
New Differential Flatness-Based Trajectory Tracking Method and Its Application to a Wheeled Mobile Robot Control

[Alexander Krasavin](#) , [Gaukhar Nazenova](#) , [Adema Dairbekova](#) , [Albina Kadyroldina](#) , [Tamás Haidegger](#) * , [Darya Alontseva](#) *

Posted Date: 14 December 2025

doi: 10.20944/preprints202512.1150.v1

Keywords: wheeled mobile robot (WMR); nonlinear control; trajectory tracking; differential flatness; computed torque control (CTC); smart mobility programming



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

New Differential Flatness-Based Trajectory Tracking Method and Its Application to a Wheeled Mobile Robot Control

Alexander Krasavin ¹, Gaukhar Nazenova ¹, Adema Dairbekova ¹, Albina Kadyroidina ^{1,2}, Tamás Haidegger ^{3,*} and Darya Alontseva ^{1,*}

¹ School of Digital Technologies and Artificial Intelligence, D. Serikbayev East Kazakhstan Technical University, 19 Serikbayev Street, 070010 Ust-Kamenogorsk, Kazakhstan

² Department of System Analysis and Management, L.N. Gumilyov Eurasian National University, Satpayev 2, 010008 Astana, Kazakhstan

³ University Research and Innovation Center (EKIK), Obuda University, Bécsi út 96/b, 1034, Budapest, Hungary

* Correspondence: haidegger@iob.uni-obuda.hu (T.H.); dalontseva@ektu.kz (D.A.)

Abstract

This article investigates the trajectory-tracking control of a differential-drive two-wheeled mobile robot (DDWMR) using its kinematic model. A nonlinear-to-linear transformation based on differential flatness is employed to convert the original nonlinear system into two fully decoupled linear subsystems, enabling a simple and robust controller design. Unlike conventional flatness-based methods that rely on exact feedforward linearization around a reference trajectory, the proposed approach performs plant linearization, ensuring reliable tracking across a wide range of trajectories. The resulting two-loop architecture consists of an inner nonlinear loop implementing state prolongation and static feedback, and an outer linear controller performing trajectory tracking of the linearized system. Simulation results on a circular reference trajectory demonstrate high tracking accuracy, with a maximum transient deviation of 0.28 m, a settling time of approximately 120 s, and a steady-state mean tracking error below 0.01 m. These results confirm that the plant-linearization-based framework provides superior accuracy, robustness, and practical applicability for DDWMR trajectory tracking.

Keywords: wheeled mobile robot (WMR); nonlinear control; trajectory tracking; differential flatness; computed torque control (CTC); smart mobility programming

1. Introduction

Mobile robot control has been an active research topic over the past three decades due to its wide range of practical applications, including mineral resource development, logistics and transportation, planetary exploration, agriculture, and defense, among others [1–4]. Beyond core algorithms and hardware, contemporary mobile robotics is increasingly shaped by sustainability and Environmental, Social, and Governance (ESG) requirements at the application level. IEEE 7000:2022 provides process-oriented guidance for value-based, ethically aligned engineering across the entire product lifecycle—from stakeholder value elicitation through design requirements, verification, and field operation of robotic systems [5]. In particular, the IEEE 7007:2021 provided an ontology based framework to support the responsible and sustainable development of autonomous systems [6]. The contribution of mobile robots has also been mapped to the United Nations Sustainable Development Goals (SDGs): improving the efficiency and safety of work, logistics, and manufacturing (SDGs 8, 9, 11); energy-efficient motion planning and resource-aware operation (battery usage, wear, maintenance) (SDGs 7, 12, 13); environmental and industrial safety monitoring (SDGs 3, 6, 13); and inclusive human–robot

interaction and service accessibility (SDGs 10, 11) [7]. In this study, these considerations are incorporated explicitly: the chosen speed-and-heading control scheme enables sustainability metrics (cycle energy expenditure, braking/acceleration intensity, predicted actuator wear) to be embedded in the objective functions and controller constraints, while preserving requirements' traceability—from value and risk levels to reference-signal filters and compensation channels. This strengthens the rationale for the selected compensation algorithms and multi-channel control aimed at improving the reliability and safety of WMR operation.

A distinctive feature of many mobile robotic systems is the presence of non-integrable (i.e., nonholonomic) kinematic constraints [8]. This, by itself, substantially complicates control synthesis; moreover, for nonholonomic mobile robots the necessary condition for smooth stabilization (Brockett's condition) is often not satisfied [9], which makes trajectory-tracking and path-following particularly challenging. In recent years, a wide range of methods has been proposed and applied to nonholonomic mobile robots, including sliding-mode control [10–12], backstepping [13], and nonlinear model predictive control (NMPC), as well as feedback-linearization and Lyapunov-based designs.

1.1. Differential-Flatness-Based Trajectory Control

Among trajectory-tracking methods for nonlinear control systems, differential-flatness-based approaches play a particularly important role. If the controlled plant is differentially flat, the trajectory-tracking problem can be feed-back-linearized—that is, a difficult nonlinear design reduces to a comparatively simple and well-studied linear control problem. Historically, flatness theory emerged from trajectory planning and tracking; interest in flatness-based control persists because flat systems possess properties that make trajectory generation and implementation especially effective.

Not every control system is flat. A flat system is characterized by the existence of a flat (linearizing) output such that, at any instant, the full state vector and the control inputs are algebraic functions of this output and a finite number of its time derivatives. Intuitively, by observing only the flat output one can reconstruct the state and the input without direct measurement. The corresponding mappings act as an inverse model of the nonlinear plant, which explains the affinity between flatness-based controllers and inverse-model designs; more generally, flat systems are dynamically equivalent to a trivial controllable linear system (e.g., Brunovsky form). In practice, flatness is established constructively by exhibiting a suitable flat output as a function of the state, the input, and finitely many input derivatives.

For wheeled mobile robots, several classes are known to be differentially flat under no-slip assumptions [14], or under appropriate inertia distributions for certain underactuated mobile manipulators [15]. In the differential-drive case, a common choice of flat output is the time trajectory of the midpoint of the axle connecting the drive wheels.

1.2. Differential-Flatness-Based Trajectory Control for Wheeled Mobile Robots

A number of differential-flatness-based trajectory-tracking methods have been proposed and experimentally validated for differential-drive wheeled mobile robots [16,17]. These methods share a common controller-design paradigm known as exact feedforward linearization. The approach, introduced by Hagenmeyer and Delaleau [18], rests on the following idea: for a flat system, there exists a mapping that reconstructs the current control input from the flat output and a finite number of its time derivatives. In practice, this mapping can be viewed as a physically realizable operator acting on the flat-output signal. In case the desired trajectory of the flat output is applied to the input of this operator, its output generates a control signal that under suitable regularity and invertibility conditions, drives the plant to follow the prescribed trajectory.

Importantly, even in the absence of disturbances, the resulting motion is determined by both the control signal and the initial state. When the plant's initial condition exactly matches that implied by the desired flat-output trajectory, the plant's flat output coincides with the reference. In realistic settings, initial conditions are only known within some tolerance; consequently, exact feedforward

linearization requires the initial state to lie within a neighborhood of the nominal initial condition associated with the designed reference trajectory.

The described feedforward trajectory controller can only maintain the plant's state locally in the vicinity of the desired trajectory. In the absence of a feedback loop, even arbitrarily small deviations caused by mismatch between the actual initial state and the reference, or by external disturbances—tend to grow over time, eventually reaching unacceptable levels; in other words, a purely feedforward scheme is unstable. Consequently, the feedforward path is complemented with a linear feedback structure (e.g., PD/PI/PID acting on tracking errors) to stabilize the closed loop and provide robustness around the reference trajectory.

Despite its inherent limitations, this flatness-based trajectory-tracking approach remains popular and often dominant in mobile robotics because it offers several practical advantages. Notably, an open-loop linearizing feedforward input can render the closed-loop behavior equivalent (locally) to a linear system in Brunovský canonical form, thereby reducing control of a nonlinear plant potentially with nonholonomic constraints to a standard linear design problem. In practice, feedforward linearization yields effective local regulation and tracking across a broad class of nonlinear plants; controllers devised in this framework typically maintain good performance even for intricate reference paths [17].

At the same time, exact feedforward linearization is only one member of the flatness-based toolkit, relying on a specific approximate linearization along the reference. In theory, any differentially flat system admits (coordinate) transformation and dynamic state feedback that convert it into an equivalent linear controllable system [19]; conversely, any nonlinear system that is dynamically feedback-linearizable is flat [20]. Trajectory-tracking architectures built on such (global) dynamic feedback linearization differ fundamentally from exact feedforward linearization: the linearized plant remains linear regardless of the current state, so these controllers do not rely on the actual trajectory staying arbitrarily close to the reference at all times, avoiding the principal drawback of the pure feedforward design.

The differences between exact feedforward linearization and linearization methods that convert a nonlinear flat plant into an equivalent linear controllable system stem from distinct underlying concepts. The “flat-output” viewpoint defining flat systems via the existence of a flat (linearizing) output—naturally aligns with exact feed-forward linearization, which reconstructs the input from the reference flat output and its finite derivatives. By contrast, the system-equivalence viewpoint central to flatness theory underpins dynamic feedback linearization, wherein a suitable change of coordinates combined with dynamic state feedback renders the plant equivalent to a linear system [19]. In fact, within this framework, the existence and properties of a flat output follow as consequences of equivalence to a trivial linear system; thus, defining flatness primarily via a flat output can be seen, theoretically, as Inverting cause and effect [20].

From a practical perspective, the equivalence-based theory of flat systems characterizes a class of transformations between control systems whose associated mathematical operations correspond to physically realizable signal processors. This perspective provides a constructive toolbox for building implementable linearizing controllers, beyond the locality inherent in exact feedforward designs.

Strikingly, although flatness theory for control systems crystallized in the 1990s (e.g., Fliess and co-authors; see the foundational work on the geometric notion of system equivalence [21]), practical trajectory-tracking methods for nonlinear flat systems predate these theoretical developments. A prominent example is CTC, also known as inverse-dynamics control, originally developed for serial robotic manipulators. As shown later in this paper, CTC realizes a static-feedback linearization that maps a non-linear plant into a trivial linear controllable form (Brunovský canonical form), thereby enabling standard linear tracking design with excellent performance and comparatively simple implementation. Notably, nonlinear flat systems such as the kinematic and dynamic models of a differential-drive wheeled mobile robot share an important structural feature with serial

manipulators: the flat output coincides with a component of the state vector, which makes the linearizing construction particularly transparent.

Sun et al. conducted a comparative study of one of the most widely used mobile-robot control techniques—Model Predictive Control (MPC)—and differential-flatness-based control for quadrotor agile flight, demonstrating that flatness-based methods provide superior tracking performance [18,22]. Ongoing research continues to explore various differential-flatness-based control techniques, including MPC-based approaches [15,16], optimal control formulations [23], and hybrid combinations of multiple techniques [24,25].

The trajectory-tracking method proposed in this paper is closely related to CTC. The linearization procedure employed herein is conceptually analogous to static-feedback linearization used in MPC and in modern robust and adaptive control frameworks for manipulators and manipulator-like systems [26–28]. The authors have previously applied inversion-based modelling techniques [29], as well as trajectory tracking based on endogenous linearization of the dynamic differential-drive WMR model [30], and differential–geometric methods for manipulator control [31–33]. Building on these results, the authors aim to generalize the CTC trajectory-control scheme so that it can be applied to differentially flat systems such as wheeled mobile robots, while preserving its simplicity and high-performance properties.

The goal of this study is to linearize the kinematic model of a DDWMR, where linearization is understood as a physically realizable transformation of the nonlinear plant into an equivalent linear controllable system (i.e., a trivial linear system), rather than an approximate linearization in a neighborhood of a reference trajectory, as in exact feedforward linearization. Of particular interest is assessing whether the CTC-inspired linearization enables DDWMR motion to be controlled analogously to the motion of a point in the plane, which, in principle, allows achieving the maximal theoretically attainable trajectory-tracking accuracy.

1.3. Relevance to Sustainable Development Goal 11 (Sustainable Cities and Communities)

Smart and sustainable urban mobility refers to “a modern approach to urban transport that aims to improve the flow of people and goods, reduce congestion and environmental impact, while increasing efficiency, accessibility, and user convenience” (<https://www.sciencedirect.com/topics/social-sciences/smart-urban-mobility>). It builds on advanced communication systems [36] and transparent and highly customizable control solution for the mobile robots within the smart city ecosystem. Regarding the latter, the proposed linearization and trajectory-control framework contributes to SDG 11 by advancing autonomous mobile robotic systems that can support safer, more efficient, and more sustainable urban environments. Differential-drive WMRs equipped with energy-aware control architectures are suitable for logistics, inspection, mobility, and monitoring tasks in smart-city infrastructures. By improving motion reliability, resource efficiency, and operational safety, the presented method aligns with the development of resilient, technology-integrated, and sustainable cities. Similar approaches have been developed for smart urban aerial mobility systems [37].

2. Problem Statement

2.1. Hardware-Implementable Transformations of Control Systems and System Equivalence

The trajectory-tracking method proposed in this work for a differential-drive wheeled mobile robot is based on a hardware-implementable transformation of the robot’s kinematic model that converts it into an equivalent trivial linear control system. To clarify the precise meaning of the terminology used, let us consider a simple example of a hardware-implementable transformation of a control system \mathbf{B} into an equivalent control system \mathbf{A} . Let \mathbf{A} be a control system with state space \mathcal{X} , control space \mathcal{U} , and dynamics given by (1), and let \mathbf{B} be a control system with state space \mathcal{Y} , control space \mathcal{W} , and dynamics given by (2).

$$\dot{x} = f(x, u) \quad x \in \mathcal{X}, u \in \mathcal{U} \quad (1)$$

$$\dot{y} = g(y, w) \quad y \in \mathcal{Y}, w \in \mathcal{W} \quad (2)$$

Let the state spaces of systems A and B equivalent to each other in the geometric sense of the term; that is, suppose there exists a continuous, invertible mapping $\mu: \mathcal{X} \rightarrow \mathcal{Y}$, that establishes a one-to-one correspondence between the points of the state spaces \mathcal{X} and \mathcal{Y} . The control spaces \mathcal{U} and \mathcal{W} of these systems are likewise assumed to be mutually equivalent. Consider now the scheme of a hardware-implementable transformation of system B , shown in Figure 1. The block ρ with inputs x and u is a signal processor that physically implements a prescribed algebraic mapping $\rho: \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{W}$, while the block μ^{-1} represents the hardware implementation of the inverse mapping $\mu: \mathcal{X} \rightarrow \mathcal{Y}$.

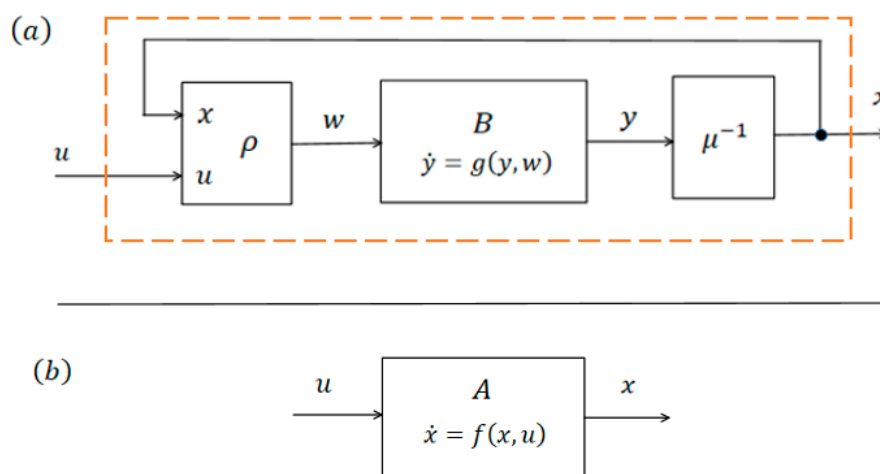


Figure 1. A hardware-implementable transformation of system B into an equivalent control system A via static state feedback.

Under certain conditions (see Note 1) imposed on the mappings μ and ρ , which define the hardware-implementable transformation, as well as on the functions f and g , that specify the dynamics of control systems A and B , the block diagram inside the “black box” with input u and output x becomes equivalent (in the usual practical sense of the term) to control system A . More precisely, if the initial state of control system B $y_0 \in \mathcal{Y}$ related to the initial state $x_0 \in \mathcal{X}$ by the relation $y_0 = \mu(x_0)$, to the system input $u(t) \in \mathcal{U}$ defined on the time interval $I = [0, T]$, $T > 0$, is applied to the input of system A and to the input of the transformed system B , then the state trajectories of the outputs of these systems, denoted in Figure 1 by x , will coincide at any moment of time $t \in I$. Thus, the object inside the “black box” behaves exactly as the control object described by the mathematical model of system A , and we may say that such a hardware-implementable transformation converts control system B into control system A . Throughout the remainder of this work, when referring to the equivalence of control systems, we shall mean a precisely defined mathematical concept associated with the sets of trajectories of control systems. Recall that a trajectory of a control system defined by the state space \mathcal{X} , the control space \mathcal{U} and the dynamical Equation (1), is an application $t \mapsto \mathcal{X} \times \mathcal{U}$, whose components satisfy the system’s dynamical equation.

To denote trajectories of control systems and their components, we will use the notation $\gamma(t) = (x_\gamma(t), u_\gamma(t))$, implying that the components of the mapping $\gamma: I \rightarrow \mathcal{X} \times \mathcal{U}$ $x_\gamma: I \rightarrow \mathcal{X}$ and $u_\gamma: I \rightarrow \mathcal{U}$, called respectively the state trajectory and the control-input trajectory, satisfy condition (3), where I is a finite or infinite time interval.

$$\forall t \in I \quad \dot{x}_\gamma(t) = f(x_\gamma(t), u_\gamma(t)) \quad (3)$$

One may observe that when referring to the equivalence of two control systems, we are in fact speaking about the equivalence of the corresponding sets of trajectories generated by these systems.

Note 1. Under Conditions 1 and 2, the control systems A and B introduced in the example at the beginning of this section are equivalent.

Condition 1. For any point $p \in \mathcal{X}$ the mapping $\rho_p: \mathcal{U} \rightarrow \mathcal{W}$ defined by formula (4), is a diffeomorphism. This requirement ensures the existence of a smooth, bijective correspondence between the admissible control inputs of the two systems.

$$\forall q \in \mathcal{U} \quad \rho_p(q) = \rho((p, q)) \quad (p, q) \in \mathcal{X} \times \mathcal{U} \quad (4)$$

Condition 2. For the functions $f(x, u)$ and $g(y, w)$ and for the mappings under consideration, relation (5) holds.

$$\forall p = (x, u) \in \mathcal{X} \times \mathcal{U} \quad d\mu_x(f(x, u)) = g(\mu(x), \rho(x, u)) \quad (5)$$

When these conditions are satisfied, the trajectories of control systems A and B are related by the diffeomorphic mapping $h: \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{Y} \times \mathcal{W}$, defined by expression (6).

$$\forall p = (x, u) \in \mathcal{X} \times \mathcal{U} \quad h(p) = (\mu(x), \rho(x, u)) \quad (6)$$

Moreover, γ is an arbitrary trajectory of system A and σ is the corresponding trajectory of system B , then for any moment of time the equalities given in $h(\gamma(t)) = \sigma(t)$ and $h^{-1}(\sigma(t)) = \gamma(t)$ (see Figure 2).

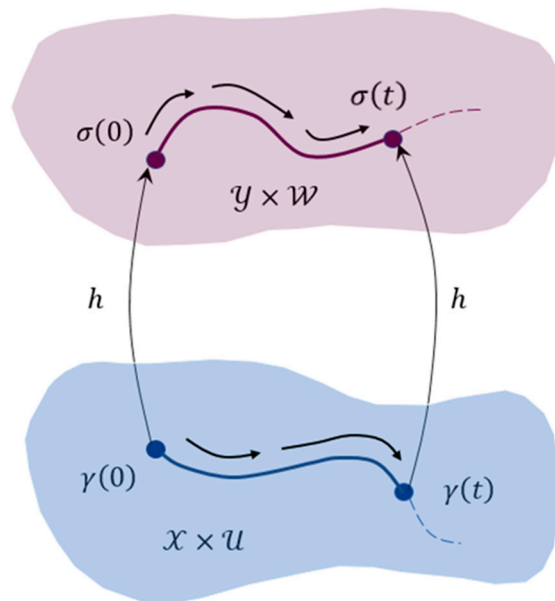


Figure 2. Correspondence between the trajectories of equivalent control systems A and B , defined by the diffeomorphic mapping $h: \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{Y} \times \mathcal{W}$.

There exist several different approaches defining the notion of equivalence between control systems. In this article, we adopt the geometric approach first introduced in [21], which is based on analogies between control systems and continuous dynamical systems. Both control systems and continuous dynamical systems can be viewed as geometric objects of the same type—so-called systems. A system is defined as a pair $\langle \mathcal{X}, F \rangle$, where \mathcal{X} – is a smooth manifold and F is a tangent vector field on \mathcal{X} . It is worth noting that the dynamical equation of system (1) corresponds to the infinite system of differential equations given in (7).

$$\dot{y} = F^{\infty}(y), \quad (7)$$

where $y = [x, u, \dot{u}, \ddot{u}, \dots]^T$ and $F^\infty(y) = [f(x, u), \dot{u}, \ddot{u}, \dots]^T$. Thus, F may be regarded as a tangent vector field defined on the infinite-dimensional manifold associated with $\mathcal{E}^\infty = \mathcal{X} \times \mathcal{U} \times \mathbb{R}^m \times \mathbb{R}^m \times \dots$, and the control system under consideration may be viewed as the infinite-dimensional dynamical system corresponding to $\langle \mathcal{E}^\infty, F^\infty \rangle$. In [21], an approach was first introduced in which control systems are defined as infinite-dimensional systems $\langle \mathcal{E}^\infty, F^\infty \rangle$, equipped with a Fréchet topology, and a definition was given for an endogenous transformation that relates equivalent control systems represented as such infinite-dimensional systems. Naturally, from a practical standpoint, the cases of greatest interest are those in which a nonlinear control system is equivalent to a linear one. A control system that is equivalent to a so-called trivial linear control system (the definition of this class of linear systems is given below) is referred to as a flat control system.

Definition 1. A linear control system with control space $\mathcal{U} = \mathbb{R}^n$ and state space $\mathcal{X} = \underbrace{\mathbb{R}^n \times \mathbb{R}^n \times \dots \times \mathbb{R}^n}_k$, whose state vector $x \in \mathcal{X}$ is of the form $x = (x_1, x_2, \dots, x_k) \forall j \in \{1, 2, \dots, k\} x_j \in \mathbb{R}^n$, and whose dynamics are given by (8), where $u \in \mathcal{U}$ is called a trivial linear control system.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_k = u \end{cases} \quad (8)$$

We denote a trivial linear control system by $T_{n,k}$, where the integer index $n \geq 1$ represents the dimension of the control space, and the index $k \geq 1$ denotes the number of components of the trivial linear system x_1, x_2, \dots, x_k . In this article, we propose a linearisation procedure for the kinematic model of a differential-drive wheeled mobile robot, by which the robot's kinematic model is transformed into the trivial control system $T_{2,2}$. The linear control system $T_{2,2}$ admits a physical interpretation as the dynamics of a material point moving in the plane under the action of a planar force vector. Thus, the linearisation method described in this work reduces the control problem for a highly nonlinear plant to a simple linear control problem with an efficient solution. The proposed linearisation approach is carried out in two stages, which are presented in Sections 3 and 4. Section 5 provides a description of the trajectory-tracking control system for the linearised kinematic model of the DDWMR.

2.2. Kinematic Model of a Differential-Drive Wheeled Mobile Robot and Class I Transformations

Consider the kinematic model of a differential-drive wheeled mobile robot moving on a horizontal plane. As state variables fully describing the robot's position and orientation, it is convenient to choose the coordinates (x_c, y_c) of point C in a fixed "world" Cartesian coordinate frame XY together with the robot's heading angle θ (see Figure 3). Let v_c denote the velocity vector of point C and let v_L and v_R denote the velocity vectors of the centers of the left wheel (point L) and the right wheel (point R), respectively.

We may regard points L and R as the endpoints of a rigid, thin rod (the imaginary "rear axle" of the robot), whose midpoint is point C . All three points therefore belong to the same rigid body—the robot chassis. As is well known [34,35], for the velocity vectors v_A and v_B of any two points A and B of a rigid body, the relation given in (9) holds, where Ω is the angular velocity vector of the body.

$$v_A - v_B = \Omega \times r_{AB} \quad (9)$$

The robot's angular velocity vector Ω is perpendicular to the floor plane. Accordingly, if we denote by \mathbf{p} the unit vector normal to the plane, then the angular velocity vector may be written as $\Omega = \omega \mathbf{p}$. In what follows, when referring to the angular velocity of the robot, we shall mean the scalar quantity ω , whose sign determines the direction of rotation of the robot's chassis in the horizontal plane. In the absence of lateral slip of the drive wheels, the vectors v_L and v_R are perpendicular to the axle LR . As follows from Equation (9), applied to points L and R , the projections of v_L and v_R onto the axis Y^* satisfy equality (10) at any moment of time.

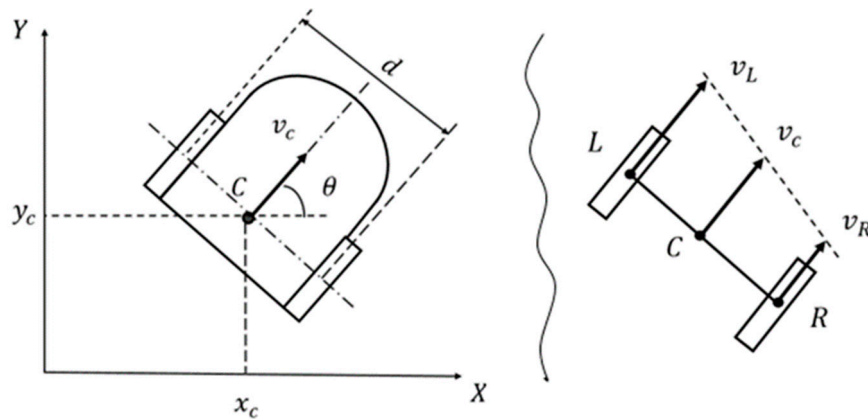


Figure 3. Geometric parameters of the robot and its kinematic variables. Point C (the nominal geometric center of the robot) lies at the intersection of the robot's longitudinal axis of symmetry and the axis connecting the centers of the drive wheels.

$$\mathbf{v}_L - \mathbf{v}_R = d\boldsymbol{\omega} \quad (10)$$

Using Equation (10), it is straightforward to show that the vector \mathbf{v}_c is parallel to the vectors \mathbf{v}_L and \mathbf{v}_R . Consequently, the velocity of point C can be written as $\mathbf{v}_c = v_c \mathbf{n}$, where $\mathbf{n} = (\cos \theta, \sin \theta)$ is the unit vector aligned with the robot's heading direction. Furthermore, from the general relation (9) applied to points R and C equality (11) follows.

$$\mathbf{v}_c - \mathbf{v}_R = \left(\frac{d}{2}\right) \boldsymbol{\omega} \quad (11)$$

Equations (10) and (11) together imply Equations (12).

$$\boldsymbol{\omega} = \frac{1}{d}(\mathbf{v}_L - \mathbf{v}_R) \quad \mathbf{v}_c = \frac{\mathbf{v}_L + \mathbf{v}_R}{2} \quad (12)$$

The above statement regarding the direction of the vector \mathbf{v}_c together with the fact that the time derivative of the heading angle, $\dot{\theta}$ is precisely the angular velocity of the robot's chassis, leads to the system of differential Equations (13).

$$\begin{cases} \dot{x}_c = v_c \cos \theta \\ \dot{y}_c = v_c \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (13)$$

Equations (13) can be regarded as the dynamical equations of a control system with state space $\mathcal{Y} = \mathbb{R}^2 \times S^1$ and control space $\mathcal{W} = \mathbb{R}^2$, representing the simplest kinematic model of a differential-drive wheeled mobile robot. The state vector of this model, $q \in \mathcal{Y}$, describing the robot's position and orientation on the plane, is given by $q = ([x_c, y_c]^T, \theta)$, and the control vector $w \in \mathcal{W}$ is $w = [v_c, \omega]^T$. In what follows, we refer to the above-defined control system as kinematic model A . This simplest DDWMR model is convenient for analytical manipulations; however, for computer simulations we employed a more detailed and realistic kinematic model, denoted as model B , which is described below. In the absence of longitudinal slip, the velocities of the wheel centers v_L and v_R satisfy relations (14), where r is the radius of the robot's wheels, and $\dot{\varphi}_R$ and $\dot{\varphi}_L$ are the angular velocities of the right and left drive wheels, respectively (with φ_R and φ_L denoting the wheel shaft rotation angles, which may be measured using incremental rotary encoders).

$$\mathbf{v}_L = r\dot{\varphi}_L \quad \mathbf{v}_R = r\dot{\varphi}_R \quad (14)$$

Taking Formulas (14) and (12) into account, system (13) can be rewritten in the form given by (15).

$$\begin{cases} \dot{x}_c = \frac{r}{2}(\dot{\phi}_L + \dot{\phi}_R) \cos \theta \\ \dot{y}_c = \frac{r}{2}(\dot{\phi}_L + \dot{\phi}_R) \sin \theta \\ \dot{\theta} = \frac{r}{d}(\dot{\phi}_L - \dot{\phi}_R) \end{cases} \quad (15)$$

We may regard system (15) as the representation of the coordinate-wise projections of the general dynamical equation of a control system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ for the state vector $\mathbf{x}(t) = [x_c(t), y_c(t), \theta(t)]^T$ and the control vector $\mathbf{u}(t) = [\dot{\phi}_R(t), \dot{\phi}_L(t)]^T$. The kinematic model of the robot, denoted as model *A* is transformed into the “advanced” robot model *B* by means of the control-signal transformation specified by formulas (16).

$$\dot{\phi}_L(v_c, \omega) = \frac{1}{r} \left(v_c + \frac{1}{2} d \omega \right) \quad \dot{\phi}_R(v_c, \omega) = \frac{1}{r} \left(v_c - \frac{1}{2} d \omega \right) \quad (16)$$

The general scheme of the transformation from system *A* to system *B* is shown in Figure 4.

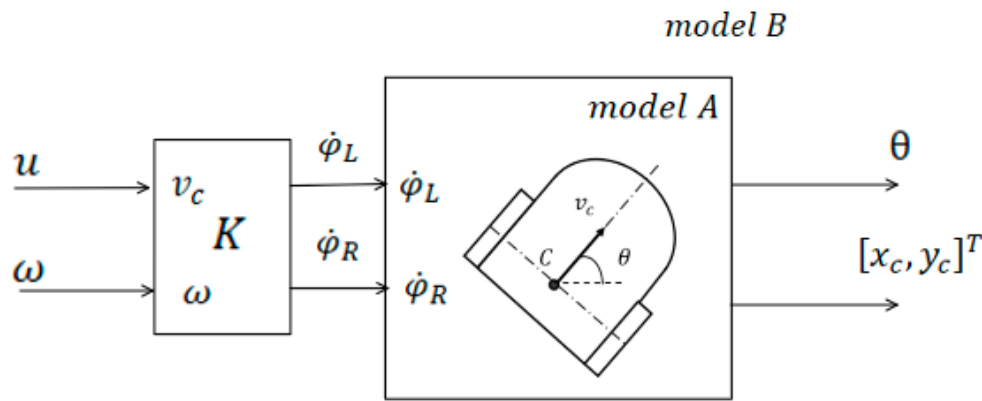


Figure 4. Transformation of the differential-drive wheeled mobile robot kinematic model *A* into model *B*.

We transform the kinematic robot model *A* into a control system *Q* with state space $\mathcal{X} = \mathbb{R}^2 \times M$, where $M = \mathbb{R} \times S^1$ and control space $\mathcal{U} = \mathbb{R}^2$ whose state vector has the form $x = (p, (u, \theta))$, and whose dynamics are given by (17).

$$\begin{cases} \dot{p}_x = u \cos \theta \\ \dot{p}_y = u \sin \theta \\ \dot{u} = a_t \\ \dot{\theta} = \omega \\ \dots \end{cases} \quad (17)$$

System *Q* is obtained by transforming the control signal $u = [u_1, u_2]^T \in \mathcal{U}$ of the robot's kinematic model (where u_1, u_2 are aliases for v_c, ω) into the control signal $w = [w_1, w_2]^T$ of system *Q* (where w_1, w_2 are aliases for a_t, ω), analytically defined by equations (18).

$$w_1 = \dot{u}_1 \quad w_2 = u_2 \quad (18)$$

This transformation is an example of a Class *I* control-system transformation, the definition of which is given below. Recall that a control system $\dot{x} = f(x, u)$, $(x, u) \in \mathcal{X} \times \mathcal{U}$ is equivalent to the system $\dot{x} = f(x, u)$, $\dot{u} = v$, since both systems are described by the same infinite-dimensional system $\langle X \times \mathcal{U} \times \mathbb{R}_m^{\infty}, F \rangle$, where the tangent vector field is defined by $F(x, u, u^1, \dots) = (f(x, u), u^1, u^2, \dots)$. Let the control signal of a control system be m -dimensional and written as $u = [u_1, u_2, \dots, u_m]^T$, and let L_j be a linear operator whose action on the signal $u(t)$ is defined by equation (19).

$$L_j u(t) = [u_1(t), \dots, \dot{u}_j(t), \dots, u_m(t)]^T. \quad (19)$$

It is evident that the control system $\dot{x} = f(x, u)$, $L_j(u) = v$ is equivalent to the original control system $\dot{x} = f(x, u)$, $(x, u) \in \mathcal{X} \times \mathcal{U}$. We shall refer to such transformations of control systems as Class I transformations.

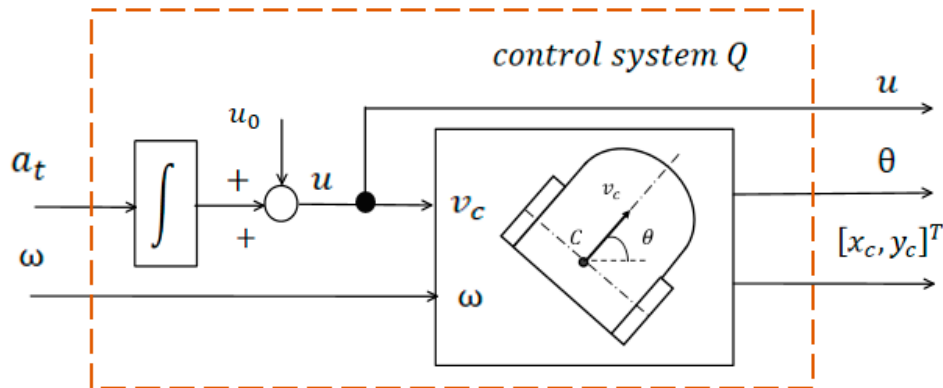


Figure 5. Hardware-implementable scheme for transforming the DDWMR kinematic model A into the control system Q .

As will be shown in the next section, the control system Q can be interpreted as a kinematic control system governing the motion of a point in the plane. Accordingly, if we construct a physically implementable transformation that maps this system into the trivial control system $T_{2,2}$, which describes force-based control of a material point moving in the plane, then we will be able to control the motion of the wheeled mobile robot in the same manner as controlling the motion of a point in the plane.

2.3. Linearisation of the DDWMR Kinematic Model

The problem of linearising the control system Q , defined in the previous section, reduces—as will be shown below—to the linearisation of a simple control system on the smooth manifold $S^1 \times \mathbb{R}$. Interestingly, the solution to this problem, presented below, leads directly to a hardware-implementable scheme for linearising the kinematic model of the wheeled mobile robot. First, observe that the kinematic equation $\dot{\mathbf{v}} = \mathbf{a}$, which describes the motion of a point in the plane, may be viewed as the dynamical equation of the trivial linear control system A with state space $\mathcal{X} = \mathbb{R}^2$ and control space $\mathcal{U} = \mathbb{R}^2$. We assume that the linear space \mathcal{X} is equipped with an orthonormal basis $\{\mathbf{i}, \mathbf{j}\}$, $\bar{\mathbf{o}}$ whose basis vectors correspond to the coordinate axes of the Cartesian frame XY in which the point moves. Any nonzero vector $\mathbf{v} \in \mathcal{X}$ can be represented in the form (20).

$$\mathbf{v} = \mathbf{un}(\theta), \quad (20)$$

where $u \in \mathbb{R}$, $\theta \in S^1$ and the mapping $\mathbf{n}: S^1 \rightarrow \mathcal{X}$ is defined for any $\theta \in S^1$ by $\mathbf{n}(\theta) = [\cos \theta, \sin \theta]^T$, with θ being the angle between the vector \mathbf{v} and \mathbf{i} . Clearly, for any nonzero vector there exist two possible representations of this form, given by expressions (21) and (22).

$$\mathbf{u}(\mathbf{v}_x, \mathbf{v}_y) = \sqrt{\mathbf{v}_x^2 + \mathbf{v}_y^2} \quad \theta = \text{atan2}(\mathbf{v}_y, \mathbf{v}_x) \quad (21)$$

$$\mathbf{u}(\mathbf{v}_x, \mathbf{v}_y) = -\sqrt{\mathbf{v}_x^2 + \mathbf{v}_y^2} \quad \theta = \text{atan2}(\mathbf{v}_y, \mathbf{v}_x) + \pi \quad (22)$$

Formulas (21) and (22) can be viewed as two coordinate representations of mappings from the Euclidean plane \mathcal{X} onto the smooth manifold $\mathcal{Y} = S^1 \times \mathbb{R}$, which we denote by $\mu^+: \mathcal{X} \rightarrow \mathcal{Y}$ and $\mu^-: \mathcal{X} \rightarrow \mathcal{Y}$. For the coordinates of a vector $\mathbf{v} = [\mathbf{v}_x, \mathbf{v}_y]^T$ expressed in the form $\mathbf{un}(\theta)$, the identities (4) clearly hold. These identities may be interpreted as defining the mapping $\pi: \mathcal{Y} \rightarrow \mathcal{X}$, which serves

as the inverse (on their respective domains of definition) to the mappings $\mu^+ : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mu^- : \mathcal{X} \rightarrow \mathcal{Y}$.

$$v_x = u \cos \theta \quad v_y = u \sin \theta \quad (23)$$

Note that the embedding of the smooth manifold $S^1 \times \mathbb{R}$ into the three-dimensional Euclidean space is represented by the lateral surface of an infinite cylinder, on which one can introduce coordinates (u, θ) , as illustrated in Figure 6. By expressing vectors of the Euclidean plane in the form (20), we effectively identify each nonzero vector with a point on the surface of this infinite cylinder. Consequently, any state trajectory γ of system A (with the exception of the degenerate trajectory of a stationary point) corresponds to two curves $\mu^+(\gamma)$ and $\mu^-(\gamma)$ on the cylinder surface. Let us define a control system B with state space $\mathcal{Y} = S^1 \times \mathbb{R}$, control space $\mathcal{W} = \mathbb{R}^2$, and dynamics $\dot{y} = w$. As will be shown below, the geometric relationship between trajectories in the phase spaces $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = S^1 \times \mathbb{R}$, determines a correspondence between the trajectories of systems A and B as a whole, thereby inducing a transformation from system B to system A . This transformation is precisely what we refer to as the linearisation of system B . Since system Q constructed on top of system B the physically realisable hardware scheme that linearises system B simultaneously linearises system Q . The smooth manifold $S^1 \times \mathbb{R}$ is not homeomorphic to the plane \mathbb{R}^2 , in particular, the cylinder cannot be continuously deformed into the Euclidean plane. For this reason alone, there cannot exist a one-to-one correspondence between the full trajectory sets of systems A and B . However, under the kinematic interpretation of system B it becomes clear that not all trajectories of system B are physically realisable. Therefore, it is meaningful to speak of an equivalence between the trajectory set of system A and the set of physically realisable trajectories of system B . To analyse the geometric relationship between the spaces \mathcal{X} and \mathcal{Y} , we use the identification of vectors of \mathcal{X} with points on the plane \mathbb{R}^2 endowed with the Cartesian frame XY . Define the region \mathcal{Y}^+ by the condition $p = (\theta_p, u_p) \in \mathcal{Y}^+ \Rightarrow u_p > 0$. The mapping $\pi : \mathcal{Y} \rightarrow \mathcal{X}$ may then be visualised as the projection of an infinite cylindrical surface, cut by the plane XY , perpendicular to the cylinder axis U , as shown in Figure 7. Each of the regions \mathcal{Y}^+ and \mathcal{Y}^- , corresponding respectively to the upper and lower halves of the cylinder, is mapped onto the plane with the origin O with coordinates $(0,0)$ removed. The intersection of the cylinder with the plane, which forms a one-dimensional manifold N homeomorphic to S^1 , is mapped to the single point O .

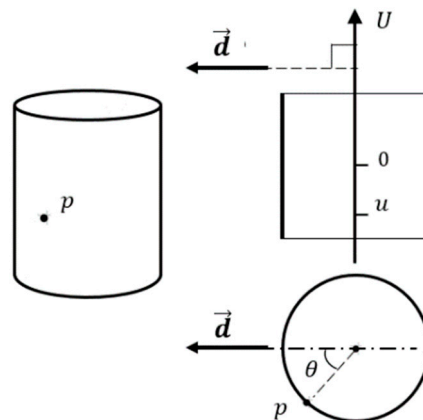


Figure 6. Geometric interpretation of the coordinates (u, θ) on the manifold $S^1 \times \mathbb{R}$.

Thus, mapping $\pi : \mathcal{Y} \rightarrow \mathcal{X}$ ends each point of the manifold $\mathcal{Y} = \mathcal{Y}^+ \cup N \cup \mathcal{Y}^-$ to a corresponding point on the Euclidean plane $\mathcal{X} = \mathbb{R}^2$. Let $\mathcal{X}^* = \mathcal{X} \setminus \{O\}$ denote the subset of the plane that is the image of regions \mathcal{Y}^+ and \mathcal{Y}^- under mapping $\pi : \mathcal{Y} \rightarrow \mathcal{X}$. Then the mappings $\mu^+ : \mathcal{X}^* \rightarrow \mathcal{Y}^+$ and $\mu^- : \mathcal{X}^* \rightarrow \mathcal{Y}^-$ are defined by equations (1) and (2) respectively. Mapping $\mu^+ : \mathcal{X}^* \rightarrow \mathcal{Y}^+$ is a diffeomorphism, and $\mu^{+^{-1}} = \pi|_{\mathcal{Y}^+}$, as $\pi|_{\mathcal{Y}^+}$ denotes the restriction of the mapping π to the region \mathcal{Y}^+ . Similarly, the mappings μ^- and $\pi|_{\mathcal{Y}^-}$ define a diffeomorphism between the regions \mathcal{X}^* and \mathcal{Y}^- . Let $\gamma : I \rightarrow \mathcal{X}$ be a parametrically defined curve on the plane \mathcal{X} , that does not pass through the

point O . Let us denote γ^+ the image of the curve γ under the diffeomorphism μ^+ , this image is a parametrically defined curve on the manifold \mathcal{Y} given by relation $\gamma^+: I \rightarrow \mathcal{Y}$. Likewise, let $\forall t \in I \gamma^+(t) = \mu^+(t)$, and as $\gamma^-: I \rightarrow \mathcal{Y}$ denote the image $\mu^-(\gamma)$ of the curve γ under the mapping $\mu^-: \mathcal{X}^* \rightarrow \mathcal{Y}^-$. The images of the curves under the mapping π are curves on the plane \mathcal{X} . Moreover, these curves are related by a symmetry transformation defined by an automorphism $\varphi: \mathcal{Y} \rightarrow \mathcal{Y}$, introduced below. Specifically, for every $t \in I$ the following relations $\varphi(\gamma^+(t)) = \gamma^-(t)$, and $\varphi(\gamma^-(t)) = \gamma^+(t)$ hold (see Figure 7). The automorphism φ is defined by formulas (24), which specify the correspondence between the coordinates (θ_p, u_p) of an arbitrary point $p \in \mathcal{Y}$ and the coordinates (θ_q, u_q) of its symmetric point $q = \varphi(p)$.

$$\theta_q = \theta_p + \pi \quad u_q = -u_p \quad (24)$$

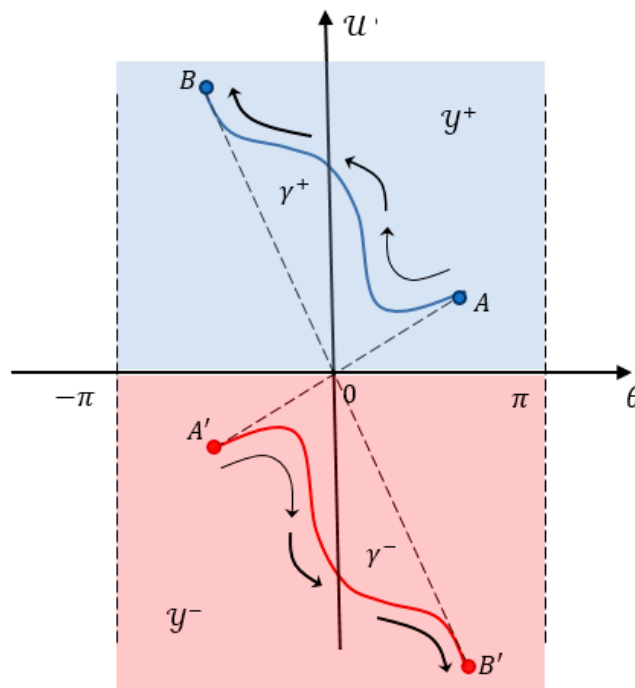


Figure 7. The symmetry relation on the phase space \mathcal{Y} induced by the automorphism $\varphi: \mathcal{Y} \rightarrow \mathcal{Y}$.

To establish the correspondence between the trajectories of control systems A and B , we represent the trajectory in the state space of system A $\mathbf{v}_\gamma(t)$ in the form (25).

$$\mathbf{v}_\gamma(t) = \mathbf{u}(t)\mathbf{n}(\theta(t)) \quad (25)$$

Differentiating both sides of equation (25) with respect to time yields Equation (26).

$$\mathbf{a}_\gamma(t) = \dot{\mathbf{u}}(t)\mathbf{n}(\theta(t)) + \mathbf{u}(t)\dot{\theta}(t)\mathbf{p}(\theta(t)), \quad (26)$$

where $\mathbf{a}_\gamma(t)$ is the control-input trajectory generating the state-space trajectory $\mathbf{v}_\gamma(t)$, and the function $\mathbf{p}: S^1 \rightarrow \mathbb{R}^2$ is defined by Equation (27).

$$\forall \theta \in S^1 \quad \mathbf{p}(\theta) = [-\sin(\theta), \cos(\theta)]^T \quad (27)$$

Obviously, for any $\theta \in S^1$ the vector $\mathbf{p}(\theta)$ is a unit vector orthogonal to the unit vector $\mathbf{n}(\theta)$. By taking the scalar product of both sides of equation (26) with the vector $\mathbf{n}(\theta(t))$ we obtain equation (28).

$$\dot{\mathbf{u}}(t) = \mathbf{a}_t(\theta(t), \mathbf{u}(t), \mathbf{a}_\gamma(t)) \quad (28)$$

where $\mathbf{a}_t: S^1 \times \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined by (29).

$$\mathbf{a}_t = T(\boldsymbol{\pi}(\boldsymbol{\theta}, \mathbf{u}), \mathbf{a}) \quad (29)$$

The function $T: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, defined by equation (31), computes the projection of a vector $\mathbf{v}, \mathbf{a} \in \mathbb{R}^2$ onto the direction specified by a vector \mathbf{v} . Accordingly, \mathbf{a}_t represents the tangential component of the acceleration vector \mathbf{a}_y , and equation (29) admits the physical interpretation of being the projection of the vector differential equation $\dot{\mathbf{v}}_y = \mathbf{a}_y$ onto the moving orthonormal basis on the plane \mathbb{R}^2 , generated by the vectors $\mathbf{n}(\boldsymbol{\theta}(t))$ and $\mathbf{p}(\boldsymbol{\theta}(t))$.

$$\forall \mathbf{v}, \mathbf{a} \in \mathbb{R}^2 \quad T(\mathbf{v}, \mathbf{a}) = \begin{cases} \frac{1}{|\mathbf{v}|}(\mathbf{v} \cdot \mathbf{a}), & \text{if } |\mathbf{v}| > 0 \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (30)$$

Similarly, by taking the scalar product of both sides of equation (30) $\mathbf{p}(\boldsymbol{\theta}(t))$ with the vector $\mathbf{u}(t)\dot{\boldsymbol{\theta}} = \mathbf{p}(t) \cdot \mathbf{a}_y(t)$ which can be conveniently rewritten in the form (31).

$$\dot{\boldsymbol{\theta}}(t) = \boldsymbol{\omega}(\boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{a}_y(t)), \quad (31)$$

where the function $\boldsymbol{\omega}: S^1 \times \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined by equation (32).

$$\boldsymbol{\omega}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{a}) = \frac{1}{\mathbf{u}}(\mathbf{p}(\boldsymbol{\pi}(\boldsymbol{\theta}, \mathbf{u})) \cdot \mathbf{a}) \quad (32)$$

The function $\boldsymbol{\omega}$ computes the instantaneous angular velocity of rotation of the velocity vector \mathbf{v}_y . It is straightforward to show that $\boldsymbol{\omega}$ can equivalently be defined by expression (33), and this alternative definition will be used throughout the remainder of the paper.

$$\boldsymbol{\omega}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{a}) = \boldsymbol{\Omega}(\boldsymbol{\pi}(\boldsymbol{\theta}, \mathbf{u}), \mathbf{a}), \quad (33)$$

where the function $\boldsymbol{\Omega}: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, defined by formulas (34), computes the instantaneous angular velocity $\boldsymbol{\omega}$ of the velocity vector of a point moving in the plane, based on its instantaneous velocity \mathbf{v} and acceleration \mathbf{a} .

$$\forall \mathbf{v}, \mathbf{a} \in \mathbb{R}^2 \quad \boldsymbol{\Omega}(\mathbf{v}, \mathbf{a}) = \begin{cases} \frac{1}{|\mathbf{v}|^2}(\mathbf{v} \wedge \mathbf{a}), & \text{if } |\mathbf{v}| > 0 \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (34)$$

The two equations (28) and (30) are equivalent to the differential Equation (35) for the trajectory $\mathbf{y}_\sigma(t)$ of system B in the phase space \mathcal{Y} .

$$\dot{\mathbf{y}}_\sigma = \boldsymbol{\rho}(\mathbf{y}_\sigma, \mathbf{a}_y), \quad (35)$$

where the function $\boldsymbol{\rho}: S^1 \times \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined by equation (36).

$$\boldsymbol{\rho}(\mathbf{y}_\sigma, \mathbf{a}) = [\mathbf{a}_t(\boldsymbol{\theta}, \mathbf{u}, \mathbf{a}), \boldsymbol{\omega}(\boldsymbol{\theta}, \mathbf{u}, \mathbf{a})]^T \quad (36)$$

For a control-input trajectory of system A prescribed on the time interval I differential equation (26) constitutes an ordinary differential equation for which the corresponding initial value problem $\mathbf{y}_\sigma(0) = \mathbf{y}_0$ admits a unique solution for any initial condition $\mathbf{y}_0 = (\boldsymbol{\theta}_0, u_0) \in \mathcal{Y}$. This initial value represents the state-space component of the trajectory $\boldsymbol{\sigma}(t) = (\mathbf{y}_\sigma(t), w_\sigma(t))$ of control system B , while the associated control-input trajectory $w_\sigma(t)$ is determined by equation $\forall t \in I \quad w_\sigma(t) = \boldsymbol{\rho}(\mathbf{y}_\sigma(t), \mathbf{a}_y(t))$. Moreover, the differential equation (35), parameterized by the control-input trajectory of system A , describes the operation of the hardware-implementable linearization scheme for the plant represented by the control system B , shown in Figure 8. The relationship between the output \mathbf{a}_t of block and the input signals \mathbf{v} and \mathbf{a} are given by formula (30), while the functional dependence between the output signal $\boldsymbol{\omega}(t)$ and the input signals \mathbf{v} and \mathbf{a} – is defined by Formula (34).

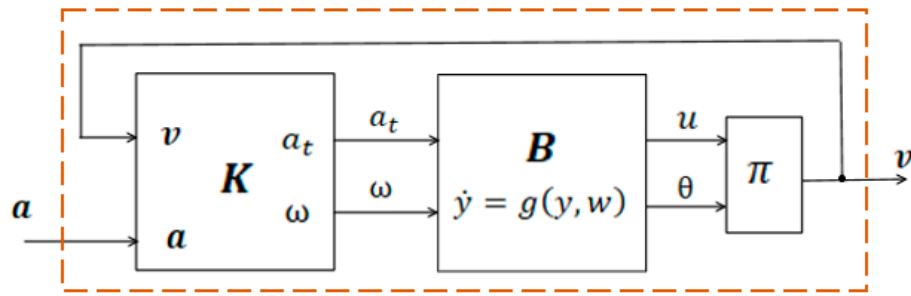


Figure 8. Linearization scheme for the control system B on the smooth manifold $S^1 \times \mathbb{R}$.

As noted above, the control system Q can be viewed as a superstructure built on top of system B or more precisely, system Q is structurally equivalent to the interconnection of system B with an integrator (see Figure 9).

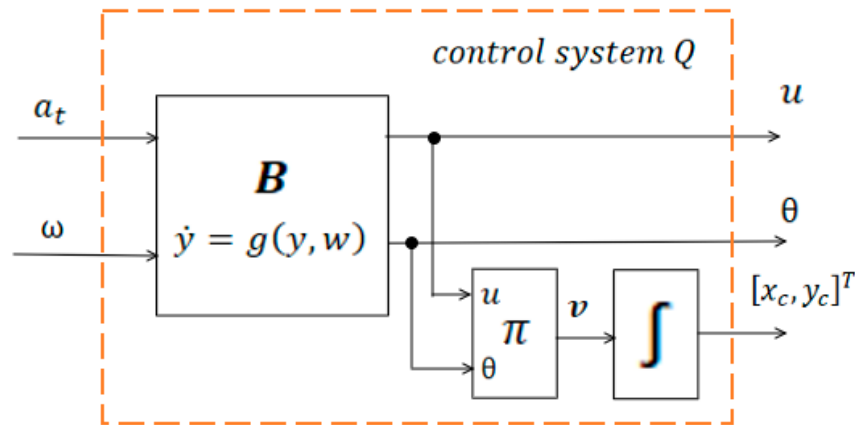


Figure 9. Structural block diagram of the control system Q .

It follows that the hardware-implementable linearisation scheme developed for the control system B (see Figure 10) is also applicable to the control system Q . Since, after linearisation, system B becomes equivalent to an integrator, the control system Q is a flat control system which flat output is $\mathbf{p} = [p_x, p_y]^T$. It should be emphasised that not all trajectories of system B admit a physical interpretation as trajectories of a material point moving in the plane; only those trajectories for which a corresponding trajectory of the control system A , exists are physically realisable. Accordingly, the trajectories of system B (and their counterparts in system Q) for which no corresponding trajectory of system A exists will be referred to as physically unrealizable trajectories.

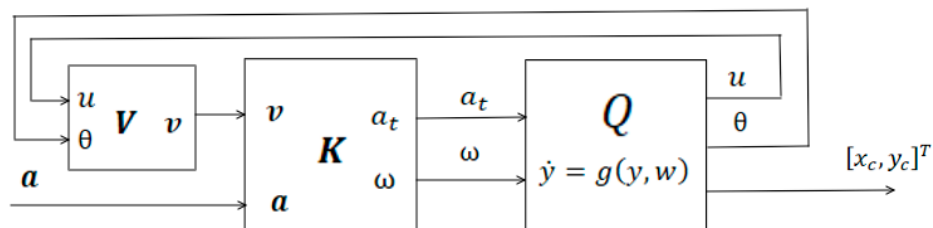


Figure 10. Linearisation of the kinematic control system governing the motion of a point in the plane Q via static state feedback.

The corresponding linearisation scheme for the simplest kinematic model of the robot A , as follows from the results presented in the previous section, is shown in Figure 11.

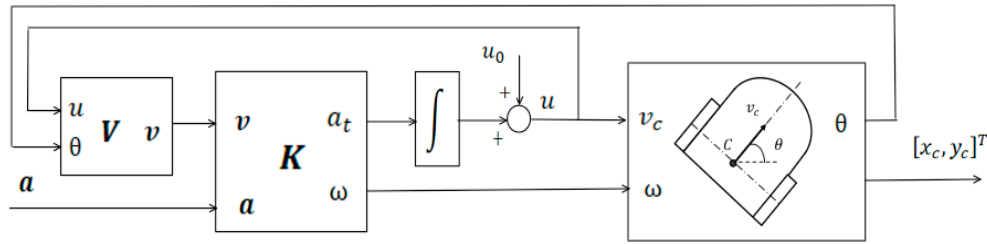


Figure 11. Linearisation scheme for the simplest kinematic model of the DDWMR.

The linearised kinematic model of the DDWMR is equivalent to the trivial linear control system $T_{2,2}$, which can be interpreted as a force-control system governing the motion of a point in the plane. Consequently, the problem of controlling a nonlinear system is reduced to a simple linear trajectory-tracking problem with a clear physical interpretation. This linear problem admits a straightforward and efficient solution, which is described in the following section.

2.4. Trajectory-Tracking Control System for the Linearised Kinematic Model of the DDWMR

The linearised kinematic model of the DDWMR is equivalent to the trivial linear control system $T_{2,2}$. Recall that this linear system can be interpreted as a force-control system governing the motion of a unit-mass point moving in the plane. Accordingly, the feedforward trajectory-tracking problem for the DDWMR may be formulated as follows: given a reference signal $\mathbf{p}_d(t) = [p_{dx}(t), p_{dy}(t)]^T$, representing the desired time evolution of the point's position in the plane, how can one transform it into a control signal $\mathbf{f}(t) = [f_x(t), f_y(t)]^T$, that ensures that the linearised system (and therefore the original DDWMR) follows the prescribed trajectory? Since for a unit-mass point the relation $\dot{\mathbf{p}} = \mathbf{f}$, holds at every instant of time, the answer becomes immediate. By setting $\mathbf{f}(t) = \ddot{\mathbf{p}}_d(t)$ we ensure that, for any $t \geq 0$ along the trajectory, the actual acceleration $\ddot{\mathbf{p}}(t)$ coincides with the prescribed reference signal $\mathbf{p}_d(t)$. Thus, the point mass follows the desired trajectory exactly, provided that the initial conditions match and no disturbances are present. Of course, the actual trajectory of the point $\mathbf{p}(t)$ is determined not only by the control signal $\mathbf{p}_d(t)$, but also by the initial position $\mathbf{p}_0 = \mathbf{p}(0)$ and the initial velocity $\mathbf{v}_0 = \dot{\mathbf{p}}(0)$. The relationship between the actual trajectory and the reference signal of the trajectory-tracking control system is given by (37).

$$\forall t \geq 0 \mathbf{p}(t) = \mathbf{p}_d(t) + (\mathbf{p}_d(0) - \mathbf{p}_0) + (\dot{\mathbf{p}}_d - \mathbf{v}_0)t \quad (37)$$

In the ideal case, when the initial state of the system exactly matches the initial state implied by the desired trajectory and when no disturbances are present, the actual trajectory coincides exactly with the desired trajectory defined by the reference signal $\mathbf{p}_d(t)$. However, in the presence of any mismatch between the initial conditions of the plant and those associated with the reference trajectory, the tracking error will inevitably grow over time. Likewise, the purely feedforward trajectory-tracking scheme is highly sensitive to external disturbances and therefore unsuitable for practical use due to its inherent open-loop instability.

To overcome these limitations, the trajectory-tracking system can be modified by introducing a feedback loop with a linear controller that stabilises the closed-loop dynamics. The resulting structure ensures robustness with respect to initial-condition mismatches and perturbations. Figure 12 shows the block diagram of the modified trajectory-tracking system for the differential-drive mobile robot.

Disturbances are modelled by adding a noise term $n(t)$ to the sum of the signals applied to the input \mathbf{a} of the linearising transformation block L . The linear signal transformer L consists of two parallel, identical linear subsystems L_x and L_y . Consequently, the linear system equivalent to the modified trajectory-tracking scheme of the mobile robot can be viewed as the combination of two independent, identical one-dimensional linear subsystems. Figure 13 presents the block diagram of the x -channel.

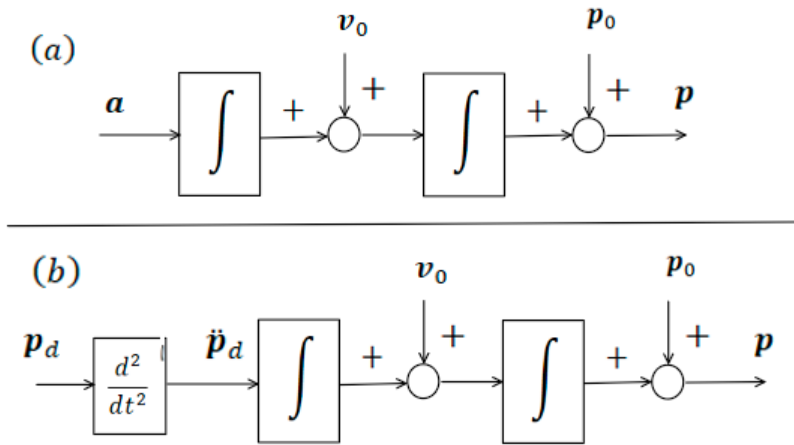


Figure 12. (a) Equivalent representation of the linearised kinematic model of the DDWMR in the form of a chain of integrators. (b) Basic feedforward trajectory-tracking control scheme.

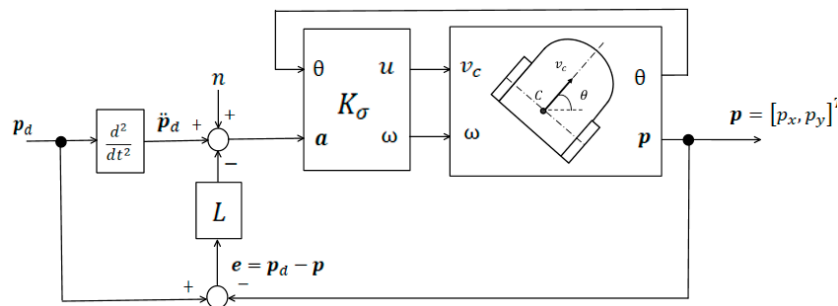


Figure 13. Equivalent block diagram of the x -channel of the trajectory-tracking control system for the linearised kinematic model of the DDWMR.

To analyse the behaviour of such a one-dimensional linear system, representing each channel of the trajectory-tracking scheme, it is convenient to employ the Laplace transform. In this framework, constant functions of the form $f(t) = c \in \mathbb{R}$ are represented as $c\eta(t)$, where η denotes the Heaviside step function. We denote by G_x the transfer function describing the linear signal transformers L_x and L_y . In addition, we introduce the notations $P_x(s) = \mathcal{L}(p_x(t))$ and $P_{dx}(s) = \mathcal{L}(p_{dx}(t))$. Taking into account that $\mathcal{L}(\ddot{p}_{dx}(t)) = s^2 P_{dx}(s) - sp_{dx}(0) - p'_{dx}(0)$ and $\mathcal{L}(\eta(t)) = \frac{1}{s}$ we obtain, for the Laplace-domain representations of the signals, the block structure describing the behaviour of each one-dimensional channel of the robot's trajectory-tracking system, as shown in Figure 14.

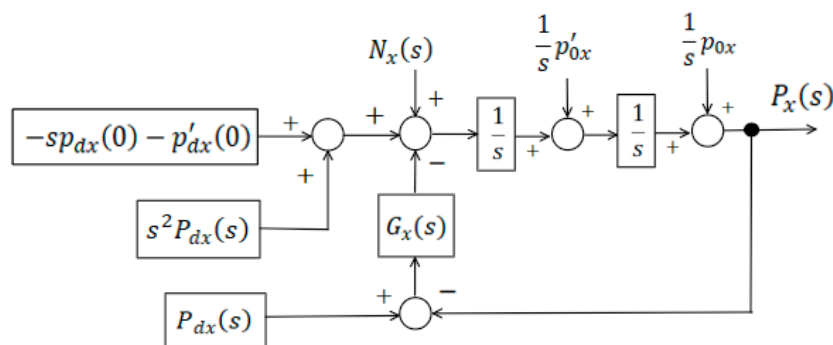


Figure 14. Laplace-domain block diagram of the x -channel of the trajectory-tracking control system for the differential-drive mobile robot.

The analysis of the one-dimensional linear trajectory-tracking subsystem in the Laplace domain leads to expression (38), which describes the relationship between the Laplace transform $P_x(s)$ of the system output component p_x and the transform $P_{dx}(s)$ of the corresponding reference component.

$$\mathbf{P}_x(s) = \mathbf{P}_{dx}(s) + \mathbf{E}_x^d(s) + \frac{N_x(s)}{s^2 - G_k(s)}, \quad (38)$$

where the complex-variable function $E_x^d(s)$ is defined by expression (39).

$$\mathbf{E}_x^d(s) = \frac{s(p_{0x} - p_{dx}(0)) + (p'_{0x} - p'_{dx}(0))}{s^2 - G_k(s)} \quad (39)$$

Naturally, the function $E_x^d(s)$ can be interpreted as the Laplace transform of the time-domain function $e_x^d(t)$. As follows from the definition of $E_x^d(s)$ in (39), the function $e_x^d(t)$ is the solution of the initial value problem for the differential equation (40) with the initial conditions given in (41).

$$\ddot{e}_x^d - U e_x^d = \mathbf{0} \quad (40)$$

$$\mathbf{e}_x^d(0) = \mathbf{p}_{0x} - \mathbf{p}_{dx}(0) \quad \dot{\mathbf{e}}_x^d(0) = \mathbf{p}'_{0x} - \mathbf{p}'_{dx}(0) \quad (41)$$

In the differential equation (40), the symbol U denotes the linear operator representing the action of the linear signal transformer defined by the transfer function G_x . Term $\frac{N_x(s)}{s^2 - G_k(s)}$ in Equation (38) represents the Laplace transform of the disturbance signal $n_x(t)$, passed through a linear filter described by the transfer function $W_f(s)$, defined in expression (42).

$$\mathbf{W}_f(s) = \frac{\mathbf{1}}{s^2 - G_k(s)} \quad (42)$$

Thus, the output of the trajectory-tracking subsystem $y_x(t)$ is a linear superposition of the three functions of the form (42).

$$\forall t \geq 0 \quad \mathbf{p}_x(t) = \mathbf{p}_{dx}(t) + \mathbf{e}_x^d(t) + \mathcal{L}^{-1}(\mathbf{W}_f N_x) \quad (43)$$

As follows from a straightforward analysis of expression (43), the stability of the trajectory-tracking system is guaranteed provided that the linear signal transformers L_x and L_y satisfy Criterion 1, formulated below. It is also evident that, in order to minimise the influence of disturbances represented by the term $\mathcal{L}^{-1}(W_f N_x)$ the superposition (43)—the transfer function $G = G_x = G_y$ of the identical signal transformers L_x and L_y must satisfy Criterion 2.

Criterion 1. The linear operator U , corresponding to the identical linear signal transformers L_x and L_y must stabilise the differential Equation (44).

$$\ddot{e} - U e = \mathbf{0} \quad (44)$$

Criterion 2. The transfer function G of the linear signal transformers L_x and L_y must satisfy the disturbance-rejection condition (45).

$$\forall \omega \geq 0 \quad |G(i\omega) + \omega^2| > 1 \quad (45)$$

As a result, the two linearisation criteria, achievability of the required dynamic form and satisfaction of the disturbance-rejection condition, ensure the correctness of the constructed transformation and form the foundation for designing the outer trajectory-tracking control loop, to be presented in the next section.

3. Analysis of the Resulting Control System

To validate the effectiveness of the proposed plant-linearization-based trajectory tracking controller, a set of simulations was conducted using a differential-drive two-wheeled mobile robot following a circular reference trajectory of radius $R = 1.5$ m. The controller receives the desired velocity and acceleration obtained from the flat output and generates the required wheel angular velocities through the nonlinear to-linear transformation.

3.1. Trajectory Tracking Performance

Figure 15 illustrates both the reference (desired) circular trajectory $z_d(t)$ and the actual robot trajectory $z(t)$ obtained from the simulation. The robot initially starts away from the reference curve, after which the nonlinear inner loop drives the system toward the desired path. Following the transient phase, the actual trajectory converges smoothly to the reference circle and remains closely aligned with it throughout the remainder of the simulation. This combined figure highlights two important effects of the proposed controller: (1) reliable convergence to the circular path, and (2) high-precision steady-state tracking, where the deviation between the desired and the actual trajectories becomes practically negligible.

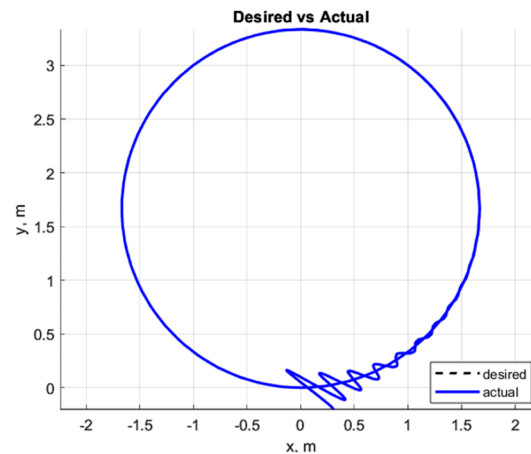


Figure 15. Desired circular trajectory and actual robot trajectory during the simulation. The robot converges to the reference path and maintains accurate tracking in steady state.

3.2. Angular Velocity and Heading Evolution

Figure 16 shows the evolution of the angular velocity $\Omega(t)$ and the heading angle $\theta(t)$. During the initial transient phase, the angular velocity exhibits expected oscillations as the controller brings the system toward the desired motion regime. After approximately 120 seconds, $\Omega(t)$ stabilizes near the nominal value corresponding to circular motion, and the heading angle increases linearly, confirming that the robot moves with a constant angular rate along the reference circle.

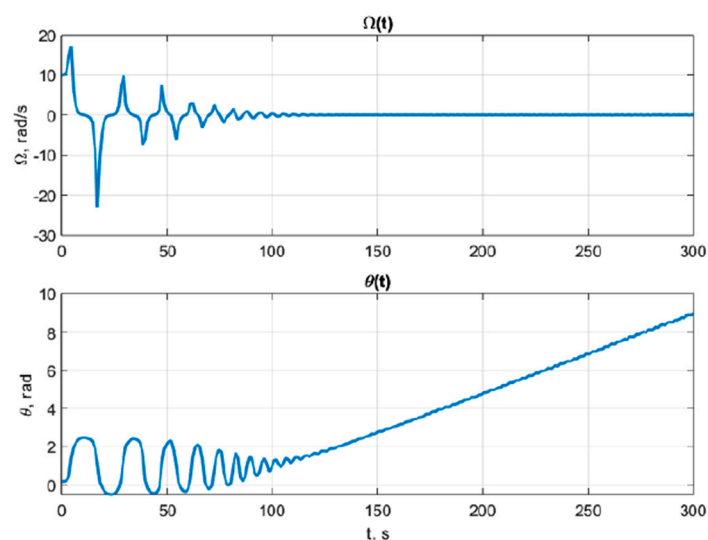


Figure 16. Angular velocity and heading angle.

These plots confirm stability of the internal nonlinear loop and proper functioning of the linearized outer loop. The evolution of the position tracking errors is presented in Figure 17. The maximum transient deviations occur at the beginning of the motion due to the initial misalignment between the robot and the reference trajectory: maximum error in the x-direction: $e_x^{max} \approx 0.22$ m; maximum error in the y-direction: $e_y^{max} \approx 0.25$ m; maximum overall positional deviation: $|e|^{max} \approx 0.28$ m. These deviations decay rapidly, and by $t \approx 120$ s the tracking errors converge to zero. In the steady-state phase, the robot maintains sub-centimeter accuracy: $mean(|e(t)|) < 0.01$ m.

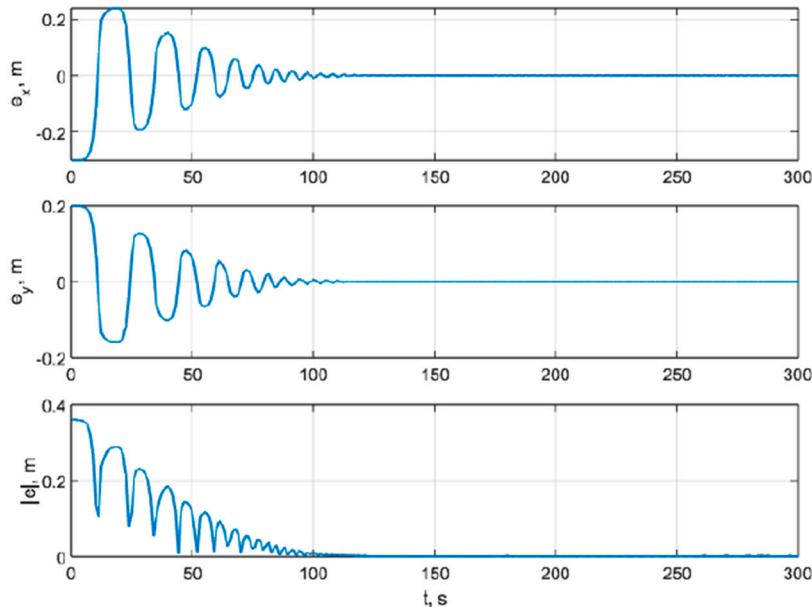


Figure 17. Position errors in x, y, and absolute error.

This figure clearly demonstrates the rapid error decay and excellent tracking precision achieved after convergence.

4. Discussion

The proposed trajectory-tracking method is closely related to the well-known CTC technique for serial manipulators, also referred to as inverse-dynamics control. At the conceptual level, both approaches rely on transforming a nonlinear plant via static-feedback linearisation into an equivalent trivial linear controllable system. As a result, the control problem for an n -dimensional nonlinear system reduces to the control of n independent one-dimensional decoupled linear subsystems. The resulting trajectory-tracking architectures are structurally identical: in both cases, a nonlinear inner loop performs linearisation, while a linear outer loop implements standard tracking laws using classical linear controllers. Consequently, together with the recognised advantages of CTC, which remains a highly active research area due to its excellent performance characteristics [26–28] the proposed method naturally inherits the limitations that are intrinsic to static-feedback linearisation.

A key drawback of CTC type methods is their limited robustness to model uncertainty: even small discrepancies between the real plant and the mathematical model used inside the linearising controller can lead to noticeable degradation of tracking performance. In this study, a simple kinematic DDWMR model was deliberately chosen to ensure transparency of the approach and ease of simulation. Nevertheless, our simulation experiments confirm the expected behaviour: the performance of the proposed controller is sensitive to parameter mismatches, similarly to classical CTC. This phenomenon is well-documented in the CTC literature, where modern research directions focus on adaptive and robust extensions, including the use of machine-learning-based compensation mechanisms [20].

Since the trajectory-tracking architecture proposed in this paper consists of two nested loops—a nonlinear inner loop that linearises the plant, and an outer loop that performs trajectory tracking of the linearised system—the outer loop can incorporate a wide range of modern control techniques, including robust, adaptive, optimal, or MPC-based controllers. This is a fundamental difference from most existing flatness-based methods for DDWMRs, which rely on approximate linearisation in the neighbourhood of a predefined reference trajectory (exact feedforward linearisation). Such schemes require a sufficiently accurate match between the initial state of the robot and the nominal initial condition of the reference trajectory, and they may exhibit performance loss or even instability in case the robot starts significantly away from the desired path.

By contrast, the method proposed here avoids these limitations because it performs plant linearisation, not trajectory linearisation. In other words, it is the nonlinear plant itself that is converted into an equivalent linear controllable system, rather than the entire trajectory-tracking closed loop. As highlighted earlier, current research trends increasingly focus on adaptive flatness-based techniques and on the application of MPC, optimal control, and hybrid control methods to flat systems [15,16,23–25]. Notably, many of these techniques become particularly effective and easier to implement when applied to differentially flat systems [20], making the proposed architecture compatible with a broad class of advanced control methods.

Another characteristic distinction of the proposed approach is that it yields a SISO (single-input-single-output) control structure for the linearised system. In contrast, most existing DDWMR trajectory-tracking methods rely on multivariable controllers with coupled control channels [25]. The SISO nature of the resulting linear system significantly simplifies controller design, tuning, and implementation, providing a conceptually straightforward yet highly effective solution.

5. Further Perspectives

Future research is expected to proceed in two directions. First, the proposed linearization-based trajectory-tracking method can be extended from the kinematic DDWMR model to its full dynamic model, as well as to other classes of mobile robots, including balancing robots, aerial vehicles, and multi-degree-of-freedom robotic platforms. Such generalizations would enable a broader assessment of the method's applicability to complex nonlinear robotic systems. Second, an important avenue for further work is the development of robust and adaptive extensions of the proposed controller. Given its sensitivity to model uncertainties—an inherent limitation shared with classical computed torque control—future research will focus on designing robust, adaptive, and learning-enhanced compensation mechanisms that preserve the advantages of the linearized structure while improving resilience to parameter variations and external disturbances.

6. Conclusions

This work has presented a linearisation-based trajectory-tracking framework for differential-drive wheeled mobile robots, in which a nonlinear kinematic model is transformed via static-feedback linearisation into an equivalent SISO linear system. This transformation allows the nonlinear trajectory-tracking problem to be reformulated as a classical linear control task for two decoupled first-order subsystems, greatly simplifying the controller design while maintaining strong theoretical guarantees.

The proposed approach is conceptually related to com CTC and inherits its principal advantages: transparency of design, explicit control of the closed-loop dynamics, and high tracking accuracy for sufficiently well-modelled systems. Simulation results confirm these benefits: the robot successfully converges to the desired circular trajectory, achieving a maximum transient deviation of 0.28 m, a settling time of approximately 120 s, and a steady-state mean tracking error below 0.01 m. These results demonstrate that linearising the plant itself, rather than the trajectory-tracking loop, provides a simple yet powerful mechanism for achieving high-precision tracking in nonlinear mobile robots.

At the same time, as with CTC, the method exhibits sensitivity to modelling inaccuracies, highlighting the importance of further research into robustness and adaptation. Nevertheless, the proposed framework establishes a general and extensible foundation for trajectory-tracking control of differentially flat systems and demonstrates strong potential for both theoretical development and practical deployment.

Author Contributions: Conceptualization, D.A., A.Kr., and G.N.; methodology, A. Kr, G.N., and A.K.; software, A. Kr. and G.N.; validation, G.N., A.Kr., A.K., and D.A.; formal analysis, D.A. and T.H.; investigation, G.N., A.Kr., A.D., A.K., and D.A.; resources, D.A.; data curation, D.A.; writing—original draft preparation, G.N. and D.A.; writing—review and editing, D.A. and T.H.; visualization, G.N.; supervision, D.A. and T.H.; project administration, D.A.; funding acquisition, D.A. and T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP19679327.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Acknowledgments: This research was funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP19679327 “Methods of machine learning for tasks of automatic control and inertial navigation of mobile robots”. T. Haidegger is a Consolidator Researcher, receiving financial support from the Distinguished Researcher program of Obuda University [38].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, J.; Li, Y.; Jin, J.; Jia, Z.; Wei, C. Design, Analysis and Control of Tracked Mobile Robot with Passive Suspension on Rugged Terrain. *Actuators* 2025, 14, 389. <https://doi.org/10.3390/act14080389>
2. Hu, H.; Long, D.; Liang, Y.; Wang, B.; Wang, X.; Su, R. Braking Control of Mobile Robots Using Integral Sliding-Mode Algorithm with Composite Convergence Regulation. *Processes* 2025, 13, 2887. <https://doi.org/10.3390/pr13092887>
3. Banu, S.; Das, O.; Venkadeshwaran, K.; Gaurav, A.; Kapoor, T.; Pandey, N. Designing an Adaptive Control System for Robust Path Planning in Autonomous Mobile Robots. *Multidiscip. Sci. J.* 2025, 7, e2025ss0120. <https://doi.org/10.31893/multiscience.2025ss0120>
4. Zeng, Z.; Chen, S.; Kong, X.; Li, X.; Zhang, C.; Yang, G. Revised Control Barrier Function with Sensing of Threats from Relative Velocity Between Humans and Mobile Robots. *Sensors* 2025, 25, 4005. <https://doi.org/10.3390/s25134005>
5. IEEE 24748-7000-2022; IEEE/ISO/IEC International Standard—Systems and Software Engineering—Life Cycle Management—Part 7000: Standard Model Process for Addressing Ethical Concerns During System Design. IEEE: New York, NY, USA, 2022.
6. Houghtaling, M.A.; Fiorini, S.R.; Fabiano, N.; Gonçalves, P.J.S.; Ulgen, O; ... & Prestes, E.. Standardizing an Ontology for Ethically Aligned Robotic and Autonomous Systems. *IEEE Trans. Syst. Man Cybern. Syst.* 2023. vol. 54. pp.1791–1804.
7. Haidegger, T.; Mai, V.; Mörch, C.; Boesl, D.; Jacobs, A.; Khamis, A.; Lach, L.; Vanderborght, B. Robotics: Enabler and inhibitor of the sustainable development goals. *Sustain. Prod. Consum.* 2023. vol.43. pp.422–434.
8. Bloch, A., Crouch, P., Bailleul, J., and Marsden, J., 2003, “Nonholonomic Mechanics and Control” (Interdisciplinary Applied Mathematics), SpringerVerlag, New York.

9. Luviano-Juárez, A.; Cortés-Romero, J.; Sira-Ramírez, H. Trajectory tracking control of a mobile robot through a flatness-based exact feedforward linearization scheme. *J. Dyn. Syst. Meas. Control* 2015, 137, 051003.
10. Chen, L.; Jia, Y. Variable-poled tracking control of a two-wheeled mobile robot using differential flatness. *J. Robot. Netw. Artif. Life* 2014, 1, 12–16.
11. Rigatos, G. *Nonlinear Control and Filtering Using Differential Flatness Theory Approaches: Applications to Electromechanical Systems*; Springer: Cham, Switzerland, 2016.
12. Chen, L.; Jia, Y. Output Feedback Tracking Control of Flat Systems via Exact Feedforward Linearization and LPV Techniques. *Int. J. Control Autom. Syst.* 2019, 17, 606–616.
13. Nicolau, F.; Respondek, W.; Barbot, J.-P. Construction of flat inputs for mechanical systems. In *Proceedings of the 7th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, Berlin, Germany, 18–21 October 2021.
14. Nicolau, F.; Respondek, W.; Barbot, J.-P. How to minimally modify a dynamical system when constructing flat inputs? *Int. J. Robust Nonlinear Control* 2021, 31(18), 9538–9561. <https://doi.org/10.1002/rnc.5790>
15. Rodriguez-Guevara, D.; Favela-Contreras, A.; Beltran-Carbajal, F.; Sotelo, C.; Sotelo, D. A Differential Flatness-Based Model Predictive Control Strategy for a Nonlinear Quarter-Car Active Suspension System. *Mathematics* 2023, 11, 1067. <https://doi.org/10.3390/math11041067>
16. Li, Y.; Zhu, Q.; Elahi, A. Quadcopter Trajectory Tracking Control Based on Flatness Model Predictive Control and Neural Network. *Actuators* 2024, 13, 154. <https://doi.org/10.3390/act13040154>
17. Hamad, Q.M.; Raafat, S.M. A Flatness-Based Trajectory Tracking Control for Chemical Reactor. In *Proceedings of the 21st International Multi-Conference on Systems, Signals & Devices (SSD 2024)*, Erbil, Iraq, 22–25 April 2024; pp. 819–825. <https://doi.org/10.1109/SSD61670.2024.10549742>
18. Lamraoui, H.C.; Bouzid, Y.; Nemra, A. Flatness-based nonlinear internal model control with disturbances compensation via improved extended state observer for remotely operated vehicle. *International Journal of Control* 2024, 98(6), 1480–1494. <https://doi.org/10.1080/00207179.2024.2428663>
19. Rigatos, G.; Wira, P.; Abbaszadeh, M.; Pomares, J. Flatness-based control in successive loops for industrial and mobile robots. In *Proceedings of the 48th Annual Conference of the IEEE Industrial Electronics Society (IECON 2022)*, 2022, pp. 1–6. <https://doi.org/10.1109/IECON49645.2022.9968538>
20. Join, C.; Delaleau, E.; Fliess, M. Flatness-based control revisited: The HEOL setting. *Comptes Rendus. Mathématique* 2024, 362, 1693–1706. <https://doi.org/10.5802/crmath.674>
21. Fliess, M.; Lévine, J.; Martin, P.; Rouchon, P. A Lie–Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Trans. Autom. Control* 1999, 44, 922–937. <https://doi.org/10.1109/9.763209>
22. S. Sun, A. Romero, P. Foehn, E. Kaufmann and D. Scaramuzza, “A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight,” in *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, Dec. 2022, doi: 10.1109/TRO.2022.3177279
23. Logan E. Beaver, Andreas A. Malikopoulos Optimal control of differentially flat systems is surprisingly easy, *Automatica*, Volume 159, 2024, 111404 <https://doi.org/10.1016/j.automatica.2023.111404>
24. J. O. Limaverde Filho, E. C. R. Fortaleza, and M. C. M. Campos, A derivative-free nonlinear Kalman filtering approach using flat inputs, 2021. *International Journal of Control Taylor and Francis*
25. Markus Bröcker, Lukas Herrmann Flatness based control and tracking control based on nonlinearity measures, *IFAC-PapersOnLine*, Volume 50, Issue 1, 2017, Pages 8250–8255, <https://doi.org/10.1016/j.ifacol.2017.08.1394>
26. Han, Y., Ma, H., Wang, Y. et al. Neural Network Robust Control Based on Computed Torque for Lower Limb Exoskeleton. *Chin. J. Mech. Eng.* 37, 37 (2024). <https://doi.org/10.1186/s10033-024-01025-5>
27. R. Luo, Z. Hu, M. Liu, L. Du, S. Bao and J. Yuan, “Adaptive Neural Computed Torque Control for Robot Joints with Asymmetric Friction Model,” in *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 732–739, Jan. 2025, doi: 10.1109/LRA.2024.3512372
28. Nguyen, Nguyet & Ba, Dang. (2022). A Computed Torque Controller for Robotic Manipulators Using Nonlinear Neural Network. 487–492. 10.1109/GTSD54989.2022.9989043

29. Shadrin, G.; Krasavin, A.; Nazenova, G.; Kussaiyn-Murat, A.; Kadyroldina, A.; Haidegger, T.; Alontseva, D. Application of compensation algorithms to control the speed and course of a four-wheeled mobile robot. *Sensors* 2024, 24, 7233. <https://doi.org/10.3390/s24227233>
30. Krasavin, A.; Nazenova, G.; Alontseva, A.; Dairbekova, A. Tracking Control of a Two-Wheeled Mobile Robot Based on En-dogenous Linearization of the Dynamic Model. In Proceedings of the 2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Antalya, Türkiye, 2025; pp. 1–6. <https://doi.org/10.1109/ACDSA65407.2025.11166132>
31. Kussaiyn-Murat, A.; Kadyroldina, A.; Krasavin, A.; Tolykbayeva, M.; Orazova, A.; Nazenova, G.; Krak, I.; Haidegger, T.; Alontseva, D. Application of discrete exterior calculus methods for the path planning of a manipulator performing thermal plasma spraying of coatings. *Sensors* 2025, 25, 708. <https://doi.org/10.3390/s25030708>
32. Alontseva D. L., Ghassemieh E., Krasavin A. L., Kadyroldina A. T. Development of 3D Scanning System for Robotic Plasma Processing of Medical Products with Complex Geometries. *Journal of Electronic Science and Technology*. - 2020. Vol 18. No 3. pp.212-222 DOI 10.1016/j.jnlest.2020.100057
33. Alontseva D. L., Ghassemieh E., Krasavin A. L., Shadrin G. K., Kussaiyn-Murat A. T., Kadyroldina A. T. Development of Control System for Robotic Surface Tracking. *International Journal of Mechanical Engineering and Robotics Research*, Vol. 9, No. 2, February 2020. – P. 280-286. doi: 10.18178/ijmerr.9.2.280-2861
34. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*; Cambridge University Press: Cambridge, UK, 2017.
35. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer: London, UK, 2009. <https://doi.org/10.1007/978-1-84628-642-1>
36. Takacs, A., & Haidegger, T. (2024). A method for mapping V2X communication requirements to highly automated and autonomous vehicle functions. *Future Internet*, 16(4), 108.
37. Takacs, A., & Haidegger, T. (2022). Infrastructural requirements and regulatory challenges of a sustainable urban air mobility ecosystem. *Buildings*, 12(6), 747.
38. Haidegger, T. P., Galambos, P., Tar, J. K., Kozlovsky, M., Zrubka, Z., Eigner, G., ... & Rudas, I. J. (2024). Strategies and outcomes of building a successful university research and innovation ecosystem. *Acta Polytechnica Hungarica* Vol 21. Issue 10. pp. 13-35.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.