Application of a Time-Varying Linear Quadratic Controller for Trajectory Tracking of a Four-Wheel Mobile Robot with Independent Steering and Drive

B. Ćaran, N. Škifić, V. Milić, M. Švaco
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, HR-10000, Croatia
branimir.caran@fsb.unizg.hr, nikoskific@gmail.com, vladimir.milic@fsb.unizg.hr, marko.svaco@fsb.unizg.hr

Abstract—**This paper is concerned with the design of a controller for trajectory tracking of a mobile robot with four steerable and four independently driven wheels (4WIS4WID). Taking into account the appropriate assumptions, the kinematic equations of the robot are converted into three-input, two-chain, single-generator chained form. After that, for a small perturbations around the reference trajectory, an approximate linearization of the chained form is performed. This procedure led to the form of a time-varying system suitable for the synthesis of a time-varying linear control law according to the quadratic optimality criterion. The version of the control law which gives the best results is determined by computer simulations in a mathematical software package MATLAB (MathWorks, Natick, MA, USA). The obtained control law is applied to the experimental set-up of a mobile robot developed at the Regional Center of Excellence for Robotic Technology (CRTA). The controller is implemented using Robot Operating System (ROS) and experimental measurements are performed using the OptiTrack system.**

Keywords*—*four-wheel steered and drive mobile robot, chained form, trajectory tracking, time-varying linear quadratic controller

I. Introduction

Over past few years in the field of mobile robotics, significant efforts have been made in the research and development of robots with four wheels that have independent drive and steer. This interest has resulted in the development and application of various control strategies for this class of mobile robot systems.

In [1], nonlinear model predictive control where constraints are expressed by applying barrier functions for a robot moving in a human-centric environment has been proposed. In order to reduce the delay in the steering angle response also the model predictive control algorithm with dynamic constraints for high-speed trajectory tracking has been used in [2]. A hybrid control strategy in which the sliding-mode control law, near-time-optimal potential function and fuzzy-based adjustment rules have been applied and integrated into one efficient method, has been proposed in [3]. In [4], direct yaw moment robust sliding-mode control scheme for simultaneous trajectory tracking and disturbance rejection of four wheel robot has been presented. Furthermore, four wheel mobile robots actually belong to the same class of nonlinear systems as autonomous vehicles and thus the same principles and methods can be applied to solve control problems of this type of robot, see, for example, [5]–[8] and references therein.

In this paper, the trajectory tracking method applied to 4WIS4WID mobile robot is based on time-varying linear quadratic (TVLQ) control. Although the theory behind TVLQ approach is well-known and can be considered standard (see for example [9], [10] and references therein), the TVLQ-based synthesis of controllers is still an intensive area of research in various applications, see for example [11]–[14]. However, to the best of the authors' knowledge, the application and experimental verification of this control law, which additionally includes the transformation into three-input, two-chain, single-generator chained form, for robots with four wheels that have independent drive and steer has not been investigated yet. The main idea of the approach proposed in this paper is to approximate the nonlinear chain model with linear time-varying (LTV) differential error equations that need to be stabilized in the vicinity of the desired trajectory.

After the results of tracking the desired trajectory of the mobile robot in a closed-loop with a TVLQ state controller are evaluated in simulations, the proposed control strategy is implemented on a real mobile robot using ROS. First, experimental measurements are carried out in which several iterations to reduce the error from the desired trajectory are used to analyse the state data obtained directly from the robot's encoders. When a sufficiently small tracking error is achieved in this way, the actual trajectory of the robot in the horizontal plane is measured using the OptiTrack system and the results are discussed.

The rest of the paper is organized as follows. In Section II, the kinematic model of the considered structure of the mobile robot is presented and its transformation into three-input, two-chain, single-generator chained form is performed. In Section III, the error equations of the LTV system are introduced and the control problem is formulated. Then, the procedure for synthesis of the optimal time-varying control law for trajectory tracking according to the quadratic criterion is proposed. The results of simulations of the mobile robot control system are given and discussed in Section IV, while the description of the experimental set-up of the robot and the results of experimental measurements with the application of the ROS and OptiTrack systems are presented and discussed in Section V. Finally, Section VI concludes the paper.

II. Mathematical Model Description

A. Kinematic Model

A schematic representation of the robot with its geometric characteristics relevant for kinematics modelling is shown in Fig. 1. We denote by x and y the coordinates of the reference point of robot in the Cartesian frame of reference, θ orientation angle with respect to the positive x-axis, v_i, δ_i and ω_i the linear velocity, steering angle and steering velocity of i-th wheel, respectively, a and b the lengths between the centroid of the robot and each wheel. Numerical values of robot's constant parameters used in simulations and experiments are: the distance from the wheel to the centre of the robot $a = b = 0.1125$ m and wheel radius $r = 0.0254$ m.

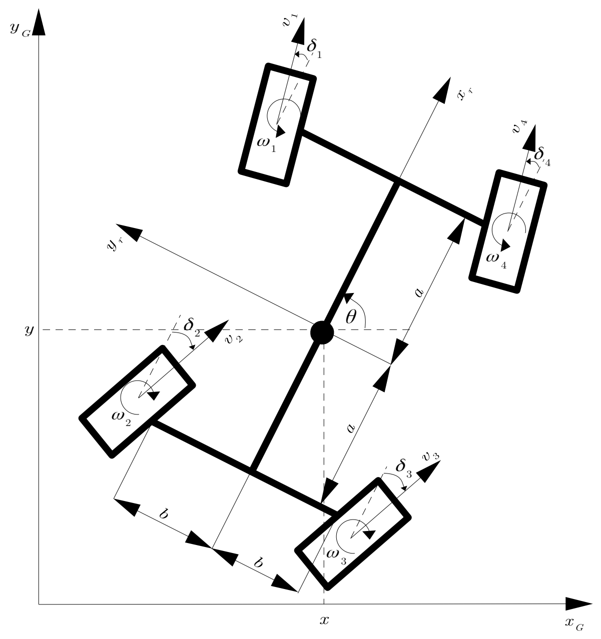

Fig. 1. Schematics of the mobile robot system.

The equations describing the kinematics of a 4WIS4WID mobile robot are given in [2] and [15], so they will not be discussed in detail here. In this paper, we will introduce appropriate assumptions for the simplification of the kinematic model, which will facilitate the subsequent transformation into the chained form.

In the approach presented in this paper we assume that $v_1 = v_2 = v_3 = v_4$ and $\delta_1 = \delta_4,\ \ \delta_2 = \delta_3$. Hence we get the following set of equations:

$$\dot{x} = \frac{1}{2}\left(\cos\left(\delta_1+\theta\right)+\cos\left(\delta_2+\theta\right)\right)v_1, \tag{1}$$

$$\dot{y} = \frac{1}{2}\left(\sin\left(\delta_1+\theta\right)+\sin\left(\delta_2+\theta\right)\right)v_1, \tag{2}$$

$$\dot{\theta} = \left(\frac{x_{w1}\sin\delta_1}{2x_{w1}^2+2y_{w1}^2}+\frac{x_{w2}\sin\delta_2}{2x_{w2}^2+2y_{w2}^2}\right)v_1, \tag{3}$$

$$\dot{\delta_1} = \omega_1, \tag{4}$$

$$\dot{\delta_2} = \omega_2, \tag{5}$$

where x_{wi} and y_{wi} are predefined as follows: $(x_{w1}, y_{w1}) = (a, b)$ and $(x_{w2}, y_{w2}) = (-a, b)$.

The system (1)-(5) can be written in standard input-affine control oriented form

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\Phi}(\boldsymbol{\xi})\boldsymbol{\omega}, \tag{6}$$

where $\boldsymbol{\xi} = \begin{bmatrix} x & y & \theta & \delta_1 & \delta_2 \end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{\omega} = \begin{bmatrix} v_1 & \omega_1 & \omega_2 \end{bmatrix}^{\mathrm{T}}$ and

$$\boldsymbol{\Phi}(\boldsymbol{\xi}) = \begin{bmatrix} \frac{(\cos(\delta_1+\theta)+\cos(\delta_2+\theta))}{2} & 0 & 0 \\ \frac{(\sin(\delta_1+\theta)+\sin(\delta_2+\theta))}{2} & 0 & 0 \\ \frac{x_{w1}\sin\delta_1}{2x_{w1}^2+2y_{w1}^2}+\frac{x_{w2}\sin\delta_2}{2x_{w2}^2+2y_{w2}^2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

B. Transformation into Chained Form

In order to transform (6) with (7) into a chained form, in this work we applied the procedure from [16], [17], where the conditions for the generalization of the chained form to systems with more than two inputs were derived. The system (6) with (7) can be transformed into the following three-input, two-chain, single-generator chained form

$$\dot{x}_1 = u_1, \tag{8}$$

$$\dot{x}_2 = u_2, \tag{9}$$

$$\dot{x}_3 = x_2 u_1, \tag{10}$$

$$\dot{x}_4 = u_3, \tag{11}$$

$$\dot{x}_5 = x_4 u_1. \tag{12}$$

First, to transform the system (6) with (7) into the system (8)-(12), the following substitutions are introduced

$$x_1 = x,\ \ x_3 = \theta,\ \ x_5 = y. \tag{13}$$

Taking the derivative of the variable x_1 with respect to time and on the basis of (1), it follows

$$\dot{x}_1 = \dot{x} = u_1 \Rightarrow \tag{14}$$

$$v_1 = \frac{2u_1}{\cos\left(\delta_1+\theta\right)+\cos\left(\delta_2+\theta\right)}. \tag{15}$$

Taking the derivative of the variable x_5 with respect to time and on the basis of (2) and including (15), it follows

$$\dot{x}_5 = \dot{y} = x_4 u_1 \Rightarrow \tag{16}$$

$$x_4 = \frac{v_1(\sin(\delta_1+\theta)+\sin(\delta_2+\theta))}{2u_1} \Rightarrow \tag{17}$$

$$x_4 = \tan\left(\frac{\delta_1+\delta_2}{2}+\theta\right). \tag{18}$$

From (11) it follows that the input variable u_3 is equal to the time derivative of the variable x_4, therefore by deriving the right side of the expression (18) and taking into account (4), (5) and (15) we get

$$\dot{x}_4 = u_3 \Rightarrow \tag{19}$$

$$u_3 = \frac{d}{dt}\left[\tan\left(\frac{\delta_1+\delta_2}{2}+\theta\right)\right] \Rightarrow \tag{20}$$

$$u_3 = \frac{\frac{2u_1\left(\frac{x_{w1}\sin(\delta_1)}{{x_{w1}}^2+{y_{w1}}^2}+\frac{x_{w2}\sin(\delta_2)}{{x_{w2}}^2+{y_{w2}}^2}\right)}{\cos(\delta_1+\theta)+\cos(\delta_2+\theta)}+\omega_1+\omega_2}{\cos(\delta_1+\delta_2+2\theta)+1} \Rightarrow \tag{21}$$

$$u_3 = \frac{v_1\left(\frac{x_{w1}\sin(\delta_1)}{{x_{w1}}^2+{y_{w1}}^2}+\frac{x_{w2}\sin(\delta_2)}{{x_{w2}}^2+{y_{w2}}^2}\right)+\omega_1+\omega_2}{\cos(\delta_1+\delta_2+2\theta)+1}. \tag{22}$$

Furthermore, by taking the time derivative of the variable x_3 and according to (3) and (15) it follows

$$\dot{x}_3 = \dot{\theta} = x_2 u_1 \Rightarrow \tag{23}$$

$$x_2 = \frac{\frac{x_{w1}\sin(\delta_1)}{x_{w1}^2+{y_{w1}}^2}+\frac{x_{w2}\sin(\delta_2)}{{x_{w2}}^2+{y_{w2}}^2}}{\cos(\delta_1+\theta)+\cos(\delta_2+\theta)}. \tag{24}$$

From (9) it follows that the input variable u_2 is equal to the time derivative of the variable x_2, therefore by deriving the right side of the expression (24) and taking into account (3), (4) and (5) we get

$$\dot{x}_2 = u_2 \Rightarrow \tag{25}$$

$$u_2 = \frac{d}{dt}\left[\frac{\frac{x_{w1}\sin(\delta_1)}{{x_{w1}}^2+{y_{w1}}^2}+\frac{x_{w2}\sin(\delta_2)}{{x_{w2}}^2+{y_{w2}}^2}}{\cos(\delta_1+\theta)+\cos(\delta_2+\theta)}\right] \Rightarrow \tag{26}$$

$$\begin{aligned} u_2 = {} & \frac{B_1\omega_1+B_2\omega_2}{(c_1+c_2)} \\ & + \frac{(D_1+D_2)(2\omega_1 s_1+2\omega_2 s_2)}{2(c_1+c_2)} \\ & + \frac{v_1(D_1+D_2)^2(s_1+s_2)}{2(c_1+c_2)^2}. \end{aligned} \tag{27}$$

where for the sake of simpler notation we introduced the following

$$c_i = \cos(\delta_i+\theta), \tag{28}$$

$$s_i = \sin(\delta_i+\theta), \tag{29}$$

$$B_i = \frac{x_{wi}\cos(\delta_i)}{{x_{wi}}^2+{y_{wi}}^2}, \tag{30}$$

$$D_i = \frac{x_{wi}\sin(\delta_i)}{{x_{wi}}^2+{y_{wi}}^2}, \tag{31}$$

$$K = 1+\cos(\delta_1+\delta_2+2\theta). \tag{32}$$

Finally, everything previously derived can be summarized in the following change of coordinates

$$x_1 = x, \tag{33}$$

$$x_2 = \frac{D_1+D_2}{c_1+c_2}, \tag{34}$$

$$x_3 = \theta, \tag{35}$$

$$x_4 = \tan\left(\frac{\delta_1+\delta_2}{2}+\theta\right), \tag{36}$$

$$x_5 = y, \tag{37}$$

together with the input transformation

$$v_1 = \frac{2u_1}{c_1+c_2}, \tag{38}$$

$$\begin{aligned} \omega_1 = {} & \frac{B_2(2u_1(D_1+D_2)-Ku_3(c_1+c_2))}{(B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2)} \\ & + \frac{u_2(c_1+c_2)^2}{(B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2)} \\ & - \frac{(D_1+D_2)Ks_2u_3(c_1+c_2)}{(c_1+c_2)((B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2))} \\ & - \frac{u_1(D_1+D_2)^2(s_1-s_2)}{(c_1+c_2)((B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2))}, \end{aligned} \tag{39}$$

$$\begin{aligned} \omega_2 = {} & \frac{B_1(-2u_1(D_1+D_2)+Ku_3(c_1+c_2))-u_2(c_1+c_2)^2}{(B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2)} \\ & + \frac{(D_1+D_2)(Ks_1u_3(c_1+c_2)-(u_1(D_1+D_2)(s_1-s_2)))}{(c_1+c_2)((B_1-B_2)(c_1+c_2)+(D_1+D_2)(s_1-s_2))}, \end{aligned} \tag{40}$$

so the system (6) with (7) is in three-input, two-chain, single-generator chained form (8)-(12).

Remark 1: Note that the chained form obtained by the transformation equations described above is actually only locally defined since (34) and (36) may be undefined for certain robot orientations. Therefore it is necessary to exclude these cases as follows

$$\cos(\delta_1+\theta)+\cos(\delta_2+\theta) \neq 0, \tag{41}$$

$$\delta_1+\delta_2+2\theta \neq \pi+2k\pi, \quad \forall k \in \mathbb{Z}. \tag{42}$$

III. Controller Synthesis for Trajectory Tracking

Based on [18], we formulate the problem of the trajectory tracking using the procedure of approximation of a nonlinear system by an LTV system. This enables the application of standard linear theory for the synthesis of control law.

First, for simplicity, let's denote system (8)-(12) in standard vector notation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{43}$$

where $\mathbf{x}(t) = [x_1\ x_2\ x_3\ x_4\ x_5]^{\mathrm{T}}$, $\mathbf{u}(t) = [u_1\ u_2\ u_3]^{\mathrm{T}}$ and $\mathbf{f} = [u_1\ u_2\ x_2u_1\ u_3\ x_4u_1]^{\mathrm{T}}$.

Suppose that the smooth desired trajectory is given in the Cartesian coordinate system by $x_d(t)$ and $y_d(t)$ and the desired orientation is computed as $\theta_d(t) = \arctan(\dot{y}_d(t)/\dot{x}_d(t))$. Furthermore, assume that for system (8)-(12) the desired trajectory is feasible, which means that desired states $\mathbf{x}_d(t) = [x_{d1}(t)\ x_{d2}(t)\ x_{d3}(t)\ x_{d4}(t)\ x_{d5}(t)]^{\mathrm{T}}$ and inputs $\mathbf{u}_d(t) = [u_{d1}(t)\ u_{d2}(t)\ u_{d3}(t)]^{\mathrm{T}}$ (i.e. reference robot) can be computed from (33)-(40).

The Taylor expansion of system (43) around $(\mathbf{x}_d(t), \mathbf{u}_d(t))$, ignoring the higher order terms, results in

$$\begin{aligned} \dot{\mathbf{x}}(t) = {} & \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t)) \\ & + \frac{\partial \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t))}{\partial \mathbf{x}}(\mathbf{x}(t)-\mathbf{x}_d(t)) \\ & + \frac{\partial \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t))}{\partial \mathbf{u}}(\mathbf{u}(t)-\mathbf{u}_d(t)). \end{aligned} \tag{44}$$

Next, let state errors and input errors be defined as $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_d(t)$, $\tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_d(t)$, respectively, then from (44) the LTV error equations are as follows

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t) &= \frac{\partial \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t))}{\partial \mathbf{x}}\tilde{\mathbf{x}}(t) + \frac{\partial \mathbf{f}(\mathbf{x}_d(t), \mathbf{u}_d(t))}{\partial \mathbf{u}}\tilde{\mathbf{u}}(t) = \\ &= \mathbf{A}(t)\tilde{\mathbf{x}}(t) + \mathbf{B}(t)\tilde{\mathbf{u}}(t),\end{aligned} \tag{45}$$

where

$$\begin{aligned}\mathbf{A}(t) &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & u_{d1}(t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_{d1}(t) & 0 \end{bmatrix}, \\ \mathbf{B}(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_{d2}(t) & 0 & 0 \\ 0 & 0 & 1 \\ x_{d4}(t) & 0 & 0 \end{bmatrix}.\end{aligned} \tag{46}$$

Remark 2: Note that for system (45) to be fully controllable, $u_{d1}(t) \neq 0$ must always hold.

Our objective is to determine the control law of the form

$$\tilde{\mathbf{u}}(t) = -\mathbf{K}(t)\tilde{\mathbf{x}}(t), \tag{47}$$

for the LTV system (45), where the matrix $\mathbf{K}(t)$ is determined according to the quadratic criterion of optimality, and thereby obtain the optimal control law of the form

$$\mathbf{u}^* = \mathbf{u}_d(t) - \mathbf{K}(t)\tilde{\mathbf{x}}(t), \tag{48}$$

for the system in chained form (8)-(12).

It is well known that local LQ trajectory stabilization [10], or TVLQ control problem [9], requires the solution of the differential Riccati equation

$$\begin{aligned}-\dot{\mathbf{P}}(t) &= \mathbf{P}(t)\mathbf{A}(t) + \mathbf{A}^{\mathrm{T}}(t)\mathbf{P}(t) \\ &\quad - \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}(t)\mathbf{P}(t) + \mathbf{Q}, \\ &\mathbf{P}(t_f) = \mathbf{Q}_f,\ \mathbf{Q} = \mathbf{Q}^{\mathrm{T}} \succ 0,\ \mathbf{R} = \mathbf{R}^{\mathrm{T}} \succ 0.\end{aligned} \tag{49}$$

Solving (49) yields a symmetric positive definite matrix $\mathbf{P}(t)$ and the controller matrix can be determined as follows

$$\mathbf{K}(t) = \mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}(t)\mathbf{P}(t). \tag{50}$$

In the research presented in this paper, in order to solve (49), first the system (45) in a closed-loop with the control law (47) on a finite time interval $[t_0,\ t_f]$ is discretized using the Euler method. Then functions from MATLAB Control System Toolbox are employed to solve the Riccati equation at each discrete point of the time interval.

IV. Simulation Results

In this section, the simulation results for control of the mobile robot, using the controller synthesis procedure that is proposed and described in the previous sections, are presented.

The desired Gaussian trajectory to evaluate the control strategy is defined as follows

$$\begin{aligned}&x_d(t) = v_m t,\ y_d(t) = Y e^{-s(x_d - x_c)^2}, \\ &\theta_d(t) = \arctan\left(\frac{\dot{y}_d(t)}{\dot{x}_d(t)}\right),\end{aligned} \tag{51}$$

The parameters of the Gaussian trajectory are set as follows: $s = 3$, $Y = 0.4$ m, $x_c = 1.5$ m, $v_m = 0.06$ m/s.

In order to obtain the reference of the robot, (51) is included in (33)-(37), from which the desired robot states are calculated as follows

$$x_{d1} = v_m t, \tag{52}$$

$$x_{d2} = \frac{2sYe^{s(x_c - v_m t)^2}\left(2s(x_c - v_m t)^2 - 1\right)}{4s^2Y^2(x_c - v_m t)^2 + e^{2s(x_c - v_m t)^2}}, \tag{53}$$

$$x_{d3} = \tan^{-1}\left(2sY(x_c - v_m t)e^{-s(x_c - v_m t)^2}\right), \tag{54}$$

$$x_{d4} = -2sY(x_c - v_m t)e^{-s(x_c - v_m t)^2}, \tag{55}$$

$$x_{d5} = Y e^{-s(v_m t - x_c)^2} \tag{56}$$

and then from (8)-(12) the desired inputs are calculated as follows

$$u_{d1} = v_m, \tag{57}$$

$$\begin{aligned}u_{d2} = &-\frac{4s^2 v_m Y(v_m t - x_c)e^{s(x_c - v_m t)^2}e^{2s(x_c - v_m t)^2}}{\left(4s^2Y^2(x_c - v_m t)^2 + e^{2s(x_c - v_m t)^2}\right)^2} \\ &\cdot \left(2s(x_c - v_m t)^2 - 3\right) \\ &- 4sY^2\left(s(x_c - v_m t)^2\left(2s(x_c - v_m t)^2 - 1\right) + 1\right)\end{aligned}, \tag{58}$$

$$u_{d3} = 2sv_m Y e^{-s(x_c - v_m t)^2}\left(2s(x_c - v_m t)^2 - 1\right). \tag{59}$$

With everything previously carried out, the matrices $\mathbf{A}(t)$ and $\mathbf{B}(t)$ from (46) are now defined, and thus the LTV system is determined, for which the controller synthesis is performed.

The entire trajectory tracking problem is discretized by applying the Euler method such that all initial states of the robot are equal to zero, the initial time is $t_0 = 0$ s, the final time is $t_f = 52$ s and the number of optimization time intervals is $N = 3250$ so that the sampling interval is 0.016 s. This corresponds to the sampling frequency of the experimental set-up, which is 62.5 Hz.

The procedure for calculating the controller matrix (47) at each discrete point of the time interval, which is based on the iterative solution of the Riccati equation (49), is written and implemented in MATLAB, using its standard functions from the Control System toolbox. Matrices $\mathbf{Q}$ and $\mathbf{R}$ from (49) are chosen as $\mathbf{Q} = \mathrm{diag}(10^5,\ 1,\ 1,\ 1,\ 10^6)$ and $\mathbf{R} = \mathrm{diag}(10^3,\ 1,\ 1)$. All calculations are performed on a standard portable (laptop) computer and simulation results are shown in Figs. 2-5.

From Fig. 2, it can be concluded that a very good accuracy of following the desired trajectory was achieved, as expected in both cases of the kinematic and chain model.

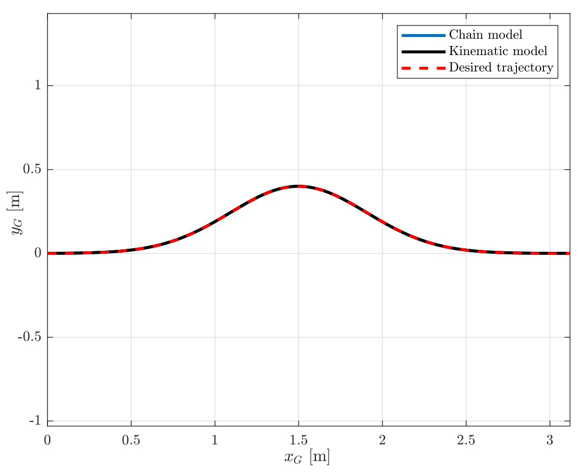

Fig. 2. Gaussian trajectory tracking simulation results in $x_G - y_G$ plane.

Fig. 3 and Fig. 4 show the control input variables (linear and steering velocities and their transformations) and from their responses and maximum values it can be seen that they are easily achievable on standard average electric motors.

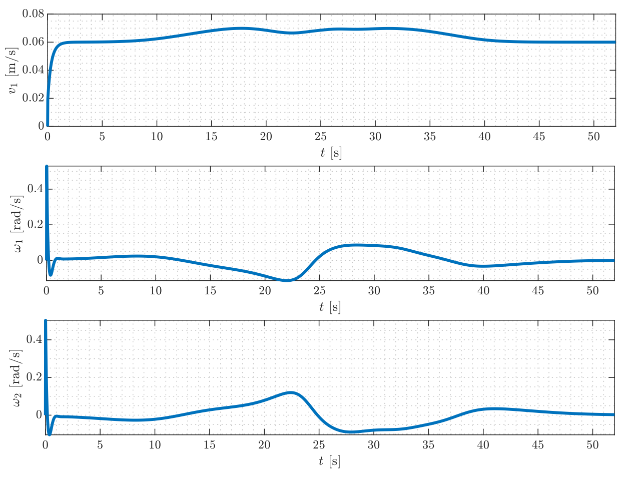

Fig. 3. Time response of control inputs for system (6) with (7) in the case of a Gaussian trajectory.

Fig. 5 shows a slightly larger error in following the reference orientation angle. This can be improved by adjusting the weight matrices $\mathbf{Q}$ and $\mathbf{R}$. However, it is well known that adjusting these matrices requires a compromise between the accuracy of the trajectory tracking and the value of the maximum control variables.

V. Experimental Results

A. Experimental Set-up Description

The mobile robotic structure considered in this paper is part of the experimental set-up designed as a suitable laboratory model for testing and verification of different control concepts and education which is preliminary described in previous works [19] and [20]. Although this mobile robot

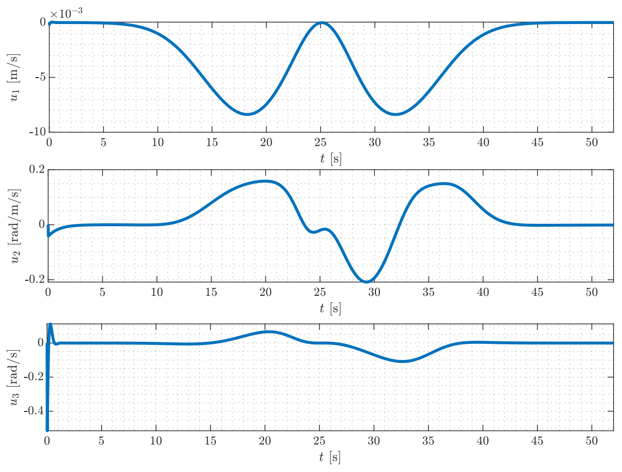

Fig. 4. Time response of control inputs for chained system (8)-(12) in the case of a Gaussian trajectory.

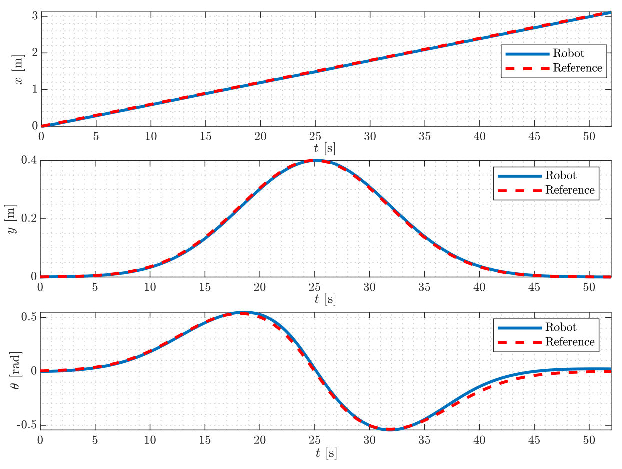

Fig. 5. Time response of state variables of system (6) with (7) in the case of a Gaussian trajectory.

was developed for climbing vertical walls, here we deal with the problem of following a reference trajectory on the ground by applying the methodology described in the previous sections. For the sake of readability, in this section we give a description of the main features and components of the system.

In Fig. 6, photos of the experimental set-up of the mobile robot and OptiTrack external measuring system are shown. The Optitrack system is used to compare the odometry with the data from the robot's motor encoders. The body of the robot is mostly made from acrylonitrile styrene acrylate (ASA) material by using a 3D printing process and some parts are made of carbon fiber tubes. The dimensions of the robot are 380×300 mm, with a total weight of 3.25 kg and a total payload of 1.5 kg. The wheels are driven and steered by Dynamixel XC430-W150 and XC430-W240 smart servo motors. The rotational positions on each motor are measured by magnetic rotary sensors AMS AS5601. Robot is also equipped with an IMU (Bosch BNO055) for relative localization. The motors are controlled by an

OpenCR board based on an STM32 microcontroller. This microcontroller communicates with the Raspberry Pi4B single-board computer via the USB protocol. Raspberry Pi4B runs on Linux and uses Robot Operating System (ROS) middleware. The mobile robot is powered by a 24 V and 3 kW stabilised laboratory power supply.

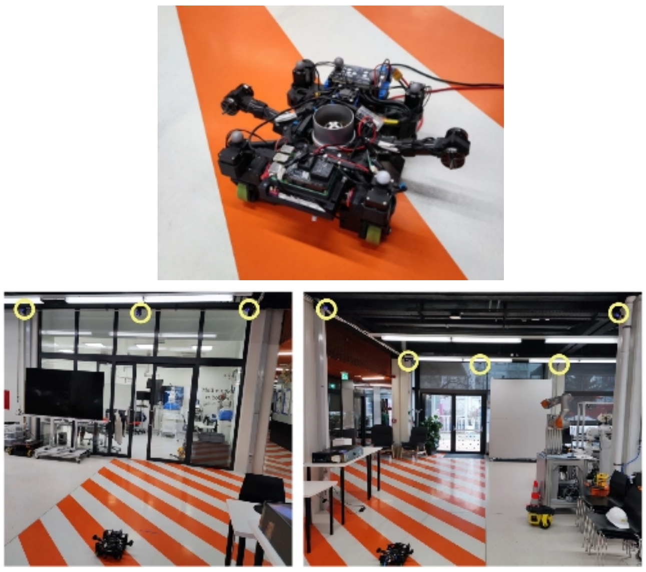
Fig. 6. Photo of the experimental set-up of the mobile robot and OptiTrack system.

B. Implementation and Measurement Results

Fig. 7 shows a block diagram of the entire control strategy for trajectory tracking, which is implemented on an experimental robot model using ROS and the Python 3 programming language. The selected version of ROS is Noetic Ninjemys for the Linux Ubuntu 20.04 operating system. The robot is controlled using WiFi in ROS Master-Slave configuration.

To create the ROS control *Node* in Python, we used following: the `rospy` package, the `String` module from the `std_msgs.msg` sub-package, the `JointState` module from the `sensor_msgs.msg` sub-package, and the `Twist` module from the `geometry_msgs.msg` sub-package, as well as the `numpy` and `pandas` packages. In order to implement the *publish/subscribe* communication mechanism typical in ROS, the basic *Topics* we used are `cmd_vel`, which contains information about linear and angular velocities in the Cartesian coordinate system, and `joint_states`, which contains information about the positions and velocities of each of the four wheels. The `JointState` module is used to *subscribe* to the odometry data, and the `Twist` module is used to *publish* the desired control input velocities. The control *Node subscribes* to the `joint_states` *Topic*, using a function whose input arguments are the values of linear velocity, steering angle and steering velocity (and consequently position and orientation) of the wheels obtained from the robot's sensors. These values are then used in the function defined according to the block diagram shown in Fig. 7 to calculate the control input values that are *published* to the `cmd_vel` *Topic*. In order to get data from the external measurement system OptiTrack in ROS, we started the Virtual Reality Peripheral Network (VRPN) with the `vrpn_client_ros` package. After the VRPN client is configured, the appropriate command is executed and a new *Topic* is *published* that contains information from OptiTrack, to which the system *subscribes* in order to start recording. More details on the previously mentioned ROS commands and functions can be found in the standard reference [21].

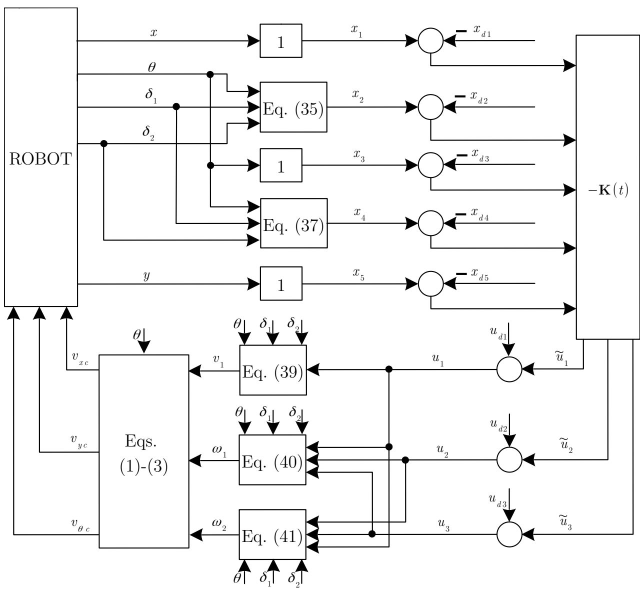

Fig. 7. Block diagram of the control strategy for implementation in ROS.

The results of the experimental evaluation of the proposed control strategy obtained both from the data collected from the measuring system of the robot and recorded by the external measurement system OptiTrack are shown in Figs. 8-12. All the desired trajectory and controller parameters used in the experiment are the same as those chosen and calculated in the simulations.

In Figs. 8-10, from the results obtained from the robot's sensors it can be seen that the robot follows the desired trajectory with an absolute error of the order of approximately 10^{-2} m, which can be considered satisfactory. However, from the measurement results obtained by the OptiTrack system the errors in the robot's odometry are evident, and this is especially emphasized in the direction of the y-axis. The reason for this lies in the mechanical structure of the robot itself, which should be improved both in terms of installing better wheels and motors with less backlash. Improvements in the design of the robot structure are planned for future research.

Fig. 11 shows the control input velocities applied to the robot, which are calculated in the feedback control loop previously explained and shown in the block diagram in Fig. 7. The wheel velocity measured directly from the

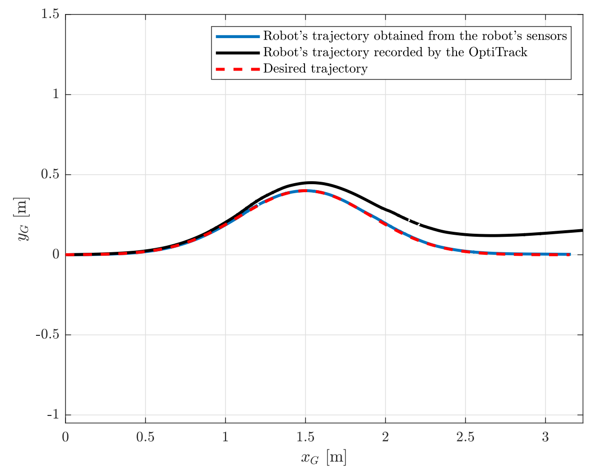

Fig. 8. Gaussian trajectory tracking experimental results in $x_G - y_G$ plane.

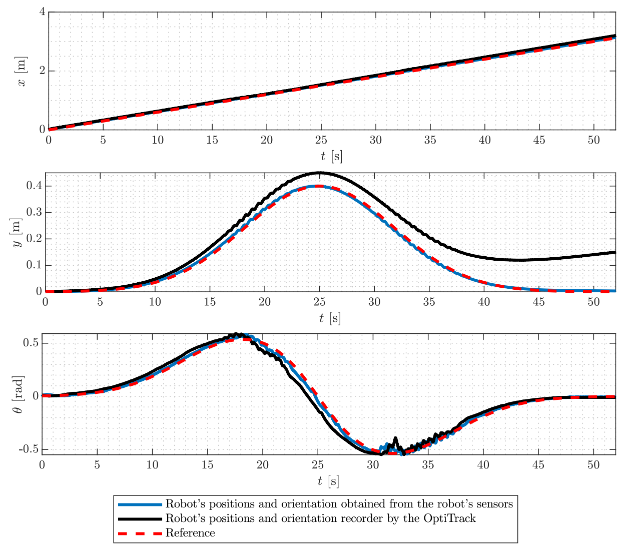

Fig. 9. Measured coordinates of the robot in the case of a Gaussian trajectory.

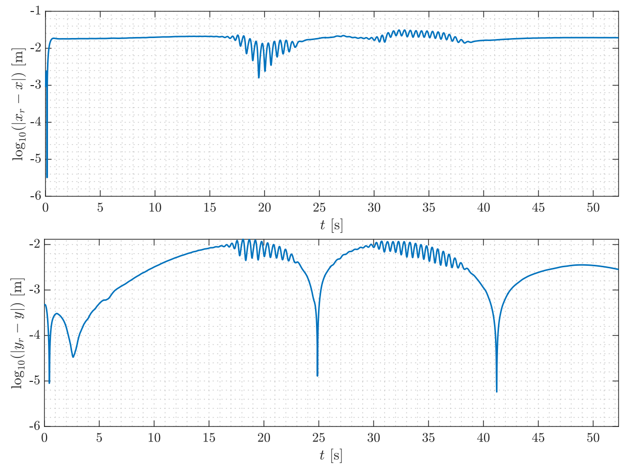

Fig. 10. Measured tracking error in the case of a Gaussian trajectory.

robot's wheel encoder is shown in Fig. 12. Figs. 11 and 12 also show the oscillations that occur at the inflection points of the Gaussian trajectory where the derivatives change direction. These oscillations can be reduced by fine-tuning the gains of the PID controller for steering and driving actuators. However, in order to achieve satisfactory tracking of the desired trajectory, the PID gains were chosen for a slightly more aggressive response.

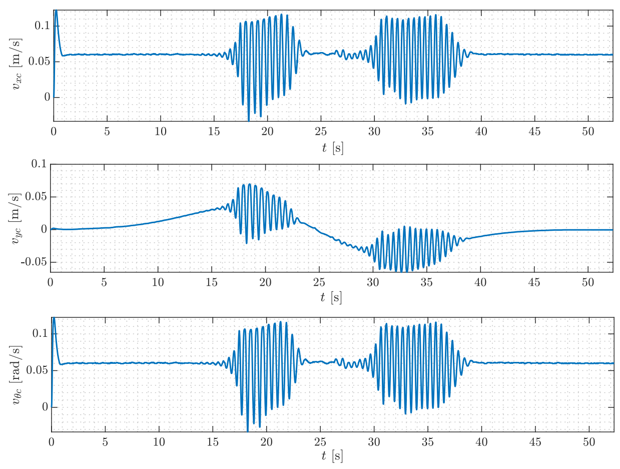

Fig. 11. Applied control input velocities to real robot in the case of a Gaussian trajectory.

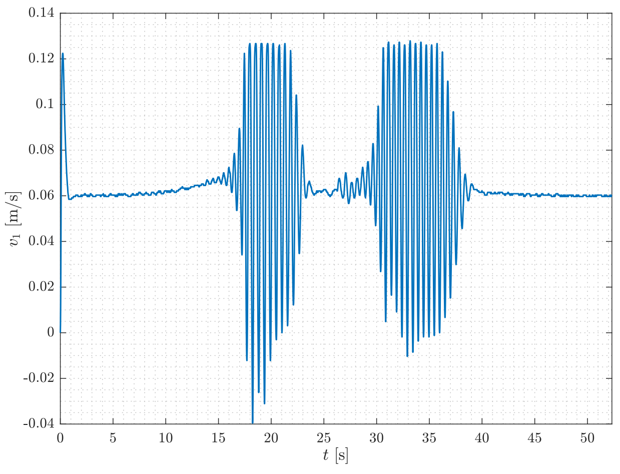

Fig. 12. Velocity obtained from the robot motor encoder in the case of a Gaussian trajectory.

VI. Conclusions

In this paper, the synthesis of a control system of a 4WIS4WID mobile robot in a closed-loop with a time-varying state controller with the aim of asymptotically following the desired trajectory has been developed and applied to a real system. The kinematic equations in the horizontal plane of the robot with independent steering and drive have been transformed into three-input, two-chain, single generator chained form for the case when each wheel rotates at the same velocity, and the front and rear wheels are steered parallel to each other and at the same steering velocity. For the purpose of controller synthesis, the chain model has been linearized in the vicinity of the desired trajectory into a system of control-oriented time-varying error equations. The corresponding Riccati equation, which gives the time-varying gain matrix of the controller, has been iteratively solved using a standard mathematical software tool.

The proposed control strategy is evaluated in simulations and verified by experiments on a real robot system. The results showed that the applied control system of the robot can follow the trajectory with satisfactory achieved errors.

Further research on this robot is aimed at solving problem of the pose estimation. Errors in odometry are caused due to the dead reckoning and inappropriate kinematic model parameters which should be calibrated. Thus our further research will be to calibrate kinematic parameters of the mobile robot, add IMU and tracking camera together with Extended Kalman Filter to get better pose estimation. OptiTrack won't be used in realistic applications, just for confirmation of chosen sensors and filtering techniques.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the project CRTA Regional Centre of Excellence for Robotic Technology, funded by the ERDF fund.

REFERENCES

[1] A. Arab, I. Hadžić, and J. Yi, "Safe predictive control of four-wheel mobile robot with independent steering and drive," in *2021 American Control Conference (ACC)*, 2021, pp. 2962–2967.

[2] X. Liu, W. Wang, X. Li, F. Liu, Z. He, Y. Yao, H. Ruan, and T. Zhang, "MPC-based high-speed trajectory tracking for 4WIS robot," *ISA Transactions*, vol. 123, pp. 413–424, 2022.

[3] Y. Xie, X. Zhang, W. Meng, S. Zheng, L. Jiang, J. Meng, and S. Wang, "Coupled fractional-order sliding mode control and obstacle avoidance of a four-wheeled steerable mobile robot," *ISA Transactions*, vol. 108, pp. 282–294, 2021.

[4] L. Jiang, S. Wang, Y. Xie, J. Meng, S. Zheng, X. Zhang, and H. Wu, "Anti-disturbance direct yaw moment control of a four-wheeled autonomous mobile robot," *IEEE Access*, vol. 8, pp. 174 654–174 666, 2020.

[5] H. Wu, Z. Li, and Z. Si, "Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control and sliding mode control," *Advances in Mechanical Engineering*, vol. 13, no. 9, pp. 1–17, 2021.

[6] P. Dai, J. Taghia, S. Lam, and J. Katupitiya, "Integration of sliding mode based steering control and PSO based drive force control for a 4WS4WD vehicle," *Autonomous Robots*, vol. 42, no. 3, pp. 553–568, 2018.

[7] P. Dai and J. Katupitiya, "Force control for path following of a 4WS4WD vehicle by the integration of PSO and SMC," *Vehicle System Dynamics*, vol. 56, no. 11, pp. 1682–1716, 2018.

[8] A. El Hajjaji, A. Ciocan, and D. Hamad, "Four wheel steering control by fuzzy approach," *Journal of Intelligent and Robotic Systems*, vol. 41, no. 2, pp. 141–156, 2005.

[9] A. E. Bryson, "Time-varying linear-quadratic control," *Journal of Optimization Theory and Applications*, vol. 100, no. 3, pp. 515–525, 1999.

[10] R. Tedrake, *Underactuated Robotics*, 2023. [Online]. Available: https://underactuated.csail.mit.edu

[11] M. Javadi, D. Harnack, P. Stocco, S. Kumar, S. Vyas, D. Pizzutilo, and F. Kirchner, "AcroMonk: A minimalist underactuated brachiating robot," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3637–3644, 2023.

[12] S. Vyas, L. Maywald, S. Kumar, M. Jankovic, A. Mueller, and F. Kirchner, "Post-capture detumble trajectory stabilization for robotic active debris removal," *Advances in Space Research*, vol. 72, no. 7, pp. 2845–2859, 2023.

[13] B. Nortmann and T. Mylvaganam, "Direct data-driven control of linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 68, no. 8, pp. 4888–4895, 2023.

[14] Y. Ling, J. Wu, W. Zhou, Y. Wang, and C. Wu, "Linear quadratic regulator method in vision-based laser beam tracking for a mobile target robot," *Robotica*, vol. 39, no. 3, p. 524–534, 2021.

[15] M.-H. Lee and T.-H. S. Li, "Kinematics, dynamics and control design of 4WIS4WID mobile robots," *The Journal of Engineering*, vol. 2015, no. 1, pp. 6–16, 2015.

[16] D. Tilbury, O. Sordalen, L. Bushnell, and S. Sastry, "A multisteering trailer system: conversion into chained form using dynamic feedback," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 807–818, 1995.

[17] G. Walsh and L. Bushnell, "Stabilization of multiple input chained form control systems," *Systems & Control Letters*, vol. 25, no. 3, pp. 227–234, 1995.

[18] A. De Luca, G. Oriolo, and C. Samson, *Feedback control of a nonholonomic car-like robot*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 171–253.

[19] M. Božić, B. Jerbić, and M. Švaco, "Development of a mobile wall-climbing robot with a hybrid adhesion system," in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2021, pp. 1136–1142.

[20] M. Božić, B. Ćaran, M. Švaco, B. Jerbić, and M. Serdar, "Mobile wall-climbing robot for ndt inspection of vertical concrete structures," in *International Symposium on Non-Destructive Testing in Civil Engineering (NDT-CE 2022)*, 2022, pp. 1–14.

[21] M. Quigley, B. Gerkey, and W. D. Smar, *Programming Robots with ROS*. Boston: O'Reilly Media, 2015.