots.

(b) Computational cost.

(c) Path following errors.

(d) Control efforts.

Table 1 Quantitative metrics ($v = 0.05$ m/s, 550 samples)

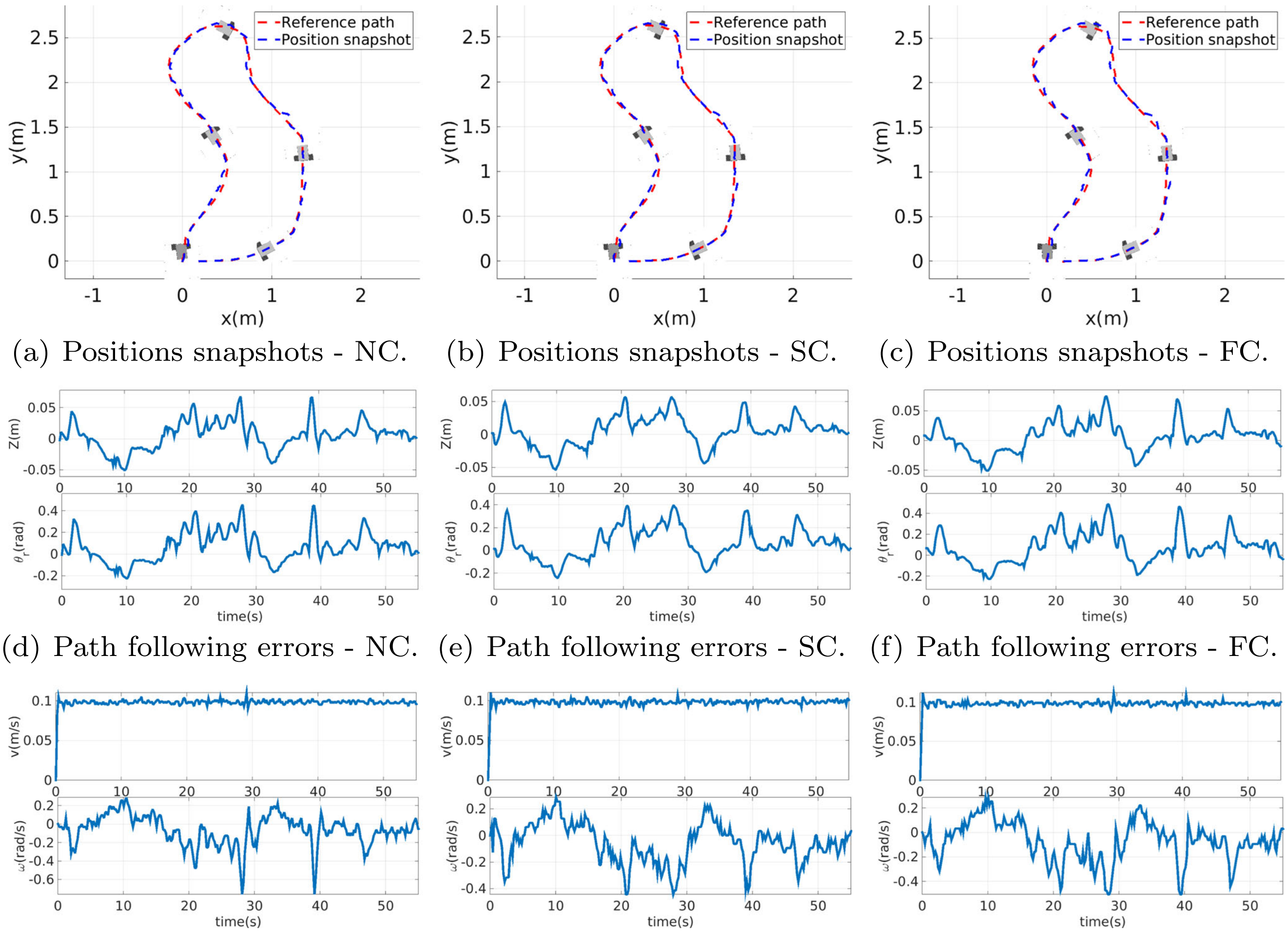
	IAE		*TV*	
	Z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TV} [m/s]	ω_{TV} [rad/s]
Nominal case	1.1126	7.2840	1.1400	17.7760
Stable case	1.1828	7.5952	1.3390	20.1809
Feasible case	1.1146	7.1702	1.2820	17.9080

It is possible to visualize small differences in the figures of the states (see Fig. 7c) and control actions (see Fig. 7d), with the current experiment presenting slightly lower performance, with higher magnitudes and change rates.

4.3 Feasible Case

Complementing this initial analysis, the stage to guarantee feasibility is enabled. For the same tuning parameters, Fig. 8 shows the results. Similar results were obtained for positions snapshots (see Fig. 8a), however, detailed analysis shows a better performance of this technique, since the state errors, especially orientation errors, have remained lower than in the two previous experiments (see Fig. 8c) even with relaxed constraints. The processing time and control actions are close to the nominal case (see Fig. 8b and d) confirming the low computational complexity of the proposal. Figure 8b also shows a more precise profile of curvature function captured in this case. This measure is quite sensitive to the tracking errors, being therefore useful to evaluate the best performance of the technique based on the constraints relaxation.

In order to compare the performance of these approaches, two quantitative metrics were used: the integral of the absolute error *(IAE)* and the total control variation *(TV)*. The *IAE* index, calculated by $\int_0^{T_{END}} |e(t)|dt$, is widely used to compare the performance of distinct strategies in similar experiments. The *TV* index, calculated by $\sum_{k=0}^{k_{END}} |u(k) - u(k-1)|$, aims to evaluate the effect of noise on the control signals, assessing the effectiveness of the controller.



(a) Positions snapshots - NC. (b) Positions snapshots - SC. (c) Positions snapshots - FC.

(d) Path following errors - NC. (e) Path following errors - SC. (f) Path following errors - FC.

(g) Control efforts - NC. (h) Control efforts - SC. (i) Control efforts - FC.

Fig. 9 Approaches comparison for $v = 0.1$ m/s

Table 2 Quantitative metrics ($v = 0.1$ m/s, 275 samples)

	IAE		*TV*	
	Z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TV} [m/s]	ω_{TV} [rad/s]
Nominal case	1.0124	6.4820	0.8170	15.8300
Stable case	1.0228	6.5784	0.9650	12.2760
Feasible case	1.1088	7.1316	0.9190	13.2920

Table 3 Quantitative metrics ($v = 0.2$ m/s, 60 samples)

	IAE		*TV*	
	Z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TV} [m/s]	ω_{TV} [rad/s]
Nominal case	0.4698	2.6262	0.5730	11.8430
Stable case	0.4196	2.3476	0.5290	11.3650
Feasible case	0.3106	1.7324	0.4160	3.0770

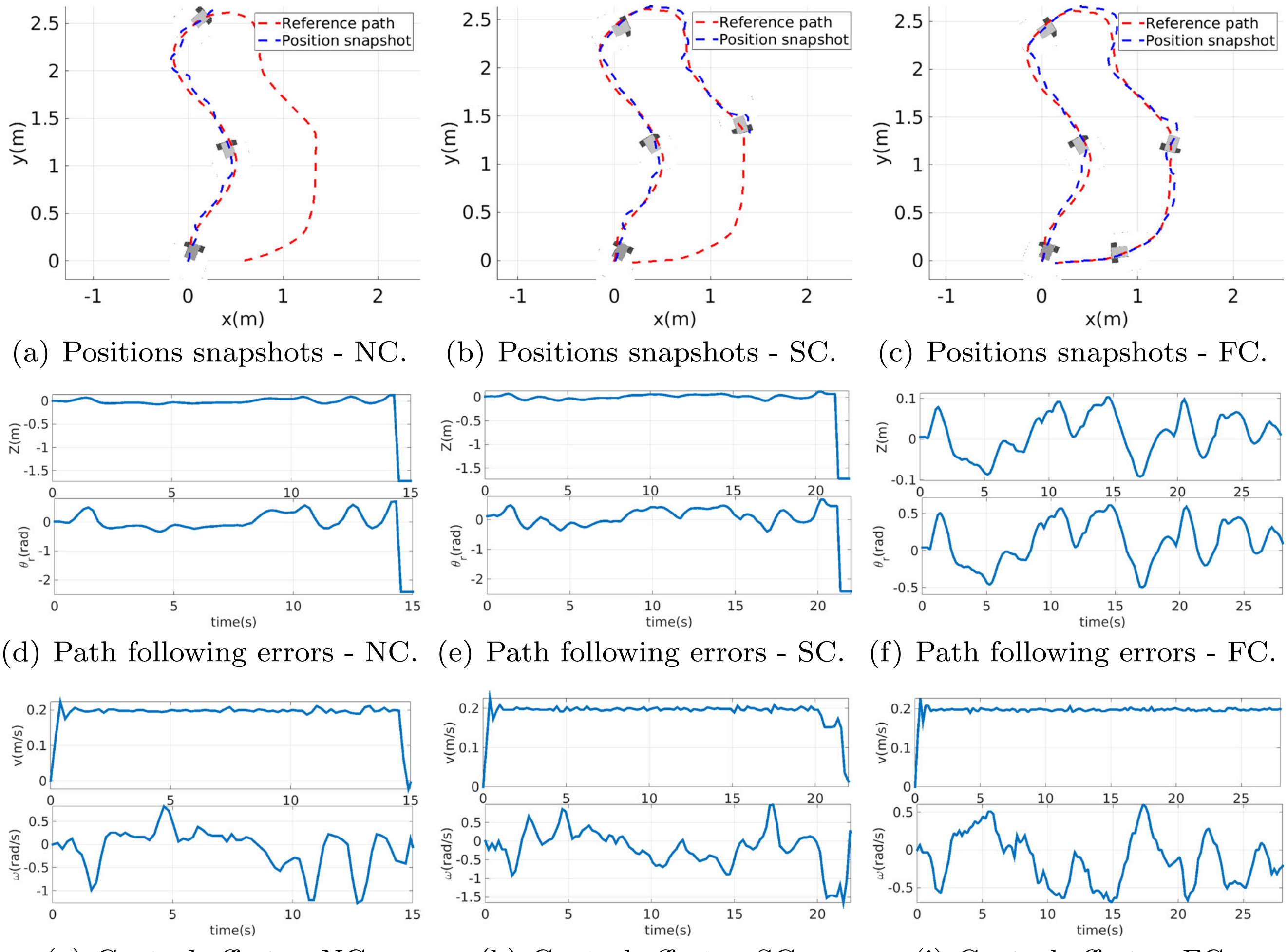
In general, the higher these indices are, the worse the performance of a given approach is.

For the problem at hand, the state errors (Z and θ_r) and the physical control actions (v and ω) were used to calculate the indices presented in Table 1.

Similar values are observed, but with a slight advantage for the Nominal Case for the majority of indices except for $\theta_{r_{IAE}}$, where the best result was in the Feasible Case. The reduction of this variable is fundamental for the actual application, since it ensures that the path remains in the camera's field of view. The Stable Case indices show that due to the experimental platform, the inherent increase in computational cost became unnecessary.

4.4 Speed Influence

The imperfections of experimental platform and the navigation environment impose limitations on the navigation speed, whose values need to be set below the robots maximum. In order to analyze the behavior of the proposed



(a) Positions snapshots - NC. (b) Positions snapshots - SC. (c) Positions snapshots - FC.

(d) Path following errors - NC. (e) Path following errors - SC. (f) Path following errors - FC.

(g) Control efforts - NC. (h) Control efforts - SC. (i) Control efforts - FC.

Fig. 10 Approaches comparison for $v = 0.2$ m/s (60 samples)

Table 4 Visual path following with feasibility guaratee for different speeds

	IAE		*TV*	
v [m/s]	Z_{IAE} [m]	$\theta_{r_{IAE}}$ [rad]	v_{TV} [m/s]	ω_{TV} [rad/s]
0.05	1.1146	7.1702	1.2820	17.9080
0.1	1.0228	6.5784	0.9650	12.2760
0.2	1.2492	7.5438	0.7960	11.9040

methods at higher speeds, more specifically $v = 0.1$ m/s and $v = 0.2$ m/s, new experiments were performed with the same reference path.

From the previous experiments it is possible to conclude that if some formal metric of speed reduction in more curved parts of the path was used, it could be possible to obtain satisfactory results for all control methods, however, it is interesting to explain the limitations of the proposals for constant navigation speeds.

Remaining with all other parameters unaltered, experimental results were acquired for $v = 0.1$ m/s (as shown in Fig. 9). In this case, the path was tracked with a total navigation time of 55 s, generating 275 samples. It is possible to note that the state error constraints were violated in some moments for the Nominal and Stable Cases. The same can not be said for the Feasible Case since the constraints were relaxed in the right measure to guarantee the optimization problem recursive feasibility. Table 2 shows the respective quantitative metrics.

It is worth to note that smaller values are obtained due to the smaller analysis interval (half of the experiments with $v = 0.05$ m/s). Close values are observed for the index relative to state errors, but for control actions total variation there is a slight advantage in the Stable Case. This was expected due to the generation of more conservative movements, necessary for the stable convergence to the terminal states. It is also possible to note that for all experiments the Feasible Case provided more uniform indices with the increase of speed.

From these results, it is natural to expect performance deterioration at higher speeds since the constraints window was fully used and the quantitative metrics were not reduced by half (ideal scenario). This is confirmed through experiments with $v = 0.2$ m/s, illustrated in Fig. 10. It is observed that only the Feasible Case follows the path completely, since at neither moment the path left the camera's field of view. With a very degraded performance, for the Nominal Case, the path was lost in approximately 14 s and for the Stable Case case in approximately 21s, confirming the importance of recursive feasibility for all NMPC approaches. Table 3 shows the quantitative metrics for the first 12 s of movement (60 samples). As can be observed, the only strategy in which there was no substantial increase in all indices was the one based on guaranteed feasibility.

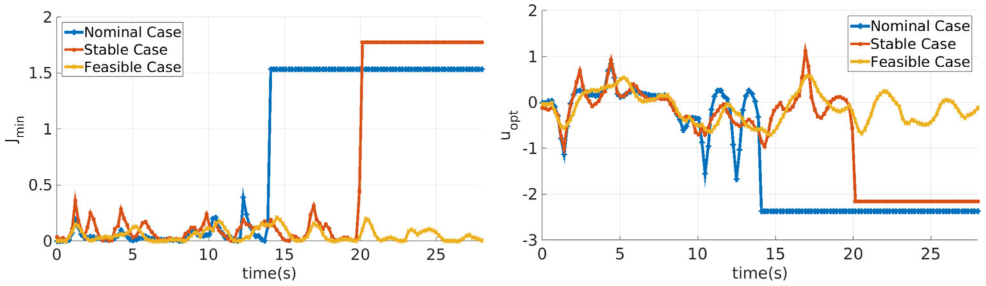
For the Nominal Case there is a natural limitation due to the lack of stability, on the other hand, the approach that provides guarantees for this characteristic suffers from too high a computational cost, not allowing the calculation of control actions by only T_s, not capturing new reference images. In this context, relaxing the constraints without penalty in the cost function arises as an appropriate approach.

The performance indices calculated for all the experiment of Feasible Case for all analyzed speeds (including the $v = 0.2$ m/s, 140 samples) are summarized in Table 4. As can be seen, at double speed, the *IAE* index got worse, on the other hand there was improvement in *TV* since for greater speeds, few variations were necessary to correct the pose. In all cases, the proposed technique for guaranteeing feasibility meets the regulation of state errors with control actions appropriate to the experimental platform and with low processing time, even without any modification in the structure of the cost function.

Figure 11 shows the minimal cost functions ($J_{\min}$) and the optimal control actions of these last experiments. As can be seen in Fig. 11a, the cost function for Nominal and Stable Cases increases few instants before the loss of the path. In these cases Fig. 11b shows the internal control action behaviour in search of a feasible solution. These figures also show the full convergence of the cost function and suitable values to optimal inputs for the Feasible Case.

The computational and visual systems performances are illustrated in Fig.12. As can be seen, the computational

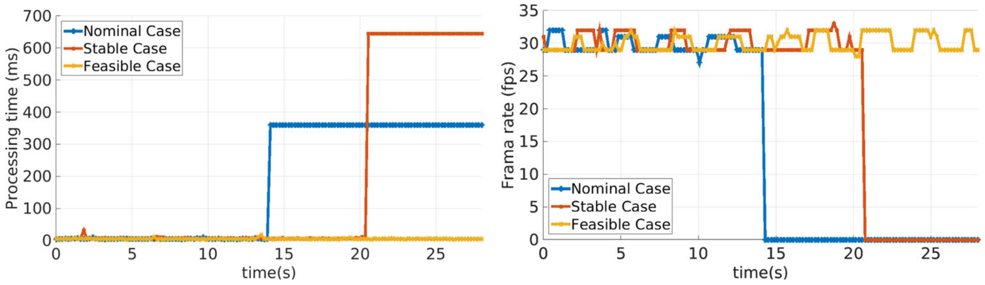
Fig. 11 Approaches comparison: optimization internal variables analysis



(a) Cost functions J_{min}.

(b) Optimal inputs u_{opt}.

Fig. 12 Approaches comparison: computational and visual systems analysis



(a) Computational Cost. (b) Frame rate.

delay due to the optimization activities of the Nominal and Stable Cases (Fig. 12a) causes irreversible information loss represented by null values of frame rate data Fig. 12b. For the Feasible Case an uniform profile is illustrated for both characteristics.

These results confirm that for path following in the image plane, the navigation speed must be specified taking into account characteristics of the experimental platform (Mobile robot + Camera). In the current evaluation, the only approach that can meet the variation of 0.05 to 0.2 m/s was based on the Feasible Case.

5 Conclusion

This paper proposed a new nonlinear model predictive visual path following control aiming to increase the autonomy of a robotic system. This task was performed by inclusion of the visual system in all stages of control architecture. A commercial nonholonomic mobile robot was used to performance evaluation.

The model was based on the extraction of simplified visual features obtained from a *Serret-Frenet* system positioned in the center of camera's field of view and the control law was calculated directly from the image plane, a fact that motivates analysis of stability and feasibility issues since a perturbation on robot structure tends to be amplified in the images, making the path potentially out of camera field of view.

Through experimental results the applicability of NMPC controllers proposed for arbitrary paths was confirmed. For this purpose a path with variable curvature (which for most visual path following techniques hinders directly the specification of navigation speed) was used.

Although the nominal NMPC and the case with a classical approach to guarantee stability based on terminal state and constraints satisfied the visual path following problem at low speeds, the need to increase this speed showed that the only approach with full capability is based on guaranteeing of feasibility. The NMPC controller addressing feasibility has increased the ability to keep the path in the image plane. This was due to the conservation of the recursive feasibility at all time instants, making the processing time low enough to acquire new reference images.

The guarantee of recursive feasibility based on constraints relaxation allows to get a superior performance with full time safe operation in all scenarios, becoming an attractive tool for validation in several other practical applications.

Compared to other works on a similar problem, we believe that our approach presents the following advantages:

- The error model generated allows to get stabilizing control actions directly from the image plane.
- The Visual Path Following based on NMPC allows handling the visual system limitations directly under constraints form, enabling the treatment of time-varying paths.
- The NMPC controller addressing feasibility increases the ability to navigate even with imperfections of surface, camera and mobile robots, improving the levels of inherent robustness.
- The full convergence of the optimization problem showed a possibility for treatment of variable visual reference paths in a simplified way which produces low computational cost solutions.

Future work includes robustness analysis and visual formation controllers based on NMPC with guaranteed stability and feasibility.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Efraim, H., Arogeti, S., Shapiro, A., Weiss, G.: Vision based output feedback control of micro aerial vehicles in indoor environments. J. Intell. Robot. Syst. **87**(1), 169–186 (2017)
2. Kucukyildiz, G., Ocak, H., Karakaya, S., Sayli, O.: Design and implementation of a multi sensor based brain computer interface for a robotic wheelchair. J. Intell. Robot. Syst. **87**(2), 247–263 (2017)

3. Ji, P., Song, A., Xiong, P., Yi, P., Xu, X., Li, H.: Egocentric-vision based hand posture control system for reconnaissance robots. J. Intell. Robot. Syst. **87**(3), 583–599 (2017)
4. Chaumette, F., Hutchinson, S.: Visual servo control. i. Basic approaches. IEEE Robot. Autom. Mag. **13**(4), 82–90 (2006)
5. Corke, P.: Robotics, Vision and Control: Fundamental Algorithms in MATLAB. Springer Tracts in Advanced Robotics. Springer, Berlin (2011)
6. Zhao, Y.-M., Xie, W.-F., Liu, S., Wang, T.: Neural network-based image moments for robotic visual servoing. J. Intell. Robot. Syst. **78**(2), 239–256 (2015)
7. Araar, O., Aouf, N., Vitanov, I.: Vision based autonomous landing of multirotor uav on moving platform. J. Intell. Robot. Syst. **85**(2), 369–384 (2017)
8. Kanellakis, C., Nikolakopoulos, G.: Survey on computer vision for uavs: current developments and trends. J. Intell. Robot. Syst. **87**(1), 141–168 (2017)
9. Frezza, R., Soatto, S., Picci, G.: Visual path following by recursive spline updating. In: Proceedings of the 36th IEEE Conference on Decision and Control, 1997, vol. 2, pp. 1130–1134 (1997)
10. Diosi, A., Remazeilles, A., Segvic, S., Chaumette, F.: Outdoor visual path following experiments. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4265–4270 (2007)
11. Delfín, J., Becerra, H.M., Arechavaleta, G.: Visual path following using a sequence of target images and smooth robot velocities for humanoid navigation. In: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 354–359 (2014)
12. Bertozzi, M., Broggi, A., Fascioli, A.: Vision-based intelligent vehicles: state of the art and perspectives. Robot. Auton. Syst. **32**(1), 1–16 (2000)
13. Cherubini, A., Chaumette, F., Oriolo, G.: An Image-Based visual servoing scheme for following paths with nonholonomic mobile robots. In: International Conference on Control, Automation, Robotics and Vision, ICARCV 2008, pp. 108–113, Hanoi, Vietnam, France (2008)
14. de Lima, D.A., Victorino, A.C.: A visual servoing approach for road lane following with obstacle avoidance. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 412–417 (2014)
15. Sabatta, D.: A vision-based error metric for path following control. 2014 PRASA, RobMech and AfLaT International Joint Symposium (PRASA/RobMech/AfLaT 2014) (2014)
16. Mehrez, M.W., Mann, G.K.I., Gosine, R.G.: An optimization based approach for relative localization and relative tracking control in multi-robot systems. J. Intell. Robot. Syst. **85**(2), 385–408 (2017)
17. Rybus, T., Seweryn, K., Sasiadek, J.Z.: Control system for free-floating space manipulator based on nonlinear model predictive control (nmpc). J. Intell. Robot. Syst. **85**(3), 491–509 (2017)
18. Cao, G., Lai, E.M.-K., Alam, F.: Gaussian process model predictive control of an unmanned quadrotor. J. Intell. Robot. Syst. **88**, 147–162 (2017)
19. Masri, M.A., Dbeis, S., Saba, M.A.: Autolanding a power-off uav using on-line optimization and slip maneuvers. J. Intell. Robot. Syst. **86**(2), 255–276 (2017)
20. Li, Z., Yang, C., Su, C.Y., Deng, J., Zhang, W.: Vision-based model predictive control for steering of a nonholonomic mobile robot. IEEE Trans. Control Syst. Technol. **24**(2), 553–564 (2016)
21. Faulwasser, T., Findeisen, R.: Nonlinear model predictive control for constrained output path following. IEEE Trans. Autom. Control **61**(4), 1026–1039 (2016)
22. Płaskonka, J.: Different kinematic path following controllers for a wheeled mobile robot of (2,0) type. J. Intell. Robot. Syst. **77**(3), 481–498 (2015)
23. Coulaud, J.B., Campion, G., Bastin, G., De Wan, M.: Stability analysis of a vision-based control design for an autonomous mobile robot. IEEE Trans. Robot. **22**(5), 1062–1069 (2006)
24. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica **36**(6), 789–814 (2000)
25. Grüne, L., Pannek, J.: Nonlinear Model Predictive Control: Theory and Algorithms. Communications and Control Engineering. 1st edn. Springer, Berlin (2011)
26. Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Trans. Autom. Control **44**(3), 648–654 (1999)
27. de Oliveira, N., Biegler, L.T.: Constraint handing and stability properties of model-predictive control. AIChE J. **40**(7), 1138–1155 (1994)
28. Microsoft. www.microsoft.com/accessories/en-us/d/lifecam-hd-3000. MIcrosoft LifeCam HD-3000 Product Guide. Accessed 18 Aug 2017
29. Park, E.J.: Exploring LEGO Mindstorms EV3: Tools and Techniques for Building and Programming Robots. 1st edn. Wiley, New York (2014)
30. Spellucci, P.: An sqp method for general nonlinear programs using only equality constrained subproblems. In: Mathematical Programming, vol. 82, pp. 413–448 (1998)

Tiago T. Ribeiro received the B.S. (2010) M.S. (2013) and Ph.D (2017) degrees in Electrical Engineering with emphasis on control and automation from Federal University of Bahia, Salvador, Brazil. In 2017, he joined the Department of Electrical Engineering, Federal University of Bahia, Salvador, Brazil, as an Assistant Professor. His main research interests include robotics, multirobot systems, predictive control, nonlinear systems, process control, and embedded systems.

André G. S. Conceição received the B.S. and M.S. degrees in Electrical Engineering from the Pontifical Catholic University of Rio Grande do Sul (Porto Alegre, Brazil), in 2001 and 2004, respectively, and the Ph.D. degree in Electrical and Computer Engineering from Oporto University (Porto, Portugal), in 2007. Since 2009, he has been an Associate Professor in the Department of Electrical Engineering, Federal University of Bahia (Salvador, Brazil), and a member of the Electrical Engineering Postgraduate Program. As a Visiting Scholar, he spent one year at the School of Electrical Engineering and Computer Science - Queensland University of Technology - QUT(Brisbane, Australia), and one month in the Department of Electronics, Information, and Bioengineering - Politecnico de Milano (Milan, Italy), in 2013 and 2016, respectively. His main research interests include autonomous mobile robots, multi-robot systems, sensor fusion, state and parameter estimation, and process control.