Robust Tube-based Predictive Control for Mobile Robots in Off-Road Conditions

Ramon Gonzalez\textsuperscript{a}, Mirko Fiacchini\textsuperscript{b}, Jose Luis Guzman\textsuperscript{a}, Teodoro Alamo\textsuperscript{c}, Francisco Rodriguez\textsuperscript{a}

\textsuperscript{a}Dep. Lenguajes y Computación, University of Almería, Almería, Spain
\{rgonzalez, joguzman, frrodrig\}@ual.es

\textsuperscript{b}Laboratory for Analysis and Architecture of Systems, Toulouse, France
fiacchini@laas.fr

\textsuperscript{c}Dep. Ingeniería de Sistemas y Automática, University of Seville, Seville, Spain
alamo@cartuja.us.es

Abstract

This paper focuses on the design of a tube-based Model Predictive Control law for the control of constrained mobile robots in off-road conditions with longitudinal slip while ensuring robustness and stability. A time-varying trajectory tracking error model is used, where uncertainties are assumed to be bounded and additive. The robust tube-based MPC is compared with other motion control techniques through simulation and physical experiments. These tests show the satisfactory behavior of the presented control strategy.

Keywords: Autonomous mobile robot, Robustness, Model Predictive Control, Slip

1. Introduction

Mobile robotics comprises an important field of application for many control techniques, mainly those related to handle constraints (i.e., physical limitations of actuators, narrow workspaces, etc.), and uncertainties (i.e., unmodelled dynamics,
simplified models, noisy measurements, etc.). Model Predictive Control (MPC) is a popular control technique to deal with constrained systems [1], [2], [3]. However, in order to tackle uncertainty, robust MPC formulations such as Min-Max MPC [4], or feedback MPC [5] must be considered.

In addition, another fundamental factor that can affect the performance of the motion of off-road mobile robots is the slip effect. Slip induces traction/velocity loss during the robot motion, which can adversely influence the mobility and controllability [6], [7]. Slip can be presented in lateral and longitudinal directions. In the case of wheeled robots, lateral slip occurs due to the deformation of the pneumatic tire surface and large centrifugal force [8]. Longitudinal slip comes from pneumatic tire compression due to load on the robot. In the case of tracked robots, lateral slip is only presented at high velocity turns and it only depends on centrifugal force [9], [10]. Longitudinal slip is mainly caused by the track/soil interaction, such as the sinkage effect. This produces a bulldozing phenomenon that makes the robot sinks in the soil (particularly on sandy soils) [6], [10].

In the past few years, some research efforts have been devoted to the application of predictive control in mobile robotics and some of them related to off-road robots. For instance, in [9], the authors use an MPC controller to enable both anticipation of approaching curvature and to compensate from lateral slip phenomena for path tracking control of an agricultural vehicle. In [11], an MPC is applied to the trajectory tracking problem. The control law is analytically derived, which permits its application to a physical mobile robot. In order to avoid vehicle slip, velocity and acceleration are bounded. The work [12] presents a predictive strategy that permits to avoid unexpected static obstacles in the robot environment. For the purpose of MPC real-time implementation, a neural network was
trained. A Smith-predictor-based generalized predictive controller is discussed in [13]. This control strategy permits dealing with dead-time uncertainties related to a mobile robot control motion.

Generally, the main issue of robust MPC strategies, which sometimes prevents its physical application, is related to the high computation burden [3], [14]. Recently, an efficient theoretical concept, called “tube-based MPC”, has been applied to robustify MPC [15], [16]. The term tube-based refers to those control techniques whose objective is to maintain all the possible trajectories of a uncertain system inside a sequence of admissible regions. These regions are determined by taking constraints satisfaction into account. In the MPC context, the possible trajectories form a sequence of sets in the state space, which is the admissible set for each step within the prediction horizon.

The main contribution of this paper is the adaptation and application of a robust tube-based MPC strategy to the mobile robotics field. From the theoretical point of view, an extension of tube-based predictive control to time-varying systems using reachable sets is provided. In this case, a trajectory tracking error model of a mobile robot in slip conditions is considered. In order to ensure stability, Linear Matrix Inequalities (LMI) formalism is employed to determine a Lyapunov function constituting the terminal cost, and a terminal robust positively invariant set following the ideas of Blanchini [17], and Kolmanovsky and Gilbert [18] is also calculated. Finally, the robust tube-based MPC has been compared to other motion control techniques through simulation and off-road physical experiments. These tests show the satisfactory behavior of the suggested control strategy and the efficient real-time execution. In the physical experiments, the visual odometry approach has been used for robot localization purposes.
This paper is organized as follows: Section 2 presents the problem statement. The control objective and the robust tube-based MPC control law are stated in Section 3. Section 4 shows a comparison of the proposed controller with other motion control techniques through simulation and physical experiments. Finally, Section 5 deals with conclusion and future research. Videos related to the physical experiments are available at http://www.ual.es/personal/rgonzalez/videosMPC.htm.

2. Problem Statement

In this section, the time-varying trajectory tracking error model of a mobile robot in slip conditions is considered. This model is discretized and an additive uncertainty term has been included.

Trajectory tracking consists in the problem in which a robot must follow a reference or virtual mobile robot representing the desired positions and velocities [19]. Hence, the objective is to find a feedback control law such that the error between the desired location and the real location of the mobile robot approaches zero (regulation problem).

These errors are expressed in the real robot frame as

\[
\begin{bmatrix}
e_x(t) \\
e_y(t) \\
e_\theta(t)
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta^{sl}(t)) & \sin(\theta^{sl}(t)) & 0 \\
-\sin(\theta^{sl}(t)) & \cos(\theta^{sl}(t)) & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x^{rf}(t) - x^{sl}(t) \\
y^{rf}(t) - y^{sl}(t) \\
\theta^{rf}(t) - \theta^{sl}(t)
\end{bmatrix},
\]

(1)

where \( t \in \mathbb{R}^+ \) is the continuous time, \( e_x \in \mathbb{R} \) is the longitudinal error, \( e_y \in \mathbb{R} \) is the lateral error and \( e_\theta \in \mathbb{R} \) is the orientation error. The rest of terms are defined in what follows.
In this work, the reference robot is defined by the kinematic model of a differential-drive robot [20]

\[
\begin{align*}
    x^{rf}(t) &= \frac{v^{rf}_r(t) + v^{rf}_l(t)}{2} \cos \theta^{rf}(t), \\
    y^{rf}(t) &= \frac{v^{rf}_r(t) + v^{rf}_l(t)}{2} \sin \theta^{rf}(t), \\
    \theta^{rf}(t) &= \frac{v^{rf}_r(t) - v^{rf}_l(t)}{b},
\end{align*}
\]

where \([x^{rf} \ y^{rf} \ \theta^{rf}]^T \in \mathbb{R}^3\) represents the location (position and orientation) of the reference mobile robot, \(v^{rf}_r \in \mathbb{R}^+\) and \(v^{rf}_l \in \mathbb{R}^+\) are the reference linear velocities of right and left wheels, respectively, and \(b \in \mathbb{R}\) is the distance between the wheel centers.

**Assumption 1.** Assume that reference robot wheel velocities are known positive \((v^{rf}_r > 0, v^{rf}_l > 0)\) and bounded \((v^{rf}_r \in [v_r^{rf,m}, v_r^{rf,M}], v^{rf}_l \in [v_l^{rf,m}, v_l^{rf,M}])\).

On the other hand, the real robot is modelled taking into account an extended kinematic model where some terms related to longitudinal slip appear. Such kinematic model is given by [10], [21]

\[
\begin{align*}
    \dot{x}^{sl}(t) &= \frac{v^{sl}_r(t) + v^{sl}_l(t)}{2} \cos \theta^{sl}(t), \\
    \dot{y}^{sl}(t) &= \frac{v^{sl}_r(t) + v^{sl}_l(t)}{2} \sin \theta^{sl}(t), \\
    \dot{\theta}^{sl}(t) &= \frac{v^{sl}_r(t) - v^{sl}_l(t)}{b},
\end{align*}
\]
where \([x^{sl}, y^{sl}, \theta^{sl}]^T \in \mathbb{R}^3\) represents the location of the mobile robot, and

\[
\begin{align*}
\nu_r^{sl}(t) &= v_r(t)(1 - i_r(t)), \\
\nu_l^{sl}(t) &= v_l(t)(1 - i_l(t)),
\end{align*}
\]

(4)

where \(\nu_r^{sl} \in \mathbb{R}\) and \(\nu_l^{sl} \in \mathbb{R}\) are the linear velocities taking into account slip effects, \(v_r \in \mathbb{R}\) and \(v_l \in \mathbb{R}\) are the linear velocities of the right and left wheels, respectively, and \(i_r \in \mathbb{R}^+\) and \(i_l \in \mathbb{R}^+\) are the terms representing the longitudinal slip component. Note that, by definition, \(i_r, i_l \in [0, 1]\), for details see [6], [21].

**Assumption 2.** It is assumed that the real mobile robot moves at low velocities \((\leq 2[m/s])\).

**Remark 1.** Note that from Assumption 2, lateral slip and forces arising from dynamic effects are neglected. As stated in [10], [22], lateral slip is zero for straight-line motions and it can be neglected when the vehicle turns “on the spot” or at low velocities. However, longitudinal slip is an unavoidable effect of the robot-terrain interaction [6].

As shown in [21], differentiation of equation (1) with respect to time and defining the following virtual control signals for linearization purposes

\[
\begin{align*}
u^{sl} &= (v_r^{sl} + v_l^{sl})/2, \\
\theta^{sl} &= (v_r^{sl} - v_l^{sl})/2,
\end{align*}
\]

(5)

where \(v^{sl} = (v_r^{sl} + v_l^{sl})/2\) is the linear velocity of the robot taking into account slip effects and \(v^{rf} = (v_r^{rf} + v_l^{rf})/2\) is the linear velocity of the reference. Previous
process leads to the following linear, time-varying, discrete-time system

\[ e(k+1) = A_{\gamma(k)} e(k) + B_d u(k) + w(k), \]

(7)

where \( k \in \mathbb{Z}^+ \) is the discrete time, \( e = [e_x \ e_y \ e_\theta]^T \in \mathbb{R}^3 \) is the current state (error), \( u = [u_1 \ u_2]^T \in \mathbb{R}^2 \) is the current input (see Remark 2), and \( w \) is a bounded additive uncertainty satisfying \( w \in W \), where \( W \) is a polytope in the state space \( \mathbb{R}^3 \) (see Remark 3). The matrices \( A_{\gamma} \) and \( B_d \) are defined as

\[
A_{\gamma(k)} = \begin{bmatrix}
1 & \varepsilon(k) & 0 \\
-\varepsilon(k) & 1 & \rho(k) \\
0 & 0 & 1
\end{bmatrix}, \quad B_d = \begin{bmatrix} T_s \ 0 \\
0 \ 0 \ T_s \\
0 \ T_s
\end{bmatrix}, \quad (8)
\]

where \( \varepsilon = T_s \frac{(1-\bar{i}_r)v_{r}'f - (1-\bar{i}_l)v_{l}'f}{b} \) and \( \rho = T_s v_{r}'f \), \( T_s \in \mathbb{R}^+ \) is the sampling time, and \( \bar{i}_r \in \mathbb{R}^+ \) and \( \bar{i}_l \in \mathbb{R}^+ \) are the nominal slip of each wheel (see Remark 4). For sake of notational simplicity, the dependence of \( A_{\gamma} \) on \( k \) is omitted.

From Assumption 1, \( \gamma = [v_{r}'f \ v_{l}'f]^T \in \mathbb{R}^2 \) is a time-varying vector such that \( \gamma(k) \in \Gamma, \forall k \in \mathbb{Z}^+ \), where \( \Gamma \subseteq \mathbb{R}^2 \) is a polytope. For any admissible realization of parameter \( \gamma \in \Gamma \), a dynamic matrix \( A_{\gamma} \) is determined. It follows that \( A_{\gamma} \in \mathcal{A} \), with \( \mathcal{A} \) being a polytope in \( \mathbb{R}^{3 \times 3} \) that represents the dynamic system matrix.

**Remark 2.** Note that according to (5)-(6), the linear velocities for each wheel are obtained as

\[ v_r(t) = \frac{v_{r}'f(t) - u_1(t) - \frac{b}{2}u_2(t)}{1 - \bar{i}_r(t)}, \quad (9) \]

\[ v_l(t) = \frac{-v_{l}'f(t) + u_1(t) - \frac{b}{2}u_2(t)}{-1 + \bar{i}_l(t)}, \quad (10) \]
where $v_r \in [v_{r}^m, v_{r}^M]$ and $v_l \in [v_{l}^m, v_{l}^M]$. This leads to bounds on the space of $u$, which are obtained from the constraints on $v_r, v_l, i_r, i_l, v_{r f}, v_{l f}$.

**Remark 3.** The set $W \subseteq \mathbb{R}^3$ represents the uncertainty affecting the state at each sampling instant. This uncertainty bounds the mismatch between the continuous-time non-linear trajectory tracking error model and the discrete-time linear model, the noise in the slip estimation, and the uncertainty in the robot localization. In this case, such mismatch has been estimated simulating random input values to both models. Furthermore, the set $W$ has been enlarged to take into account both the noise in the slip estimation and the uncertainty in the robot localization based on physical experiments (estimating the deviation between the robot localization and the ground-truth through several physical experiments).

**Remark 4.** Notice that the nominal slip in the model is used, since the real values of $i_r$ and $i_l$ are not known for the predictions.

Finally, the states and inputs are subject to the following constraints

$$e(k) \in E, \quad u(k) \in U,$$

where $E \subseteq \mathbb{R}^3$ and $U \subseteq \mathbb{R}^2$ are polytopes that contain the origin. Recall that state and input constraints represent the physical limitations such as narrow spaces of operation and saturations on the actuators of the problem.

### 2.1. Definitions

A polyhedron is the (convex) intersection of a finite number of open and/or closed half-spaces and a polytope is a closed and bounded polyhedron. The convex
hull of a set of points is defined as the smallest convex set containing the points, and it is denoted as \( \text{co} \). Given two sets \( X, Y \subseteq \mathbb{R}^n \), the Minkowski sum is defined by \( X \oplus Y \triangleq \{x + y | x \in X, y \in Y\} \) and the Pontryagin set difference is \( X \ominus Y \triangleq \{x | x \oplus Y \subseteq X\} \). For a given \( \lambda \geq 0 \), \( \lambda X = \{\lambda x : x \in X\} \).

### 3. Robust MPC Strategy

In this section, the implementation of the robust tube-based MPC is discussed, adapting the ideas of Chisci et al. [15], and Langson et al. [16]. First, a local control law that compensates the effect of uncertainty is calculated. This leads to reachable sets containing the real state for every possible realization of the uncertainty and every admissible parameter occurrence. Finally, as explained in [15], online computation is only devoted to the solution of a deterministic MPC for a nominal system with restricted constraints (see Figure 1).

The nominal system is defined as

\[
\tilde{e}(k + 1) = A_{\tilde{e}} \tilde{e}(k) + B_d g(k),
\]

where \( \tilde{e} \in \mathbb{R}^3 \) is the nominal state, and \( g \in \mathbb{R}^2 \) is the control input for the nominal system. Observe that this is equivalent to system (7) with \( W = \{0\} \).

The control objective is to design a state feedback control law of the form [15]

\[
\begin{align*}
    u(k) &= K \tilde{e}(k) + g(k), \\
    \tilde{e}(k) &= e(k) - \tilde{e}(k),
\end{align*}
\]

where \( K \in \mathbb{R}^{2 \times 3} \) is a local controller whose goal is to compensate the mismatch between the real (7) and the nominal (12) systems, \( \tilde{e} = [\tilde{e}_x \quad \tilde{e}_y \quad \tilde{e}_\theta]^T \) is the mis-
match state vector and \( g \) is the online MPC control input.

Replacing (7) and (12) into (14), the local uncertain closed-loop system follows as

\[
\dot{e}(k+1) = e(k+1) - \bar{e}(k+1) = (A_T + B_T K) \bar{e}(k) + w(k). \tag{15}
\]

### 3.1. Control Objectives

In this paper, the major control objectives are:

- **Robustness**: Following the tube-based MPC policy, original constraints (11) are replaced with more restricted ones, which take into account additive uncertainties and time-varying dynamics.

- **Performance**: An optimization problem (QP) is solved at each sampling instant obtaining the proper control actions as a compromise between small deviations from the reference trajectory and suitable control actions.

- **Input and state constraints fulfillment**: This requirement is guaranteed ensuring constraints satisfaction in the minimization of the MPC control law.

- **Asymptotic stability**: It is assured through a quadratic Lyapunov function determined using LMI and a robust positively invariant set for the terminal region of the nominal MPC.

- **Efficient real-time execution**: A standard nominal MPC is solved online to control the nominal system since the effect of the uncertainties and system dynamics are already included in the restricted constraints. This fact implies that the robust tube-based MPC strategy fits properly to mobile robotics applications, where high sampling frequencies are employed.
Figure 1 summarizes the robust tube-based MPC strategy. First, the reachable sets are calculated for the local uncertain closed-loop system (15) (see Subsection 3.2), then the state and input constraints are replaced using the previously calculated reachable sets (Subsection 3.3). Finally, an MPC control law to the nominal system is applied (detailed in Subsection 3.4). In order to ensure the stability of the nominal MPC, a Lyapunov function and a terminal robust positively invariant set are determined (Subsection 3.5). We would like to point out that both reachable sets calculation and the replacement of state and input constraints are carried out offline, so they do not influence online computation burden. Online computation is only devoted to solve the standard MPC controller handling the nominal system for which tighter constraints are considered.

3.2. Local Compensation of System Dynamics

In this subsection, the control law that compensates the mismatch between the real and nominal systems and the reachable sets, that will be used in what follows,
are calculated.

First, in order to implement the tube-based MPC, the system matrix, $A_{\gamma}$, is decomposed into the time-invariant and time-varying parts

$$A_{\gamma} = I_3 + A^r v_{r}^{rf} + A^l v_{l}^{rf},$$

(16)

where $I_3 \in \mathbb{R}^{3 \times 3}$ is the identity matrix, and

$$A^r = \begin{bmatrix} 0 & \xi & 0 \\ -\xi & 0 & \beta \\ 0 & 0 & 0 \end{bmatrix}, \quad A^l = \begin{bmatrix} 0 & -\zeta & 0 \\ \zeta & 0 & \beta \\ 0 & 0 & 0 \end{bmatrix},$$

(17)

with $\xi = T_s \left( \frac{1-\bar{\mu}}{\eta} \right)$, $\beta = \frac{T_s}{2}$, $\zeta = T_s \left( \frac{1-\bar{\mu}}{p\eta} \right)$.

From Assumption 1, it follows

$$v_{r}^{rf} = \bar{v}_{r}^{rf} + \Delta v_{r}^{rf} \quad \Rightarrow \quad \Delta v_{r}^{rf} \in [\delta v_{r}^{rf,m}, \delta v_{r}^{rf,M}],$$

(18)

$$v_{l}^{rf} = \bar{v}_{l}^{rf} + \Delta v_{l}^{rf} \quad \Rightarrow \quad \Delta v_{l}^{rf} \in [\delta v_{l}^{rf,m}, \delta v_{l}^{rf,M}],$$

(19)

where $\bar{v}_{r}^{rf}$ and $\bar{v}_{l}^{rf}$ are the reference nominal velocities of the right and left wheels, respectively, and $\Delta v_{r}^{rf}$ and $\Delta v_{l}^{rf}$ can be seen as the new ranges in the reference velocities. Hence, the matrix $A_{\gamma}$ can be expressed as

$$A_{\gamma} = A^n + A^r \Delta v_{r}^{rf} + A^l \Delta v_{l}^{rf},$$

(20)

where $A^n = I_3 + A^r \bar{v}_{r}^{rf} + A^l \bar{v}_{l}^{rf}$ represents the time-invariant part of matrix $A_{\gamma}$ and the rest of terms are time-varying, which must be bounded. Replacing previous
equation into (15), it is obtained

\[ \bar{e}(k + 1) = [A^n + A^r \Delta v^{rf}_r + A^l \Delta v^{rf}_l + B_d K] \bar{e}(k) + w(k). \]  

In order to compensate the system dynamics, the local control gain $K$ is defined as (see Remark 5)

\[ K = K^n + K^r \Delta v^{rf}_r + K^l \Delta v^{rf}_l, \]  

which implies that equation (21) becomes

\[ \bar{e}(k + 1) = A^n_{cl} \bar{e}(k) + A^r_{cl} \Delta v^{rf}_r \bar{e}(k) + A^l_{cl} \Delta v^{rf}_l \bar{e}(k) + w(k), \]  

where $A^n_{cl} = (A^n + B_d K^n)$, $A^r_{cl} = (A^r + B_d K^r)$, and $A^l_{cl} = (A^l + B_d K^l)$.

Local controller $K^n$ is straightforwardly obtained solving an LQR problem [23]. Local controllers $K^r$ and $K^l$ can be chosen such that the effect of the system dynamics in matrices $A^r$ and $A^l$ is partially compensated in closed-loop. Choosing

\[ K^r = \begin{bmatrix} 0 & -\xi & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad K^l = \begin{bmatrix} 0 & \xi & 0 \\ 0 & 0 & 0 \end{bmatrix}, \]  

it is obtained that

\[ A^r_{cl} = \begin{bmatrix} 0 & 0 & 0 \\ -\xi & 0 & \beta \\ 0 & 0 & 0 \end{bmatrix}, \quad A^l_{cl} = \begin{bmatrix} 0 & 0 & 0 \\ \xi & 0 & \beta \\ 0 & 0 & 0 \end{bmatrix}. \]

**Remark 5.** Notice that $K = K^n + K^r \Delta v^{rf}_r + K^l \Delta v^{rf}_l$ can be seen as an adaptive
local control law which tries to compensate the local mismatch system (15). However, as can be checked from (25) it is not possible to completely compensate the effects on $\hat{e}_y$, but it can be done for the components $\hat{e}_x$ and $\hat{e}_b$. Furthermore, $K^n$ is chosen solving an LQR problem, then leading to stability of the nominal closed-loop dynamics $A^n_{cl}$. On the other hand, we would like to remark that $K$ could be obtained solving online an optimization problem in terms of LMI similarly to the approach detailed in [7]. However, this issue would lead to increase the online computation burden at each step within the prediction horizon.

We have to compute the sequence of sets that can be reached by the closed-loop local uncertain system (15) iteratively with $R_0 = \{0\}$ and the following iteration

$$R_{j+1} \triangleq \bigcup_{A_T \in \gamma} (A_T + B_d K) R_j \oplus W, \quad \forall j = 0, \ldots, N - 1,$$

(26)

where $R_j$ is the reachable set of the $j$-th step in the prediction horizon $N$. From Remark 5, the reachable sets obtained from (26) can be computed as

$$R_{j+1} = A^n_{cl} R_j \oplus \text{co}\{(A^n_{cl} \delta v^f_m R_j) \cup (A^n_{cl} \delta v^f M R_j)\} \oplus$$

$$\oplus \text{co}\{(A^n_{cl} \delta v^f M R_j) \cup (A^n_{cl} \delta v^f M R_j)\} \oplus W,$$

(27)

$$\forall j = 0, \ldots, N - 1.$$

Notice that this formulation is slightly different from that exposed in [15], since now the parameter dependence of the system dynamics is taken into account in the reachable sets calculation.
3.3. Robust Tubed-based MPC

When dealing with uncertain systems, deterministic MPC is limited -although some degree of robustness is achieved due to feedback [24]- since the uncertainties are not explicitly considered in the synthesis of the control law to guarantee robust stability [4]. As explained above, the concept of tube-based MPC implies the replacement of the original constraints (11) with more restricted ones [15]. Following the ideas of Chisci et al., imposing that \( \tilde{e}_j \in \tilde{E}_j \), \( \forall j = 0, \ldots, N \), where \( \tilde{E}_j \) is defined as

\[
\tilde{E}_j = E \ominus R_j, \quad \forall j = 0, \ldots, N - 1,
\]

then constraints satisfaction is ensured and feasibility is also preserved in the presence of uncertainties in the system (7). In addition, the input constraints are also replaced by

\[
\tilde{U}_j = U \ominus \left( K^n R_j \oplus K^r \Delta u^f_i R_j \oplus K^l \Delta u^f_i R_j \right), \quad \forall j = 0, \ldots, N - 1.
\]

Remark 6. It is important to point out that reachable sets sequence carries all the information about every possible trajectory, which is a function of the realization of the uncertainty. For that reason, large system uncertainty could lead to empty or tiny restricted state and input constraints (sets \( \tilde{E}_j \) and \( \tilde{U}_j \)). However, it is noticed that the problem of feasibility is hardly avoidable, and this fact constitutes an important issue of tube-based MPC approaches [15], [16], [18]. For the case under consideration, a trade-off has been realized, mainly varying the prediction horizon, in order to achieve an acceptable size of the reachable sets.
3.4. MPC Strategy

Assume that a measurement of the state $\tilde{e}$ is available at the current time $k$. Then, the optimization problem is stated as follows [15], [16]

$$\min_{G(k) \triangleq \{g_k, \ldots, g_{k+N-1}\}} J_N(G(k), \tilde{e}(k)),$$

subject to

$$\tilde{e}_{k+j|k} \in \tilde{E}_j \quad \forall j = 1, \ldots, N,$$

$$g_{k+j} \in \tilde{U}_j \quad \forall j = 0, \ldots, N - 1,$$

$$\tilde{e}_{k+N|k} \in \Omega \otimes \mathcal{B}_N,$$

where $\tilde{e}_{k+j|k}$ denotes the predicted state vector at time $k+j$, obtained by applying the input sequence $G(k) \triangleq \{g_k, \ldots, g_{k+N-1}\}$ to model (12) starting from the state $\tilde{e}(k)$. Notice that the MPC includes the new state and input constraints (28)-(29). The considered cost function is given by

$$J_N(G(k), \tilde{e}(k)) = \sum_{j=0}^{N-1} \tilde{e}_{k+j|k}^T Q \tilde{e}_{k+j|k} + g_{k+j}^T R g_{k+j} + \Upsilon(\tilde{e}_{k+N|k}),$$

(31)

where $Q = Q^T = [q_x \ q_y \ q_\theta]I_3 \geq 0$ and $R = R^T = [r_r \ r_t]I_2 > 0$. Notice that $Q$, $R$, constitute parameters to be tuned for the MPC control law. Depending on their values more attention can be given to the states or to the control signals (see discussion in Section 4). The terminal cost $\Upsilon(\cdot)$, and the terminal constraint set given by the region $\Omega$ in (30), are both calculated as illustrated in the subsequent section.

To summarize, the MPC control law is based on the following idea [25]: At
time \( k \), compute the optimal solution \( G^*(k) = \{ g_k^*, \ldots, g_{k+N-1}^* \} \) to problem (30) and apply

\[
u(k) = K\tilde{e}(k) + g_k^*
\]

as input to system (7); repeat the optimization (30) at time \( k + 1 \) based on the new state \( \tilde{e}(k + 1) \), and continue iteratively.

3.5. Terminal Constraints for nominal MPC

A common approach to ensure the stability of deterministic MPC (30) consists in incorporating both a terminal cost, \( \Upsilon \), and a terminal constraint set, \( \Omega \) [1]. Notice that we are ensuring asymptotic stability for nominal MPC. The stability for the overall controlled system is not demonstrated since it is beyond the scope of the paper. Stability for similar approaches is demonstrated in [15] and [26].

The purpose of the terminal cost is to ensure closed-loop stability. To this end, it requires the use of a Lyapunov function with a stabilizing control law. In this case, given a quadratic function \( \Upsilon(\tilde{e}) = \tilde{e}^TP\tilde{e} \), the stability of the system is guaranteed if there exists a matrix \( P > 0 \) such that \( \Upsilon(\tilde{e}(k + 1)) - \Upsilon(\tilde{e}(k)) < 0 \), for all \( \tilde{e}(k) \neq 0 \) (Lyapunov function) [27], [28].

Using the LMI formulation [27], [28] to obtain matrix \( P \) and optimizing with respect to the LQR, the following inequality is defined

\[
\tilde{e}^T((A_{cl}^i)^T P A_{cl}^i)\tilde{e} - \tilde{e}^T P\tilde{e} \leq \tilde{e}^T(-Q - (\kappa^i)^T R\kappa^i)\tilde{e}, \quad \forall i = 1, \ldots, n_\gamma.
\]

Notice that the previous inequality is only imposed at the extremal values of the polytopic set \( \Gamma \), i.e. at the \( n_\gamma = 2^2 \) vertices of \( \Gamma \). It can be proved that the ful-
fillment of such conditions at the vertices yields the satisfaction at any point in $\Gamma$ [7].

The closed-loop matrices $A_{cl}^i$ are given by

$$A_{cl}^i = A^i + B_d \kappa^i, \quad \forall i = 1, \ldots, n_y,$$  

(34)

where $\kappa^i$ are the gains ensuring the stability of the closed-loop system, and

$$A^1 = (A^n + A^r \delta v_{r}^{rf, m} + A^l \delta v_{l}^{rf, M}),$$  

(35)

$$A^2 = (A^n + A^r \delta v_{r}^{rf, m} + A^l \delta v_{l}^{rf, m}),$$  

(36)

$$A^3 = (A^n + A^r \delta v_{r}^{rf, M} + A^l \delta v_{l}^{rf, m}),$$  

(37)

$$A^4 = (A^n + A^r \delta v_{r}^{rf, M} + A^l \delta v_{l}^{rf, M}).$$  

(38)

Using (33) and applying the Schur complement [27], it becomes

$$\begin{bmatrix}
P - Q - (\kappa^i)^T R \kappa^i & (A_{cl}^i)^T \\
A_{cl}^i & P^{-1}
\end{bmatrix} \geq 0,$$  

(39)

for the $i$-th vertex. From previous inequality, it holds

$$\begin{bmatrix}
P & (A_{cl}^i)^T & Q^{\frac{1}{2}} & (\kappa^i)^T R^{\frac{1}{2}} \\
A_{cl}^i & P^{-1} & 0 & 0 \\
Q^{\frac{1}{2}} & 0 & I & 0 \\
R^{\frac{1}{2}} \kappa^i & 0 & 0 & I
\end{bmatrix} \geq 0,$$  

(40)

for the $i$-th vertex, with $I$ being the identity matrix. Now, pre- and post- multiply-
ing by

\[
\begin{bmatrix}
P^{-1} & 0 & 0 & 0 \\
0 & I & 0 & 0 \\
0 & 0 & I & 0 \\
0 & 0 & 0 & I
\end{bmatrix},
\]

and replacing \( S = P^{-1}, Y^i = \kappa^i P^{-1}, \) and \( A_{cl}^i = A^i + B_d \kappa^i, \) it is obtained

\[
\begin{bmatrix}
S & S(A^i)^T + (Y^i)^T B_d^T & SQ_1^+ & (Y^i)^T R_1^+ \\
A^i S + B Y^i & S & 0 & 0 \\
Q_1^+ S & 0 & I & 0 \\
R_1^+ Y^i & 0 & 0 & I
\end{bmatrix} \geq 0.
\]

Then, the optimization problem to be solved is given by

\[
\begin{align*}
\min_{S \geq 0, \ Y^i} & \quad \text{Tr}(P) \\
\text{subject to} & \quad (42) \quad \forall i, i = 1, \ldots, 4,
\end{align*}
\]

where \( \text{Tr}(\cdot) \) denotes the trace of a matrix [27]. The solution of this optimization problem produces four gains \((\kappa^1, \kappa^2, \kappa^3, \kappa^4)\) and the matrix \( P \) which defines the terminal cost.

Finally, in order to calculate the terminal region \( \Omega \) as a robust positively invariant set for the system, the ideas in [17] and [18] have been adapted to this particular case. Details are presented in Appendix.
4. Results and Discussion

The aim of this section is to validate the performance of the robust tube-based MPC control law and to compare it with existing time-varying control techniques. In this case, the well-known linear time-varying controller described in [19] has been implemented (denoted in figures as “No slip comp.”). Furthermore, in order to compare the new proposed formulation with a controller that compensates slip effects, the work presented in [21] has also been considered (referred to as “Slip comp.” in figures). This control strategy constitutes an easy extension of the former feedback control law, but now slip explicitly appears as a parameter in determination of the feedback gains. For details about this strategy, see [21]. The reference trajectories have been calculated based on unicycle kinematics.

Simulations have been carried out in Matlab® suite using the LMI toolbox [29] and MPT toolbox [30]. Physical experiments were programmed using LabVIEW® and Matlab®.

Some tuned parameters used for simulation and physical experiments are: $b = 0.5[m]$, $T_s = 0.35[s]$, and $N = 5$. This prediction horizon has been selected as a trade-off between satisfactory performance and an acceptable size of the reachable sets (see Remark 6).

As commented in Section 3.4, the values of matrices $Q$ and $R$ influence the performance of the MPC controller. For that reason, an important trade-off has been carried out to obtain the proper values. The most interesting conclusions are:

- For small values of $q_\theta$ and $r_I (< 1)$, the controller achieves small deviations in longitudinal and lateral directions, but not appropriate orientation error and oscillatory signals.
• Large values of $q_\theta$ and $r_l (> 10)$ produce a large error in the longitudinal direction and in the orientation, and a really oscillatory behavior in the control signals.

• Values of $q_\theta$ smaller than 1 and $R = I_2$ yield a satisfactory error in robot orientation and satisfactory control signals.

Therefore, $Q = diag([1 \quad 1 \quad 0.0001])$ and $R = I_2$ have been chosen in order to obtain an appropriate performance, small errors and smooth control signals. The sampling period has been empirically selected taking into account the desired closed-loop performance. The parameters of the two selected controllers are $\beta_c = 1$ and $\delta_c = 0.6$ to reach a soft overdamped closed-loop behavior. For more details about these parameters see [19], [21].

4.1. Experimental mobile robot

The vehicle under consideration is a TMR available at the University of Almería (Spain) called Fitorobot, which has been used as the testbed in this work (see Figure 2). This TMR has been designed to operate inside greenhouses for spraying purposes. As presented in [31], greenhouse terrain is composed of loose sandy soils, which can produce significant slip phenomena in vehicles. Furthermore, greenhouses are characterized by narrow corridors between plants, in which this robot has to operate. Hence, hard constraints satisfaction becomes an essential factor of the robust control problem, that means, small lateral error should be ensured to avoid that the robot collides with the plants.

The mobile robot has a mass of 500 [Kg] and its dimensions are 1.5 [m] long $\times$ 0.7 [m] wide. It is driven by a 20 [HP] gasoline engine. More details about the TMR can be found in [31].
4.2. Simulations

In this subsection, some simulations carried out before the physical experiments are detailed. Due to the complexity of the physical system (track-soil interactions, resistance forces, inertia forces, etc.), the kinematic models have been simulated. However, to make more realistic simulations, a small random noise (Gaussian distributed noise) was added to the states.

In order to check the robustness of the controllers, nominal slip for each wheel is set to \(10\%\), and it is supposed that slips vary within \(\pm 15\%\). The uncertainty set is given by \(W = \{ w_1, w_2 \in \pm 0.0035 [m], w_3 \in \pm 0.25 [deg] \} \). State constraints

\[ \]

\[ ^1 \text{Notice that, for notational convenience, slip is shown in per cent.} \]

\[ ^2 \text{Recall that } W \text{ represents the uncertainty on the state for each time instant. In particular, it bounds the mismatch between the continuous-time non-linear trajectory tracking error model and} \]

22
are $E = \{ e_x, e_y \in [0.5, \pm 0.5][m], e_\theta \in [\pm 20][deg] \}$ and reference linear wheel velocities are restricted to $\{ v_{r}^{rf}, v_{f}^{rf} \in [0.1, 1.4][m/s] \}$. Real linear track velocities were also experimentally determined as $\{ v_r, v_l \in [-2.2, 2][m/s] \}$. In all cases, the initial location of the mobile robot was the same that the reference.

Although many trajectories have been tested, a reference trajectory that comprises the full reference velocity range has been tested. In this case, the reference velocities are always changing between the bounds and never become constant. The total traveled distance is close to 70 [m]. As depicted in Figure 3b, slips have been simulated between 0-25 [%], and reference velocities vary between 0.1-1.4 [m/s].

Figure 3a shows the reference trajectory and the followed trajectories using the three compared control strategies. In this case, the predictive controller fixes properly the reference.

The errors between the reference trajectory and those steered by the compared controllers are displayed with respect to the traveled distance in Figure 4. Notice that, although a small random noise was added to the states, the MPC controller achieves an almost zero error in the longitudinal and lateral directions and a maximum orientation error of 0.80 [deg]. The rest of controllers have a maximum lateral error of -0.10 [m] for the slip compensation controller, and -0.34 [m] for the no slip compensation controller; a maximum longitudinal error of 0.07 [m] for the slip compensation controller, and 0.27 [m] for the no slip compensation controller, and finally, a maximum orientation error of -3.16 [deg] for the slip

the discrete-time linear model, the noise in the slip estimation, and the uncertainty in the robot localization. These values were obtained experimentally. Then, a value of $w_1 \in [\pm 0.0035][m]$ represents an uncertainty in the $e_x$ state of $\pm 1[cm]$ each second.

Figure 5 displays the control inputs or linear velocities of the wheels. Notice
the non-static reference velocities comprising all the range, that is, from 0.1 [m/s]
to 1.4 [m/s]. In this figure, it can be observed how controllers compensate the
effect of slip, that is, since slip decreases the velocity of the wheels, the motion
controllers increase the set-points in order to compensate such negative effect. In
this figure, it is possible to observe the oscillatory control actions generated by the
no slip compensation controller. For instance, at sampling instant 45 [s] for the
right track and at sampling instant 27 [s] for the left track.

As commented in Remark 6, restricted state and input constraints are not
empty or tiny for the proposed control law. In Figure 6, the state constraints and
the terminal robust positively invariant set are shown.
4.3. Physical experiments

In this subsection, the results of the physical experiments carried out using the real mobile robot are presented. Before discussing these experiments, some considerations about practical issues are pointed out:

- A typical four-layer navigation architecture [32] has been designed and implemented in order to attempt physical experiments (see Figure 7). The first layer is devoted to path planning. The second layer includes the motion controller presented in this paper. In the third layer, two low-level controllers have been implemented in order to ensure that the set-points generated by
the motion controllers are reached by the robot tracks. For that purpose, two PI controllers with an anti-windup scheme have been tuned. The fourth layer is related to the robot localization.

- For localization purposes, the extended kinematic model (3), which was
extended considering slip, was employed. In this case, slip was estimated using a visual-odometry-based approach [33], [34]. Visual odometry is defined as the incremental online estimation of robot motion from an image sequence by an on-robot camera [34]. Notice that it means that typical and undesirable effects of wheel-based odometry are minimized, since visual information gives the actual robot velocity. In particular, a camera pointing at the ground beneath the robot was employed. Visual odometry was implemented using a template matching approach. Furthermore, a threshold filter was employed to attenuate outliers due to mainly false matches [35].

The tests were carried out on an off-road gravel terrain. Although this is not an ideal site for the experiments, since it is a partially bumpy terrain, the results presented here are illustrative of the performance of the compared controllers. Figure 8 details the soil of the experiment site. Notice the sinkage effect in both views. This sinkage leads to longitudinal slip.

The experiments were carried out with sunlit conditions, and thus some shadows that could lead to false matches in the visual odometry algorithm were observed. Nevertheless, these false matches were minimized selecting a proper position for the camera looking at the ground and an appropriate filter.

In order to validate the followed trajectories and orientations, a DGPS (R100, Hemisphere, Calgary, Canada) and a magnetic compass (C100, KVH, Middletown, USA) have been employed as ground-truth. The DGPS has an accuracy of 0.20 [m] and the magnetic compass of 0.1 [deg]. The rest of sensors were: one consumer-grade camera (Quickcam Sphere AF, Log-
itech, Apples, Switzerland) and two incremental encoders attached to the track drive sprockets (DRS61, Sick AG, Waldkirch, Germany).

Figure 7: Control architecture of the testbed

(a) Robot on gravel terrain

(b) Soil detail

Figure 8: Experiment site (gravel soil)

- Although many experiments have been carried out, a circular trajectory and an U-shaped trajectory have been selected in this case. In the former, the
robot moved at middle/high velocities ($\geq 0.5[m/s]$) and in the second one it moved at low velocities ($\leq 0.4[m/s]$). Notice that similar trajectories to these selected here are usually employed in off-road mobile robotics, see for instance [10], [36].

The same parameters used for the simulation were employed during the physical experiments.

4.3.1. Experiment 1. Circular trajectory

In this first experiment, a reference circular trajectory was tested. The motivation for selecting this trajectory is because it is a closed trajectory, what helps in a better comparison of the controllers, since the robot has to come back to the starting point. Furthermore, the reference track velocities were 0.6 [m/s] for the right track and 0.5 [m/s] for the left track. For the testbed, these can be considered as middle/high velocities. In this first selected experiment, the total traveled distance was close to 18 [m].

Figure 9a shows the reference trajectory and the followed trajectories using the three control strategies. Figure 9b shows the reference trajectory, the trajectory followed using the MPC controller, and the ground-truth (DGPS). In the former plot, it is possible to observe that the MPC controller achieves the best behavior. From Figure 9b, it is possible to notice that there is a small mismatch between the trajectory obtained using the MPC controller and the ground-truth. Nevertheless, the maximum lateral deviation is 0.53 [m], and the mean lateral deviation is 0.18 [m], which means 1 [%] of the total traveled distance. It must bear in mind that since a kinematic model is employed, the error-growth effect of kinematic-model-based solutions is not fully avoided, although it is minimized using the visual
odometry approach. This explains why there is a small mismatch between the trajectory followed using the MPC controller and the ground-truth.

![Followed trajectories](image1.png)

(a) Followed trajectories

![MPC-trajectory vs ground-truth](image2.png)

(b) MPC-trajectory vs ground-truth

Figure 9: Experiment 1. Followed trajectories during the experiment

Figure 10a displays the orientations of the compared controllers. It can be observed the proper behavior of the MPC controller with few oscillations. Figure 10b plots the reference orientation, the orientation obtained using the MPC controller, and the ground-truth (magnetic compass). As expected from the previous discussion, the orientation obtained using the MPC controllers follows the ground-truth with small deviation. Particularly, the mean deviation is 7.62 [deg].

Figure 11 shows the slip estimation for each track. Recall that the slip is obtained as the relation between the actual robot velocity (visual odometry) and the theoretical velocities of the tracks (incremental encoders). In this particular experiment, the tracks move at different velocities. For that reason, the obtained slip estimations are different. Notice that this fact can lead to a mistake since the same actual velocity is being supposed for both tracks. In order to minimize this
mistake presented in circular trajectories, the mean value of both slip estimations was considered. In further works, two independent cameras will be employed to estimate the actual track velocity. The mean slip is 4.41 [%] (gravel soil).

Figure 10: Experiment 1. Orientations

Figure 11: Experiment 1. Slip estimation for the case of the MPC controller
The errors between the reference trajectory and those steered by the compared controllers are displayed in Figure 12. In these plots, the proper behavior of the predictive controller is verified. The slip compensation controller also achieves a satisfactory result in comparison to the no slip compensation controller. As in simulations, a positive longitudinal error is observed. This means that the real robot is pursuing the reference robot. The mean longitudinal errors are: 0.05 [m] with standard deviation 0.05 [m] for the MPC controller, 0.09 [m] with standard deviation 0.10 [m] for the slip compensation controller, and 0.13 [m] with standard deviation 0.10 [m] for the no slip compensation controller. The mean lateral errors are: -0.002 [m] with standard deviation 0.05 [m] for the MPC controller, -0.01 [m] with standard deviation 0.08 [m] for the slip compensation controller, and -0.02 [m] with standard deviation 0.07 [m] for the no slip compensation controller. The mean orientation errors are: 0.50 [deg] with standard deviation 5.80 [deg] for the MPC controller, 1.80 [deg] with standard deviation 4.84 [deg] for the slip compensation controller, and 2.27 [deg] with standard deviation 4.73 [deg] for the no slip compensation controller.

Figure 13 shows the reference velocities (denoted as $v_r^f$ and $v_l^f$), the control inputs (referred to as $v_r$ and $v_l$), and the real track velocities (labelled as “Right Enc.” and “Left Enc.”) for the case of the MPC controller. As in simulation, the motion controller increases the control inputs to compensate the negative slip effect. It is interesting to point out the proper behavior of the low-level PI controllers, which are responsible to achieve the set-points given by the motion controllers.

Finally, Figure 14a displays the pixel displacement values between consecutive images for the case of the MPC controller (visual odometry), and the compu-
Figure 12: Experiment 1. Longitudinal, lateral, and orientation errors in the physical experiment.
Furthermore, it is checked that sampling period is always ensured. In this case, the mean computation time for the MPC controller is 0.25 [s] and for the slip compensation controller is 0.20 [s] (the no slip compensation controller achieves a similar value). Notice that motion controllers run on the computer on-board the mobile robot (Intel Pentium III 1GHz, 512 MB RAM).

4.3.2. Experiment 2. U-shaped trajectory

In this second experiment, an U-shaped trajectory has been selected. It constitutes an interesting trajectory which combines straight-line motions and two 90-degrees turns. The total traveled distance was close to 30 [m]. In order to check the navigation architecture at low velocities, a maximum reference velocity of 0.3 [m/s] was selected.

Figure 15a shows the reference trajectory and the followed trajectories using the three control strategies. Figure 15b displays the reference trajectory, the trajec-
Figure 14: Experiment 1. Pixel displacements (MPC controller) and computation time (MPC vs Slip comp. controllers)

tory followed using the predictive controller and the ground-truth (DGPS). In the former plot, it is possible to observe that the trajectory obtained using the predictive controller follows satisfactorily the reference. The trajectories obtained using the linear feedback controllers have a smooth oscillatory behavior, especially after the 90-degrees turns. As in the previous experiment, the trajectory followed using the predictive controller slightly diverges from ground-truth. Particularly, the maximum lateral deviation is 0.72 [m], and the mean lateral deviation is -0.11 [m], which means 0.40 [%] of the total traveled distance.

Figure 16a shows the orientations. From this figure, it is clearly observed the smooth oscillatory behavior of the linear feedback controllers, especially at the end of the experiment. The orientation obtained using the predictive controller follows the reference with very small oscillations. Figure 16b plots the reference orientation, the orientation obtained using the predictive controller, and
the ground-truth (magnetic compass). As expected, the orientation obtained using the predictive controller does not fix the ground-truth. In this case, the mean deviation is $-0.11$ [deg] with standard deviation $6.47$ [deg].

Figure 17 displays the slip estimations. In this case, the median slip value for the right track was $6.32$ [%], and for the left track was $2.17$ [%]. As in previous experiment, both slip values are different and the reason is that the same actual velocity has been used for both tracks.

The errors between the reference trajectory and those steered by the compared controllers are plotted in Figure 18. Especially remarkable is the almost zero lateral error achieved by the MPC controller. The mean longitudinal errors are: $0.02$ [m] with standard deviation $0.03$ [m] for the MPC controller, $0.09$ [m] with standard deviation $0.05$ [m] for the slip compensation controller, and $0.13$ [m] with standard deviation $0.06$ [m] for the no slip compensation controller. The mean lateral errors are: $0$ [m] with standard deviation $0.04$ [m] for the predictive
controller, 0.02 [m] with standard deviation 0.17 [m] for the slip compensation controller, and 0.01 [m] with standard deviation 0.18 [m] for the no slip compensation controller. The mean orientation errors are: 0.43 [deg] with standard devi-
ation 4.74 [deg] for the MPC controller, 0.66 [deg] with standard deviation 8.16 [deg] for the slip compensation controller, and 1.09 [deg] with standard deviation 8.51 [deg] for the no slip compensation controller. As in the previous experiment, the predictive control strategy obtains the best results. Furthermore, note that the rest of compared controllers oscillate between maximum values closer or greater to 0.3 [m], which could be unacceptable for some kind of applications.

Figure 18: Experiment 2. Longitudinal, lateral, and orientation errors in the physical experiment

Figure 19 displays the reference velocities, the control inputs and the real track velocities for the case of the predictive controller. These plots notice the proper
tuning of the PI controllers, since the real velocities of the tracks follow satisfactorily the set-points. It is important to clarify that the smooth oscillatory behavior of the right track is stressed by that fact that the mobile robot was moving on a partially bumpy gravel soil. In this scenario, little stones produce vibrations to the vehicle, and hence to the tracks. Notice that in the previous experiment, a similar behavior was observed.

Figure 19: Experiment 2. Control signals (track velocities) in the physical experiment

Finally, Figure 20a displays the pixel displacement between consecutive images for the MPC controller (visual odometry). As in previous experiment, few outliers appear due to the proper downward camera position and the threshold filter. Notice that in this experiment the mean value of the pixel displacement is around -50 [pixel]. This confirms that the mobile robot was moving at a slower velocity than in previous experiment. Figure 20b shows the computation time employed by the predictive controller and the slip compensation controller. The average computation time for the predictive controller is 0.24 [s] and for the slip
compensation controller is 0.19 [s] (the no slip compensation controller achieves a similar value).

![Image](image_url)

(a) Visual odometry data  
(b) Computation time

Figure 20: Experiment 2. Pixel displacements (MPC controller) and computation time (MPC vs Slip comp. controllers)

5. Conclusions

This paper proposes a robust tube-based predictive control law for constrained mobile robots in off-road conditions.

From the theoretical point of view, an extension of tube-based predictive control to time-varying systems is provided (via reachable sets). From a practical point of view, the designed control strategy compensates slip and guarantee that state and input constraints are fulfilled, since these constraints are taken into account in the solution of the optimization problem. The time-varying trajectory tracking error model with additive uncertainties has been addressed. Real-time execution has been ensured using tube-based MPC formulation and it has been
tested through simulations and physical experiments. The comparative study (simulations and physical experiments) with other control laws illustrates the proper behavior of the robust tube-based MPC strategy.

Summing up, the main advantages of the proposed control scheme are: robustness, efficient online computation, state and input constraints fulfillment and compensation of longitudinal slip. It is interesting to point out that the proposed approach can be easily applied to other robot configurations. The main limitation of the current strategy is a certain degree of conservativeness, mainly related to calculation of reachable sets. Then, future efforts will be devoted to reduce such degree of conservatism. Furthermore, a more precise estimation of the uncertainty set $W$ and new physical experiments will be tested.

Acknowledgments

This work has been supported in part by the Spanish CICYT under grants DPI 2007-66718-C04-01, DPI 2007-66718- C04-04, and DPI2010-21589-C05-04. The authors would like to thank professors R. Siegwart and C. Pradalier from the Autonomous System Lab, ETH Zürich (Switzerland) for introduce us the visual odometry localization technique.

References


Appendix A. Robust positively terminal invariant set

In what follows, once a stabilizing control law is given, we provide a procedure to obtain the maximal robust invariant set for the uncertain system (trajectory tracking error model with slip) contained in the state constraint set. In addition, constraints on the input are considered.

Let $A_\gamma \in \mathcal{A}$, with $\mathcal{A}$ a polytope in $\mathbb{R}^{3 \times 3}$ holds for the system (7), and the input to be constrained, $u(k) \in U$. Given a control feedback gain $\kappa(A_\gamma) = \lambda_1 \kappa^1(A_\gamma^1) + \ldots + \lambda_4 \kappa^4(A_\gamma^4)$ (each gain is calculated in (42)), and a convex set $\Omega \subseteq \mathbb{R}^3$, the concept of one-step operator is employed as

$$Q_\mathcal{A}(\Omega) = \{ e \in E : \kappa(A_\gamma) e \in U, \quad (A_\gamma + B_d \kappa(A_\gamma)) e + w \in \Omega, \forall w \in W, \forall A_\gamma \in \mathcal{A} \}.$$

Notice that the one-step operator is a standard tool for the invariant sets calculation through iterative procedures [18]. Parameters $\lambda_j$ with $j = 1, \ldots, 4$, can be obtained by solving a linear programming problem (LP).

Finally, taking into account the one-step operator previously defined, the maximal robust invariant set for the uncertain system (7) is obtained by means of the following iterative procedure:

1. Initialization: $\Omega_0 = E \cap \{ \omega \in \mathbb{R}^3 : \kappa(A_\gamma) \omega \in U, \forall A_\gamma \in \mathcal{A} \}$.

2. Iteration: $\Omega_{k+1} = \Omega_k \cap Q_{\mathcal{A}}(\Omega_k)$.

3. Termination condition: stop when $\Omega_{k+1} = \Omega_k$ or $\Omega_{k+1} = \emptyset$. Set $\Omega = \Omega_\infty = \Omega_{k+1}$.

An important issue when dealing with algorithmic procedure for computing the robust invariant sets is its finite determinedness, that is, the conditions under which the algorithm provides a solution after a finite number of iterations. Results regarding the problem of finite determination can be found in [18], [17]. Finite determinedness has not been proved for this work, since it goes beyond the aim of the paper. Nevertheless, the application of the algorithm to the case under analysis provides a result after a finite number of iterations.