Abstract

This paper presents a control law for the trajectory tracking of mobile robots under slip conditions and subject to both system constraints and varying dynamics. First, a control law is obtained based on a Lyapunov function to guarantee closed-loop asymptotic stability, resulting in a set of feedback gains, one for each extreme model realization. On-line computation is devoted to determine an adaptive feedback control law for the current realization of the state as a convex combination of the gains previously obtained. Simulations comparing the proposed control law with other strategies under slip conditions are provided. They show the satisfactory behavior of the proposed control strategy.

Keywords: Adaptive control, Autonomous Mobile robot, Linear Matrix Inequalities, Slip
I. INTRODUCTION

Mobile robot control systems must deal with both state and input constraints, i.e. physical limitations of actuators, non-holonomic constraints, narrow workspace, etc. Furthermore, mobile robots constitute non-holonomic systems, which cannot be stabilized by smooth static state feedback laws [8]. These systems fail in the Brockett’s Condition for the existence of a continuously differentiable control law, since the dimension of the state space is three and the number of control signals is only two [6]. In order to solve this problem, discontinuous feedback control laws [4] and adaptive continuous feedback control laws have been commonly used [26], [9], [20].

On the other hand, mobile robots operating on off-road conditions present some phenomena as slip or sliding which cause that rolling of a wheel is not perfect [31]. Thereby, the guidance and the controllability of the mobile robot is considerably influenced by the condition of terrain [13]. For that reason, one key issue is to design motion controllers which compensate slip effects. For example, a study for four generic wheeled mobile robots in the presence of wheel skidding and slipping from a control perspective is developed in [30]. Disturbances due to skidding and slipping are categorically classified as input-additive, input-multiplicative, and / or matched / unmatched perturbations. A linear feedback control law for a Tracked Mobile Robot (TMR) is presented in [13], where gains are adapted according to the longitudinal slip measured in real-time. In [10], [21], the problem is addressed for an Ackermann-type agricultural vehicle in which adaptive and predictive control techniques are used to face the lateral slip effects. The work presented in [22] proposes a control for a TMR based on a kinematic approach using the different values of the instantaneous rotation center (ICR) of the tracks. The ICR position depends on the track-soil interactions.

This paper focuses on the synthesis of an adaptive control law which guarantees asymptotic stability for a linear, time-varying, discrete-time system subject to both constraints and varying dynamics. The constraints are imposed supposing that the mobile robot operates in narrow spaces and since the actuators are physically saturated. The objective is to determine an adaptive control law and a Lyapunov function guaranteeing the asymptotic stability of the closed-loop system. We formulate the problem in terms of Linear Matrix Inequalities (LMI) optimization problem [5], [17], in order to obtain a positive definite matrix $P$ determining the Lyapunov function, and a set of feedback gains composing the control law. This problem is solved off-line for each extreme realizations of the system dynamics. Afterwards, an on-line adaptive feedback control law is used depending on the current system realization. This adaptive
control law is obtained as a convex combination of the previously determined gains.

It is important to remark that the control law presented here tries to compensate the slip effect, considering the slip in the control design. An interesting alternative is to include a velocity/acceleration limiter that would prevent the robot’s wheel from slip [16], [25]. However, as discussed in [31], this solution could become unsuitable in practice. The reason is that off-road terrains are intrinsically loose, producing a noncontrollable slip, that is, the robot will slip although velocity and acceleration are limited.

LMI-based solutions have been satisfactorily applied in other mobile robotics problems. For instance, backing control of simulated mobile robots with multiple trailers by fuzzy modelling and control is presented in [29]. LMI are used to solve the problem of finding stable feedback gains and a common Lyapunov function. In [1], a feedback path controller for an articulated mining vehicle based on LMI techniques to guarantee stability of the closed-loop system is proposed. In [32], a robust tracking problem of wheeled mobile robots subject to non-holonomic constraints and input constraints is discussed. In the framework of LMI, the suggested tracking scheme is formulated as an on-line controller which is obtained solving a constrained $H_\infty$ control law. The work [23] shows a method for motion planning of mobile robots. The free configuration space is decomposed into Delaunay triangles, and an optimum channel from initial to goal configurations is found by solving an LMI system.

The paper is organized as follows: a modification of the standard kinematic model including slip effects, and the trajectory tracking error model, are presented in section 2. The following section is devoted to obtain the adaptive control law using the LMI-based approach to guarantee stability under input and state constraints. Simulations of the proposed control law with other control strategies are detailed in section 4. Finally, conclusions and future trends are summarized in section 5.

II. TRAJECTORY TRACKING BASED ON KINEMATIC MODEL WITH SLIP

In this section, we present a modified formulation of the well-known kinematic model of a differential-drive wheeled mobile robot [28], [7]. For that purpose, this kinematic model has been extended with a parameter which weights the slip factor of the terrain [13]. In this case, we suppose that the mobile robot will operate at low velocities, and we only consider longitudinal slip. As stated in [15], [27], [19], 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 pneumatic tire compression
reaction to soil shearing due to the own characteristics of wheeled / tracked locomotion [31], [13], [30].

Furthermore, the trajectory tracking or posture tracking problem is also described and the error state space system is obtained using the modified kinematic model.

A. Kinematic Model under slip conditions

When wheel slip is not considered, the linear velocity of the wheels is [28]

\[ v_r(t) = \rho \phi_r(t), \]
\[ v_l(t) = \rho \phi_l(t), \] (1)

where \( t \in \mathbb{R} \) is the continuous time, \( \rho \) is the wheel (or track) rolling radius, and \( v_r \) / \( \phi_r \) and \( v_l \) / \( \phi_l \) are the linear / angular velocities of the right and left wheels respectively.

As commented above, (longitudinal) slip can be considered as a penalizing factor of the wheel velocity [31], [13]

\[ v_r^{\text{slip}}(t) = \rho \phi_r(t)(1 - i_r(t)), \]
\[ v_l^{\text{slip}}(t) = \rho \phi_l(t)(1 - i_l(t)), \] (2)

where \( i_r \) and \( i_l \) are the terms representing the (longitudinal) slip component of each wheel on a terrain. As shown in [13], slip can be estimated in real-time using the appropriate sensors.

Using this knowledge in the classical kinematic model of a differential-drive robot [28], we obtain

\[ \dot{x}(t) = \frac{v_r(t)(1 - i_r(t)) + v_l(t)(1 - i_l(t))}{2} \cos \theta(t), \]
\[ \dot{y}(t) = \frac{v_r(t)(1 - i_r(t)) + v_l(t)(1 - i_l(t))}{2} \sin \theta(t), \]
\[ \dot{\theta}(t) = \frac{v_r(t)(1 - i_r(t)) - v_l(t)(1 - i_l(t))}{b}, \] (3)

where \([x \ y \ \theta]^T\) represents the location (position and orientation) of the mobile robot, and \( b \) is the distance between the wheels’ centers.

B. Trajectory tracking error model

Trajectory tracking problem can be seen as a problem in which a robot must follow a virtual mobile robot representing the desired positions and velocities, as shown in Figure 1. Hence, the objective is to
find a feedback control law \[ [8], [12] \]

\[ \mathbf{v}(t) = (v_r(t), v_l(t)) = f(p(t), p_{ref}^r(t), v_{ref}^r(t), v_{ref}^l(t)), \]

such that

\[ \lim_{t \to \infty} e(t) = \lim_{t \to \infty} [p(t) - p_{ref}^r(t)] = 0, \]

where the location of the real mobile robot is denoted as \( p = [x \ y \ \theta]^T \), \( p_{ref} = [x_{ref}^r \ y_{ref}^r \ \theta_{ref}^r]^T \) and, \( v_{ref}^r, v_{ref}^l \) are the reference trajectory and linear velocities respectively.

As stated in (5), the control objective is to steer the error, between the desired location and the real location of the mobile robot, close to zero (regulation problem). To express this error with respect to the real robot frame, the following change is considered

\[
\begin{bmatrix}
    e_x(t) \\
    e_y(t) \\
    e_{\theta}(t)
\end{bmatrix} =
\begin{bmatrix}
    \cos \theta(t) & \sin \theta(t) & 0 \\
    -\sin \theta(t) & \cos \theta(t) & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_{ref}^r(t) - x(t) \\
    y_{ref}^r(t) - y(t) \\
    \theta_{ref}^r(t) - \theta(t)
\end{bmatrix},
\]

where \( e_x \) is the longitudinal error, \( e_y \) is the lateral error, and \( e_{\theta} \) is the orientation error. These errors are graphically presented in Figure 1 where the virtual robot is represented in dotted lines and the real robot in solid ones.

Fig. 1. Graphical representation of the trajectory tracking problem
To determine the error along the time, the equation (6) is differentiated producing [13]

\[
\begin{align*}
\dot{e}_x(t) &= \alpha(t)e_x(t) + \cos \theta(t) \frac{\nu^r_{ref}(t) + \nu^l_{ref}(t)}{2} - v_r(t) + v_l(t) + v_r(t)i_r(t) + v_l(t)i_l(t) \\
\dot{e}_y(t) &= -\alpha(t)e_x(t) + \sin \theta(t) \frac{\nu^r_{ref}(t) + \nu^l_{ref}(t)}{2} \\
\dot{e}_\theta(t) &= \left(\frac{\nu^r_{ref}(t) - \nu^l_{ref}(t)}{b} - \frac{v_r(t) - v_l(t)}{b}\right) + v_r(t)i_r(t) - v_l(t)i_l(t),
\end{align*}
\]

where \(\alpha(t) = \left(\frac{v_r(t) - v_l(t)}{b} - \frac{v_r(t)i_r(t) - v_l(t)i_l(t)}{b}\right)\).

In order to linearize the previous equation around the reference trajectory, a first-order Taylor expansion has been used. Furthermore, we have defined the following virtual control signals to eliminate some of the non-linear terms,

\[
\begin{align*}
u_1(t) &= \frac{-1 + i_r(t)}{2} v_r(t) + \frac{-1 + i_l(t)}{2} v_l(t) + \frac{\nu^r_{ref}(t)}{2} + \frac{\nu^l_{ref}(t)}{2}, \\
u_2(t) &= \frac{-1 + i_r(t)}{b} v_r(t) + \frac{1 - i_l(t)}{b} v_l(t) + \frac{\nu^r_{ref}(t)}{b} - \frac{\nu^l_{ref}(t)}{b}.
\end{align*}
\]

Afterwards, equation (6) becomes

\[
\begin{bmatrix}
\dot{e}_x(t) \\
\dot{e}_y(t) \\
\dot{e}_\theta(t)
\end{bmatrix} =
\begin{bmatrix}
0 & \alpha(t) & 0 \\
-\alpha(t) & 0 & \frac{\nu^r_{ref}(t) + \nu^l_{ref}(t)}{2} \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
e_x(t) \\
e_y(t) \\
e_\theta(t)
\end{bmatrix} +
\begin{bmatrix}
1 & 0 \\
0 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
u_1(t) \\
u_2(t)
\end{bmatrix},
\]

where \(\alpha_r(t) = \left(\frac{\nu^r_{ref}(t) - \nu^l_{ref}(t)}{b} - \frac{\nu^r_{ref}(t)i_r(t) - \nu^l_{ref}(t)i_l(t)}{b}\right)\).

Remark 1: Notice that the mismatch between the linearized model and the nonlinear system grows for values of \(e_\theta\) far from 0. It will be shown that, in practice, after the transient, \(e_\theta\) remains very close to zero. Then, the problem could be present at the first instants, due to the initial condition. For that reason, we assume that, in practice, the real robot and the reference virtual robot start close. In that case, the linearization is successful.

Equation (10) is expressed in state space representation as a linear, time-varying, continuous-time system

\[
\dot{e}(t) = A_{\gamma,c}(t)e(t) + B_cu(t),
\]

where \(e = [e_x, e_y, e_\theta]^T\) is the state, \(u = [u_1, u_2]^T\) is the (virtual) control input, and \(\gamma\) is the parameter,
characterized in Remark 2. Matrices $A_{\gamma,c}$ and $B_c$ are defined as

$$A_{\gamma,c}(t) = \begin{bmatrix} 0 & \alpha_r(t) & 0 \\ -\alpha_r(t) & 0 & \frac{v_{r,ref}(t) + v_{l,ref}(t)}{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{12}$$

Then, the trajectory tracking error model (11) is discretized, obtaining the following linear, time-varying, discrete-time system

$$e(k+1) = A_{\gamma}(k)e(k) + B_d u(k) \tag{13}$$

where $k \in \mathbb{Z}^+$ is the discrete sample, and matrices $A_{\gamma}$ and $B_d$ are now defined as

$$A_{\gamma}(k) = \begin{bmatrix} 1 & T_m \alpha_r(k) & 0 \\ -T_m \alpha_r(k) & 1 & T_m \frac{v_{r,ref}(k) + v_{l,ref}(k)}{2} \\ 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} T_m & 0 \\ 0 & 0 \\ 0 & T_m \end{bmatrix}, \tag{14}$$

being $T_m$ the sampling period.

**Assumption 1:** Assume that slip factors and reference robot wheel velocities are known at each time and bounded, i.e. $i_r \in [i_r^m, i_r^M]$, $i_l \in [i_l^m, i_l^M]$, $v_{r,ref} \in [v_{r,ref}^m, v_{r,ref}^M]$, and $v_{l,ref} \in [v_{l,ref}^m, v_{l,ref}^M]$.

**Remark 2:** From Assumption 1, we can define a time-varying vector of parameters $\gamma(k) = [v_{r,ref}^r(k) \quad v_{l,ref}^r(k) \quad i_r(k) \quad i_l(k)]^T \in \mathbb{R}^4$, and a bounding set $\Gamma \subseteq \mathbb{R}^4$, such that $\gamma(k) \in \Gamma, \forall k \in \mathbb{R}$. For any admissible realization of parameter $\gamma \in \Gamma$, a dynamic matrix denoted as $A_{\gamma}$ is determined. Notice that, from Assumption 1, it follows that $A_{\gamma} \in \mathcal{A}$ where $\mathcal{A}$ is a polytope in $\mathbb{R}^{3 \times 3}$.

Note that the model is composed by a family of linear systems (defined by matrix $A_{\gamma}$), each of them is controllable providing that $0 \leq i_r, i_l < 1$ and $v_{r,ref}^r, v_{l,ref}^r > 0$.

For sake of notational simplicity, we omit to express the dependence of $A_{\gamma}(k)$ on $k$, employing $A_{\gamma}$ to refer to it.

**Remark 3:** Note that, according to (8)-(9), the control signals (velocities) are obtained through

$$v_r(k) = \frac{v_{r,ref}^r(k) - u_1(k) - \frac{b}{2} u_2(k)}{1 - i_r(k)} \tag{15},$$

$$v_l(k) = \frac{-v_{l,ref}^l(k) + u_1(k) - \frac{b}{2} u_2(k)}{-1 + i_l(k)} \tag{16},$$

where $v_r \in [v_r^m, v_r^M]$ and $v_l \in [v_l^m, v_l^M]$. 
As commented above, states and inputs of the system are subject to constraints, that is
\[ e(k) \in E, \quad u(k) \in U, \] (17)
where \( E \subseteq \mathbb{R}^3 \) and \( U \subseteq \mathbb{R}^2 \) are polytopes and contain the origin.

Remark 4: Note that, equations (15)-(16) lead to bounds on the space of \( u \), obtained from the constraints on \( v_r, v_l, i_r, i_l, v_r^{ref}, v_l^{ref} \).

III. ADAPTIVE CONTROL USING LINEAR MATRIX INEQUALITIES

LMI are known as an efficient tool for solving convex optimization problems. LMI have an extensive application in the field of automatic control involving robust and optimal control [5], [17], [24], [14].

A linear matrix inequality is a matrix inequality of the form [5]
\[ F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0, \] (18)
where \( x \in \mathbb{R}^m \) is the decision variable and the symmetric matrices \( F_i = F_i^T \in \mathbb{R}^{n \times n}, i = 0, \ldots, m \), are given. The inequality \( F(x) > 0 \) means that \( F(x) \) is positive-definite. For a complete description of LMI see [5] and the references therein.

Now, we pose the trajectory tracking control problem in terms of an LMI optimization problem, in order to obtain a positive definite matrix \( P \) determining a Lyapunov function and a set of feedback gains which provide the control law. This problem is solved off-line considering all the extreme realizations of the parameter. Afterwards, an adaptive feedback control law is obtained on-line depending on the current realization of the parameter \( \gamma \), which is obtained as a convex combination of the extremal feedback gains.

The control scheme implemented in this paper is summarized in Figure 2. The adaptive controller uses an estimation of the slip and the reference to determine the feedback gain guaranteeing asymptotic stability.

The control strategy is designed to satisfy different specifications, in particular, it is required to provide:

- Input and state constraints fulfillment: this requirement is guaranteed through the determination of an invariant set [2], i.e. a set in the state space in which the system state can be confined by the control law. An ellipsoidal invariant set is computed ensuring constraints satisfaction and providing
Asymptotic stability: it is achieved by means of a quadratic Lyapunov function, i.e. a positive definite function decreasing along the trajectories of the closed-loop system.

Performance: the quadratic Lyapunov function provides an upper bound on the cost-to-go as close as possible to the optimal LQR cost. We recall that, with cost-to-go, we consider the sum of the stage cost, from the present to the infinite time. Our solution provides a cost-to-go function which is an overbound of the optimal LQR one. The objective of the optimization problem is to minimize such overbounding function, to make the cost-to-go as close as possible to the optimal one, i.e. the LQR cost.

Adaptivity: the control has to fulfill the specifications for any of the admissible realization of the parameter in the bounded set $\Gamma$. For that reason, a feedback gain is designed for any extremal realization of the linear system, such that the quadratic function is a common Lyapunov function. The set of controllers induces a time-varying feedback gain.

Performance region: we define a target set of the state space where the performance is considered. The objective is that the controller obtained solving LMI has higher performance inside this set, which is the region of the state space in which the system is confined in practice.

Fast real-time implementation: once the feedback gains have been calculated for each extremal parameter realization, it is sufficient to solve on-line a linear programming problem to obtain the stabilizing control law. This fact supposes that the presented control strategy fits very well to mobile robotics applications, where small sampling rates are employed.

Given a linear, time-varying, discrete-time system in the form (13) and subject to the constraints (17), we look for a matrix $P > 0$ and a set of parameter-dependent controllers $K(A^i_j)$, one for each vertex $\gamma^i$.
of $\Gamma$, such that for all $e \in \mathcal{E}(P)$ the system is asymptotically stable, and the input and state constraints are fulfilled.

Note that, in order to assure the stated specifications, it is required that the state evolves inside an invariant set $\mathcal{E}(P)$, i.e., $e \in \mathcal{E}(P) = \{ e \in \mathbb{R}^3 : e^T \mathcal{P} e \leq 1 \}$. In our case, invariance is assured by the fact that the ellipsoid is a level set of a Lyapunov function. This region has the property that if the initial state belongs to it, all the following states are contained in that set for any possible realization of the parameter $[2]$.

**Remark 5**: In the following, all the conditions required are imposed only at the extremal values of the polytopic set $\Gamma$, i.e. at the $N_\gamma$ vertices of $\Gamma$. In our case, $N_\gamma = 2^4$, since as shown in Assumption 1, matrix $A_\gamma$ is determined by the admissible realization of four variables $(v_r^r, v_l^r, i_r, i_l)$. Fulfillment of such conditions at the vertices yields the satisfaction at any point in $\Gamma$, as stated in Property 1, that will be presented in Section III-E.

In the following section, we detail the design of the control strategy used in this work to assure previous specifications, regarding to adaptivity and constraints fulfillment.

### A. Stability and performance

When dealing with the problem of determining asymptotically stable controllers, one classical way to proceed is to look for a Lyapunov function determined by a definite positive matrix $P > 0$, i.e. $V(e) = e^T \mathcal{P} e$, such that $V(e(k+1)) - V(e(k)) < 0$, for all $e \neq 0$ [5], [17]. In general, invariant ellipsoid and Lyapunov function are both determined by $P$. We decouple this fact introducing a scaling factor in the definition of the Lyapunov function $V(e) = e^T \mu \mathcal{P} e, \mu \in \mathbb{R}^+.$

On one hand, $\mu$ represents a minimization parameter of the convex optimization problem. That means, small values of $\mu$ lead to low bounds of the cost-to-go, which implies to high performances over a narrow region.

On the other hand, matrix $P$ determines the “shape” of the Lyapunov function ensuring state and input constraints fulfillment.

Finally, in order to avoid that the ellipsoidal invariant set determined by $V(e)$ is too tight, we have included an inner geometrical constraint in the solution of the minimization problem (see Section III-D).

As explained previously, the LMI to be solved is formulated as

$$e^T ((A_\gamma^i + B_dK(A_\gamma^i))^T \mu P(A_\gamma^i + B_dK(A_\gamma^i))) e - e^T (\mu P)e \quad (19)$$
for every vertex $\gamma^j$ of $\Gamma$, with $j = 1, \ldots, N_\gamma$, where $Q > 0$, $R > 0$ are symmetric matrices weighting the state and input signals. Notice that inequality (19) gives a guaranteed cost function, for details see [17].

Denoting $\bar{A}^j_\gamma = A^j_\gamma + B_dK(A^j_\gamma)$, we get

$$e^T((\bar{A}^j_\gamma)^T \mu P(\bar{A}^j_\gamma))e - e^T(\mu P)e \leq -e^T(Q + K(A^j_\gamma)^T R K(A^j_\gamma))e, \quad \forall e \in \mathbb{R}^3,$$

for all $\gamma^j$, with $j = 1, \ldots, N_\gamma$. The previous inequality is equivalent to the following LMI

$$(\bar{A}^j_\gamma)^T \mu P(\bar{A}^j_\gamma) - \mu P \leq -Q - K(A^j_\gamma)^T R K(A^j_\gamma),$$

for all $\gamma^j$, with $j = 1, \ldots, N_\gamma$. Using the Schur complement [5], it results that the previous inequality is equivalent to

$$\begin{bmatrix}
    P - \frac{Q}{\mu} - K(A^j_\gamma)^T R K(A^j_\gamma) & (\bar{A}^j_\gamma)^T \\
    \bar{A}^j_\gamma & P^{-1}
\end{bmatrix} \begin{bmatrix}
    Q^\frac{1}{2} & K(A^j_\gamma)^T R^\frac{1}{2} \\
    0 & \mu I
\end{bmatrix} \begin{bmatrix}
    \bar{A}^j_\gamma^T \\
    P^{-1}
\end{bmatrix} \geq 0,$$

for all $\gamma^j$. Rearranging the previous LMI, we obtain

$$\begin{bmatrix}
    P & (\bar{A}^j_\gamma)^T & Q^\frac{1}{2} & K(A^j_\gamma)^T R^\frac{1}{2} \\
    \bar{A}^j_\gamma & P^{-1} & 0 & 0 \\
    Q^\frac{1}{2} & 0 & \mu I & 0 \\
    R^\frac{1}{2} K(A^j_\gamma) & 0 & 0 & \mu I
\end{bmatrix} \begin{bmatrix}
    P^{-1} & 0 & 0 & 0 \\
    0 & I & 0 & 0 \\
    0 & 0 & I & 0 \\
    0 & 0 & 0 & I
\end{bmatrix} \geq 0,$$

for all $\gamma^j$. In order to remove the non-linear terms on $P$, the previous matrix inequality is pre- and post-multiplied by $S = P^{-1}, Y^j_\gamma = K(A^j_\gamma) P^{-1}$, and $\bar{A}^j_\gamma = A^j_\gamma + B_dK(A^j_\gamma)$, the linear matrix inequality to
solve is given by

$$
\begin{bmatrix}
S & S(A_j^g)^T + (Y_j^g)^TB_d^T & SQ_j^g & (Y_j^g)^TR_j^g \\
(A_j^g)S + B_d(Y_j^g) & S & 0 & 0 \\
Q_j^gS & 0 & \mu I & 0 \\
R_j^gY_j^g & 0 & 0 & \mu I \\
\end{bmatrix} \geq 0,
$$

(25)

for all $\gamma^j$. This LMI is imposed for each vertex of the set $\Gamma$, and the solution of the optimization problem determine a feedback gain for each vertex. For that reason, we determine off-line $N_\gamma = 2^4$ control gains.

B. Input Constraints

Physical limitations in the actuators of the mobile robot impose constraints on the input variables. We have to impose, in LMI form, that no point of the invariant ellipsoid causes input constraints violations. Knowing that $v_r \leq v_r^M$ and $v_l \leq v_l^M$, we find that such restriction in terms of the virtual control inputs is

$$
\left(\frac{v_r^{\text{ref}} - u_1 - \frac{b}{2}u_2}{1 - i_r}\right) \leq v_r^M,
$$

(26)

we formulate the LMI for the case of $v_r$, the case for $v_l$ is obtained in a similar way.

Now, rearranging equation (26), we get

$$
C_j^g u + d_j^g \leq v_r^M,
$$

(27)

for all $\gamma^j$, where $C_j^g = \left[ \begin{array}{cc} -\frac{1}{1-i_r} & -\frac{b}{2(1-i_r)} \end{array} \right]$, $d_j^g = \frac{v_r^{\text{ref}}}{1-i_r}$, and clearly, $i_r$ and $v_r^{\text{ref}}$ are those related to the particular extremal realization $\gamma^j$ of the parameter.

Defining $\eta_j^g = v_r^M - d_j^g$ and substituting $u = K(A_j^g)e$, previous inequality becomes

$$
C_j^gK(A_j^g)e \leq \eta_j^g, \quad \forall e \in \mathcal{E}(P),
$$

(28)

for all $\gamma^j$. Considering the following problem to determine the maximum of a linear function with ellipsoidal constraints, i.e.

$$
a^* = \max_{e} C_j^gK(A_j^g)e \quad \text{s.t.} \quad e^TPe \leq 1,
$$

(29)

Notice that, in practice, the wheels can also move backward. However, these negative velocities are rarely reached, since the reference virtual robot always moves forward and for that reason no lower bounds have to be added to the optimization problem.
the solution to the previous maximization problem is (see [5])

\[ a^* = \sqrt{C_j^i K(A_j^i) P^{-1}(K(A_j^i))^T (C_j^i)^T}. \] (30)

Hence, a necessary and sufficient condition for (28) to be fulfilled is that

\[ C_j^i K(A_j^i) P^{-1}(K(A_j^i))^T (C_j^i)^T \leq (\eta_j^i)^2, \] (31)

for all \( \gamma_j^i \). Notice that a quadratic term \((\eta_j^i)^2\), depending on \( \gamma_j^i \) appears. In order to assure the convexity properties of LMI, it should be substituted for the upper bound \( \bar{\eta} = \bar{v}_r M r_{ref,M} \frac{1}{1-r_{ref}^M} \). Thus, it produces

\[ C_j^i K(A_j^i) P^{-1}(K(A_j^i))^T (C_j^i)^T \leq (\bar{\eta})^2, \] (32)

for all \( \gamma_j^i \). Then, applying the Schur complement, it becomes

\[
\begin{bmatrix}
(\bar{\eta})^2 & C_j^i K(A_j^i) \\
(K(A_j^i))^T (C_j^i)^T & P
\end{bmatrix} \succeq 0,
\] (33)

for all \( \gamma_j^i \). Finally, previous equation is pre- and post-multiplied by

\[
\begin{bmatrix}
I & 0 \\
0 & P^{-1}
\end{bmatrix},
\] (34)

and substituting \( S = P^{-1} \) and \( Y_j^i = K(A_j^i) P^{-1} \), the input constraints result in the following LMI form

\[
\begin{bmatrix}
(\bar{\eta})^2 & C_j^i Y_j^i \\
(Y_j^i)^T (C_j^i)^T & S
\end{bmatrix} \succeq 0,
\] (35)

for every vertex \( \gamma_j^i \) of \( \Gamma \), with \( j = 1, \ldots, N_\gamma \).

C. State Constraints

In addition, we have to impose that the invariant set is contained in the admissible state space region. This guarantees no state constraints violations, provided that the initial state is confined in \( \varepsilon(P) \). Using (13), state constraints can be formulated as

\[ |e| \leq T, \quad \forall e \in \varepsilon(P), \] (36)

where \( | \cdot | \) denoted the element-wise absolute value and \( T = [e_x^M e_y^M e_\theta^M]^T \).
Similarly to the case of input constraints, and substituting \( S = P^{-1} \), it produces

\[
\begin{align*}
    h^T_1 S h_1 & \leq (e^M_1)^2, \\
    h^T_2 S h_2 & \leq (e^M_2)^2, \\
    h^T_3 S h_3 & \leq (e^M_3)^2,
\end{align*}
\]

(37)

where \( h_1 = [1 \ 0 \ 0]^T \), \( h_2 = [0 \ 1 \ 0]^T \), and \( h_3 = [0 \ 0 \ 1]^T \). Finally, due to symmetry of LMI, this equation is also achieved for the lower bounds.

D. Performance region

We establish a target region, denoted as \( \Psi \), which can be considered as a performance region, in which the system evolves mostly. The objective is that the controller obtained solving LMI has higher performance inside this set. We define \( \psi^j \in \Psi \) as the \( j \)-th vertex of \( \Psi \). The set \( \Psi \) is the parallelotope in the state space determined by the intervals of interest, i.e. \( \Psi = \{ e : \psi^m \leq e \leq \psi^M \} \), and we impose \( \Psi \subseteq \varepsilon(P) \).

In LMI form, it must be imposed that all the vertices of \( \Psi \) belong to the invariant ellipsoid, i.e.

\[
1 - (\psi^j)^T P(\psi^j) \geq 0,
\]

(38)

which is equivalent to

\[
\begin{bmatrix}
1 & (\psi^j)^T \\
\psi^j & S
\end{bmatrix} \geq 0.
\]

(39)

In conclusion, the off-line design process is aimed to obtain a family of control gains fulfilling the LMI constraints and such that the resulting Lyapunov function is induced minimizing with respect to \( \mu \), that is

\[
\min_{S>0, \ \mu, \ Y_j \forall \gamma^j} \mu \quad \text{s.t.}
\]

\[
(25), (35), (37), \quad \forall \gamma^j,
\]

\[
(39), \quad \forall \psi^j.
\]

Figure 3 plots the invariant ellipsoid obtained solving (40) using the LMI toolbox [11] and MPT toolbox [18] both for Matlab ® Suite. The small box inside the ellipsoid depicts the set \( \Psi \). As expected, the ellipsoid is constrained by the performance region. Furthermore, input and state constraints are also
drawn, the values of these constraints are also employed in the simulations, see Table I. Recall that, once the feedback law is defined, the input constraints are projected into the state space (see (28)). This can be noticed in the top right cuts of the outer box in Figure 3.

![Fig. 3. Ellipsoidal invariant set, input and state constraints, and performance region](image)

E. On-line adaptive control strategy

Finally, in order to assure the performance and stability of the gain determined on-line (adaptive controller), given an $A_\gamma \in \mathcal{A}$, we have to compute a vector of coefficients $\lambda \in \mathbb{R}^{N_\gamma}$ such that $A_\gamma$ is a convex combination of the extreme matrices of the set $\mathcal{A}$. For that purpose, we can express

$$A_\gamma = \sum_{j=1}^{N_\gamma} \lambda_j A_{\gamma}^j, \quad \sum_{j=1}^{N_\gamma} \lambda_j = 1, \quad \lambda_j \geq 0, \quad \forall j = 1, \ldots, N_\gamma,$$

(41)

where $A_\gamma$ is the current matrix (on-line).

The problem to determine $\lambda$ is a Linear Programming feasibility problem (LP) with respect to $\lambda$, that is, we only need to find a feasible solution to equation (41).

Then, the proposed adaptive feedback gain for the current realization state is calculated as

$$K(A_\gamma) = \lambda_1 K(A_{\gamma}^1) + \ldots + \lambda_{N_\gamma} K(A_{\gamma}^{N_\gamma}).$$

(42)

Finally, the resulting feedback control law is

$$u(k) = K(A_\gamma)e(k).$$

(43)
In the following property, we establish that adaptive control law (43) assures the specifications given in Section III for any $A_{I} \in \mathcal{A}$.

**Property 1:** Suppose that Assumption 1 holds. Consider the linear, time-varying, discrete-time system (13) with constraints $e(k) \in E$, $u(k) \in U$. The adaptive control law defined in (43), is such that, for each $\gamma \in \Gamma$:

- The function $V(e) = e^T \mu Pe$ is a local Lyapunov function for the system inside $\epsilon(P) = \{e : e^T Pe \leq 1\}$ ensuring stability.
- The set $\epsilon(P) = \{e : e^T Pe \leq 1\}$ is an invariant set for the closed-loop system satisfying input and state constraints.
- The function $V(e)$ is an upper bound of the cost-to-go, and of the cost of the LQR, i.e.,

$$V(e(0)) \geq \min_{u_{[0,\infty)}} \sum_{k=0}^{\infty} e^T(k) Qe(k) + u(k)^T Ru(k), \quad (44)$$

where $u_{[0,\infty)}$ denotes the infinite sequence of $u(k)$ for $k \in \mathbb{Z}^+$ and $\forall e \in \epsilon(P)$.

The proof can be found in the Appendix A.

**IV. SIMULATIONS**

This subsection analyzes the performance of the proposed adaptive control law, and it provides a comparison with existing time-varying control techniques. For this purpose, the well-known linear, time-varying controller described in [8] has been implemented. Furthermore, in order to compare our new formulation with a controller that compensates slip effect, the control law presented in [13] have been also implemented.

Although, many different trajectories have been tested, in this case, we show a reference trajectory which is not too typical in mobile robotics, but it has been included in order to check the full velocity and slip ranges. In order to make more realistic the simulations, we have added a small random noise to the measurements of the robot position and to the slips. The initial location of the mobile robot is also different from the desired one. The parameters of the controllers developed in [8], [13] are set to $\beta = 1$ and $\delta = 0.6$ in order to reach a soft overdamped closed-loop behavior, see [8], [13] for more details. The rest of parameters are: $T_m = 0.1[s]$, $b = 0.5[m]$ and the parameters for the proposed adaptive control law are $Q = diag([1, 1, 0.1])$ and $R = 10I_2$. Reference velocities are restricted to $\{v^{ref}_r, v^{ref}_l \in [0.4, 1.5][m/s]\}$ and slip is restricted to $\{i_r, i_l \in [10, 30][\%]\}$. The state and input constraints
are summarized in Table I.

Figure 4a shows the trajectories. It is possible to observe that the control laws considering the slip effect have a better behavior than the controller proposed in [8]. In Figure 4b, we notice that the simulated slip varies over the whole range previously defined. The errors between the reference trajectory and those steered by the compared controllers are plotted in Figure 5. As expected, the controller presented in this work achieves the smallest error due to the adaptivity of the control law for each reference inside the specifications. It is possible to note that although some noise has been added to the slip and the robot position, the adaptive control law obtains the smallest error mainly in the lateral direction. Large lateral errors could cause crashes with obstacles in its workspace. In the proposed controller, it is assured that lateral errors are always smaller than in the other cases.

Fig. 4. Simulated trajectories and Slip

Fig. 5. Errors along the simulations

As explained above, longitudinal slip decreases the linear velocity, what means that controllers must increase this component of the velocity to compensate this negative effect. For that reason, the velocities displayed in Figure 6a are greater than the references. Finally, Figure 6b shows the virtual control signals
for the three controllers. The adaptive controller presents control signals which are much smoother than the other control laws.

![Graphs showing control and virtual signals along simulations](image)

Fig. 6. Control and virtual signals along the simulations

V. CONCLUSIONS

This paper presents the synthesis of an adaptive control law guaranteeing asymptotic stability for mobile robots under slip conditions subject to both constraints and varying dynamics. LMI are used to solve this convex optimization problem. This problem is solved off-line for each extreme system realization. On-line computation is devoted to determine an adaptive feedback control law for the current realization of the system as a convex combination of the extremal gains obtained off-line. Finally, a comparative study with other control laws have been addressed through simulations. These simulations show the appropriate behavior of the proposed formulation, state and input constraints are assured, and longitudinal slip is compensated. In future, we are planning to test this control strategy in a real mobile robot.

VI. ACKNOWLEDGMENTS

This work has been supported by the Spanish CICYT under grants DPI 2007-66718-C04-01 and DPI 2007-66718-C04-04.

REFERENCES


APPENDIX

Proof of Property 1

In the following, we assume that \( \lambda \) are obtained such that \( A_\gamma \) is a convex combination of \( A_{\gamma i} \), with \( \gamma_i \) the \( N_\gamma \) vertices of \( \Gamma \) as in (19) and \( K(A_\gamma) \) is determined as in (42). The three statements of the property are proved. We start proving the third point, since the first one is a direct consequence of it.

To prove the third point we have to show that inequality,

\[
e^T ((A_\gamma + B_dK(A_\gamma))^T \mu P(A_\gamma + B_dK(A_\gamma))) e - e^T (\mu P)e \
\leq -e^T (Q + K(A_\gamma)^T R K(A_\gamma)) e, \quad \forall e \in \mathbb{R}^3,
\]

is satisfied for any \( \gamma \in \Gamma \) if it is satisfied at the vertices \( \gamma_i \), with \( j = 1, \ldots, N_\gamma \). To simplify the notation, we define

\[
M_\gamma = \begin{bmatrix}
S & S(A_\gamma)^T + (Y_\gamma)^T B_d^T & SQ^{1/2} & (Y_\gamma)^TR^{1/2} \\
(A_\gamma)S + B_d(Y_\gamma) & S & 0 & 0 \\
Q^{1/2}S & 0 & \mu I & 0 \\
R^{1/2}Y_\gamma & 0 & 0 & \mu I
\end{bmatrix},
\]

where the dependence on the parameter \( \gamma \) is explicit, and note that \( M_\gamma = \sum_{i=1}^{N_\gamma} \lambda^j M_{\gamma i} \). Since, as illustrated in subsection III-A, condition (45) is equivalent to \( M_\gamma \geq 0 \), and from (25) and the non-negativeness of \( \lambda \), we have that

\[
M_\gamma = \sum_{i=1}^{N_\gamma} \lambda^j M_{\gamma i} \geq 0,
\]
it means that (45) is fulfilled. To prove that $V(e)$ is a Lyapunov function for the closed-loop system regardless on the realization of parameter $\gamma$, we have to show that for any $\gamma \in \Gamma$ condition $V(e(k + 1)) - V(e(k)) < 0$ is satisfied, i.e.,

$$
e^T ((A_{\gamma} + B_dK(A_{\gamma})))^T \mu P(A_{\gamma} + B_dK(A_{\gamma})))e - e^T (\mu P)e < 0, \quad \forall e \in \mathbb{R}^3, \tag{48}$$

since $V(e)$ is positive definite. From $Q > 0$, $R > 0$ and (45), the condition follows. Furthermore, since we prove it for the entire space, $V(e)$ is, in particular, a Lyapunov function in the ellipsoid centered in the origin. Finally, the set $\mathcal{E}(P)$ is an invariant set since it is the level set of a Lyapunov function (see [3]), and it fulfills the state constraints by construction. For what concern the input constraints, we have that

$$
C_{\gamma}K(A_{\gamma})e = \sum_{j=1}^{N_{\gamma}} \lambda^j C_{\gamma}K(A_{\gamma})e \leq \sum_{i=1}^{N_{\gamma}} \lambda^j \bar{\eta} = \bar{\eta} \quad \forall e \in \mathcal{E}(P), \tag{49}
$$

is satisfied by convexity of $\Gamma$ and fulfillment of input constraints at its vertices.
<table>
<thead>
<tr>
<th>Input</th>
<th>Max</th>
<th>Min</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_r$</td>
<td>2.5</td>
<td>0</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$v_l$</td>
<td>2.5</td>
<td>0</td>
<td>[m/s]</td>
</tr>
<tr>
<td>State</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_x$</td>
<td>1</td>
<td>-1</td>
<td>[m]</td>
</tr>
<tr>
<td>$e_y$</td>
<td>1</td>
<td>-1</td>
<td>[m]</td>
</tr>
<tr>
<td>$e_{\theta}$</td>
<td>0.52</td>
<td>-0.52</td>
<td>[rad]</td>
</tr>
<tr>
<td>Performance region</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\psi_{e_x}$</td>
<td>0.3</td>
<td>-0.3</td>
<td>[m]</td>
</tr>
<tr>
<td>$\psi_{e_y}$</td>
<td>0.3</td>
<td>-0.3</td>
<td>[m]</td>
</tr>
<tr>
<td>$\psi_{e_{\theta}}$</td>
<td>0.17</td>
<td>-0.17</td>
<td>[rad]</td>
</tr>
</tbody>
</table>

**TABLE I**

Input and state constraints, and performance region for simulations.
FIGURE LEGENDS

Figure 1. Graphical representation of the trajectory tracking problem

Figure 2. Control scheme for the adaptive control strategy

Figure 3. Ellipsoidal invariant set, input and state constraints, and performance region

Figure 4. Simulated trajectories and Slip

Figure 5. Errors along the simulations

Figure 6. Control and virtual signals along the simulations