WMR Control With UKF-Based Wheel Slippage Estimation

Francomano Bonifacio Marco, Gallo Luigi, Tancredi Alessia
Sapienza, University of Rome

SAPIENZA
UNIVERSITÀ DI ROMA

Contents

1 Introduction

Tracked mobile robots are extensively studied due to their ability to operate in unstructured environments and their large ground contact patch provides stability and traction across various terrains, making them valuable in agriculture, forestry, mining, military, search and rescue, and space exploration. Essential to their functionality is accurate motion tracking and control, enabling them to perform tasks efficiently, adapt to changing conditions, and optimize resource usage. However, implementing an efficient solution to the robot navigation problem remains a significant challenge, particularly in trajectory tracking, which involves designing control inputs to stabilize the robot along a reference trajectory. These challenges are related to factors such as indirect measurement methods, environmental variability, and dynamic conditions, necessitating the development of robust estimation algorithms. Trajectory tracking is especially challenging due to the nonholonomic constraints typical of tracked robots. Moreover, controlling tracked robots in unstructured environments is difficult due to slip phenomena, which must be considered during control design. When navigating different terrain, not considering wheel slippage in our model could greatly decrease the controller's ability to follow a defined trajectory, leading to inaccuracies in wheel velocities and forces, and consequently affecting the robot's overall performance. However, direct measurement of slippage is not feasible, and therefore methods must be introduced to estimate this quantity. Inspired by the work presented in [8], this report explores the application of a UKF (Unscented Kalman Filter) to estimate parameters that are not directly measurable.

1.1 State of the art

Significant works have been published related to the challenges posed by slip phenomena in the navigation of mobile robots. Researchers have explored various methodologies and techniques aimed at understanding, modeling, and compensating for slip effects to improve the autonomy and performance of mobile robotic systems. Wang and Low [16] presented a comprehensive overview of the modeling of wheeled mobile robots in the presence of wheel skid and slip, emphasizing the importance of control design in mitigating slip effects. Sidek and Sarkar [13] contributed to the field by proposing a theoretical framework to incorporate slip dynamics into the overall system dynamics of wheeled mobile robots, laying the foundation for more accurate modeling and control strategies. Ward and Iagnemma [16] proposed a model-based approach for estimating longitudinal wheel slip and detecting immobilized conditions of autonomous mobile robots operating in outdoor terrains, addressing practical challenges in slip detection and compensation. Song et al. [14] introduced a nonlinear sliding mode observer for estimating tracked-vehicle slip parameters based on kinematic equations and sensor measurements, contributing to the development of robust slip estimation techniques. Gonzales et al. [2] synthesized a control law for wheeled mobile robots under slip conditions using an LMI-based approach,

demonstrating the efficacy of control strategies in managing slip-induced disturbances. Michalek et al. [11] proposed a nonlinear feed-forward loop to compensate for skid-slip effects in mobile robots, highlighting the importance of proactive control strategies in mitigating slip-related challenges. Building upon previous works, Iossaqui et al. [5] introduced an adaptive tracking control strategy based on the robot's kinematic model to compensate for longitudinal slip, paving the way for adaptive control approaches in slip compensation. Subsequent works by Iossaqui et al. [4] and [7] extended these approaches to account for varying slip conditions, demonstrating the adaptability of control strategies in diverse operating environments. Furthermore, Iossaqui et al. [6] demonstrated that kinematic and dynamic controllers could maintain satisfactory performance even when state estimation is performed using advanced techniques such as the unscented Kalman filter and the extended Kalman filter, underscoring the robustness of control strategies in the face of uncertainties. Additional contributions to the field are found in the works of Iagnemma and Ward [3] and Tarakameh et al. [15], increasing our understanding of slip phenomena and advancing the state of the art in slip estimation and control in mobile robotics.

1.2 Problem addressed

The problem addressed in our project regards the effective control of mobile robots in the presence of slip phenomena. The efficient control and navigation of autonomous robots in the context of this project involves four main objectives:

1. **Localization**: in the context of mobile robotics, refers to the process of determining the precise position and orientation of a robot within its environment.

2. **Trajectory tracking**: is the ability of a robot to follow a predefined path or trajectory accurately. This predefined path could be a straight line, a curved path, or a sequence of waypoints that the robot needs to traverse. Precise trajectory tracking ensures that the robot can reach the destinations and achieve tasks effectively.

3. **Slip parameters estimation**: the main challenge lies in accurately estimating the longitudinal slip parameters for each wheel of the robot. This is needed for compensating the deviations between commanded and actual motion caused by slip, which can significantly affect the robot's ability to navigate and follow desired trajectories. To address this challenge, the advanced estimation technique of Unscented Kalman Filter (UKF) is employed to estimate slip parameters from the measured states (pose and velocity) of the robot.

4. **Integration with control strategy**: the effective slip estimation technique must be integrated with control strategies to enable robots to adapt dynamically to changes in slip conditions and maintain stable and reliable motion.

2 Theoretical Background

In this section we delve into the studies and theory behind the proposed solution to the problem of wheel slip in wheeled robots. We start by explaining the kinematic and dynamic models, showing how robots maneuver and respond to various inputs. Moreover, we delve into fundamental control theory methodologies by centering our discussion on:

- **Backstepping Technique**, a method known for its efficacy in nonlinear control, designed for a special class of systems. These systems are built from subsystems that radiate out from an irreducible subsystem that can be stabilized using some other method. Because of this recursive structure, the designer can start the design process at the known-stable system and "back out" new controllers that progressively stabilize each outer subsystem.
- **Trajectory Tracking** with State Error Feedback and **Approximate Linearization**. For handling the kinematic aspect, we employ trajectory tracking techniques tailored for unicycle models. These methods use state error feedback to adjust control inputs dynamically and incorporate approximate linearization to simplify control design and enhance trajectory tracking accuracy.

Additionally, we examine trajectory tracking specifically tailored for unicycle models[1], a prevalent mobile robot variant, to illustrate how robots can adeptly pursue predefined paths with precision. Finally, we introduce the Unscented Kalman Filter (UKF), a sophisticated method to predict the future state of a robot, particularly adapted to handling nonlinear scenarios. The incorporation of the UKF offers robust predictions to handle both dynamic and uncertain environments and the presence of high nonlinearities in the model of the robot.

2.1 Kinematic and Dynamic model

In this section, we introduce the kinematic and dynamic model of our robot when the presence of wheel slip is considered. The reference model is the differential drive robot, depicted in Figure 1, a mobile robot whose movement is based on two separately driven wheels placed on either side of the robot body. Its motion is described by its position (x, y) and its orientation θ in an inertial coordinate frame $F_1(x_w, y_w)$. The robot's position is defined by the coordinates of its geometric center O_m, which is also the origin of the local coordinate frame $F_2(x_m, y_m)$. The robot's orientation is given by the rotation of frame F_2 relative to frame F_1. The distance between the two tracks is b, and the equivalent radius of each track is r. Additionally, the robot's motion consists of translational velocity v and rotational velocity $\omega = \partial\theta/\partial t$.

[1] We opt for a unicycle model given our objective to tailor the controller to the platform of the TIAGo robot by PAL Robotics.

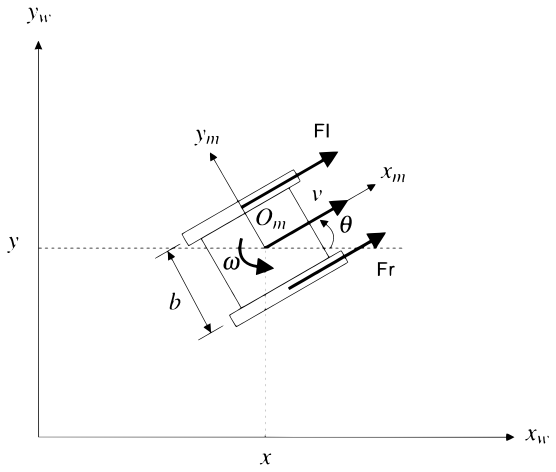


Figure 1: Geometric model of a tracked mobile robot.

Kinematic Model

As presented in [18], the kinematic equation of the tracked robot when we consider the presence of longitudinal slip is given by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \frac{1}{2b}\begin{bmatrix} br(1-i_L)\cos\theta & br(1-i_R)\cos\theta \\ br(1-i_L)\sin\theta & br(1-i_R)\sin\theta \\ -2r(1-i_L) & 2r(1-i_R)\end{bmatrix}\begin{bmatrix}\omega_L\\ \omega_R\end{bmatrix} \quad \Leftrightarrow \quad \dot{q} = S(q)\xi \qquad (1)$$

where $q = (x, y, \theta)^T$ is the robot configuration, $\xi = (\omega_L, \omega_R)^T$ is the vector composed of the angular velocities of the left and right tracks, i_L and i_R denote the longitudinal slip ratio of the left and right wheels, respectively, which are assumed unknown. As discussed in [17], the unknown slip parameters can be defined as:

$$i_L = \frac{r\omega_L - v_L}{r\omega_L}, \quad i_R = \frac{r\omega_R - v_R}{r\omega_R}, \quad 0 \leq i_L, i_R < 1$$

where v_L and v_R are respectively the linear velocities of the left and right wheels with respect to the terrain.

Dynamic Model

The dynamic model of the tracked robot, as presented in [5], is given by

$$\dot{q} = S(q)\xi$$
$$\bar{M}\dot{\xi} = \bar{B}(q)\tau$$

where $q = (x, y, \theta)^T$, $S(q)$ and ξ are defined in (1), $\tau = (\tau_L, \tau_R)^T$ is the input vector representing generalized forces on the left and right wheel, and $\bar{M}$ and

$\bar{B}$ are, respectively

$$\bar{M} = S^T(q)MS(q), \quad \bar{B}(q) = S^T(q)B(q)$$

with the matrices M and $B(q)$ defined as

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, \quad B(q) = \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ -b/2 & b/2 \end{bmatrix}$$

where m and I of M are respectively the robots's total mass and rotational Inertia (I_{zz}) around the vertical axis through O_m.

2.2 Integrator Backstepping

We deal with the special case of integrator backstepping [9]. Consider the system:

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\xi \\ \dot{\xi} &= u \end{aligned}$$

where $[\eta, \xi] \in \mathbb{R}^n + 1$ is the state and $u \in \mathbb{R}$ is the control input. The functions $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^n$ are smooth in a domain $D \subset \mathbb{R}^n$ that contains $\eta = 0$ and $f(0) = 0$. We aim to design a state feedback control law to stabilize the origin ($\eta = 0, \xi = 0$). We assume that both f and g are known.

System Description

This system can be viewed as a cascade connection of two components:

1. The first component is $\dot{\eta} = f(\eta) + g(\eta)\xi$ with ξ as input.
2. The second component is the integrator $\dot{\xi} = u$.

Assumptions

Suppose the component $\dot{\eta} = f(\eta) + g(\eta)\xi$ can be stabilized by a smooth state feedback control law $\xi = \alpha(\eta)$ with $\alpha(0) = 0$; that is, the origin of

$$\dot{\eta} = f(\eta) + g(\eta)\alpha(\eta)$$

is asymptotically stable. Further, suppose we know a smooth, positive definite Lyapunov function $V(\eta)$ that satisfies the inequality

$$\frac{\partial V}{\partial \eta}\dot{\eta} = \frac{\partial V}{\partial \eta}\left[f(\eta) + g(\eta)\alpha(\eta)\right] \leq -W(\eta), \quad \forall \eta \in D$$

where $W(\eta)$ is positive definite.

Backstepping Process

By adding and subtracting $g(\eta)\alpha(\eta)$ on the right-hand side of $\dot{\eta} = f(\eta) + g(\eta)\xi$, we obtain the equivalent representation:

$$\dot{\eta} = [f(\eta) + g(\eta)\alpha(\eta)] + g(\eta)\left[\xi - \alpha(\eta)\right]$$

The change of variables $z = \xi - \alpha(\eta)$ results in the system:

$$\begin{aligned}\dot{\eta} &= [f(\eta) + g(\eta)\alpha(\eta)] + g(\eta)z \\ \dot{z} &= u - \dot{\alpha}(\eta)\end{aligned}$$

And here is where we "backstep" the subtraction of alpha through the integrator. So now we have in the first subsystem an asymptotically stable origin when the input is persistently zero (we assumed that $f(\eta) + g(\eta)\alpha(\eta)$ has asymptotically stable origin). Since f, g, and α are known, the derivative $\dot{\alpha}$ can be computed by using the expression

$$\dot{\alpha} = \frac{\partial \alpha}{\partial \eta}\dot{\eta} = \frac{\partial \alpha}{\partial \eta}\left[f(\eta) + g(\eta)\alpha(\eta)\right]$$

Taking $v = u - \dot{\alpha}$ reduces the system to the cascade connection:

$$\begin{aligned}\dot{\eta} &= [f(\eta) + g(\eta)\alpha(\eta)] + g(\eta)z \\ \dot{z} &= v\end{aligned}$$

2.3 Lyapunov Stability Analysis

Using the Lyapunov function candidate $V_e(\eta, z) = V(\eta) + \frac{1}{2}z^2$, we obtain:

$$\dot{V}_e = \frac{\partial V}{\partial \eta}\left[f(\eta) + g(\eta)\alpha(\eta)\right] + \frac{\partial V}{\partial \eta}g(\eta)z + zv \leq -W(\eta) + zv$$

Choosing

$$v = -\frac{\partial V}{\partial \eta}g(\eta) - kz, \quad k > 0$$

yields

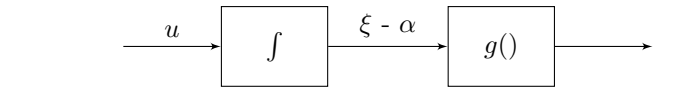
$$\dot{V}_e \leq -W(\eta) - kz^2$$

which shows that the origin $(\eta = 0, z = 0)$ is asymptotically stable. Since $\alpha(0) = 0$, we conclude that the origin $(\eta = 0, \xi = 0)$ is asymptotically stable. Substituting for v, z, and α, we obtain the state feedback control law:

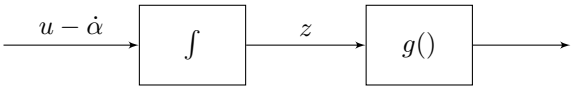
$$u = \left[\frac{\partial \alpha}{\partial \eta}\left(f(\eta) + g(\eta)\xi\right)\right] - \frac{\partial V}{\partial \eta}g(\eta) - k(\xi - \alpha(\eta))$$

Conclusion

If all the assumptions hold globally and $V(\eta)$ is radially unbounded, we can conclude that the origin is globally asymptotically stable. Below the block scheme of the second subsystem before and after the effect of the backstepping control technique:



Before Backstepping



After Backstepping

2.4 Control design

In this section, we introduce an UKF-based controller for the dynamic model. This controller takes in input two reference velocities v_r and ω_r, representing the reference linear and angular velocities, and produces force control τ, taking into account the dynamic of the system. The dynamic model is represented as an integrator backstepping form [1]. To design the control for the mobile robot, we start by defining the input force τ as

$$\tau = \bar{B}(q)^{-1}\bar{M}u$$

where u is an auxiliary control input. Our objective is to transform the system dynamics into a form suitable for control design. To achieve this, the backstepping technique can be applied, leading to the backstepping form (described in section 2.2)

$$\dot{q} = S(q)\xi, \quad \dot{\xi} = u \tag{2}$$

hence to design the control law u using the backstepping technique, a desired velocity input $\xi_d = (\omega_{Ld}, \omega_{Rd})^T$ must be specified. To obtain a control input u that guarantees the velocity applied to the system ξ to follows the desired one ξ_d, we need to define a velocity error $e_d = \xi - \xi_d$ to be converged at zero by the feedback controller

$$u = \dot{\xi}_d - \begin{bmatrix} k_4 & 0 \\ 0 & k_5 \end{bmatrix} (\xi - \xi_d)$$

with k_4 and k_5 positive. Next, to obtain the desired velocity ξ_d, we consider an auxiliary velocity $\eta = (v, \omega)^T$, related to ξ_d by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r}{2b} \begin{bmatrix} b\omega_{Ld}(1 - i_L) + \omega_{Rd}(1 - i_R) \\ -2\omega_{Ld}(1 - i_L) + 2\omega_{Rd}(1 - i_R) \end{bmatrix} = T \begin{bmatrix} \omega_{Ld} \\ \omega_{Rd} \end{bmatrix}$$

and, by inverting this relation, we obtain the desired velocity

$$\begin{bmatrix} \omega_{Ld} \\ \omega_{Rd} \end{bmatrix} = \frac{1}{2r} \begin{bmatrix} 2(1 - i_{Le})^{-1} & -b(1 - i_{Le})^{-1} \\ 2(1 - i_{Re})^{-1} & b(1 - i_{Re})^{-1} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

giving us a method to derive the desired velocity $\xi_d = (\omega_{Ld}, \omega_{Rd})^T$ from a given auxiliary velocity $\eta = (v, \omega)^T$. The auxiliary velocity η will be computed to solve a tracking problem for the kinematic model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \Longleftrightarrow \dot{q} = S_a(q)\eta \tag{4}$$

The tracking problem statement is the following:

- Find velocity $\eta = (v, \omega)^T$ such that:

$$\lim_{t \to \infty} (q_r - q) = 0$$

- The reference trajectory $q_r = (x_r, y_r, \theta_r)^T$ is generated by:

$$\dot{q}_r = S_a(q_r)\eta_r$$

with the reference velocity $\eta_r = (v_r, \omega_r)^T$ constant. We now have all the elements to address the kinematic tracking problem. First, rather than using directly the state error $q_d - q$, we can use $e = (e_1, e_2, e_3)^T$ defined in a frame rotated by θ as

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{5}$$

Now, neglecting the slip, it can be shown [10] that the velocity input

$$\begin{aligned} v &= v_r \cos e_3 - k_3 e_3 \omega + k_1 e_1 \\ \omega &= \omega_r + \frac{v_r}{2} \left[k_3(e_2 + k_3 e_3) + \frac{1}{k_2} \sin e_3 \right], \quad k_i > 0 \end{aligned}$$

can drive the error defined in (5) to zero. Now, if we don't know the slipping parameters i_L, i_R in (3), the desired velocity ξ_d cannot directly be computed, hence an estimation procedure is needed. For this purpose, we use a non-linear filtering algorithm (UKF) to estimate the slipping parameters, and replace the estimated i_{Le}, i_{Re} in (3) to obtain the desired reference velocity ξ_d.

2.5 Trajectory Tracking Controller for a Unicycle

In the context of mobile robots, trajectory tracking is the main task to achieve for ensuring that the robot follows a predefined path or trajectory accurately. For a unicycle model [12], trajectory tracking involves controlling the linear and angular velocities of the robot to minimize the Cartesian error e_d between the desired and actual positions and orientations. Instead of using the direct state error $(q_d - q)$, we utilize its rotated version defined in (5) as $e = (e_1, e_2, e_3)$,

expressed in a frame rotated by an angle θ around the z-axis. The dynamic of the error in (5) is then

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 + v_r \cos e_3 - v \\ -\omega e_1 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \tag{6}$$

and, in this representation, it's nonlinear and time-varying, as the reference inputs v_r and ω_r depend on time, indeed they vary along the trajectory ($v_r = v_r(t)$, $\omega_r = \omega_r(t)$). A simple idea to handle this complexity is that, to stabilize a nonlinear system at an equilibrium, we can simply stabilize the approximate linearization around it. That said, for a generic non-linear system of the form

$$\dot{x} = \varphi(x, u) = f(x) + G(x)u$$

assume the origin is the desired unforced equilibrium, so that $\varphi(x, u)$ in $x = 0$, $u = 0$ is $\varphi(0, 0) = 0$ (or, equivalently, $f(0) = 0$). The Taylor expansion of φ around $x = 0$, $u = 0$ is

$$\varphi(x, u) \approx \varphi(0, 0) + \left.\frac{\partial \varphi}{\partial x}\right|_{\substack{x=0 \\ u=0}} x + \left.\frac{\partial \varphi}{\partial u}\right|_{\substack{x=0 \\ u=0}} u$$

where higher-order terms have been neglected. The approximate linearization of the non-linear system at the origin is then defined as

$$\dot{x} = \left.\frac{\partial \varphi}{\partial x}\right|_{\substack{x=0 \\ u=0}} x + \left.\frac{\partial \varphi}{\partial u}\right|_{\substack{x=0 \\ u=0}} u = Ax + Bu$$

Now, let $u = Kx$ so that $\dot{x} = Ax + BKx = (A + BK)x$. If K is chosen in such a way that $A + BK$ is Hurwitz (eigenvalues in the left hand plane, certainly possible if (A, B) is controllable), then the approximate linearization is asymptotically stable. Moreover, by Lyapunov indirect method, this ensures the origin is locally asymptotically stable for the nonlinear system. By now applying the approximate linearization technique to stabilize the error dynamics in (6), we notice that $e = (0, 0, 0)$ is not an unforced equilibrium, in fact

$$\begin{aligned} \dot{e}_1 &= v_r \cos e_3 - v + e_2\omega = v_r - v \\ \dot{e}_2 &= v_r \sin e_3 - e_1\omega = 0 \\ \dot{e}_3 &= \omega_r - \omega \end{aligned}$$

however, this problem can be easily solved by using the following (invertible) input transformation:

$$\begin{aligned} u_1 &= v_d \cos e_3 - v \\ u_2 &= \omega_d - \omega \end{aligned} \tag{7}$$

hence, inserting u_1 and u_2 according to (7) in (6) we get

$$\begin{aligned} \dot{e}_1 &= \omega_d e_2 + u_1 - e_2 u_2 \\ \dot{e}_2 &= -\omega_d e_1 + v_d \sin e_3 + e_1 u_2 \\ \dot{e}_3 &= u_2 \end{aligned}$$

and expressing $\dot{e} = \varphi(e, u)$ as a decomposition of drift term and input term, we obtain:

$$\dot{e} = \varphi(e,u) = f(e) + G(e)u = \begin{pmatrix} \omega_d e_2 \\ -\omega_d e_1 + v_d \sin e_3 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

The linear approximation of the error dynamics is then:

$$\dot{e} = \left.\frac{\partial \varphi}{\partial e}\right|_{\substack{x=0 \\ u=0}} e + \left.\frac{\partial \varphi}{\partial u}\right|_{\substack{x=0 \\ u=0}} u = A(t)e + Bu$$

with

$$A(t) = \begin{pmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

where $A(t)$ is time-varying due to the dependence of the reference inputs v_d and ω_d on time. To stabilize the error dynamics, we define a linear feedback controller:

$$u = Ke = \begin{pmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & -k_3 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

The time-varying closed-loop error dynamics is then

$$\dot{e} = (A(t) - BK(t))e = \begin{pmatrix} -k_1 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & -k_2 & -k_3 \end{pmatrix} e$$

The characteristic polynomial of $A(t)$ can be made time-invariant and Hurwitz by choosing $K(t)$ appropriately. Specifically, letting $k_1 = k_3 = 2\zeta a$ and $k_2 = (a^2 - \omega_d^2)/v_d$, with $a > 0$ and $\zeta \in (0, 1)$, the system will have a pair of complex eigenvalues with negative real part and a real negative eigenvalue. It's important to note that this do not ensure asymptotic stability unless v_d and ω_d are constant (i.e., rectilinear and circular trajectories). Even in these cases, the stability of the unicycle is not globally guaranteed (indirect Lyapunov method). The actual velocity inputs v and ω are obtained by plugging the feedback controls u_1 and u_2 into the input transformation. As the error e approaches zero, the actual inputs converge to the desired inputs v_d and ω_d, demonstrating pure feedforward control. Global stability can be guaranteed by employing a nonlinear version of the controller with bounded, positive gains k_1 and k_3 and bounded derivatives.

$$\begin{aligned} u_1 &= -k_1(v_d, \omega_d)e_1 \\ u_2 &= -k_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3(v_d, \omega_d)e_3 \end{aligned}$$

2.6 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a variant of the Kalman Filter (KF) used for state estimation in systems with nonlinear dynamics. Unlike the Extended Kalman Filter (EKF), which linearizes the system dynamics and observation models using first-order Taylor expansion, the UKF employs a deterministic sampling technique called the Unscented Transformation to capture the statistical properties of the system's state and observation variables more accurately. The UKF is particularly well-suited for nonlinear systems where the assumptions of linearity underlying the KF and EKF may not hold. By using a set of carefully chosen sample points (sigma points) to approximate the probability distribution of the state variables, the UKF can handle nonlinearities more effectively. Moreover, since the EKF linearizes the system dynamics and observation models, this can introduce errors, especially when dealing with highly nonlinear systems. The UKF avoids the linearization errors associated with the EKF, resulting in more accurate state estimates.

Initialization

$$\hat{x}_0 = \bar{x}_0, \quad P_0 = \bar{P}_0$$

Prediction

The process model $f(x_k, u_k)$ is defined by the kinematic equations (obtained by first order Taylor integration) along with the process equation of the additional parameters that are part of the augmented state $(x_e, y_e, \theta_e, \omega_{Le}, \omega_{Re}, i_{Le}, i_{Re})$. The accelerations of the wheels are obtained by inverting the process of the control design described above (from the torques as input). In the end what we have, given the S in (1), and M and $B(q)$ in (2.1):

$$\overline{B}(q) = S^T(q)B(q), \quad \overline{M} = S^T(q)MS(q), \quad \dot{\xi} = \begin{pmatrix} \dot{\omega}_{Le} \\ \dot{\omega}_{Re} \end{pmatrix} = \overline{M}^{-1}\overline{B}\tau$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T \begin{bmatrix} \omega_{Ld} \\ \omega_{Rd} \end{bmatrix}, \quad T = \frac{r}{2b} \begin{bmatrix} b(1-i_L) & b(1-i_R) \\ -2(1-i_L) & 2(1-i_R) \end{bmatrix}$$

and

$$
\begin{aligned}
x_{k+1} &= x_k + \delta t v \cos\theta_k \\
y_{k+1} &= y_k + \delta t v \sin\theta_k \\
\theta_{k+1} &= \theta_k + \delta t \omega \\
\omega_{L_{k+1}} &= \omega_{L_k} + \delta t \dot{\omega_L} \\
\omega_{R_{k+1}} &= \omega_{R_k} + \delta t \dot{\omega_R} \\
i_{Le_{k+1}} &= i_{Le_k} + \nu_{i_{Le_k}} \\
i_{Re_{k+1}} &= i_{Re_k} + \nu_{i_{Re_k}}
\end{aligned}
$$

with $\nu_{i_{Le_k}}$ and $\nu_{i_{Re_k}}$ being the additive process noise. This set of equations represents the process model f which takes as input the previous state and the control input (torques) and outputs the new state. The Unscented Kalman Filter (UKF) algorithm utilizes $2n+1$ sigma points, where n is the dimension of the state vector, to capture the statistical properties of the system's state and observation variables more accurately. Generate and propagate sigma points through process model:

$$
\begin{aligned}
X^i_{k-1} &= \hat{x}_{k-1}, \quad i = 0 \\
X^i_{k-1} &= \hat{x}_{k-1} + \sqrt{(n+\lambda)P_{k-1}}, \quad i = 1, \ldots, n \\
X^i_{k-1} &= \hat{x}_{k-1} - \sqrt{(n+\lambda)P_{k-1}}, \quad i = n+1, \ldots, 2n
\end{aligned}
$$

Estimate predicted state and covariance:

$$
\begin{aligned}
X^i_{k|k-1} &= f(X^i_{k-1}, u_{k-1}), \quad i = 0, \ldots, 2n \\
\hat{x}_k &= \sum_{i=0}^{2n} W_i^{(m)} X^i_{k|k-1} \\
P_k &= \sum_{i=0}^{2n} W_i^{(c)} \left[X^i_{k|k-1} - \hat{x}_k \right] \left[X^i_{k|k-1} - \hat{x}_k \right]^T + Q
\end{aligned}
$$

Update (Correction)

Propagate sigma points through measurement model:

$$
Y^i_{k|k-1} = h(X^i_{k-1}), \quad i = 0, \ldots, 2n
$$

Estimate corrected state and covariance:

$$
\begin{aligned}
\hat{y}_k &= \sum_{i=0}^{2n} W_i^{(m)} Y^i_{k|k-1} \\
P_{\tilde{y}_k \tilde{y}_k} &= \sum_{i=0}^{2n} W_i^{(c)} \left(Y^i_{k|k-1} - \hat{y}_k \right) \left(Y^i_{k|k-1} - \hat{y}_k \right)^T + R
\end{aligned}
$$

Compute cross-covariance and Kalman gain:

$$P_{x_k y_k} = \sum_{i=0}^{2n} W_i^{(c)} \left(X_{k|k-1}^i - \hat{x}_k \right) \left(Y_{k|k-1}^i - \hat{y}_k \right)^T$$
$$K_k = P_{x_k y_k} P_{\tilde{y}_k \tilde{y}_k}^{-1}$$

Update state estimate and covariance:

$$\hat{x}_k = \hat{x}_k + K_k \left(y_k - \hat{y}_k \right)$$
$$P_k = P_k - K_k P_{\tilde{y}_k \tilde{y}_k} K_k^T$$

where $\hat{x}_k$is the updated (corrected) state estimate at time step k, P_k is the updated (corrected) state covariance matrix at time step k, Q is the process noise covariance matrix, R is the measurement noise covariance matrix, $f(\cdot)$ and $h(\cdot)$ are the process and measurement models respectively, u_k is the control input at time step k, y_k is the measured output at time step k, and $W_i^{(m)}$ and $W_i^{(c)}$ are the weights for computing the mean and covariance, respectively.

2.7 Bézier Curves

Bézier curves (depicted in Figure 2), extensively employed in robotics, serve as an effective method for describing and generating motion trajectories for robots. In robotics, Bézier curves are often utilized to represent smooth and precise motion trajectories for robots. They are particularly useful for defining complex trajectories that robots need to traverse smoothly. Bézier curves can be of different orders, but quadratic and cubic curves are the most commonly used in robotics. A quadratic Bézier curve is defined by three control points P_0, P_1, P_2 and is given by the parametric equation:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2 \quad 0 \leq t \leq 1$$

A cubic Bézier curve is defined by four control points P_0, P_1, P_2, P_3 and is given by the parametric equation:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 tP_1 + 3(1-t)t^2 P_2 + t^3 P_3 \quad 0 \leq t \leq 1$$

The shape of a Bézier curve is determined by the positions of its control points. The curve starts at the first control point, ends at the last control point, and is influenced by the intermediate control points. Bézier curves offer smoother interpolation due to their continuous and differentiable nature. By smoothly transitioning between control points, these curves minimize abrupt changes in direction or velocity. In robotics, this smoothness reduces stress on mechanical components and enhances overall motion stability and precision. In our robotics application, we employed Bézier curves to obtain smoother velocity profiles, aiming to eliminate any torque spikes during the robot's startup phase.

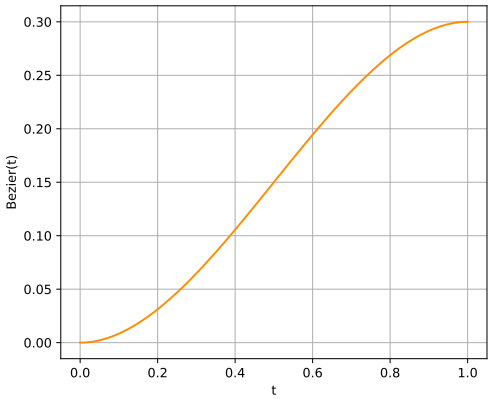


Figure 2: Bézier curve.

3 Problem Statement and Methodology

In this section, we describe the methodology in terms of practical implementations using simulation tools. We detail how we use the theoretical models described in section 2 to simulate realistic slip scenarios and adjust our control strategies to maintain accurate navigation.

3.1 Methodology

The primary objective of this work is to estimate the slipping parameters of the two wheels using the Unscented Kalman Filter (UKF). In this subsection, we detail the simulation methodology, which was implemented entirely in Python without the utilization of additional software such as ROS and Gazebo. We began by simulating the behavior of the real robot using a fourth-order Runge-Kutta method for numerical integration. The differential equations governing the motion of the robot were discretized and solved iteratively to predict the robot's state over time. The integration scheme is given by

$$
\begin{aligned}
s_1 &= f(M, \tau, x_0, i_L, i_R), \quad & x_1 &= x_0 + \frac{\delta t}{2} s_1 \\
s_2 &= f(M, \tau, x_1, i_L, i_R), \quad & x_2 &= x_0 + \frac{\delta t}{2} s_2 \\
s_3 &= f(M, \tau, x_2, i_L, i_R), \quad & x_3 &= x_0 + \delta t s_3 \\
s_4 &= f(M, \tau, x_3, i_L, i_R), \quad & x &= x_0 + \frac{\delta t}{6}(s_1 + 2s_2 + 2s_3 + s_4)
\end{aligned}
$$

where M represents the inertia matrix of the robot, x_0 is the initial state vector, δt is the time step, τ is the control input vector, i_L and i_R denote the input reference slipping parameters for the left and right wheels respectively. Here, f

is the process model defined before, and it takes as input the current state x_0 and control inputs τ, as well as the real slipping parameters. The inclusion of the real slipping parameters as input to the process model allows us to simulate the robot's motion with specific slipping characteristics manually set to mimic various real-world scenarios. This integration method enables us to incorporate the effects of wheel slipping into the simulation accurately. Additionally, we implemented the control strategy and the UKF for slip parameter estimation explained in the section 2, in a separate Python module. The control strategy is designed to regulate the robot's motion based on feedback from the manually set slipping parameters.

3.2 Integration with ROS and Gazebo

The controller and the Unscented Kalman Filter (UKF) were integrated into a Catkin package to work within a Gazebo environment controlled by ROS. This allowed seamless communication with Gazebo and other ROS nodes, enabling effective control of the simulation environment. The integration with ROS and Gazebo was undertaken to evaluate the effectiveness of the filter and controller within a more realistic simulation environment, such as the Tiago robot by PAL Robotics. The Tiago robot, akin to a unicycle differential drive with multiple payloads, offers a dynamic XML representation in Gazebo, providing all necessary parameters to compute the model matrices. This integration facilitated rigorous testing and validation of the filter and controller algorithms in a simulated environment closely resembling real-world conditions.

4 Simulations

This section discusses simulations performed with Python, both in a simpler environment using ODE, and integrating the controller with Gazebo and ROS for simulation on TIAGo. In the two environments, the effects of several key factors were analyzed, including the α parameters, and the covariance matrices Q and R of the UKF, and the k_i parameters of the controller for $i = 1, \cdots, 5$. Key specifications regarding the robot include wheel distance $b = 0.2022$ m, wheel radius $r = 0.0985$ m, mass of the robot $m = \sum_i m_i$ kg, and inertia $I(t)$, time-dependent, computed using `pinocchio`. The input for the UKF algorithm are the forces τ_L, τ_R and the measurements $(x_m, y_m, \theta_m, \omega_{L,m}, \omega_{R,m})^T$, which are corrupted by an additive noise with covariance matrix R. The initial state for the filter is chosen as $x_0 = (1, 1, \pi/4, 0, 0, 0, 0)^T + 0.01(1, 1, 1, 1, 1, 1, 1)^T\epsilon$, where ϵ is a random value drawn from a normal distribution with mean zero and standard deviation one. The initial state covariance is the identity matrix $P_0 = I_d$. During simulations, both a circular ($v_r = 0.3$ m/s, $\omega_r = 0.1$ rad/s) and a linear ($v_r = 0.3$ m/s) reference trajectory are used. In the simulations, the robot starts from a stationary position and gradually reaches cruising speeds through similar Bézier profiles (section 2.7) for both linear and angular velocities. The reason behind this choice is observable in Figure 5. Starting from a standstill

and having to apply an instantaneous velocity creates a significant discontinuity that results in an initial spike in torques. However, as shown in Figure 4, by incorporating a Bézier profile, this initial spike is eliminated. Generally, the presence of spikes is due to the sudden change of a factor in the system, as can be observed in Figure 4, by analyzing the trend of slipping parameters. The profiles defined in (8) and (9) exhibit discontinuities at the instants $t = 10$ for i_R and $t = 20$ for i_L, and these discontinuities cause spikes in the torques, which is not the case with the linear trend descending from 0.3 to 0 for i_R and from 0.2 to 0 for i_L.

4.1 ODE Tests

This section examines simulations conducted without the aid of Gazebo and ROS, but using a simple simulator written in Python using ODE (Ordinary Differential Equation). In the following tests, the parameters of the controller were found empirically and set to $k_1 = 1$, $k_2 = 15$, $k_3 = 1$, $k_4 = 10$ and $k_5 = 10$. The slip parameters i_L and i_R are taken as

$$i_L(t) = \begin{cases} 0.0 & \text{if } 0 \leq t < 20 \text{ s} \\ 0.2 & \text{if } 20 \leq t < 60 \text{ s} \\ 0.2 - \frac{0.2}{30}(t - 60) & \text{if } 60 \leq t < 90 \text{ s} \\ 0.0 & \text{if } 90 \leq t \leq 100 \text{ s} \end{cases} \tag{8}$$

$$i_R(t) = \begin{cases} 0.0 & \text{if } 0 \leq t < 10 \text{ s} \\ 0.3 & \text{if } 10 \leq t < 50 \text{ s} \\ 0.3 - \frac{0.3}{10}(t - 50) & \text{if } 50 \leq t < 60 \text{ s} \\ 0.0 & \text{if } 60 \leq t \leq 100 \text{ s} \end{cases} \tag{9}$$

Note that the parameter profiles are as a function of time, in seconds, while in the plots the values are as a function of iterations, calculated as $i = \lceil 50/\delta t \rceil$. To obtain satisfactory results, we selected the filter parameters $\beta = 2$ and $\gamma = 0$. When it comes to the parameter α, we conducted simulations considering various values of alpha belonging to the interval $(0, 1]$. The covariance of the process and the measurement noise, needed by the UKF algorithm, are selected respectively as $Q_k = I_d \cdot 10^{-5}$ and $R_k = I_d \cdot 10^{-5}$. We have assumed that the signals τ_L and τ_R are measured without noise. In our simulation experiments, we investigated the impact of different values of some parameters on the effectiveness of our slipping parameter estimation technique. Recognizing the influence of these parameters is important for improving the control by tuning. Before analyzing the test results, it is interesting to observe how the controller behaves when the UKF's estimation of the slipping parameters is not used, but these are replaced with $i_{Le} = 0$, $i_{Re} = 0$. As can be seen in Figure 3, although the UKF correctly estimates the slipping parameters, not having these values in the controller results in a significant increase in error with respect to the desired trajectory, and poor tracking of it.

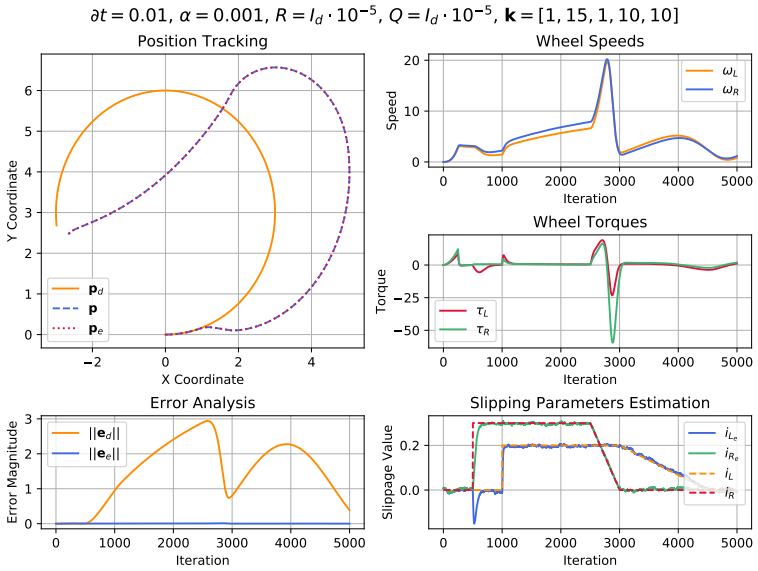


Figure 3: Simulation with slipping estimation set to $i_{Le} = i_{Re} = 0$ in the controller.

Impact of α

The parameter α affects the spread of sigma points used in the prediction and update steps. By adjusting α, we control the degree of spread of these points, which impacts the filter's ability to capture the true distribution of the state variables. Tests conducted on this parameters considers the range $\alpha \in (0, 1]$. When we adjust the parameter alpha in the Unscented Kalman Filter from a small value like 0.001 (depicted in Figure 4), to its upper limit of 1 (depicted in Figure 5), we're essentially changing how the filter explores and updates its estimation of the robot's state. We can observe that as α increases, there is a fall in the trajectory tracking performance. This can be attributed to the big spread of sigma points in the UKF prediction and update steps. So with a higher alpha, the filter explores a larger region of the state space during estimation. While this increased exploration can help capture non-linearities and uncertainties in the system dynamics more effectively, it may also result in a less precise estimation of the true state, particularly in regions of high curvature or rapid change. Despite the slight deterioration in trajectory tracking, the profile of slipping parameters remains close to the reference values. This indicates that the Unscented Kalman Filter is still able to estimate the slipping parameters accurately, even with the increased exploration introduced by higher α values. However, the slipping parameter profile may appear more noisy as alpha increases. This is likely due to the filter's increased sensitivity to measurement noise and disturbances.

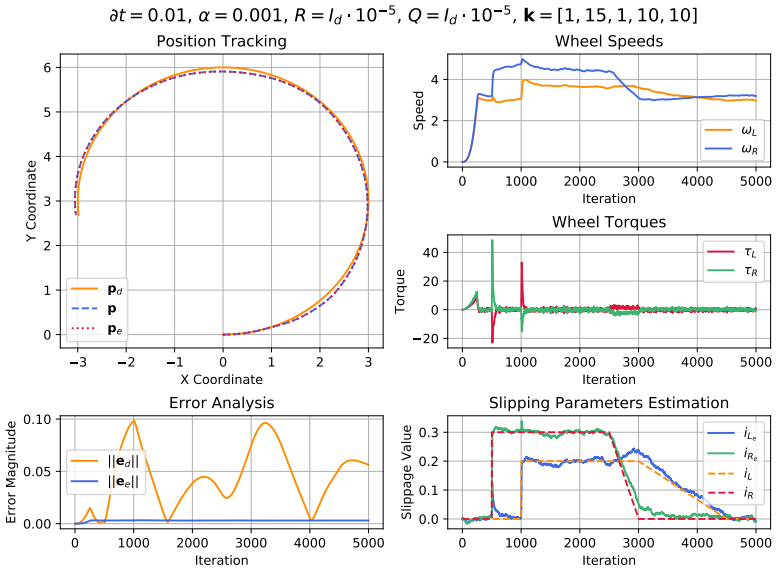


Figure 4: Impact of $\alpha = 0.001$.

Overall, increasing α from a small value to its maximum limit involves a trade-off between exploration and precision in the filter's estimation process.

Impact of covariance matrices R and Q

One important problem that needs to be addressed in using the UKF is how to properly set up the covariance matrices of measurement noise R and process noise Q. The measurement noise[2] covariance matrix, denoted as R, quantifies the uncertainty or variability associated with the sensor measurements used in the state estimation process. The process noise[3] covariance matrix instead, denoted as Q, quantifies the uncertainty or variability associated with the process noise in the system dynamics. The graphical results will show us how much R and Q influence the UKF's performance. Improper choice of R and Q may significantly degrade the UKF's performance. First, let's examine the results of the simulation with $R = I_d \cdot 0.01$, depicted in Figure 6, meaning after increasing the measurement noise covariance matrix by three orders of magnitude. By analyzing the error and trajectory tracking, it is evident that even with an increase in the values of the noise covariance matrix, the trajectory continues to run correctly, although there is a slight increase in the error $||e_d||$ with respect

[2]Measurement noise refers to random errors, biases, or inaccuracies in the sensor measurements, which deviate from the true values of the observed variables.

[3]Process noise refers to the random disturbances or uncertainties in the evolution of the system over time, which are not explicitly accounted for by the state transition model.

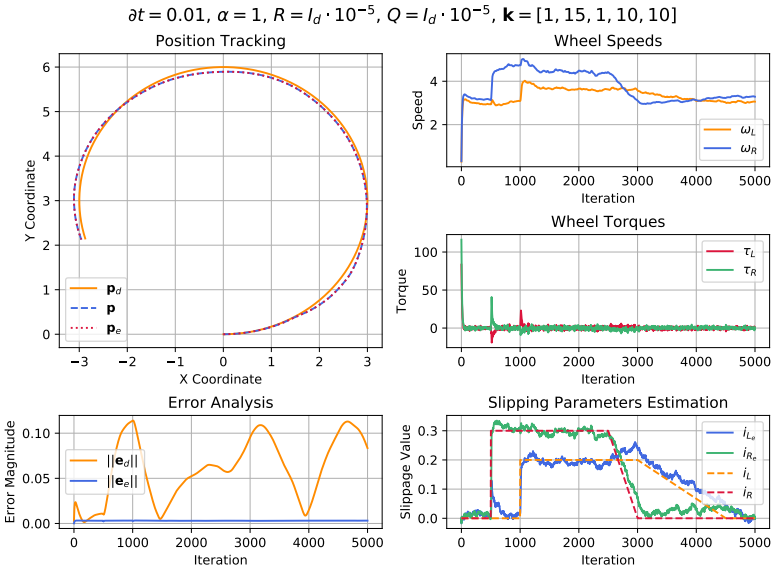


Figure 5: Impact of $\alpha = 1$. For demonstration purposes, there is no initial Bézier profile on the velocities, and this is reflected by the initial spike on τ_L and τ_R.

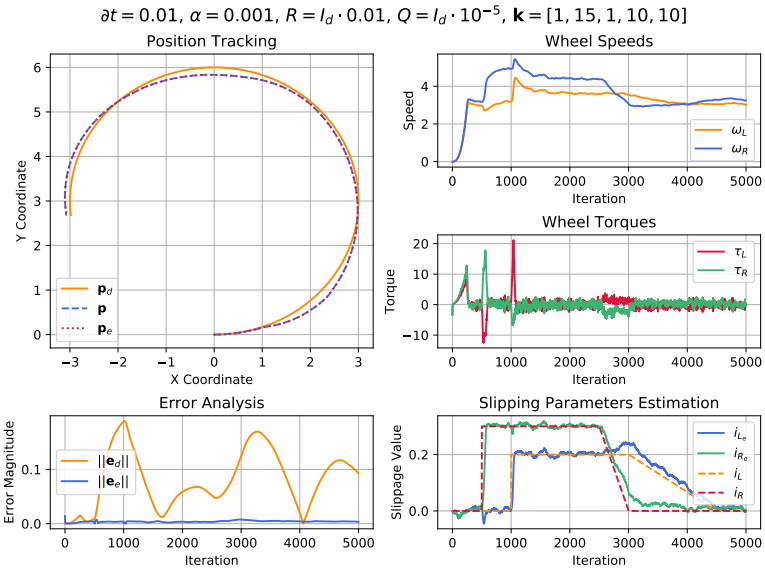


Figure 6: Impact of covariance matrix $R = I_d \cdot 0.01$.

to the desired trajectory. In addition, the filter consistently estimates the slip parameters, as evidenced by the profiles almost mirroring the reference values. Similarly, tests conducted on the Q matrix depicted in Figure 6 were obtained by increasing the values from $I_d \cdot 10^{-5}$ to $I_d \cdot 10^{-3}$. Analyzing the outcomes, we observe a significant degradation in trajectory tracking, which may be attributed to the fact that when we increase Q, we're essentially introducing more randomness or unpredictability into how the system evolves. This uncertainty makes it more challenging for the state estimation filter to accurately predict the future state of the system. Consequently, we observe significant deviations from the expected trajectory, resulting in a evident deterioration in trajectory tracking performance. Focusing our attention to slipping parameter profiles, the impact of increasing Q is similar: slipping parameters are derived from the dynamics of the system, so any increase in the uncertainty of these dynamics directly affects the accuracy of slipping parameter estimation. The observed results lead us to conclude that increased uncertainty in the system dynamics results in the filter generating noisy or wrong estimates of slipping parameters and in relation to this, a bad trajectory tracking.

Impact of the initial state of the UKF

In our experimentation, we also explored various initial states for the UKF, ranging from predefined positions like $[1, 1, \pi/4, 0, 0, 0, 0]$, to randomly selecting

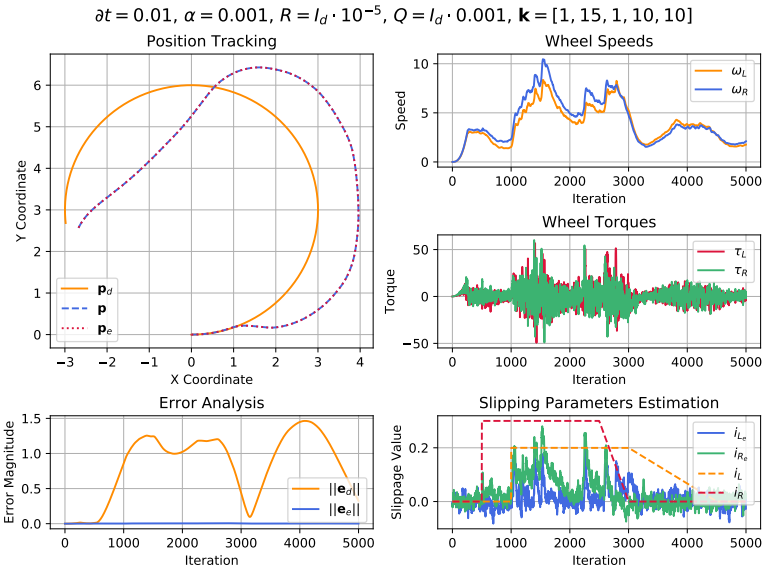


Figure 7: Impact of covariance matrix $Q = I_d \cdot 0.001$.

initial positions within certain ranges for x and y. This approach aimed to show how the initial position of the UKF influences its convergence and performance. One main aspect of our results is the consistency observed in the filter's behavior across the spectrum of starting points tested. Despite the diversity in initial positions, including both predetermined and randomly selected ones, we did not observe any substantial differences in the filter's performance. The absence of differences underscores the filter's ability to adapt and converge effectively to the true state of the system, regardless of where it begins its estimation process. This consistency across different starting points is a highly desirable characteristic in state estimation algorithms, particularly in dynamic and uncertain environments typical of mobile robotics.

4.2 ROS and Gazebo tests

In this section we analyze the results obtained by applying controller and UKF in combination with Gazebo, a 3D simulation environment, and ROS, to receive and communicate information to TIAGo. Also in this case the controller parameters were determined through empirical testing and set as follows: $k_1 = 1$, $k_2 = 8$, $k_3 = 2$, $k_4 = 3$, and $k_5 = 3$. For the UKF, we configured the parameters $\alpha = 0.001$, $R = I_d \cdot 10^{-4}$ (measurement noise covariance matrix), and $Q = I_d \cdot 10^{-5}$ (process noise covariance matrix). We specify that time is shown in the plots, but the values refer to iterations of ROS. In the following tests, we

have assumed, based on the results obtained from the ODE simulations, that the slipping parameters estimated by the UKF correspond to the real ones. Unlike the previous simulation in ODE, the tools we use in these tests, particularly Gazebo, allow us to introduce an additional customizable factor: the friction coefficients μ along the principal directions. This enables us to evaluate the effect of different terrains on the slipping parameters. To modify these parameters, it is necessary to edit the SDF (Simulation Description Format) files, which define a wide variety of environments, including terrain modification. Specifically, by acting on the file detailing the collision properties of the links, we modified the parameters μ and μ_2, which represent the unitless maximum ratio of force in the first and second friction pyramid directions to the normal force, respectively. By default, these parameters are set at $\mu = 100$ and $\mu_2 = 50$ in an empty world; for our tests, we have reduced both by a factor of 50, bringing them to $\mu = 2$ and $\mu_2 = 1$. Compared to the previous tests, the speeds applied to the robot were increased from $v_r = 0.3$ m/s and $\omega_r = 0.1$ rad/s in the empty world to $v_r = 0.8$ m/s and $\omega_r = 0.2$ rad/s in the reduced friction world, to have a more noticeable effect on the slipping parameters. The initial Bézier curve on the velocity profiles was also removed to further accentuate slipping from a standstill. The expected result is that by reducing the μ parameters, and thus the friction coefficients, the wheels would slip more compared to the empty world, and therefore the estimation of their value by the UKF should increase. As can be seen from the plots in Figures 8 and 9, conducted in the empty world, the estimate of the slipping parameters settles around a value of 0.1 for both parameters i_{Le} and i_{Re}. In these tests, the trajectories are executed correctly, with an error norm of about 0.3. However, in tests depicted in Figures 10 and 11 conducted in the modified environment by reducing the μ parameters, the estimate of the slipping parameters assumes higher values, reaching an average of about 0.5 (about five times higher), reflecting exactly the result we would have expected. However, it can be observed that with less wheel grip, the performance on trajectory tracking is worse, leading to an error norm relative to the desired trajectory of 0.75.

5 Conclusions

In this project, we have effectively demonstrated the use of the Unscented Kalman Filter (UKF) to enhance the control and navigation of mobile robots dealing with wheel slippage. Our simulations show that the UKF can provide reliable slip parameter estimations that contribute to improved trajectory tracking of mobile robots across different terrains. The use of theoretical models and practical simulations helped us understand and address the challenges posed by wheel slippage. By incorporating the UKF into our control strategy, we were able to estimate non-measurable parameters that are crucial for maintaining accurate robot navigation under slip conditions. Our results validate the effectiveness of the UKF over traditional methods, particularly in handling the nonlinear dynamics of mobile robots. Simulations carried out in both simplified

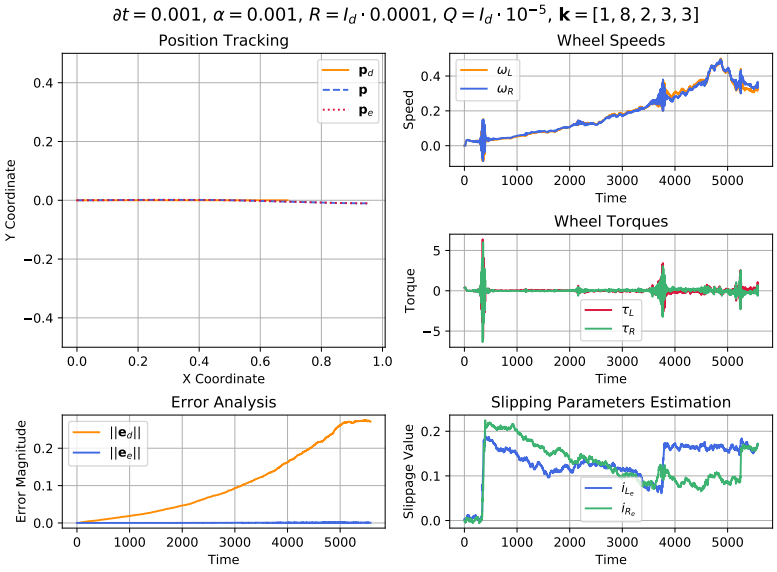


Figure 8: Linear trajectory in the empty world, with $\mu = 100$, $\mu_2 = 50$.

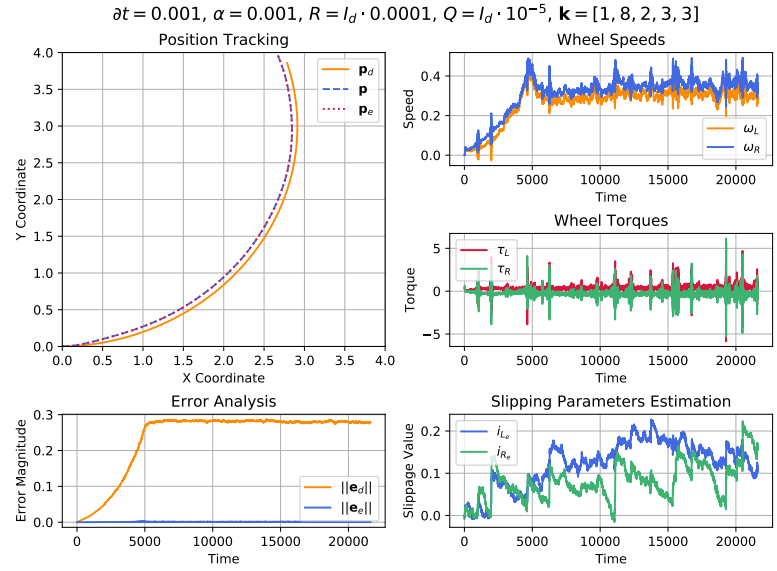


Figure 9: Circular trajectory in the empty world, with $\mu = 100$, $\mu_2 = 50$.

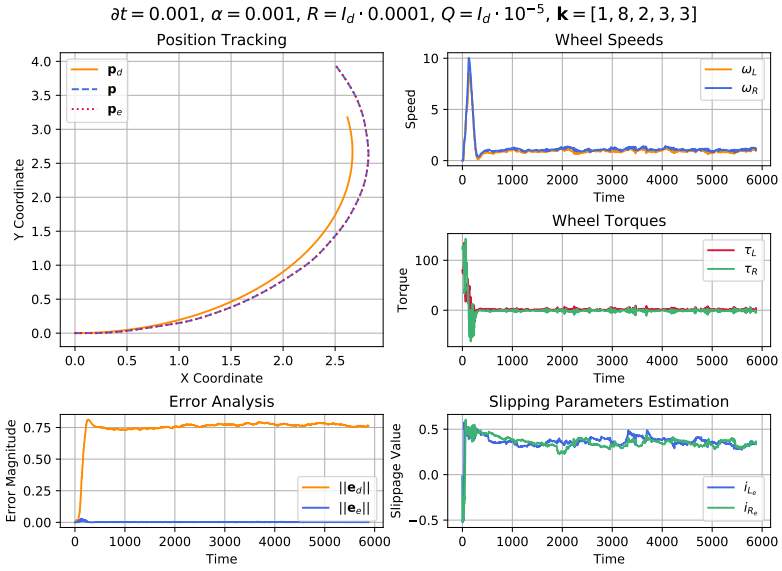


Figure 10: Circular trajectory in the reduced friction world, with $\mu = 2$, $\mu_2 = 1$.

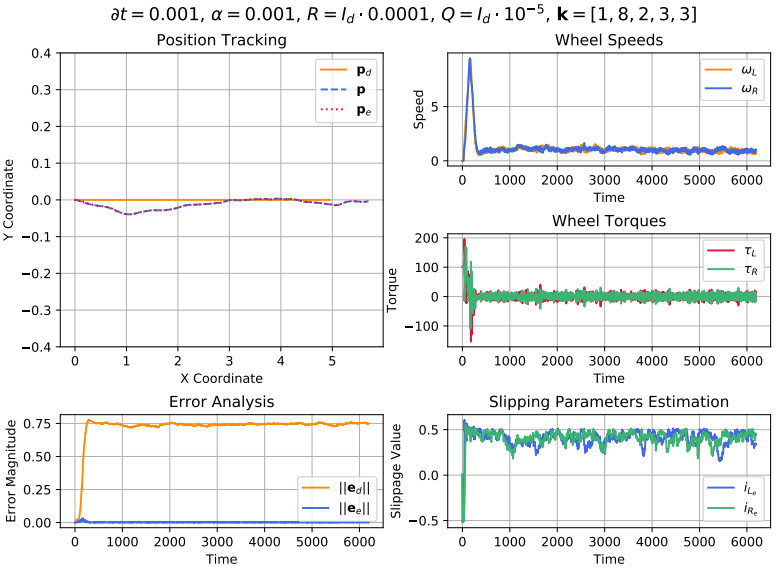


Figure 11: Linear trajectory in the reduced friction world, with $\mu = 2$, $\mu_2 = 1$.

environments and more complex settings using ROS and Gazebo confirmed that our approach could adapt to real-world conditions effectively.

References

[1] R. Fierro and F. L. Lewis. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. *Journal of Robotic Systems*, 14(3):149–163, 1997.

[2] R. Gonzales, M. Fiacchini, T. Alamo, J. Guzmán, and F. Rodriguez. Adaptive control for a mobile robot under slip conditions using lmi-based approach. In *Proceedings of the European Control Conference*, pages 1251–1256, Budapest, Hungary, 2009.

[3] K. Iagnemma and C. Ward. Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain. *Autonomous Robot*, 26:33–46, 2009.

[4] J. Iossaqui, J. Camino, and D. Zampieri. Adaptive torque-based control of tracked mobile robots with unknown longitudinal slip parameter. In *Proceedings of the VI Congresso Nacional de Engenharia Mecânica*, pages 1–10, Campina Grande, Brazil, 2010.

[5] J. Iossaqui, J. Camino, and D. Zampieri. Adaptive tracking control of tracked mobile robots with unknown slip parameters. In *Proceedings of the XVIII Congresso Brasileiro de Automática*, pages 1846–1851, Bonito, Brazil, 2010.

[6] J. Iossaqui, J. Camino, and D. Zampieri. An application of filtering techniques for the tracking control of mobile robots with slipping. In *Proceedings of the XIV International Symposium on Dynamic Problems of Mechanics*, pages 194–203, São Sebastião, Brazil, 2011.

[7] J. Iossaqui, J. Camino, and D. Zampieri. A nonlinear control design for tracked robots with longitudinal slip. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control*, Milano, Italy, 2011.

[8] J. G. Iossaqui. Slip estimation using the unscented kalman filter for the tracking control of mobile robots. 2011.

[9] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[10] D. H. Kim and J. H. Oh. Globally asymptotically stable tracking control of mobile robots. In *Proceedings of the IEEE International Conference on Control Applications*, pages 1297–1301, Trieste, Italy, 1998. IEEE.

[11] M. Michalek, P. Dutkiewicz, M. Kielczewski, and D. Pazderski. Trajectory tracking for a mobile robot with skid-slip compensation in the vector-field-orientation control system. *International Journal of Applied Mathematics and Computer Science*, 4(19):547–559, 2009.

[12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*, volume 46. 2009.

[13] N. Sidek and N. Sarkar. Dynamic modeling and control of nonholonomic mobile robot with lateral slip. In *Proceedings of the 7th WSEAS International Conference on Signal Processing, Robotics and Automation*, pages 66–74, Cambridge, UK, 2008.

[14] Z. Song, Y. Zweiri, L. Seneviratne, and K. Althoefer. Non-linear observer for slip estimation of tracked vehicles. *Journal of Automobile Engineering*, 222(4):515–533, 2008.

[15] A. Tarakameh, K. Shojaei, and A. Shahr. Adaptive control of nonholonomic wheeled mobile robot in presence of lateral slip and dynamic uncertainties. In *Proceedings of the 18th Iranian Conference on Electrical Engineering*, Isfahan, Iran, 2010.

[16] D. Wang and C. Low. Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective. *IEEE Transactions on Robotics*, 24(3):676–687, 2008.

[17] J. Y. Wong. *Theory of Ground Vehicles*. John Wiley and Sons, New York, USA, 2001.

[18] B. Zhou, Y. Peng, and J. Han. Ukf based estimation and tracking control of nonholonomic mobile robots with slipping. In *IEEE International Conference on Robotics and Biomimetics*, pages 2058–2063, Sanya, China, 2007. IEEE.