Abstract:
The main objectives of this work are the development of simulation models and the synthesis of controllers for path following, in the scope of mobile robotics. The mathematical model of the mobile robot is nonlinear and subject to a set of constraints at the inputs of the system, as well as in its states. Therefore, hybrid control techniques integrated in a model predictive control strategy are proposed. The proposed methodology allows the respect of the constraints and guarantees the asymptotical stability of the controlled system. The obtained results show that the methodology used for controller synthesis is promising and deserves future development.

Keywords: Mobile robots, hybrid control, piecewise affine systems, model predictive control, multi-parametric optimization

1. INTRODUCTION

In the past few years, the number of applications for mobile robots has increased, therefore these dynamic systems have a high interest of study. The focus of this work is the path following problem, where a camera fixed on the mobile robot is used to measure position and orientation errors relative to the path.

On the class of terrestrial mobile robots, it is often used the wheel configuration named differential drive or unicyle. This robot configuration is very simple, with two independently actuated fixed wheels and one free wheel for balancing the structure. This type of robot presents a superior performance comparing with others configurations He [2005]. Despite all the advantages, this type of robot is a nonholonomic and nonlinear system, which implies a challenging control task. The techniques of linear control do not offer a good performance for a nonlinear system (for all state space) neither reassurance of stability. To overcome this obstacle, control strategies that deal with nonlinearities of the system, in order to obtain good performance for all state spaces, and that guarantee stability, should be considered.

Beyond the soft nonlinearities associated to the robot dynamics, there are hard nonlinearities (for example: motor dead zone), as well as input constraints (for example: saturations) and state constraints (for example: maximum position error relative to the path) of the system, which imply the adoption of modelling/control techniques capable of expressing/dealing with this kind of nonlinearities and constraints.

1.1 Model predictive control

The strategy known as model predictive control (MPC) is commonly applied in the industry, because it is capable to deal with multivariable control problems, subject to constraints Richalet [1993], Camacho and Bordons [1999]. This type of control is used in this work, and a brief resume about the fundamental concepts of MPC are presented next.

The main objective of MPC is to obtain the control action by repeatedly solving an on-line optimal control problem based on the system model, using the actual states of the system as initial states. The optimization problem gives a sequence of optimal inputs but only the first is applied to the system Mayne et al. [2000]. At the next sampling time, the state of the system is measured again and a new optimization problem is solved. With this strategy, we obtain a feedback control law. This control strategy is often denoted in literature as Receding Horizon Control (RHC).

Figure 1 presents the basic structure of MPC, being the main components of the predictive model, the cost function to minimize, the optimizer to obtain a control law, and the performance constraints which the system is subjected to.

In fact, MPC has great potential, compared with other control strategies, to deal with constraints in the states.
and inputs of the systems, as well as with nonlinearities Michalska and Mayne [1993].

1.2 Definition of a piecewise affine system

Consider the following discrete time piecewise affine system,

\[ x(k+1) = A_i x(k) + B_i u(k) + e_i \]

where \( x(k) \) is a polytope set. \( A_i, B_i, F_i \) and \( G_i \) are real matrices with appropriate dimensions, \( e_i \) is a real vector, and \( A_i \) is a polytope set. \( A_i, B_i, F_i \) and \( G_i \) are real matrices with appropriate dimensions, \( e_i \) is a real vector, and \( A_i \) is a polytope set. \( x(k) \) is a polytope set.

The initial state is known (measurable) and is represented, respectively, the input and the states at instant \( k \).

\[ x(0) = x_t \]

The index \( i \) belongs to a finite set, i.e. \( i \in \{1, \ldots, s\} \), and represent the system mode (dynamic) or the active partition.

Each partition \( \Omega_i \) is defined as the intersection of linear planes at state/input space, so each \( \Omega_i \) is a polytope set. \( A_i, B_i, F_i \) and \( G_i \) are real matrices with appropriate dimensions, \( e_i \) is a real vector, and \( e_i \) is a polytope set.

Also consider that,

\[ \Omega = \bigcup_{i=1}^{s} \Omega_i \]

where \( \Omega_i \) denotes polytope interior \( \Omega_i \). i.e. partitions interiors are disjoints.

In order to use the piecewise affine system (1)-(2) for controllers synthesis, the performance constraints must be respected.

Considering that the piecewise affine system is also subjected to performance constraints set in state/input space defined as

\[ \left[ \begin{array}{c} x(k) \\ u(k) \end{array} \right] \in \mathbb{C} \triangleq \left\{ \left[ \begin{array}{c} x(k) \\ u(k) \end{array} \right] : Kx(k) + Lu(k) \leq n \right\} \]

where \( K \) and \( L \) are real matrices and \( n \) is a real vector, all of them with appropriate dimensions.

Obviously, we assume that

\[ \mathbb{C} \subseteq \Omega \] (5)

which means that performance constraints must be contained in the region where the dynamic is defined.

1.3 Modelling and control for mobile robots

The approach of Guerra et al. [2004] to model the mobile robot is different from classic modelling, where the state variables are the Cartesian coordinates of the position of the mobile robot, resulting in a nonlinear system. To avoid the nonlinearities, they purpose a discrete model based on the distance taken at a sampling time \( \Delta l \). The resulting model is linear, time invariant and can be identified by classic methods of system identification. Although distance \( \Delta l \) cannot be directly measured, this problem can be solved by an estimation of \( \Delta l \), based on an approximation to a second order curve.

Asensio and Montano [2002] presents a model where the parameters of the dynamics allows to separately adjust the translation and rotation behaviors of the mobile robot. The model considers the kinematics and the dynamics constraints of the robot, resulting into velocities and accelerations with realistic values. Therefore, due to the precision of the model it is possible to elaborate a controller based on it.

In He [2005] a feedback linearization controller for mobile robots is presented. The mobile robot used is a differential-drive wheeled robot, which results on a highly nonlinear system. Concluding that classic control methods are not suitable to this kind of control strategy.

The authors of Fierro and Lewis [1998] presented a structured control for nonholonomic mobile robots in which a kinematic controller can be integrated with a controller based on neural networks for binary control. The combined controller kinematic/neuronal networks is developed using backstepping and stability is guaranteed by Lyapunov theory. The neuronal network controller can deal with uncertainty and/or perturbation to the dynamical system.

1.4 Objectives and organization

The main objectives of this work are the construction of simulation models of the mobile robot with a camera and synthesis of controllers for path following based in model predictive control. The proposed approach for synthesis of controllers has two important steps. The first step is the approximation of a nonlinear system to a piecewise affine system. This method has the advantage of dealing with hard nonlinearities, for example camera limitation. The second step, we apply the methodology of model predictive control dual mode. This control strategy has the advantages of dealing with multivariable problems and associated constraints.

This article is organized as followed: Section 2 describes the platform and mathematical modelling of the used mobile robot. Section 3 presents the simulation models developed in Simulink. Section 4 deals with the approximation of nonlinear systems to piecewise-affine system and synthesis of controllers based on model predictive control, using multi-parametric programming to obtain a control law that is an explicit function of the variable states. The results obtained in simulation for path following are presented in Section 5. Finally, in Section 6 the conclusions and recommendations for future works are presented.

2. SYSTEM PLATFORM AND MODELLING

This section presents the platform and mathematical modelling of the mobile robot used in this work which is named “Rasteirinho”.

2.1 System platform

The mobile robot has two independently actuated wheels and one free wheel to balance the structure. This configuration is called differential-drive, which offers a good mobility and simple construction. Each motor need a gearbox to decrease the angular velocity and increase the torque. In the mobile robot one (or more) batteries are installed to provide electrical energy to each motor. The motor DC used are nonlinear systems, due a dead zone. This zone is characterized by lack of response when a tension different from zero is applied to the terminals of the DC motor.

The mobile robot chassis is a square aluminium plate, where all components are installed. The wheels have a
radius of 0.035m and the weight of the mobile robot, without the notebook, is approximately 2kg.

The interface with the computer (where the path following controller is implement) and the electrical motors is done by a data acquisition and control card. This card has a USB port and is very simple to connect a computer, allowing data transfer between the computer and the actuators of the mobile robot. The interface with the card and the actuators, it was developed a dispositive with a microprocessor capable to transform the signal PWM output of the card to a tension at the terminals of the electrical motor. The data acquisition and control card have various inputs and outputs, so it is possible to install various sensors on the mobile robot (analog or digital).

In Cardeira and da Costa [2005], this robot is described in detail.

2.2 Mathematical modelling

DC motor The mobile robot is actuated by electrical motors and the mathematical modelling of this subsystem is important for a better understanding of the system’s behavior, such as the controller design.

Figure 2 shows a simplified electrical circuit equivalent to a electrical motor. The armature is modelled as series circuit with a resistance \( R_a \) and an inductor \( L_a \). The voltage source \( e_b \) represents the back-emf of the armature when the rotor spins. The magnetic flux is represent by \( \phi \). Based on figure 2, it is possible to obtain the transfer function between angular displacement of the motor and the tension applied at the terminals.

\[
\theta_m(s) = \frac{E_a(s)}{I_a(s)} = \frac{L_a J_m s^3 + (R_a J_m + B_m L_a) s^2 + (K_b K_i + R_a B_m) s}{K_i}
\]

where \( L_a \) is the inductance of the armature (T), \( R_a \) is the resistance of the armature (Ω), \( \theta_m(t) \) is the rotor position (rad), \( K_b \) is the back-emf constant, \( \omega_m(t) \) is the rotor angular velocity (rad/s), \( J_m \) is the rotor inertial and \( B_m \) is the viscous friction coefficient. This transfer function do not take in account the load torque \( T_l(t) \).

The transfer function in 6 shows that the DC motor is a integrator between two variables. If \( e_a \) is a constant input of the system, output \( \theta_m \) will behave like an integrated output.

Mobile robot This section presents the kinematic Wairyudi [2004] and dynamic Asensio and Montano [2002] models of the mobile robot.

Figure 3 shows the geometric parameteres of the model \((r, b, s)\), global referential \(W\) and the referential \(M\) associated to the mobile robot. In the kinematic model, we assume there is no slipping between the floor and wheels of the mobile robot and the motion equations are defined as rigid body. With this assumptions and define the position and orientation of the mobile robot relative to global referential \(W\), and obtain the following kinematic model

\[
v_m = \frac{r}{2} (\Omega_{right} + \Omega_{left}) (7)
\]

\[
\dot{\theta}_m = \frac{r}{2b} (\Omega_{right} - \Omega_{left}) (8)
\]

\[
w_m = \frac{sr}{2b} (\Omega_{right} - \Omega_{left}) (9)
\]

With the kinematic model it becomes possible calculate the position and orientation of the mobile robot in the global referential,

\[
\begin{align*}
\theta_m &= \int \dot{\theta}_m dt (10) \\
x_m &= \int \dot{x}_m dt (11) \\
y_m &= \int \dot{y}_m dt (14)
\end{align*}
\]

where \( x_m \) and \( y_m \) are the cartesian coordinates of the mobile robot relative to the global referencial and \( \theta_m \) is the orientation of the mobile robot, given by the angle between axis-xx of the referential global and the linear velocity of the mobile robot (see figure 3).

To obtain the dynamic model, we consider that the forces are applied on the wheels and the moments are calculated at mass center of the mobile robot. Applying the dynamic equations (sum of forces and moments), neglecting the slipping between the floor and the wheels and assuming that the mobile robot only steers in turn of the axis-zz, we obtain the following equations,

\[
m \ddot{v}_m + m s \dot{\theta}_m^2 + \mu mg = F_{right} + F_{left} (15)
\]

\[
(I_{zz} + m s^2) \ddot{\theta}_m - 2m s v_m \dot{\theta}_m = (F_{right} - F_{left}) b (16)
\]

where \( F_{right} \) and \( F_{left} \) are the forces applied by the motor right and left, respectively, \( m \) is the mass of the mobile robot, \( I_{zz} \) is the inertial moment on \( zz \) and \( \mu \) is the friction coefficient of Coulomb.
obtained, the relation is given by

\[ d_{lat} = \left( \frac{X \sin (\pi/2) + \tan \phi + \tan \alpha}{\frac{n_y}{2} (\tan \phi + \tan \alpha) - Y \tan \alpha} \right) \frac{p}{\tan (\phi + \alpha)} \]  

(20)

\[ d_{log} = \left( \frac{Y \sin (\pi/2) \sqrt{1 + \tan^2 \phi}}{\frac{n_y}{2} (\tan \phi + \tan \alpha) - Y \tan \alpha} \right) \frac{p}{h} + \frac{h}{\tan (\phi + \alpha)} \]  

(21)

where \( d_{lat} \) and \( d_{log} \) are the lateral and longitudinal distance of the point capture by the camera, in metric units.

Assumptions We assume that the electrical part of the DC motor is a lot faster than the mechanical part and we neglect the poles of the system that are inherent to the electrical part. The simplified transfer function of the DC motor is given by

\[ \frac{\Omega_m(s)}{E_a(s)} = \frac{K_m}{\tau_m s + 1} \]  

(22)

From 22, we estimate the gain \( K_m \) and constant time \( \tau_m \).

Other important parameters to identify of each actuator is the dead zone and the saturation zone. All parameters are obtained in load, i.e., the resulting dynamic equation contains, implicitly, the contribution of the friction force.

We assume that the force exerted by each motor is applied to half of the mass of the mobile robot and we also assume that the mass center of the mobile robot is located between the two wheels. With this assumptions, the dynamic model of the mobile robot is given by the following expressions

\[ \dot{v}_m = \frac{r}{2} \left( \dot{\Omega}_{right} + \dot{\Omega}_{left} \right) \]  

(23)

\[ \ddot{\theta}_m = \frac{mbr}{2 I_{zz}} \left( \dot{\Omega}_{right} - \dot{\Omega}_{left} \right) \]  

(24)

where \( \Omega_{right} \) and \( \Omega_{left} \) are the accelerations of the right and left wheels, respectively.

3. SIMULATION MODELS

3.1 Kinematic model

Consider a mobile robot whose coordinates \((x, y, \theta)\) are related by the following kinematic equations,

\[
\begin{bmatrix}
\dot{x}_m \\
\dot{y}_m \\
\dot{\theta}_m \\
\end{bmatrix} = \begin{bmatrix}
v_m \cos \theta_m \\
v_m \sin \theta_m \\
\omega_m \\
\end{bmatrix}
\]  

(25)

where \( x_m, y_m \) are the coordinates of point \( M \) in the Cartesian space, \( \theta_m \) is the orientation given by the angle between de x-axis and axis of the robot, \( v_m \) and \( \omega_m \) are, respectively, the linear and angular velocity of the robot. Consider a virtual mobile robot which moves on a trajectory \( p \) whose coordinates \((x, y, \theta)\) are related by the following kinematic equations,

\[
\begin{bmatrix}
\dot{x}_t \\
\dot{y}_t \\
\dot{\theta}_t \\
\end{bmatrix} = \begin{bmatrix}
v_t \cos \theta_t \\
v_t \sin \theta_t \\
\omega_t \\
\end{bmatrix}
\]  

(26)

where \( x_t \) and \( y_t \) are the coordinates of point \( T \) in the Cartesian space, \( \theta_t \) is the orientation, \( v_t \) and \( \omega_t \) are the linear and angular velocity of the virtual mobile robot, respectively.

Consider that a camera is installed on the mobile robot with a certain height \( h \) and inclination \( \phi \) and is coincident with the center mass of the mobile robot. The intersection
between the optical axis of the camera and floor’s plane results in point C, with coordinates \( x_c \) and \( y_c \). We assume that the referential with origin in C has the same orientation of the optical axis of the camera. The coordinates of the point C related with point M are given by

\[
x_c = x_m + l \cos \theta_m \\
y_c = y_m + l \sin \theta_m
\]  

where \( l \) is the distance between point C and point M and is given by

\[
l = \frac{h}{\tan(\phi)}
\]  

The kinematic equations of the point C are given by

\[
\begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{\theta}_c 
\end{bmatrix} =
\begin{bmatrix}
\dot{x}_m - l \dot{\theta}_m \sin \theta_m \\
\dot{y}_m + l \dot{\theta}_m \cos \theta_m \\
\omega_c
\end{bmatrix}
\]  

where \( \theta_c = \theta_m \), which implies \( \dot{\theta}_c = \dot{\theta}_m \), because the camera is fixed to the mobile robot.

Consider the following error coordinates of the point T related with point C,

\[
\begin{bmatrix}
x_e \\
y_e \\
\dot{\theta}_e
\end{bmatrix} :=
\begin{bmatrix}
\cos \theta_m \sin \theta_m 0 \\
-\sin \theta_m \cos \theta_m 0 \\
0 0 1
\end{bmatrix}
\begin{bmatrix}
x_t - x_c \\
y_t - y_c \\
\theta_t - \theta_m
\end{bmatrix}
\]  

Deriving 30 and using the equations 26 and 29, the kinematics error are given by

\[
\begin{bmatrix}
\dot{x}_e \\
\dot{y}_e \\
\dot{\theta}_e
\end{bmatrix} =
\begin{bmatrix}
\dot{\theta}_m y_e - v_m + v_t \cos \theta_e \\
-\dot{\theta}_m (x_c + l) + v_t \sin \theta_e \\
\theta_t - \theta_m
\end{bmatrix}
\]  

3.2 Dynamic model

Consider the dynamic model of the mobile robot described by the equations 23–24 and the dynamic model of the electrical motor 22. The differential equation of the electric motor is given by,

\[
\tau_i \dot{\Omega}_i(t) + \Omega_i(t) = K_i u_i(t)
\]  

where \( i \in \{\text{esq}, \text{dir}\} \). \( \tau_i \), \( K_i \), \( \Omega_i(t) \) and \( u_i(t) \) are the time constant, gain, output and input of the motor \( i \), respectively. The input is a electrical tension and the output is returned in angular velocity.

Deriving the equation system 31,

\[
\begin{bmatrix}
\dot{x}_e \\
\dot{y}_e \\
\dot{\theta}_e
\end{bmatrix} =
\begin{bmatrix}
\dot{\theta}_m y_e + \dot{\theta}_m y_c - \dot{v}_m + \dot{v}_t \cos \theta_c - v_t \dot{\theta}_c \sin \theta_c \\
-\dot{\theta}_m (x_c + l) - \dot{\theta}_m x_c + \dot{v}_t \sin \theta_c + v_t \dot{\theta}_c \cos \theta_c \\
\theta_t - \theta_m
\end{bmatrix}
\]  

3.3 Tracking of a spatial trajectory

For the simulation of a mobile robot tracking a path, we assume that the value of the position error is the distance between point T and point C and the orientation error is given by the difference between the orientation of the mobile robot and orientation of the virtual mobile robot. We assume that the segment \( CT \) is always perpendicular to the optical axis of the camera. Figure 3.3 shows the assumptions considered.

Fig. 7. Path following with camera

Due to the assumption that segment \( CT \) is always perpendicular to the optical axis of the camera, it is necessary to define the constraints \( x_c = 0, \dot{x}_c = 0 \) and \( \dot{x}_c = 0 \). With those constraints, the kinematics and dynamical equations of the error are the following,

\[
\begin{bmatrix}
\dot{x}_e \\
\dot{y}_e \\
\dot{\theta}_e
\end{bmatrix} =
\begin{bmatrix}
\dot{\theta}_m y_e + \dot{\theta}_m y_c - \dot{v}_m + \dot{v}_t \cos \theta_e \\
-\dot{\theta}_m l + \dot{v}_t \sin \theta_e \\
\theta_t - \theta_m
\end{bmatrix}
\]  

From the first equation of system 34, we obtain the following expression

\[
v_t = \frac{1}{\cos \theta_e} \left( v_m - \dot{\theta}_m y_c \right)
\]  

which allows us to calculate the linear velocity of the virtual mobile robot. From the first equation of system 35 it is possible to calculate the linear acceleration of the virtual mobile robot.

\[
\ddot{v}_t = \frac{1}{\cos \theta_e} \left( \ddot{\theta}_m y_c - \dot{\theta}_m \dot{y}_c + v_t \dot{\theta}_c \sin \theta_c \right)
\]
4. SYNTHESIS OF CONTROLLERS

This section presents a method of linearization for nonlinear systems and project of model predictive controllers. The methodology project of controllers is based on Kvasnica [2008] and it is used the toolbox Kvasnica et al. [2004].

4.1 Approximation of nonlinear systems to piecewise-affine systems

Piecewise-affine systems are a special class of hybrid systems where the continuous dynamics correspond to a mode and are affine. The change of mode occurs always in specific subsets of the state space and these subsets are known a priori.

The approximation of nonlinear systems to piecewise-affine presented in this section is based on Rodrigues [2002] and Johansson [1999].

A fundamental concept is the definition of simplex.

**Definition 1.** The convex hull of a set $S$ is the smallest convex set that contains $S$. A simplex in $\mathbb{R}^n$ is defined as the convex hull of $n + 1$ affinely independent points.

Intervals in $\mathbb{R}$ or triangles in $\mathbb{R}^2$ are examples of simplex.

Using this definition, the following states the assumptions used in the problem formulation.

**Assumption 1.** The nonlinear systems considered in this section are described by

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + a + f(x) + B(x)u(t) \\
y(t) &= Cx(t)
\end{aligned}
$$

(38)

where $x \in \mathbb{R}^n$ contains the state variables, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^p$ is the output, $a \in \mathbb{R}^n$ is a constant vector, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$ are matrices and $f(x)$ is a nonlinear function.

**Assumption 2.** The right hand side of 38 is locally Lipschitz (guarantees existence and uniqueness of solutions to the differential equation 38) and that the function $f(x)$ is continuous.

**Assumption 3.** The support of function $f(x)$ is a compact hypercube and that a uniform rectangular grid is defined in that compact support.

**Representation in a piecewise-affine system** A piecewise-affine system has two important components: the partition $\{\mathcal{X}_i\}$ of the state space into regions and the equations which describe the dynamic within each region. For a good understanding of the global dynamics, it is necessary to take in account both components. The term piecewise-affine do not impose any kind of restrictions on the geometry of the regions, but it is frequent to impose restrictions for better results.

Piecewise-affine system consists on the division of the state space and specification of a valid dynamic in each region. Therefore, the piecewise-affine system can be described as a set of ordered pairs

$$
\{\{\Sigma_i, \mathcal{X}_i\}\}_{i \in \mathcal{I}}
$$

(39)

where each region $\mathcal{X}_i$ is associate to a dynamic $\Sigma_i$. The index set is denote as $\mathcal{I}$.

The system dynamic is given by

$$
\Sigma_i = \left\{ \begin{array}{ll}
\dot{x}(t) &= A_i x(t) + a_i + B(x_{\text{cheb}})u(t) \\
y(t) &= C_i x(t)
\end{array} \right. 
$$

(40)

for all $x(t) \in \mathcal{X}_i$ and $x_{\text{cheb}}$ represents the center of Chebychev of region $\mathcal{X}_i$. The matrices $A_i$, $a_i$, $B_i$ e $C_i$ are invariant in time and with compatible dimensions. We assume that the sets $\mathcal{X}_i \subseteq \mathbb{R}^n$ are convex hull and possibly unbounded. We assume that the various ordered pairs with affine dynamic and corresponding partition are given. The convex set $\mathcal{X}_i$ results from the intersection of finite closed hyper planes, that implies for each $\mathcal{X}_i$ exists a matrix $H_i$ and a vector $g_i$ where

$$
\mathcal{X}_i = \{ x \mid H_i^T x - g_i < 0 \}
$$

(41)

The partition $\mathcal{X} = \{ \mathcal{X}_i \}_{i \in \mathcal{I}}$ covers a subset of the state space, $\mathcal{X} \subseteq \mathbb{R}^n$ where

$$
\bigcup_{i=1}^M \overline{\mathcal{X}_i} = \mathcal{X}
$$

(42)

where $\overline{\mathcal{X}_i}$ denotes the closure of $\mathcal{X}_i$ and $M$ is the cardinality of the index set $\mathcal{I}$. Each cell has $n + 1$ vertices and $n + 1$ facets. The cells have disjoint interior, i.e.

$$
\mathcal{X}_i \cap \mathcal{X}_j = \emptyset
$$

(43)

for $i \neq j$, where two cells would have only in common the frontier between them. Two cells sharing a common facet arc called level-1 neighboring cells. A number finite of cells sharing a common vertex are called level-2 neighboring cells. Let $\mathcal{N}_i$ be the set of the neighboring cell of type level-1. We assume that vectors $c_{ij} \in \mathbb{R}^n$ and scalars $d_{ij}$ exists such that the facet boundary between cells $\mathcal{R}_i$ and $\mathcal{R}_j$ is contained in the hyperplane described by

$$
\{ x \in \mathbb{R}^n \mid c_{ij}^T x - d_{ij} = 0 \}
$$

for $i = 1, \ldots, M$ and $j \in \mathcal{N}_i$. A parametric description of the boundaries can be obtained as

$$
\overline{\mathcal{X}_i} \cap \overline{\mathcal{X}_j} \subseteq \{ l_{ij} + F_{ij}s \mid s \in \mathbb{R}^{n-1} \}
$$

(44)

for $i = 1, \ldots, M$ and $j \in \mathcal{N}_i$, where $F_{ij} \in \mathbb{R}^{n \times (n-1)}$ is the matrix whose columns span the null space of $c_{ij}$ and $l_{ij} \in \mathbb{R}^n$ is a particular solution of $c_{ij}^T x = d_{ij}$ given by

$$
l_{ij} = c_{ij} \left( c_{ij}^T c_{ij} \right)^{-1} d_{ij}.
$$

It is important to analyze the results in terms of the equilibrium point. We assume that the equilibrium point is located at $x = 0$. Let $\mathcal{I}_0 \subseteq \mathcal{I}$ be the index set that contains the origin and $\mathcal{I}_1 \subseteq \mathcal{I}$ be the index set that does not contains the origin. We assume $a_i = 0$ for $i \in \mathcal{I}_0$.

**Piecewise-affine approximation algorithm** We assume that the control objective is to stabilize the system to a desired closed-loop equilibrium point $x_{af}$. Therefore, the piecewise-affine approximation algorithm has as input the system description, the desired closed-loop equilibrium point and a uniform rectangular grid for the domain of the nonlinearity involved in the dynamical system. With these inputs, the algorithm returns a partition of the state space into a finite number of simplex cells. Given a nonlinear system 38 the following problems are solved,

1. Given a uniform rectangular grid for the domain of the nonlinearity, set a index $i \in \mathcal{I}$ to the vertices of that grid;
2. Group the vertices into simplicial cells;
3. Find a polytopic description for each cell;
Consider the following dynamic discrete time piecewise affine system time-invariant

\[ x(k + 1) = A_i x(k) + B_i u(k) + f_i \]

subject to

\[ L_i x(k) + E_i u(k) \leq W_i, \quad i \in \mathcal{I} \]

where \( A_i \in \mathbb{R}^{n \times n}, B_i \in \mathbb{R}^{n \times m}, f_i \in \mathbb{R}^n, C_i \in \mathbb{R}^{p \times n} \) and \( g_i \in \mathbb{R}^p \). Let \( x(k) \) be the measured state at time \( k \), \( x_k \) be the predicted state at time \( k \) and \( u_k \) be the predicted input at time \( k \), given \( x(0) \). We assume that the states and inputs of the system 45 are subject to the following constraints

\[ x(k) \in \mathbb{X} \subset \mathbb{R}^n, \quad u(k) \in \mathbb{U} \subset \mathbb{R}^m, \quad \forall k \geq 0 \]

where \( \mathbb{X} \) and \( \mathbb{U} \) are polytope sets containing the origin in their inside. Consider the constrained time-finite optimal control problem

\[
J^*_N(x(0)) = \min_{u_0, \ldots, u_{N-1}} \|Q_f x_N\|_I + \sum_{k=0}^{N-1} \|R u_k\|_I + \|Q x_k\|_I
\]

subject to

\[
L_i x_k + E_i u_k \leq W_i, \quad i \in \mathcal{I}, \quad \forall k \in \{0, \ldots, N - 1\}
\]

\[ x_{N} \in X_{\text{set}} \]

\[ x_{k+1} = f_{\text{W}A}(x_{k}, u_{k}) , \quad x_0 = x(0), \quad \forall k \in \{0, \ldots, N - 1\} \]

The expression 52 is a constraint which defines a terminal set that can be used to guarantee stability. As an alternative, the solution with an infinite horizon guarantees stability.

**Receding horizon control** The strategy known as receding horizon control consists on solving a time finite optimal control problem at each sampling time to obtain an optimal control law \( U^*_N \) and only the first element of the input vector is applied on the system. At the next sampling time, the states of the system are measured again and the optimal control problem is repeated for the new initial condition \( x_0 \). However, the execution of the optimization problem repeatedly does not guarantee stability for all time. To guarantee stability it becomes necessary choose an appropriate terminal set \( X_{\text{set}} \).

If the optimization problem is solved by multi-parametric programming to obtain an explicit function, given an initial condition \( x_0 \), the optimal control law \( u^* = f(x_0) \) takes a form of a lookup-table. The online implementation of such table is done by searching the set which contains the measured states.

**Positively invariant set** In this section, a method is presented to obtaining stabilizing controllers for piecewise affine systems. The objective is to calculate a positively invariant set \( O^P_{\infty} \) with an associate Lyapunov function such that stability is guaranteed. We assume that the origin is an equilibrium point to the piecewise affine system and if this assumption is not satisfied, this approach will fail.

In the first step, it is stabilized each dynamics of the piecewise affine system which contains the origin by a linear state feedback controller \( F_i \) of the form \( u = F_i x \) if \( x \in D_i \). Let \( D_0 \) be the set of indices of dynamics which contain the origins in their respective interiors.

\[ D_0 \triangleq \{ i \in \mathcal{I} | 0 \in D_i \} \]

The search for the stabilizing control law and an associated common quadratic Lyapunov function \( V(x) = x^T P x \) can be posed as

\[
x^T P x \geq 0, \quad \forall x \in \mathbb{X}
\]

\[
x^T (A_i + B_i F_i)^T P (A_i + B_i F_i) x - x^T P x \leq -x^T Q x + x^T F_i^T RF_i x, \quad \forall x \in D_i, \forall i \in D_0
\]

To obtain the matrix \( P \) and the linear state feedback controller \( F_i \) a LMI problem is solved.

The second step is to compute the positively invariant set \( O^P_{\infty} \) to the piecewise affine system subject to the linear controller \( F_i \).

**Minimum-time controller (dual mode controller)** The minimum-time controller objective is to steer the states \( x(k) \) to a positively invariant set \( O^P_{\infty} \) in the minimum number of time steps. Consider the following optimization problem of the minimum time controller

\[
J^*_N(x(0)) = \min_{u_0, \ldots, u_{N-1}} N
\]

subject to

\[
L_i x_k + E_i u_k \leq W_i, \quad i \in \mathcal{I}, \quad \forall k \in \{0, \ldots, N - 1\}
\]

\[ x_{N} \in O^P_{\infty} \]

\[ x_{k+1} = f_{\text{W}A}(x_{k}, u_{k}) , \quad x_0 = x(0), \quad \forall k \in \{0, \ldots, N - 1\} \]

The cost function for this type of problem assumes only integer values and generally the target sets which need to be considered at each iteration step are larger and fewer in number than those which would be obtained if an optimal controller with a different cost objective were to be computed.

When the minimum-time algorithm terminates, the associated linear feedback controller will cover the N-step stabilizable set.

**Definition 2.** **N-step stabilizable set** \( K^P_{\infty} (O^P_{\infty}) \)

The set \( K^P_{\infty} (O^P_{\infty}) \) denotes the N-step stabilizable set for a piecewise affine system, i.e., it contains all states which can be steered into \( O^P_{\infty} \) in N steps. Specifically,
\[ \{ x(0) \in \mathbb{R}^n \mid \exists u(k) \in \mathbb{R}^m, \text{subject to } x(N) \in \mathcal{O}_{\infty}^{PWA} \}, \]
\[ x(k+1) = f_{PWA}(x(k), u(k)), \forall k \in \{0, \ldots, N-1\} \}
\]
\[ (61) \]

The set \( \mathcal{O}_{\infty}^{PWA} \) denotes the maximal stabilizable set for \( N \to \infty \).

The minimum time controller calculation is based on solving a sequence of multi-parametric program at each iteration step. The number of iterations correspond to the number of time steps which are needed to reach the target set. At each iteration, a controller partition is computed which steers the states into the partition that on which was obtained in the previous iteration.

5. RESULTS

This section presents the results obtained in the simulation models.

5.1 Kinematic simulation model

This section presents the results of the kinematic simulation model. The control objective is to steer the point that result from the intersection between the optical axes of the camera with the plane’s floor, to follow a track with determined linear velocity. To fulfill this objective, a model predictive controller is designed, using multi-parametric programming to obtain a control law which is an explicit function of the position and orientation errors.

To design the controller minimum-time algorithm is used and the cost function is quadratic. Since the state equations of the system are nonlinear, the system is linearized into three partitions.

**Constant linear velocity** Consider a constant linear velocity of 1m/s and the following \( Q \) and \( R \) weights for the cost function,

\[ Q = \begin{bmatrix} 125000 & 0 \\ 0 & 9118,9 \end{bmatrix} \]
\[ R = 10 \]
\[ (62) \]
\[ (63) \]

Figure 8 shows the projection of the control law at the state space. For the project of the controller the following limits for the states \( y_l \in [-0.2, 0.2] \) and \( \theta \in [-\pi/3, \pi/3] \) were chosen. The input of the system is the linear velocity of the mobile robot.

Observing figure 8, the resultant controller does not fill all of the defined state space, i.e., for a constant linear velocity of 1m/s the controller is not capable to control the kinematic model for determined states. This incapacity in controlling the kinematic model in all space state is due to the reduced limits of the position error which makes the model quickly lose sight of the track to follow. To avoid this situation the mobile robot must lower the linear velocity, augment the limits of the position error and/or change the parameters of the camera, more precisely the height and the inclination.

The controller was tested for various sinusoidal track and was capable following a sinusoidal track with 2.5m of amplitude and a spatial frequency of \( \pi/5 \) with maximum position and orientation errors of 0.021m and 0.338rad, respectively.

**Reference for the linear velocity** Consider a reference for the linear velocity of 1m/s and the following weights for the cost function

\[ Q = \begin{bmatrix} 125000 & 0 \\ 0 & 9118,9 \end{bmatrix} \]
\[ R = \begin{bmatrix} 4444,4 & 0 \\ 0 & 100 \end{bmatrix} \]
\[ (64) \]
\[ (65) \]

Figure 9 shows the projection of the control law into space state. For the design of the controller the following constraints for the states \( y_l \in [-0.2, 0.2] \) and \( \theta \in [-\pi/3, \pi/3] \) are used. The inputs of this system are the linear and angular velocities of the mobile robot.

Comparing with figure 8, it is possible to observe that the state space is more filled up than the previous case. The robustness is augmented due to the unconstrained linear velocity. The controller was tested for various sinusoidal tracks and was capable following a sinusoidal track with an amplitude of 1m and a spatial frequency of \( \pi/2 \). This controller is capable following a tighter track than the controller with a constant linear velocity.

5.2 Dynamic simulation model

Initially, we wanted to design a controller based on the dynamic equations, but this task has a high computational weight and would result in a highly complex control law. To overcome this inconvenient, we split the problem in two. First, we designed controllers for each motor to track
a determined reference in angular velocity. With these controllers, we assume that the robot dynamic’s is fast and following a track is solved with the kinematic model. Once again, the kinematics equations are nonlinear and had to be linearized into three partitions. For design the controller to following a track is considered time-minimum algorithm and cost function is quadratic.

Validation of the dynamic model Figure 10 show the results obtained in open-loop for the mobile robot and the dynamic simulation model. The objective of this figure is to validate the simulation model.

The mobile robot has a behavior similar to a non-minimum phase system, due to its undershoot. This behavior exists because of the difference of time constant of each motor. It is also possible to observe that the mobile robot moves away from the track because of the difference in gain between each motor.

Tracking control to the motors For the task of controlling the motors, a predictive controller was designed, using multi-parametric programming. The controller applies a tension to the motor in order to reach a certain angular velocity. For the cost function the following weights \( Q = 40000 \) and \( R = 0.0016 \) were chosen. The cost function is quadratic, prediction horizon is three and method for the solving of the optimization problem is the optimal with the option tracking. If there exist any uncertain in the model, the controller can follow the reference in angular velocity but with an associated stationary error. To remove this error, we integrate the angular velocity error.

Figure 11 presents the results obtained for the following of a reference for both motors.

Constant linear velocity Consider a constant angular velocity for each motor of 30rad/s (approximately 1m/s) and with the following weights

\[
Q = \begin{bmatrix}
158520 & 0 \\
0 & 9118.9
\end{bmatrix}
\]  (66)

\[
R = \begin{bmatrix}
12,245 & 0 \\
0 & 12,245
\end{bmatrix}
\]  (67)

The states of the system are the position and orientation errors. The inputs of the system are the right and left angular velocity motors. The controller is tested for various sinusoidal tracks, with different amplitudes and spatial frequency. The toughest track which this controller can do is a track with amplitude of 3m and spatial frequency of \( \pi/6 \). The maximum position error is 0.015m and the maximum orientation error is 0.305rad.

Reference for linear velocity Consider a reference for angular velocidade for each motor of 30rad/s and with the following weights

\[
Q = \begin{bmatrix}
158520 & 0 \\
0 & 9118.9
\end{bmatrix}
\]  (68)

\[
R = \begin{bmatrix}
12,245 & 0 \\
0 & 12,245
\end{bmatrix}
\]  (69)

The states of the system are the position and orientation errors. The inputs of the system are the right and left angular velocity motors. The controller is tested for various sinusoidal tracks, with different amplitudes and spatial frequency. The toughest track which this controller can do is the track with amplitude of 1m and spatial frequency of \( \pi/2 \). The maximum position error is 0.178m and the maximum orientation error is 0.887rad. Once again, this controller shows great robustness in comparison with the controller in which the linear velocity is constant.

6. CONCLUSIONS AND FUTURE WORK

This section presents the main conclusions of this work. The two main objectives of this work were the mathematic modeling and construction of simulation models, as well the presentation of a methodology to design controllers for mobile robot to follow paths. The main conclusions are:

1. The construction of the simulation model allowed us to know in greater detail the dynamic system of the mobile robot called “Rasteirinho”, which was given to know that if a mobile robot will have motors with different response time and different gain, the global system may have a behavior similar to a non-minimum phase system, which difficulties the controller project.
(2) After obtaining the nonlinear state equation, based on the position and orientation errors, it was proceeded to his linearization around of various work points, to obtain a good approximation of the real dynamic system. The approximation used in this work is based on Rodrigues [2002], who developed a toolbox for Matlab.

(3) To control of the mobile robot, the model predictive control strategy in association with multi-parametric programming was chosen. The utilization of the model predictive control seems not be feasible, once this strategy implies the resolution of optimization problem at each sampling time. To overcome this difficulty, with the aid of multi-parametric programming it becomes possible to obtain an explicit function of the states and is not necessary the resolution of the optimization problem at each sampling time. A disadvantage that this control strategy may carry is the high number of resulting partitions which can result in a very complex look-up table. To avoid this problem, the dual-mode strategy applied to model predictive control was used which results in simpler controllers.

6.1 Future works

In this section we give some directions to continue this work.

(1) The results obtain of this work guarantee stability for the piecewise affine and not for nonlinear system. To guarantee stability of the nonlinear system must be projected a robust controller which takes into account the uncertainties of the system must be projected.

(2) The controllers that were projected in this work assume constant linear velocity or reference to a determined linear velocity. It is interesting to study the variation of the reference of the linear velocity along time. The proposed solution is the project of various controllers to various linear velocity references, creating a bank of controllers. It would also become necessary to construct a control law that commutes between controllers, guaranteeing stability of this commutation law.

(3) Study the problem of target following, being this type of problem very important, for example, to maintain a determined formation of various mobile robots. The difference between this problem and path following is the introduction of a new state $x_e$, constrain $x_e = 0$ is removed. For the problem of formation control, we recommend a lecture of Dunbar and Murray [2005], Vidal et al. [2004], Soria et al. [2006], Spletzer et al. and Öğren [2003].

REFERENCES


