VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODELOVÁNÍ A ŘÍZENÍ MOBILNÍCH ROBOTŮ S NĚKOLIKA ŘÍZENÝMI KOLY

MODELLING AND CONTROL OF MULTI-STEERED WHEELED MOBILE ROBOTS

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE / AUTHOR — Ing. JAKUB HRABEC

VEDOUCÍ PRÁCE / SUPERVISOR — prof. Ing. PAVEL JURA, CSc.

BRNO 2009

Abstrakt
Dizertační práce se zabývá problematikou kinematického modelování a řízení mobliních kolových robotů. Přináší sumarizaci problematiky kinematického modelování mobilních robotů obecně a popis vlastností kolových mobilních robotů s několika řízenými koly. Použitý aparát z matematiky, fyziky je vysvětlován s důrazem na pohled teorie řízení.

Dále je prezentován nový řídicí algoritmus pro mobilní kolové roboty s více řízenými koly, vhodný pro úlohu stabilizace v bodě i sledování trajektorie, tedy obě nejčastěji řešené úlohy pohybu mobilních robotů.

Summary
The dissertation deals with the kinematic modelling and control of wheeled mobile robots. It summarizes the problems of kinematic modelling of wheeled mobile robots in general and examines the properties of the multi-steered wheeled mobile robots. The theoretical background is explained from control theory viewpoint.

A new control algorithm for multi-steered wheeled mobile robots is presented. It is suitable for set-point stabilization as well as trajectory tracking, the two most common tasks.

Klíčová slova
Kinematické modelování, řízení, mobilní roboty, několik řízených kol

Keywords
Kinematic modelling, control, wheeled mobile robots, multi-steered

HRABEC, J.*Modelování a řízení mobilních robotů s několika řízenými koly.* Brno: Vysoké učení technické v Brně, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, 2009. 92 s. Vedoucí prof. Ing. Pavel Jura, CSc.

Prohlášení

„Prohlašuji, že svou dizertační práci na téma *Modelování a řízení mobilních robotů s několika řízenými koly* jsem vypracoval samostatně pod vedením školitele a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené dizertační práce dále prohlašuji, že v souvislosti s vytvořením této dizertační práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **11. září 2009** ....

podpis autora

Acknowledgments

I would like to thank to my supervisor, prof. Ing. Pavel Jura, CSc., for his guidance and inspiring suggestions. I would like to thank to prof. Ing. František Šolc, CSc. for consultations and proof-reading the text. Also, I would like to thank to doc. RNDr. Miroslav Kureš, Ph.D. and doc. RNDr. Sergej Čelikovský, CSc. for valuable help.

Many thanks goes also to my family for their support and patience.

Jakub Hrabec
Brno
September 11, 2009

Contents

Chapter 1

Goal of Thesis

The goal of this dissertation is to

- summarize and co-ordinate the problems of kinematic modelling of wheeled mobile robots,
- examine the properties of multi-steered wheeled mobile robots,
- develop a new control scheme for multi-steered wheeled mobile robots.

There are many theoretical works on the topic of kinematic modelling and control of wheeled mobile robots. They use scientific language at different levels of complexity. Sometimes, this strict mathematical viewpoint prevents a control engineer to understand the core of the solution of a given problem. The first part of this thesis tries to be a contribution to solution of this situation.

The mathematical apparatus from the first part is then used to examine properties and modelling of multi-steered wheeled mobile robots, namely bi-steered wheeled mobile robot.

Development of a new control strategy for the multi-steered wheeled mobile robots concludes the work.

Chapter 2

Introduction

Wheeled mobile robots (WMRs) are increasingly present in industrial and service robotics, particularly when autonomous motion capabilities are required over reasonably smooth grounds and surfaces. This property is important from a few aspects with respect to robot motion: **wheeled mobile robots (WMRs)**

- chassis construction—the chassis, and the wheels, don't need to be constructed to overcome terrain roughnesses. This simplifies the design of the robot significantly.
- control problems—under certain assumptions (e.g., in low speeds), only kinematics can be taken into account, neglecting the dynamics of the robot. This results in easier design of control algorithms.

Several mobility configurations (wheel number and type, their location and actuation, single- or multibody vehicle structure) can be found in applications. Single-body robots can be used as carriages for manipulators, service robots, household robots, automatic lawn mowers etc. Typical use for multi-body setup is in transportation tasks (material transportation, luggage transportation).

It should be noted that mobile robots are still[1] not very common in the industry, at least compared to stationary robots (manipulators)—specifically in automobile industry (for handling materials, welding, spraying etc.).

On the other hand, there are many areas where the mobile robots become indispensable. Their main use is in situations dangerous for humans, like explosives deactivation, fire extinguishing, exploring sites of accidents or havocs (e.g., unstable buildings damaged by earthquake) or in space exploration. However, in these situations, the mobile robots usually run in *telepresence* mode, i.e., they are controlled remotely by a human operator. **telepresence** The sensory subsystem of the robot gives then as much information as possible about the surrounding environment to the operator (real-time stereo image in combination with positioning of the cameras based on operator's head movements, force-feedback, etc.). In ideal case, the operator should feel like he was in the place of the robot.

The situations described above can also involve motion of the robot in rough terrain. If a good mobility through terrain is the main objective, other types of locomotion than wheels can be used, including legged, caterpillar etc. chassis.

The opposite case—moving on flat surfaces—can typically be found in indoor applications, such as industry (material transport, storages), services (luggage transport),

[1] at the time of writing this text

home (household robotics) or medical (patient assistance) to name a selection of possible uses. In the industry and services (transportation), the multi-body robots (i.e., tractor-trailer systems) are usually deployed. A typical task is to autonomously follow a predefined trajectory. In most mentioned situations, the human individuals can be present in the workspace (factory, home, hospital). This brings necessity of safe behaviour of the robots. The maximum velocity is usually reduced and the vehicles are equipped with sensory subsystem to enable obstacle detection (including eventual human individuals) and avoidance (or emergency stop).

Chapter 3

Control Theory, Mathematical and Physical Background

In this chapter, selected terms and concepts from control theory, mathematics and physics (namely kinematics) required in the following text will be reviewed. The definitions and terms are explained with focus on technical viewpoint rather than strictly mathematical one.

Note on notations: In (mathematical) texts on differential geometry, the indexes of individual components are usually written as upper indexes (superscripts), whereas the lower indexes are reserved for summation variables. In this text, however, this will not be adopted to keep coherency with engineering-oriented literature.

3.1 State, State Space and Configuration Space

A *state* is a minimum set of variables completely describing the condition of the system in given time instant. In simple words, it describes a situation, which the WMR finds in. Usually, a state of a WMR is described by its position and orientation in the Cartesian coordinates. **state**

A *state space* is a set of all possible states (all possible situations of a WMR). It is an m-dimensional smooth manifold. **state space**

A *manifold* is an abstract space locally similar to an Euclidean-like m-dimensional space but globally it can have different topology. As a simplified example, $\mathbb{R}^n$ can be taken, i.e., a state is represented by n-tuple of real numbers.

A *configuration space* is a state space for motion planning, in which it plays an important role. An in-depth description can be found, e.g., in [24]. For control of a WMR in obstacle-free environment, the state space and configuration space blend. It holds that dimension of configuration space equals to the number of DOFs of the WMR [24].

3.2 Nonlinear Systems

Systems can be divided into two main groups—linear and nonlinear ones.

Nonlinear systems can be used to model general and complex real physical systems. The most straightforward definition is that they are systems that are not linear, therefore it can be useful to recapitulate the feature of linear systems.

linear system

Linear systems are characterized by the following property: for given initial condition(s) and inputs(s) and measured corresponding output(s), it is possible to determine (compute) also outputs on sums and multiples of these initial conditions and outputs. This results in fact that linear system can be fully determined by finite number of experiments and behaviour of the system in a small neighbourhood of any point in its state space determines the behaviour in all points of the state space.

Nonlinear system can be neither characterized by finite number of experiments nor behaviour near selected point in its state space.

nonlinear system

A nonlinear system can be expressed in the form

$$\dot{\mathbf{q}} = \mathbf{g}(\mathbf{q}) + \sum_{i=1}^{m} \mathbf{h}_i(\mathbf{q}) u_i \tag{3.1}$$

where $\mathbf{q} \in \mathcal{Q}$ with $\mathbf{q}$ denoting the state of the system and $\mathcal{Q}$ its state space. Variables (more precisely vector) $\mathbf{u} \in \mathcal{U} = \mathbb{R}^n$, $\mathbf{u} = (u_1, \ldots, u_i, \ldots u_m)$ are the control inputs, with $m \leq n$. If the functions $\mathbf{g}$ and $\mathbf{h}_i$ are *smooth*[1], then the system is called *control-affine system* or *affine-in-control system*. These systems are linear in actions but nonlinear w.r.t. the state [24].

control-affine system

drift

The term $\mathbf{g}(\mathbf{q})$ is called a *drift* and if it is nonzero, for some $\mathbf{q} \in \mathcal{Q}$ there doesn't exist $\mathbf{u} \in \mathcal{U}$ such that $\mathbf{h}(\mathbf{q}, \mathbf{u}) = 0$ [24].

A special class of nonlinear systems can be obtained from (3.1) by letting $\mathbf{g}(\mathbf{q}) \equiv 0$. These systems are called *driftless systems*. Kinematic models of WMRs represent such systems. By contrast, dynamic models are systems with drift—also under zero control the vehicle continues to move (momentum conservation).

driftless systems

3.3 Degrees of Freedom

Considering a free mass point in three-dimensional space, three quantities (x, y, z) are needed to describe its position in Cartesian coordinates. In case of any other coordinate system (like the polar one, (r, φ, ϑ)) three coordinates are still needed. This is because of the three-dimensional space [51]. Therefore, in the presence of external force, the point can generally move in three directions and so, it has three *degrees of freedom (DOFs)*. However, there can be one or more conditions restricting the movement of the point and thus reducing the number of degrees of freedom. For example, the movement can be restricted to a plane ($f(x, y, z) = 0$) or a curve ($f_1(x, y, z) = 0$, $f_2(x, y, z) = 0$). Three conditions would restrict the mass point to stay at one point—the intersection of the planes. In this degenerative case, the mass point would have zero DOFs.

degree of freedom (DOF)

Differential degrees of freedom (DDOFs) are independent *velocities* of motion (in Cartesian coordinates $\dot{x}, \dot{y}, \dot{z}$).

differential DOF (DDOF)

[1] A smooth function is a function that has derivatives of all orders defined and continuous.

3.4 Vector Fields and Spaces

A *vector* is an object which, as opposed to a scalar, has its magnitude as well as a direction. Let n be a fixed number. Then an n-tuple $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, where $a_i \in \mathbb{R}$, is an *n-dimensional real vector* (complex vectors are defined similarly but their components are complex numbers). **vector - real, complex**

A *vector field* can be defined using smooth functions $h_i(q_1, \ldots, q_n)$ of state $\mathbf{q}$ **vector field**

$$\mathbf{h}(\mathbf{q}) = \begin{pmatrix} h_1(q_1, \ldots, q_n) \\ \vdots \\ h_m(q_1, \ldots, q_n) \end{pmatrix}, \tag{3.2}$$

because it defines a vector at each point of state space [56]. In general, the vector field defines a vector at each point of (locally) Euclidean space. The (locally) Euclidean space can be, e.g., $\mathbb{R}^n$. In physics, the vector fields are used to model, e.g., strength and direction of force in electric or magnetic fields.

Vector spaces are defined over fields[2] S, called *scalars*. If the scalars are real numbers $\mathbb{R}$, then the vector space V is $\mathbb{R}^n$. There exist more general vector fields but they are not considered further. **vector spaces**

A *vector space* V is a set of vectors with operations of *addition* and *multiplication*. The multiplication needed for definition of vector fields is a *scalar multiplication* (i.e., by a number), and it differs from a vector multiplication, which is another type of operation. **vector space–definition**

To precisely define a vector space, it is necessary to begin more generally. Let M be a *smooth manifold*. In the case of real vector fields considered in this text it is sufficient to put $M = \mathbb{R}^n$, $\dim M = n$. Smooth means it is of class C^∞ (derivatives of an arbitrary order are continuous).

A smooth map $\gamma : I \to M$ is called a *curve* in M, where I is an interval in $\mathbb{R}$. The initial point of the curve is $\gamma(0) = P_0$ (a fixed point $P_0 = (p_0, p_1, \ldots, p_n)$). Then, $\frac{d\gamma}{dt}(0)$ is a *tangent vector* to γ in P_0. All tangent vectors to all curves γ going through the point P_0 form an n-dimensional vector space, tangent space in P_0, denoted $T_{P_0}M$. Let TM be a disjoint union **tangent vector**

$$TM = \bigcup_{P_0 \in M} T_{P_0}M. \tag{3.3}$$

The vector field is a smooth section $X : TM \to M$. Each vector field has a *basis*. A basis is a set of linearly independent vectors. Using a linear combination of basis vectors, any vector from the vector field can be obtained. **vector field–basis**

[2]A *field* is an object from algebra that has two operations—addition and multiplication along with additional axioms (associativity, commutativity, distributivity, identities, additive and multiplicative inverse). As an example, real numbers $\mathbb{R}$ can be mentioned.

3.4.1 Vector Fields—Axioms

Along with the two operations of addition and multiplication, additional axioms need to be satisfied. It should be noted that different subsets of the following axioms can be found in the literature in definitions of vector fields (the symbol $\cdot$ is used to emphasize multiplication where necessary):

vector fields–axioms

1. Commutativity — $\forall\, \mathbf{a}, \mathbf{b} \in V$, $\mathbf{a}+\mathbf{b} = \mathbf{b}+\mathbf{a}$ (the vector space V is a commutative group under addition operation);
2. Associativity — $\forall\, \mathbf{a},\mathbf{b},\mathbf{c} \in V$, $\mathbf{a} + (\mathbf{b}+\mathbf{c}) = (\mathbf{a}+\mathbf{b}) + \mathbf{c} = (\mathbf{a}+\mathbf{c}) + \mathbf{b}$;
3. Null element (Null vector) — there exists such a vector (null vector) $\mathbf{0} = (0, \ldots, 0)$ that $\mathbf{a} + \mathbf{0} = \mathbf{a}$ (null vector can be added to any other vector resulting in the same vector); also $\mathbf{a} \cdot 0 = \mathbf{0}$, $c \cdot \mathbf{0} = \mathbf{0}$;
4. Additive inverse — $\forall$ $\mathbf{a}$ always exists $\mathbf{b}$, $\mathbf{a}+\mathbf{b} = \mathbf{0}$;
5. Distributivity of scalar addition — $\mathbf{a} \cdot (c+d) = c \cdot \mathbf{a} + d \cdot \mathbf{a}$;
6. Distributivity of vector addition — $c(\mathbf{a}+\mathbf{b}) = c \cdot \mathbf{a} + c \cdot \mathbf{b}$;
7. Associativity of scalar multiplication — $c(d \cdot \mathbf{a}) = (c \cdot d)\mathbf{a}$;
8. Equality $c \cdot \mathbf{a} = \mathbf{0}$ is true iff $c = 0$ or $\mathbf{a} = \mathbf{0}$;
9. $-(c \cdot \mathbf{a}) = (-c)\mathbf{a} = c(-\mathbf{a})$.

3.4.2 Vector Fields and Spaces in Control of WMRs

The vector fields and vector spaces introduced above are important for the theory of control systems (or WMRs, respectively).

If a driftless control system in the form

$$\dot{\mathbf{q}} = \sum_{i=1}^{m} \mathbf{h}_i(\mathbf{q})u_i \tag{3.4}$$

is considered, then $\mathbf{h}_1, \ldots, \mathbf{h}_m$ are the *system vector fields*. The action variables (controls) $u_i \in \mathbb{R}$ can be considered as coefficients (or weights) used to combine the system vector fields $\mathbf{h}_i$ to $\dot{\mathbf{q}}$. Therefore they determine how each of the system vector fields affects (contributes to) the time development of the system state variables ("velocity" of each state). An important condition is that the control space $\mathcal{U}, \mathbf{u}_i \in \mathcal{U}$, contains at least an open set that contains the origin of $\mathbb{R}^m$. Otherwise, the system is not driftless [24] (an intuitive explanation is that the zero control input is impossible and thus the system cannot "stop").

system vector fields

The equation (3.4) expresses all allowable velocities of the system and is referred to as the *configuration transition equation*.

configuration transition eqn.

Kinematic models of WMRs (introduced in Sect. 4.2.3) are usually driftless systems (no motion occurs under zero control). The situation is different in case of dynamic models—the dynamics introduces a drift, due to the momentum conservation law (a moving mass point (i.e., the abstraction of a moving WMR) continues to move also under zero control action).

3.5 Distributions and Foliations

Information in this section are taken from seminar text [22] and the seminar itself.

distribution

Distribution of a vector field is a map that assigns to each point $x_0 \in M$ a k-dimensional vector subspace $D_{x_0}M$ of $T_{x_0}M$, where $k < n$.

If a distribution is formed as[3] $D(\mathbf{q}) = \mathrm{span}(\mathbf{h}_1(\mathbf{q}), \ldots, \mathbf{h}_n(\mathbf{q}))$, where $\mathbf{h}_1(\mathbf{q}), \ldots, \mathbf{h}_n(\mathbf{q})$ are *linearly independent* vector fields, then the dimension of the distribution is constant everywhere and D is called a $k-$*distribution*. Dimension of a vector space equals to the number of its basis vectors. Therefore, $\dim D = n$.

Let N be a n-dimensional submanifold of M, $n \leq k$. Then, if $T_{x_0}N \subseteq D(x_0)$, N is an *integral manifold* of D. If D is not contained in any strictly larger integral manifold, then it is called *maximal integral manifold*.

k-distribution

An integrable distribution is the $k-$distribution D on M, if each point of M lies in some integral manifold of D.

In general case, the generating vector fields (the basis) can be arbitrary and the dimension k of $D(\mathbf{q})$ can vary in different points. Then, the distribution is not a k-distribution and is called *distribution* only.

involutive distribution

A distribution D is called *involutive*, if any of the Lie brackets $[X, Y]$, $X, Y \in D$, also belong to D. This means that the Lie brackets do not generate any new motions not belonging to D. More on Lie bracket in Sect. 3.8.2.

3.6 System Constraints

types of constraints

A system (including WMR) can be subject to *constraints*, which affect its possible motions. There are many types of constraints, which a vehicle can be subject to (e.g., configuration constraints, dynamical constraints, integral constraints). A brief overview can be found in [50]. The *configuration constraints* limits the possible configuration of a vehicle by delimiting forbidden areas in its configuration or state space. *Dynamical constraints* restricts differential quantities, such as velocities or accelerations. *Dynamical bounds* are inequalities on the maximum possible speeds.

[3] $\mathrm{span}(\mathbf{h}_1, \ldots, \mathbf{h}_n)$ is a linear span or linear hull. It can be defined as a set (union) of all linear combinations of the vectors $\mathbf{h}_1, \ldots, \mathbf{h}_n$. Therefore the vectors $\mathbf{h}_i$ constitute a basis of the vector space defined by $\mathrm{span}(\mathbf{h}_1, \ldots, \mathbf{h}_n)$.

3.7 Differential Constraints

The last group of constraints mentioned here will be the *differential constraints* that can be divided into two subgroups—*holonomic* and *nonholonomic*. These constraints will be discussed in the next part of the text. **differential constraints**

The conditions can also explicitly depend on time t. This is more general case, **time-dependent constraints**

$$f(x, y, z, t) = 0. \tag{3.5}$$

Nevertheless, this condition also reduces the number of degrees of freedom by one, the same way as the time-independent condition does[4].

Differentiation of (3.5) gives

$$\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + \frac{\partial f}{\partial z}\dot{z} + \frac{\partial f}{\partial t} = 0. \tag{3.6}$$

This equation possesses the coordinates (x, y, z) and time t, but also derivatives $\dot{x}, \dot{y}, \dot{z}$, which represents the components of velocity. This equation is integrable, and the result of integration is

$$f(x, y, z, t) = C, \tag{3.7}$$

with constant C independent on time t.

Further generalization of this equation results in **generalization**

$$f(x, y, z, t, \dot{x}, \dot{y}, \dot{z}) = 0, \tag{3.8}$$

which is in general nonlinear in $\dot{x}, \dot{y}, \dot{z}$. This is the most general form of this type of constraint.

3.7.1 Implicit and Parametric Form

These are the two ways of expressing the constraints on velocities. The intuitive difference is that from the *implicit form* (e.g., $\dot{x} \geq 0$) of constraint the velocities that are *prohibited* can be determined in straightforward manner, whereas the *parametric form* **two forms – differences**

$$\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u}),\ \mathbf{q} \in \mathcal{Q},\ \mathbf{u} \in \mathcal{U} \tag{3.9}$$

the velocities that are *allowed*. The manifold $\mathcal{Q}$ is a state space, $\mathcal{U}$ is a set of allowable actions.

The parametric form is especially interesting for kinematic modelling. The equation (3.9) is called *state transition equation* or *configuration transition equation* or simply *system* and the vector form presented represents a set of k scalar equations, where $k = \dim(\mathcal{Q})$.

[4]Examples: A point fixed by one constraint condition to a plane that moves in space; any point on the Earth's surface that has two degrees of freedom (longitude and latitude) regardless movements of the Earth (rotation (spinning on its axis) and revolution (orbit around the Sun))

3.7.2 Holonomic and Nonholonomic Constraints

The constraints introduced in Sect. 3.7 are integrable or nonintegrable, time-dependent or time-independent. They can be sorted as follows:

Integrable, time-independent constraint is *holonomic* and *scleronomous*

Integrable, time-dependent constraint is *holonomic* and *rheonomous*[5]

Nonintegrable, time-independent is *nonholonomic* and *scleronomous*

Nonintegrable, time-dependent is *nonholonomic* and *rheonomous*

It should be noted that the integrability property is in the sense of ability to express the constraint in position (non-derived) variables only.

The type of constraint represented by the equation (3.7) is called *semiholonomic*, because the constant resulting from integration is arbitrary. Holonomic constraint needs a specific, not an arbitrary constant. **semiholonomic constraint**

The terms *holonomic* and *nonholonomic (also: anholonomic)* have been introduced by Heinrich Hertz (1857 - 1894) (from Greek words *holos* = whole (= integrable), *nomos* = law) for description of constraints in classical mechanics. From here, the idea had spread to many other fields of physics (thermodynamics, quantum mechanics etc.).

The terms scleronomous and rheonomous (from Greek *scleros* = hard, *rheo* = (I) flow) have been brought off by Ludwig Boltzmann (1844 - 1906) [51].

The properties of holonomic and nonholonomic constraints are summarized in the following text.

Holonomic Constraints

A *holonomic constraint* is a geometric constraint. This means it limits allowable positions (configurations) of the system and therefore *reduces the number of degrees of freedom* of the system. The equation of the constraint contains the generalized coordinates only ($\mathbf{A}(\mathbf{q})\mathbf{q} = \mathbf{0}$), there are no generalized velocities ($\dot{\mathbf{q}}_n$). **holonomic constraint**

From control theory viewpoint, holonomic constraint represents a *static system.* This means, that no dynamics is present (there are no integrators) and it is a function of the position variables only.

Nonholonomic Constraints

A *nonholonomic constraint* is a kinematic constraint that cannot be integrated. If this was true, the constraint would become holonomic. The fact that nonholonomic constraints are kinematic means that they limit allowable velocities. They don't limit allowable geometric configurations and thus don't reduce the number of degrees of freedom. **nonholonomic constraint**

[5]Some literature sources use the term "rhenomorous", which, as seen consequently, doesn't comply with original Greek term from which it is derived.

From control theory viewpoint, a nonholonomic constraint can be regarded as a *dynamic system.* In this type of system, integrators (with states) are present and thus it has a memory (i.e., the current state depends also on the previous ones).

3.7.3 Distinction of Constraints

Once the two types of differential constraints have been defined, a question how to distinguish them arises.

If the constraints are integrable, they are holonomic, if not, they are nonholonomic. As has been noted above, the mentioned integrability *doesn't mean numeric integration*, but expressing the constraint using non-derived variables only (i.e., using positions, not velocities). **integrability**

Three methods can be used to distinguish if the constraints are holonomic or nonholonomic [30]:

- Integration—holonomic constraints are integrable, therefore it is possible to try to integrate the constraint (please note the remark on integration above)
- Reduction of DOFs—checking the number of DOFs of the system, holonomic constraints reduce the number of DOFs
- Frobenius' theorem (using Lie brackets)—Sect. 3.8.4

There exists another method for constraints classification, that could be denoted "practical". This method follows from Frobenius' theorem. To demonstrate the method, one holonomic and one nonholonomic representative system will be used. Holonomic systems are represented by a two-link planar manipulator, as depicted in Fig. 3.1. The manipulator is driven by two motors (1 and 2) located at the rotary joints, while the first rotary joint is located at the origin of coordinate system. The state variables are the angles of both joints, selected as depicted in Fig. 3.1, without loss of generalization. **practical method**

Nonholonomic systems are represented by a car, as depicted in Fig. 3.2. For the purposes of this example, two variables are important, the distance travelled by the car, denoted s, and the steering angle ϑ.

The method is based on checking whether change of sequence of control inputs results in different final state. For the planar manipulator, the control inputs are commands for its motors such as $\alpha_1 = \alpha_{1F}$ and $\alpha_2 = \alpha_{2F}$ (as can be seen the real commands (control signals) are not important). For the car, the control commands are "go by the distance s_F" and "turn the front wheel by the angle ϑ". **method principle**

The sequence of these control inputs is applied in normal and then reversed order. If the final state of the system is the same in both cases, the system is *holonomic*. If not, the system is *nonholonomic*. **control sequences**

For planar manipulator, the first sequence is $\alpha_1 \to \alpha_{1F}$, then $\alpha_2 \to \alpha_{2F}$ (Fig. 3.3) and the second sequence is reversed, i.e., $\alpha_2 \to \alpha_{2F}$, then $\alpha_1 \to \alpha_{1F}$ (Fig. 3.4). As the resulting state is the same for both sequences, it can be concluded that the planar manipulator is a holonomic system.

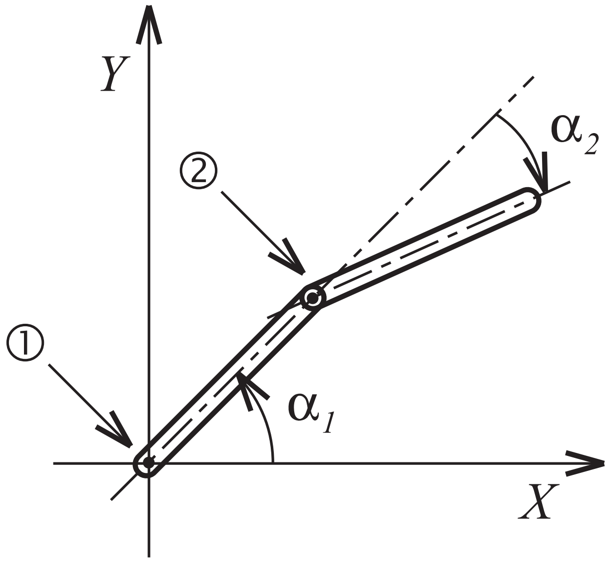


Figure 3.1: *Holonomic system—a two-link planar manipulator*

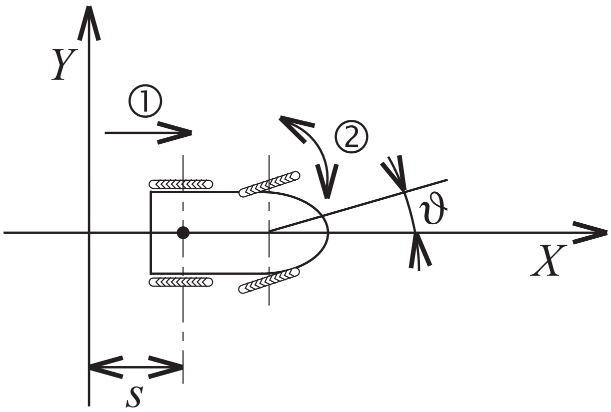


Figure 3.2: *Nonholonomic system—a car*

For the car, the first control sequence is $s \to s_F$, then $\vartheta \to \vartheta_F$ (Fig. 3.5), and the reversed one $\vartheta \to \vartheta_F$, then $s \to s_F$ (Fig. 3.6).

The resulting state is different for each sequence and therefore the system is nonholonomic.

Note: a set of constraints is called Pfaffian if it is linear in velocities, i.e., **Pfaffian constraints**

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}, \tag{3.10}$$

where $\mathbf{A}(\mathbf{q})$ *is the* constraint matrix*.*

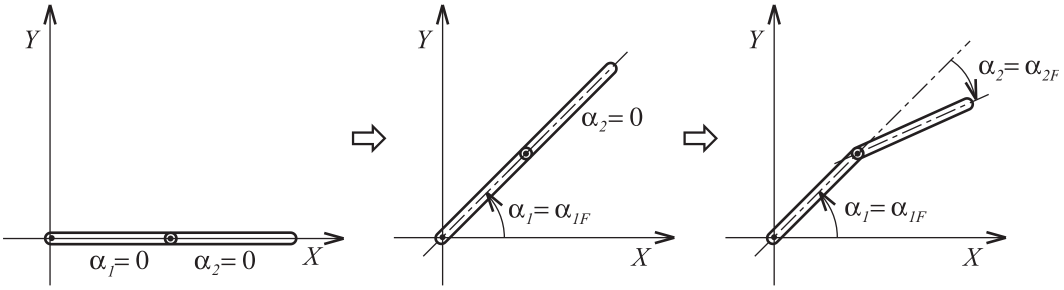


Figure 3.3: *Planar manipulator—control sequence 1*

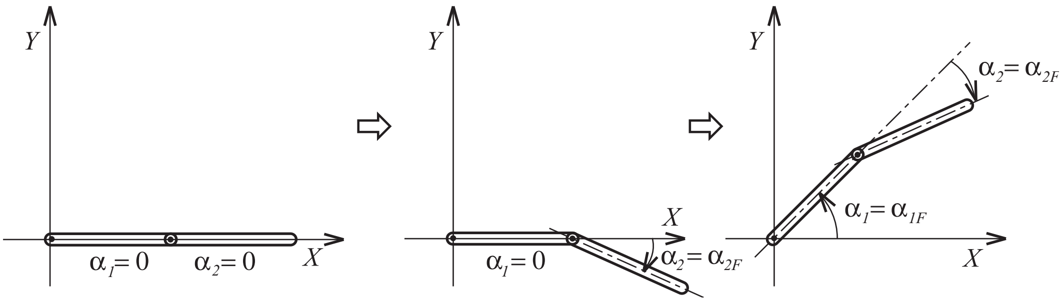


Figure 3.4: *Planar manipulator—control sequence 2*

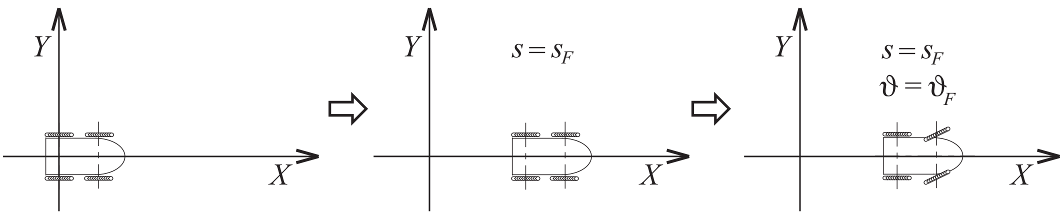


Figure 3.5: *Car—control sequence 1*

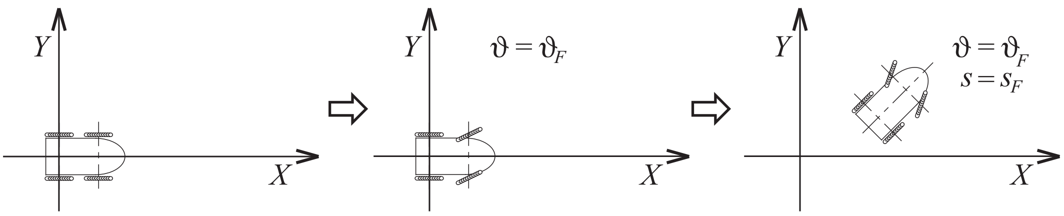


Figure 3.6: *Car—control sequence 2*

3.7.4 WMR Classification by Constraints

WMRs are classified based on number n of *nonholonomic constraints* present:

$n = 0$ characterizes *holonomic* WMR,

$n > 0$ characterizes *nonholonomic* WMR.

Examples of holonomic robots are omnidirectional robot ($n = 0$), train and car with **holonomic robots** locked steering. A question may arise why, e.g., the last robot is holonomic, while the (non-locked) car is nonholonomic. The answer is that the constraints are integrable. The position of the robot is a function of an angle of rotation of the wheels (traveled distance directly corresponds to the angle of which the wheels are turned). Moreover, turning the wheels by $\varphi(t)$ and $-\varphi(t)$ brings the robot to the same configuration. This is true also for the train.

nonholonomic robots

Nonholonomic robots contain fixed or steered standard wheels. The presence of these wheels introduces the nonholonomic constraints into the system. The number of the constraints is not necessarily equal to the number of the wheels present on the robot. For example, two co-axial wheels introduce only one constraint.

3.8 Differential Geometry in Mobile Robotics

In this section, a concept of Lie bracket, Lie algebra and Frobenius' theorem will be presented. These tools from differential geometry are useful for checking controllability of WMRs.

3.8.1 Lie Derivative

Lie derivative

A *Lie derivative* represents a derivative of a scalar function in direction of a vector field[6]. Given a smooth function of state $\boldsymbol{f}(\mathbf{q})$ in addition to a vector field $\mathbf{h}$ as defined in Sect. 3.4, the Lie derivative of function $\boldsymbol{f}(\mathbf{q})$ along a vector field $\mathbf{h}$ is a new function defined as

$$L_{\mathbf{h}}\mathbf{f}(\mathbf{q}) = \sum_{i=1}^{n} \frac{\partial \mathbf{f}(\mathbf{q})}{\partial q_i} h_i(\mathbf{q}). \tag{3.11}$$

Note: In mathematical texts (like [56], [24]) it is common to denote the state x, the vector field f and the function $h(x)$. Then the definition of the Lie derivative is

$$L_f h(x) = \sum_{i=1}^{n} \frac{\partial h(x)}{\partial x_i} f_i(x) \tag{3.12}$$

and in compact vector notation (Einstein summation, refer to Note on notation),

$$L_f h(x) = \frac{\partial h(x)}{\partial x_i} f_i(x), \frac{\partial h(x)}{\partial x} = \left[\frac{\partial h(x)}{\partial x_1}, \ldots, \frac{\partial h(x)}{\partial x_n}\right]. \tag{3.13}$$

3.8.2 Lie Bracket

A nonholonomic system (e.g., a car) clearly cannot produce certain moves instantaneously (e.g., cannot move sideways). However, by appropriate control, the desired

[6] It should be noted that there exist more definitions of Lie derivative that are equivalent, but this variant will be used here.

sideways motion can be achieved (parallel parking[7]). Therefore, the system can produce movements that are not directly allowed and that cannot be discovered by examining the distribution of the original vector fields only.

To check these additional possible motions[8], a Lie bracket can be used. A Lie bracket checks for possible motions of a system considering its motion on two distinct vector fields. **Lie bracket purpose**

The Lie bracket is defined as

$$[X,Y]^i = \sum_{j=1}^{n} \left(X^j \frac{\partial Y^i}{\partial x^j} \right) - \left(Y^j \frac{\partial X^i}{\partial x^j} \right). \tag{3.14}$$

In differential geometry notation, the automatic summation over j is assumed (so-called *Einstein summation*)—please refer to note about differential geometry notation in Chap. 3.

The Lie bracket possesses the following properties

1. $[.,.]$ is *bilinear*,
2. $[.,.]$ is *antisymmetric* ($[X,Y] = -[Y,X]$),
3. Jacobi identity holds, $[X,[Y,Z]] + [Y,[Z,X]] + [Z,[X,Y]] = 0$.

The important condition is that the motion on the fields *never* occur *simultaneously*. For a driftless system

$$\dot{\mathbf{q}} = \mathbf{h}_1(\mathbf{q})u_1(t) + \mathbf{h}_2(\mathbf{q})u_2(t) \tag{3.15}$$

where $\mathbf{u} = (u_1(t), u_2(t))$ is defined as

$$\mathbf{u}(t) = (u_1(t), u_2(t)) = \begin{cases} (1,0) & \text{for} \quad t \in [0,\varepsilon] \\ (0,1) & \text{for} \quad t \in [\varepsilon, 2\varepsilon] \\ (-1,0) & \text{for} \quad t \in [2\varepsilon, 3\varepsilon] \\ (0,-1) & \text{for} \quad t \in [3\varepsilon, 4\varepsilon] \end{cases} \tag{3.16}$$

it is possible to compute the Taylor expansion of the first part of the motion

$$\mathbf{q}(\varepsilon) = \mathbf{q}(0) + \varepsilon \dot{\mathbf{q}}(0) + \frac{1}{2}\varepsilon^2 \ddot{\mathbf{q}}(0) + O(\varepsilon^3) \tag{3.17}$$

and for the second part as

$$\begin{aligned} \mathbf{q}(2\varepsilon) &= \mathbf{q}(\varepsilon) + \varepsilon \dot{\mathbf{q}}(\varepsilon) + \tfrac{1}{2}\varepsilon^2 \ddot{\mathbf{q}}(\varepsilon) + O(\varepsilon^3) = \\ &= \mathbf{q}(\varepsilon) + \varepsilon \mathbf{h}_2(\mathbf{q}(\varepsilon)) + \tfrac{1}{2}\varepsilon^2 \tfrac{\partial \mathbf{h}_2}{\partial \mathbf{q}}(\mathbf{q}(\varepsilon))\mathbf{h}_2(\mathbf{q}(\varepsilon)) + O(\varepsilon^3). \end{aligned} \tag{3.18}$$

Substituting (3.17) into (3.18) and using the fact that for the infinitesimal ε

[7]Not only when a driver parks a car into a lane of cars but in infinitesimal sense, the (seemingly) sideways motion can be produced.

[8]Here, the *infinitesimal* motions are assumed. They are infinitesimal in differential sense, i.e., infinitesimally small.

$$\mathbf{h}_2(\mathbf{q}(0)+\varepsilon\mathbf{h}_1(\mathbf{q}(0))) = \varepsilon\frac{\partial\mathbf{h}_2}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_1(\mathbf{q}(0)) \tag{3.19}$$

holds, it is obtained

$$\begin{aligned}
&\mathbf{q}(2\varepsilon) = \mathbf{q}(0) + \varepsilon\mathbf{h}_2(\mathbf{q}(0)) + \\
&+\varepsilon^2\left(\tfrac{1}{2}\tfrac{\partial\mathbf{h}_1}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_1(\mathbf{q}(0)) + \tfrac{\partial\mathbf{h}_2}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_1(\mathbf{q}(0)) + \tfrac{1}{2}\tfrac{\partial\mathbf{h}_2}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_2(\mathbf{q}(0))\right) \\
&+O(\varepsilon^3).
\end{aligned} \tag{3.20}$$

The same approach is used for $\mathbf{q}(3\varepsilon)$ and $\mathbf{q}(4\varepsilon)$ to obtain

$$\mathbf{q}(4\varepsilon) = \mathbf{q}(0) + \varepsilon^2\left(\frac{\partial\mathbf{h}_2}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_1(\mathbf{q}(0)) - \frac{\partial\mathbf{h}_1}{\partial\mathbf{q}}(\mathbf{q}(0))\mathbf{h}_2(\mathbf{q}(0))\right) + O(\varepsilon^3), \tag{3.21}$$

which is the same result as in [22] and [24].

The Lie bracket (and the computation above) shows that the motion is not possible in the directions of the original vector fields only but also in the directions of their Lie brackets. Moreover, it is possible to obtain motions in directions of higher-order Lie brackets, e.g., $[\mathbf{h}_1, [\mathbf{h}_1, \mathbf{h}_2]]$.

The reason for relatively complicated definition of Lie bracket is that it is coordinates independent, i.e., the change of coordinates does not affect the Lie bracket [56].

3.8.3 Lie Algebra

Lie brackets can determine possible motions from given configuration $\mathbf{q}$ that are not allowed by original system vector fields. Using higher-order Lie brackets, it is possible to find additional linearly independent vector fields. This set of vector fields is called the *Lie algebra*. It is limited in size, and the maximum number of vector fields equals to the dimension of the state space Q. If the last condition is true, then the system is small-time local controllable (STLC) (as described in Sect. 3.9).

Lie algebra

Philip Hall Basis

A Lie algebra defines a vector space. Its basis can contain the original system vector fields and/or their Lie brackets, including nested brackets (such as $[\mathbf{h_1}, [\mathbf{h_2}, \mathbf{h_3}]]$). The difficulty is with determining if the Lie brackets will produce linearly-independent vectors. Also it is not possible in advance to determine the necessary depth of the Lie brackets needed for the basis. Therefore an iterative method should be used [24].

basis of Lie algebra

An intuitive solution could be as follows. Start forming the basis from the original vectors $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$ and then add their Lie brackets to the basis (use nested brackets if needed). With each addition, check for the linear independency.

The outlined process leads to the Philip Hall basis (or P. Hall basis). Its construction is a breadth-oriented procedure. The *order* or *depth*, denoted d, means the number of

order of depth

nested levels of Lie brackets used and is defined as follows. For the system vector fields, let $d(\mathbf{h}_i) = 1$, for Lie brackets $[\mathbf{g}_1, \mathbf{g}_2]$ let $d([\mathbf{g}_1, \mathbf{g}_2]) = d(\mathbf{g}_1) + d(\mathbf{g}_2)$.

The order of a Lie bracket thus equals to level of nesting increased by one (this follows from definition) or it is equal to the number of the vector fields involved in the bracket, i.e., $d([\mathbf{h}_1, \mathbf{h}_2]) = 2$ and $d([\mathbf{h}_1, [\mathbf{h}_2, \mathbf{h}_3]]) = 3$ (more empirical result).

The breadth-oriented search must be accompanied by pruning process. This involves removal of redundant vector fields using skew-symmetry and Jacobi identity properties of the Lie brackets (as described in Sect. 3.8.2).

3.8.4 Frobenius' Theorem

This theorem, introduced by Fredinand Georg Frobenius (1849-1917), a German mathematician, can be used to determine whether a system is holonomic (completely integrable) or nonholonomic (not completely integrable). For more information on holonomic and nonholonomic constraints and systems please refer to Sect. 3.6.

Required tool from differential geometry for Frobenius' theorem is the Lie bracket (introduced in Sect. 3.8.2).

This theorem states that

A system is completely integrable if and only if it is involutive. **Frobenius' theorem**

For involutivity please refer to Sect. 3.5.

For a driftless system in the form

$$\dot{\mathbf{q}} = \sum_{i=1}^{m} \mathbf{h}_i(\mathbf{q}) u_i, \tag{3.22}$$

the Lie brackets $[\mathbf{h}_i, \mathbf{h}_j], i < j$ are created. If these brackets produce motions that already belong to the system distribution (and thus they do not produce any new motions), the system is integrable and consequently holonomic. This is true if these Lie brackets are linear combination of original system vector fields $\mathbf{h}_i$.

These brackets are not all possible combinations. As $[\mathbf{h}_i, \mathbf{h}_j] = -[\mathbf{h}_j, \mathbf{h}_i]$ and $[\mathbf{h}_i, \mathbf{h}_i] = 0$, it is necessary and sufficient to check the Lie brackets created as the combinations without repetitions from the system vector fields. The number of such combinations is $C(m, 2)$.

The aforementioned property of holonomic systems that their Lie brackets are linear combinations of the original system vector fields can be used to check the holonomic/nonholonomic nature of a system. If any of formed Lie brackets is *not* a linear combination of the system vector fields, then the system is *nonholonomic* (and the checking can be stopped) [24].

Easiest way is to form a matrix $\mathbf{H}(\mathbf{q})$ from the system vector fields $\mathbf{h}_i$, $i = 1, \ldots, m$ and then to append each created Lie bracket to it. If the rank of such extended matrix $\mathbf{H}_{ext}(\mathbf{q})$ is $m + 1$, then the system is nonholonomic.

3.9 Stabilizability and Controllability

Stabilizability within context of this text is a simplified term that refers to a property of a WMR that it can be stabilized by a smooth, time-invariant feedback. A tool for checking this property is called Brockett's theorem (please refer to Sect. 3.10). **stabilizability**

Controllability is a term from control theory. The intuitive definition is that given initial and goal configurations $\mathbf{q}_i$ and $\mathbf{q}_g$, the system is controllable if a control signal (this includes also a sequence of control signals) can be found such that it brings the system from $\mathbf{q}_i(0)$ to $\mathbf{q}_g(t)$ in a finite time $t > 0$. The control signal is from space of admissible control inputs $\mathcal{U}$. **controllability**

All types of WMRs presented in this text are controllable. However, it should be noted that the controllability can be restricted by obstacles in the configuration space. This is important when solving path planning problems. For example, for Dubins' car[9] it can be impossible to reach the goal configuration in the presence of obstacles.

The intuitive definition of controllability presented above can be easily understood, however checking it on the mathematical basis requires appropriate apparatus.

For controllability characterization of a driftless control-affine system it is possible to use the *Lie algebra rank condition* (LARC). It is assumed that the control space $\mathcal{U}$ contains an open set that includes the origin of $\mathbb{R}^m$. Then, the Chow-Rashevskii theorem states: **LARC**

A driftless control-affine system is small-time locally controllable (STLC) at a point $\mathbf{q} \in \mathcal{Q}$ *if and only if* $\dim \mathcal{L}(\Delta) = n$*, the dimension of* $\mathcal{Q}$ *(*$n = \dim \mathcal{Q}$*).* **Chow-Rashevskii theorem**

If this condition is true $\forall \mathbf{q} \in \mathcal{Q}$, then the whole system is STLC.

For a driftless system with $\mathcal{U} \in \mathbb{R}^m$, $\mathcal{Q} \in \mathbb{R}^n$, $m < n$, the integrability can be characterized based on $\dim \mathcal{L}(\Delta)$ as follows [24]:

1. $\dim \mathcal{L}(\Delta) = m$ system is completely integrable (holonomic),
2. $m < \dim \mathcal{L}(\Delta) < n$ system is nonholonomic, but not STLC,
3. $\dim \mathcal{L}(\Delta) = n$ system is nonholonomic and STLC.

3.10 Brockett's Theorem

This theorem has been introduced in [9] by R. W. Brockett (Roger Ware Brockett (*1938), control theorist). It deals with necessary and sufficient conditions for feedback stabilizability of continuously differentiable systems at a point by continuously differentiable feedback. **purpose**

For a continuously differentiable system

$$\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u}); \, \mathbf{q} \in \mathcal{Q}, \mathbf{u} \in \mathcal{U} \tag{3.23}$$

it is necessary to find a continuously differentiable feedback law

[9] Dubins' car [17]—this is a special vehicle, for which an optimal path planning was introduced by Dubins. Its motions are restricted to lines or arcs of upper-bounded radius and in forward direction only (the car cannot reverse). Originally used to model motion of a particle in electrical field.

$$\mathbf{u} \equiv \mathbf{u}(\mathbf{q}) \tag{3.24}$$

that brings the system (3.23) to a state $\mathbf{q_0}$ with $f(\mathbf{q}_0, 0) = 0$ and also makes $\mathbf{q}_0$ asymptotically stable. For such a law the following conditions must hold (excerpt from [50]):

stabilizing law conditions

1. There exists a neighbourhood ε of $\mathbf{q}_0$ within which each point $\mathbf{q}$ has a control sequence $\mathbf{u_q}(t)$ steering it to $\mathbf{q}_0$ for $t \to \infty$.

2. The linearized system must have no uncontrollable unstable modes at $\mathbf{q_0}$.

3. The mapping $f : (\mathbf{q}, \mathbf{u}) \to f(\mathbf{q}, \mathbf{u})$ must be onto an open neighbourhood at 0 (this means that all state space directions near $\mathbf{q}_0$ must be spanned by possible controlled motions).

Important is the third condition which results in the fact that *no* system with differential constraints can be stabilized to a point with a continuously differentiable feedback law, due to the limitation of the constrained controls (or actions, more precisely). They cannot span $T_{\mathbf{q}_0}M$, where $T_{\mathbf{q}_0}M$ is tangent space of linear vectors in $\mathbf{q}_0$.

For a driftless system in the form

$$\dot{\mathbf{q}} = \sum_{i=1}^{m} \mathbf{h}_i(\mathbf{q}) u_i, \tag{3.25}$$

the equivalent condition is that rank(span $\mathbf{h}_1 | \dots | \mathbf{h}_n) = n$. If the rank is smaller than n (the state dimension), the system fails to be stabilizable by a continuous time-invariant feedback [3], [27].

For example, the unicycle has $\dim(\mathcal{Q}) = 3$ and $\dim(\mathcal{U}) = 2$. This means that $\dim(\mathcal{Q}) \neq \mathcal{U}$ and therefore it is not stabilizable by continuously differentiable static feedback.

Generally, WMRs that contain standard fixed or steered wheels are subject to differential constraints. This implies that they do not satisfy the Brockett's theorem. However, these systems are open-loop controllable [10].

Chapter 4

Modelling of WMRs

For control algorithm design, it is suitable to create a model of the system (WMR). There are two basic types of models—mathematical and simulation one. In the successive text, the mathematical model is understood when referred to a "model". **basic models**

When modelling a WMR, one can resort, on principle, to one of two basic types of models—*kinematic* or *dynamic* ones. These two types of models differ significantly. The appropriate model type should be selected according to the nature and properties of the modelled system (WMR), expected simulation results, desired computational complexity of the model etc. **types of models**

To recapitulate, kinematic models do not consider the forces/torques acting on the system, whereas dynamic models describe the system considering them. Therefore, for the latter it is possible to model the force-related effect, like momentum of the vehicle and skid and slip of the wheels, which could be considered as the most significant from the viewpoint of motion study.

This work is focused on the kinematic models, and this type of modelling is considered in the following text, unless otherwise specified.

4.1 Terms Definition

In this section, the basic terms for modelling are defined—reference point, instantaneous centre of rotation, coordinate systems and homogenous transforms, generalized coordinates and velocities, state and configuration systems and kinematic simplification.

4.1.1 Reference Point

Because each WMR has certain physical dimensions, it is necessary to select a *reference point*. This point represents then the motion of the whole WMR. Its selection (its position within the WMR body, i.e., the position in local coordinates of the WMR) is essential for the resulting equations of motion. The point can lie within the body of the WMR as well as off the WMR body (the latter case is presented, in e.g., [15]). **reference point**

4.1.2 Instantaneous Centre of Rotation (ICR)

To introduce the Instantaneous Centre of Rotation (ICR), it is advantageous to start from Chasles' theorem. The ICR follows from this theorem.

Chasles' Theorem

The Chasles' theorem[1] (Giulio Mozzi 1773, Augustus-Louis Cauchy 1827, Michel Chasles 1830) is a basic theorem that describes kinematic motion of a rigid body. This theorem states (as presented in [51])

The most general rigid body displacement equals to a helical motion. The axis of this motion remains unchanged during the motion. For each motion it is possible to find a straight line connected to the rigid body which position is the same after the motion as before it. If the motion is not reduced to a pure translation, such a straight line is only one.

In other words, the most general rigid body displacement can be produced by a translation along a line followed (or preceded) by a rotation about that line [21].

Instantaneous Curvature Radius

For purposes of this work it is sufficient to discuss a planar case, for which there always exists a point connected to the rigid body that is fixed (remains unchanged) by the given motion.

instantaneous curvature radius

Resulting from the Chasles' theorem, a WMR always moves on a circular trajectory with the radius R—going on a straight line is a limit case of a circle ($R = \infty$). Therefore $R \in [0, \infty)$. This radius is usually not constant and it is changing over time rapidly along with the position of the centre of the instantaneous circle of motion. The radius of the circle is therefore called *instantaneous curvature radius.*

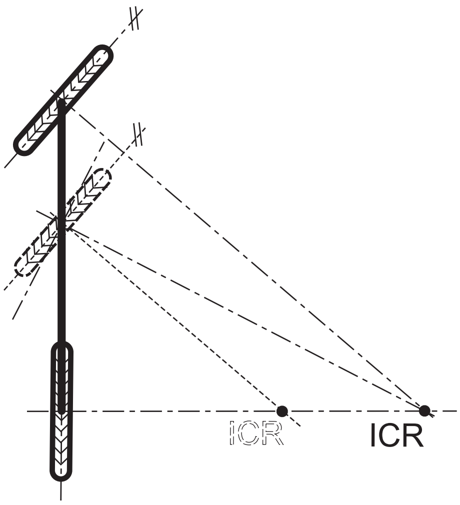


Figure 4.1: *Improper wheel steering*

[1] Michel Chasles, French mathematician, 1793-1880

ICR Existence

The position of the ICR is determined by orientation of the wheels. All horizontal axes are required to intersect at the (*only one*) point, the ICR (the *ICR existence condition*). If this requirement is not fulfilled, then the wheels would slip in lateral direction and thus violate the no lateral movement constraint. Considering the situation depicted in Fig. 4.1, the first front wheel is driven and therefore it is considered governing the velocity of the vehicle. The actual (incorrect) orientation of the second wheel with its ICR line (zero-motion line) is represented by the dashed line, whereas the desired (correct) one is represented by the dash-dotted line. Thus, a slip of this incorrectly-oriented wheel occurs (if dynamics is considered). In kinematics, the motion of the WMR is not possible. **ICR existence**

The explanation could also be based on geometric interpretation. A single wheel can have the ICR anywhere on its horizontal axis (an *ICR line*). Translational velocity of the wheel is perpendicular to the ICR line. If more wheels are present, then the ICR lies at the intersection of the ICR lines. If any of the ICR lines does not pass the ICR, then translational velocity of this wheel is forced to lie outside of the plane of the wheel with the effect described above. **ICR geometric interpretation**

4.1.3 Coordinate Systems and Homogenous Transforms

This section introduces important terms—coordinate systems and homogenous transforms. Coordinate system is a way to describe position of points in space. Homogenous transforms are useful for moving from one coordinate frame to another.

Coordinate Systems

In 3D case, the position of a point is usually determined by a triplet of real numbers (*coordinates*) and conversely, each triplet represents only one point [41]. **coordinates**

In $n-$dimensional space, position of a point is determined by an $n-$tuple. This $n-$tuple consists of numbers or scalars. In most cases (including the kinematic modelling), the scalars represent real numbers, in more general case they could be complex numbers or even members of a commutative ring[2] (the previous cases are also commutative rings). **n-dimensional case**

There exists many coordinate systems that can be used, such as Cartesian coordinate system, polar coordinate systems (circular, spherical, cylindrical) and Plücker coordinates. However, not all of them are advantageous to use in WMR modelling and control. **types**

The WMRs are in most cases modelled in 2D working environment. The reason is that the common tasks require control of the robot position and therefore a top-view (or bird's eye view) is sufficient. **2D case**

Well-known and widely-used are the first two systems, Cartesian and polar. Unless otherwise specified, the Cartesian coordinate system is considered within this text.

[2] A *ring* is an algebraic structure, that contains two operations, *addition* and *multiplication*. It generalizes algebraic properties of integers.

Cartesian coordinate system (also rectangular coordinate system) is named after René Descartes (Lat.: *Cartesius*) (1596-1650), the French mathematician and philosopher. Each point in a plane is determined by a pair of coordinates, commonly denoted x (*abscissa*) and y (*ordinate*). The 2D coordinate system is defined by two axes (the x and y axis) which are perpendicular, and a unit length (can differ on each axis), that enables measurement of distances. **Cartesian system**

This coordinate system can be extended to higher-dimensional spaces (generally $n-$dimensional), namely the 3D case with x, y and z axis.

Polar coordinate system is a coordinate system, in which a position of a point in a plane is determined by two coordinates, an *angle* and a *distance*. Use of this system can be advantageous in situations where distance-orientation pairs are known (such as radars, antenna characteristics etc.). Transformation between the Cartesian and polar coordinate system is possible using trigonometric (i.e., transcendental) functions. **polar system**

Local vs. global coordinate system of the robot. The *local coordinate system* $X_L Y_L$ is tied with the robot body and moves along with it. This coordinate system is usually oriented in the way such that the positive translational velocity is equal to $\dot{x}_L$. Using such a coordinate system, the no lateral motion condition (e.g., for a unicycle) can be expressed as $\dot{y}_L = 0$. **local system**

The *global coordinate system* XY is connected with the world, for kinematic modelling and control of WMRs usually represented by a 2D plane. The state (posture) of the robot is expressed in this coordinate system. For example, the translational velocity of the robot is expressed in global coordinate system as $v = \sqrt{\dot{x}^2 + \dot{y}^2}$. **global system**

Homogenous Transforms

Homogenous transforms are a powerful tool to express rigid motions. They combine two basic motions of a rigid body into single matrix multiplication.

If a coordinate system is considered, there exist two possible motion (or transforms) that can be applied to it—*rotation* and *translation*. The important property is that both of them *conserve scale* of the coordinate system. This means that by rotation and/or translation the position of the points are not changed relatively to the coordinate frame. Therefore, the resulting motion is called a *rigid motion*. **two basic motions**

A homogenous transform therefore consists of translation and/or rotation. The transforms can be sequenced in order to move between more coordinate frames, like in Denavit-Hartenberg[3] notation well-known in the are of robotic manipulators. If two coordinate frames i and j are defined, then it is possible to define a 4×4 matrix, usually denoted T_i^j, that converts a point in coordinates i to coordinates j. The coordinates of the point are expressed as a vector $(x, y, z, 1)^T$. Computation of resulting coordinates is determined easily by matrix multiplication.

[3] It should be noted that there exist two versions of D-H notation: the original one introduced by Denavit and Hartenberg, and the modified one introduced in [14] by Craig.

Development of homogenous transformation matrices is omitted and only the final form is presented herein. More complete coverage of the topic can be found in, e.g., [45].

Rotation

Rotation matrix can be used to express relation between two mutually rotated coordinate frames. A point $\mathbf{p}$ can be expressed in terms of two different coordinate frames, $0_{X_0Y_0Z_0}$ (fixed, nonrotated) and $0_{X_1,Y_1,Z_1}$ (rotated), as depicted in Fig. 4.2. Unit vectors for the fixed coordinate frame are denoted $(\mathbf{i}_0, \mathbf{j}_0, \mathbf{k}_0)$, unit vectors for the rotated frame are not depicted. **rotation matrix**

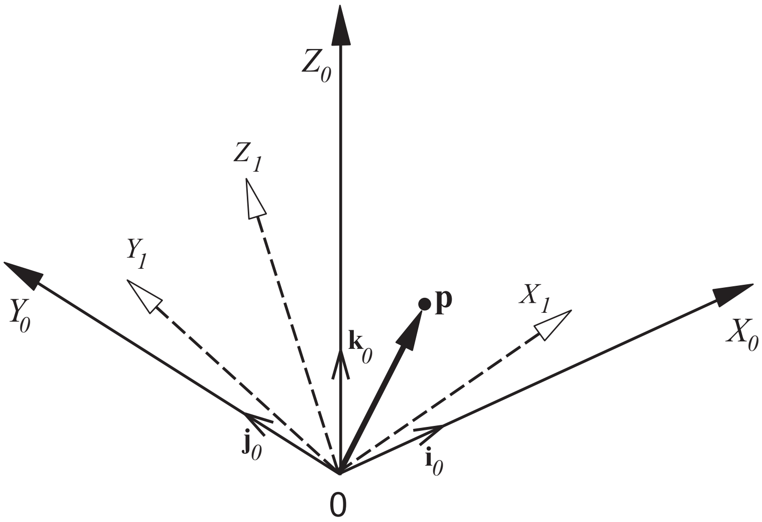


Figure 4.2: *Homogenous transform—a point in two rotated coordinate systems*

If the position of $\mathbf{p}$ is expressed in both coordinate systems, then it can be converted to the matrix form

$$\mathbf{p}_0 = \mathbf{R}_0^1 \mathbf{p}_1, \tag{4.1}$$

where $\mathbf{R}_0^1$ is the *rotation matrix* from the coordinate frame $0_{X_0Y_0Z_0}$ to the coordinate frame $0_{X_1,Y_1,Z_1}$ defined as

$$\mathbf{R}_0^1 = \begin{pmatrix} \mathbf{i}_1.\mathbf{i}_0 & \mathbf{j}_1.\mathbf{i}_0 & \mathbf{k}_1\mathbf{i}_0 \\ \mathbf{i}_1.\mathbf{j}_0 & \mathbf{j}_1\mathbf{j}_0 & \mathbf{k}_1\mathbf{j}_0 \\ \mathbf{i}_1.\mathbf{k}_0 & \mathbf{j}_1\mathbf{k}_0 & \mathbf{k}_1\mathbf{k}_0 \end{pmatrix} \tag{4.2}$$

and $\mathbf{p}_0 = (p_{0_x}, p_{0_y}, p_{0_z})$ and $\mathbf{p}_1 = (p_{1_x}, p_{1_y}, p_{1_z})$, or $\mathbf{p}_0 = p_{0_x}\mathbf{i}_0 + p_{0_y}\mathbf{j}_0 + p_{0_z}\mathbf{k}_0$ and $\mathbf{p}_1 = p_{1_x}\mathbf{i}_1 + p_{1_y}\mathbf{j}_1 + p_{1_z}\mathbf{k}_1$, respectively. The above matrix equation is derived from these representations.

It is possible to obtain the inverse rotation matrix $\mathbf{R}_1^0$ similarly. Moreover, it can be shown that

$$\mathbf{R}_1^0 = (\mathbf{R}_0^1)^{-1} = (\mathbf{R}_0^1)^T. \tag{4.3}$$

Therefore, the rotation matrix is *orthogonal*, this means that its inverse equals to its transpose. The set of all rotation matrices 3×3 is usually referred to as $SO(3)$—Special Orthogonal group of order 3 [45].

composition

An important property of rotations is that they can be *composed*. Composition of rotation expresses result of subsequent rotations. Starting from fixed frame $0_{X_0Y_0Z_0}$, going through intermediate frame $0_{X_1Y_1Z_1}$ to the final frame $0_{X_2Y_2Z_2}$, the resulting final rotation matrix is

$$\mathbf{R}_0^2 = \mathbf{R}_0^1\mathbf{R}_1^2. \tag{4.4}$$

It should be noted that it is important to observe the order of multiplications, as it determines the order of individual rotations.

An arbitrary rotation can be obtained by three successive rotations. There are two common ways to describe such compositions: *Euler angles* and *Roll-pitch-yaw angles.*

Euler angles

Euler angles (θ, ϕ, ψ) express orientation of the rotated frame $0_{X_1Y_1Z_1}$ relative to the fixed frame $0_{X_0Y_0Z_0}$. The final rotation (4.5) is a result of rotation about the z axis by the angle ϕ, then about the *current* y axis by the angle θ and finally by the *current* z axis by the angle ψ. The resulting rotation matrix is

$$\mathbf{R}_0^1 = \mathbf{R}_{z,\phi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\psi} = \begin{pmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta s_\psi & s_\theta c_\psi & c_\theta \end{pmatrix}. \tag{4.5}$$

RPY angles

Roll-pitch-yaw angles (RPY angles) represent a variant of Euler angles and are commonly used in aeronautics (Fig. 4.3). The angles are denoted ϕ, θ, ψ in this order. The final rotation is a result of rotation about the x_0 axis by the angle ψ, then about the y_0 axis by the angle θ and finally by the z_0 axis by the angle ϕ. This is yaw-pitch-roll about the fixed frame axes. Another possibility is to roll-pitch-yaw about the current axes. The resulting matrix is the same in both cases (4.6). The resulting rotation matrix is

$$\mathbf{R}_0^1 = \mathbf{R}_{z,\phi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\psi} = \begin{pmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{pmatrix}. \tag{4.6}$$

For common use, the rotation matrices for rotations about the x, y and z axes are usually needed (4.7).

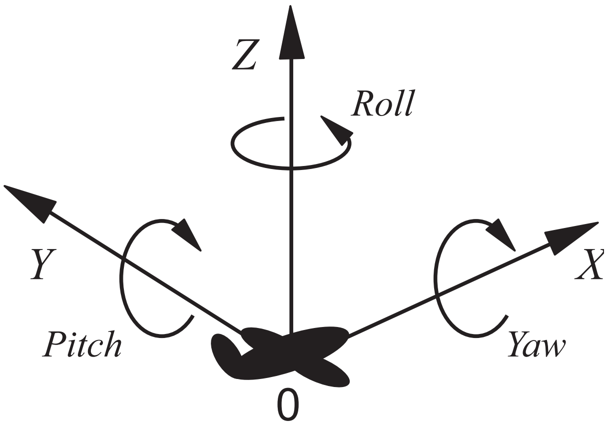


Figure 4.3: *Roll-pitch-yaw (RPY) angles*

$$\mathbf{R}_{x,\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}; \quad \mathbf{R}_{y,\beta} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}$$
$$\mathbf{R}_{z,\gamma} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

Translation

Translation means moving the coordinate frame a given distance in a specified direction. The direction and distance is determined by the translation vector $\mathbf{d} = (d_x, d_y, d_z)$. In fact, the direction is given by the combination of the respective motions by d_x, d_y and d_z.

translation

Rigid Motion and Homogenous Transforms

The above two components (pure translation and pure rotation) can be combined together to one *rigid motion* (recall the Chasles' theorem—Sect. 4.1.2). The related transform is

rigid motion

$$\mathbf{p}_0 = \mathbf{R}\mathbf{p}_1 + \mathbf{d}, \tag{4.8}$$

assuming that $\mathbf{R}$ is orthogonal.

The rigid motion can be expressed by *homogenous transformation matrix.* The most general form of homogenous transforms matrix $\mathbf{H}$ is [45]

$$\mathbf{H} = \begin{pmatrix} \mathbf{R}_{3\times 3} & \mathbf{d}_{3\times 1} \\ \mathbf{p}_{1\times 3} & \mathbf{s}_{1\times 1} \end{pmatrix}, \tag{4.9}$$

where $\mathbf{R}$ is rotation matrix, $\mathbf{d}$ is translation vector, $\mathbf{p}$ is perspective vector and $\mathbf{s}$ scale factor with dimensions specified by the subscripts (rows $\times$ columns).

As has been already mentioned, the homogenous transform preserve scale, therefore $s = 1$. Perspective also is not taken into account, therefore $\mathbf{p} = (0, 0, 0)$. However, it can be useful in, e.g., vision systems and computer graphics applications.

To be able to use homogenous transform matrices, it is necessary to augment the vectors for representing points by fourth component, 1, as follows

$$\mathbf{p}_0 = \begin{pmatrix} p_{0_x} \\ p_{0_y} \\ p_{0_z} \\ 1 \end{pmatrix}. \tag{4.10}$$

The same process needs to be applied to $\mathbf{p_1}$. This representation is known as a *homogenous representation* of the vectors $\mathbf{p}_0$ and $\mathbf{p}_1$, respectively [45].

homogenous representation

Now it is possible to present a set of *basic homogenous transforms*[4]. There are three basic translation matrices

$$\begin{gathered} Trans_{x,a} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad Trans_{y,b} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ Trans_{z,c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{gathered} \tag{4.11}$$

and three basic rotation matrices

$$\begin{gathered} Rot_{x,\alpha} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad Rot_{y,\phi} = \begin{pmatrix} c_\phi & 0 & s_\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s_\phi & 0 & c_\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ Rot_{z,\theta} = \begin{pmatrix} c_\theta & -s_\theta & 0 & 0 \\ s_\theta & c_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{gathered} \tag{4.12}$$

for movement and rotation about respective axes. In (4.12), $\sin\square$ and $\cos\square$ are abbreviated as $s_\square$ and $c_\square$, respectively, where $\square$ represents argument of the respective function.

[4]This set generates $E(3)$—Euclidean group of order 3, that is the symmetry group of 3D Euclidean space.

4.1.4 Generalized Coordinates and Velocities

Generalized coordinates and velocities play important role in modelling.

Generalized coordinates (also *Lagrangian coordinates*) are any quantities describing the system (such as position, joint coordinates, ...) in any coordinate system (not necessarily in the usual Cartesian system, therefore generalized). The selection of state variables is not unique for a given problem (e.g., position of a point on a plane with assigned coordinate system can be fully described by Cartesian coordinates (x and y) or in polar coordinates (distance-angle pair)). It is advantageous for generalized coordinates to be independent, as there are no additional conditions on the coordinates that would be necessary to describe the dependencies.

Lagrangian coords (generalized coords)

Generalized velocities are derivatives of the generalized coordinates with respect to time. They represent the evolution of the individual generalized coordinates over time. Analogically to holonomic system and generalized coordinates, nonholonomic constraints imposed on the system define dependencies between the generalized velocities.

generalized velocities

4.1.5 Kinematic Simplification

A *kinematic simplification* is a method for creating kinematically equivalent models of WMRs with axles (a differential drive, a car etc.). The heart of the method is in "compression" of the axles. The (usually) two wheels on each axle collapse into one. This reduces the complexity of the model.

method description

Unicycle Example

The first example could be a simplification of the differential drive. This robot with two independently driven wheels sharing a common axle is transformed to an equivalent model called a *unicycle* (please refer to Fig. 4.4).

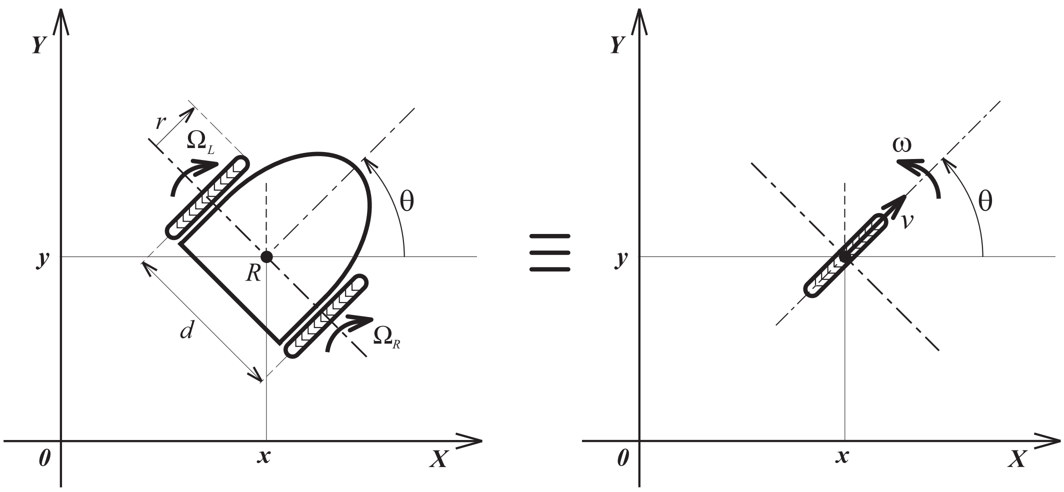


Figure 4.4: *Unicycle—kinematic simplification*

The state of the differential drive can be descried using three state variables x, y and θ. The first two variables represent the position of the reference point in Cartesian coordinates and the third one orientation w.r.t. the X^+ axis. The usual inputs (controls) of a real vehicle should be the rotational velocities of the left and right wheel, Ω_L and Ω_R. This follows from the fact that they are the motors which drive the robot and that they can be controlled directly. However, there exists another intuitive selection of the inputs—translational (v) and rotational (ω) velocity of the robot. There exists a one-to-one mapping between the former and the latter selection of inputs

$$v = \frac{r(\Omega_L + \Omega_R)}{2}, \omega = \frac{r(\Omega_R - \Omega_L)}{d}, \tag{4.13}$$

where r $[m]$ is the radius of the wheels and d $[m]$ is the distance between them.

The simplification of the differential drive is done by collapsing the two wheels into one upright wheel, which resembles a unicycle. In this case, the balancing issues are neglected and under this assumption, these two models are equivalent.

The advantages of the unicycle over the differential drive are more intuitive control inputs and freeing from the physical dimensions (r, d) of the vehicle. This conclusion may seem arguable or not very marked, unlike in the case of a car presented hereafter.

Car Example

In case of the car (or car-like—this refers to the same) model, the simplification is also accomplished by collapsing the axles. The resulting model is a bicycle model. For the original car, it is necessary to ensure the existence of the ICR, as in Fig. 4.5. This is complicated mainly by the nonzero width of the car and the therefore the front wheels have to be steered to different angles (one of the possible solutions is called the Ackermann steering). It would clearly cause unnecessary complication for modelling. Therefore the bicycle model as a simplification is contributive.

4.2 WMR Motion and Modelling

For the possibility of WMR model creation, its motion should be examined. It is necessary to select a point that represents the whole WMR, then examine the properties of different types of wheels that can be placed on a WMR (or, precisely, how they affect the motion capabilities of a WMR).

4.2.1 Creation of a Model

To create a model of a system, the following steps should be followed: **model creation**

- select state variables,
- select input (control) variables,
- write appropriate equations (state transition equation).

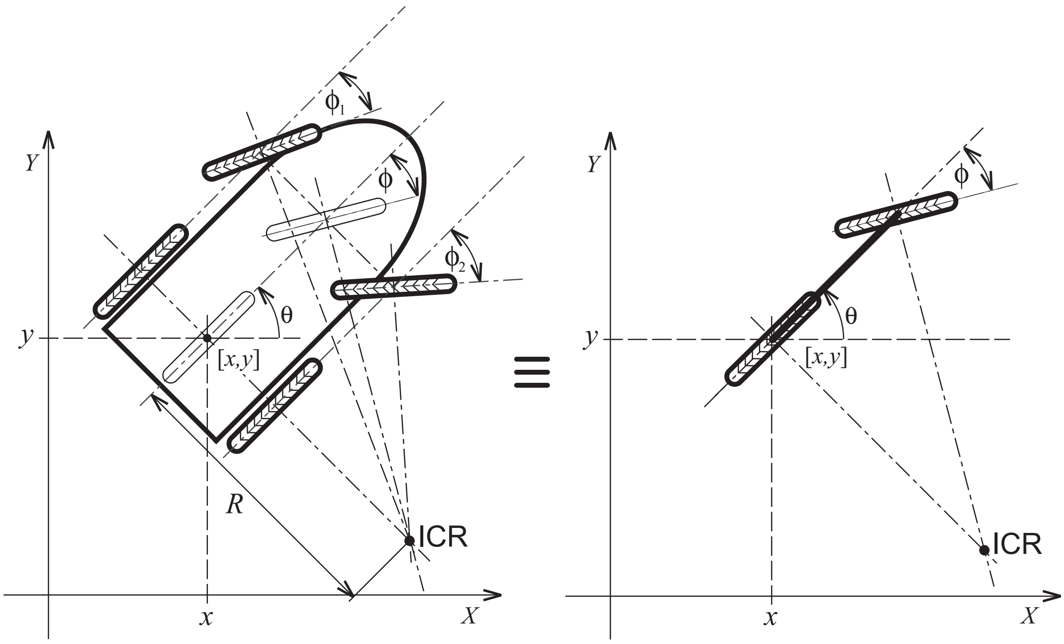


Figure 4.5: *Car with Ackermann steering—kinematic simplification*

state variables selection

The selection of state variables $q_1, \ldots, q_n$ is *not unique*. A system, in general, can be described by different sets of state variables (state vectors $\mathbf{q}$). The basic requirement is that the system is described *uniquely* by the selected state vector.

control variables selection

The control (input) variables (forming vector $\mathbf{u}$) are usually given by (mechanical/electrical) construction of the system. In case of WMRs, generally the velocities (translational and/or rotational) of the wheels can be controlled. However, sometimes it is advantageous to use transformed control inputs, like in the case of a unicycle—in this case, the speed of the wheels can be controlled directly (via control of the motors) but it is more appropriate to use the translational and angular velocity of the whole robot for modelling and control purposes. The important point is that there exists one-to-one mapping between the original and new control inputs in this case (Sect. 4.1.5).

state transition eqn.

A state transition equation (which is a differential equation) is in a general form $\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u})$.

The procedure will be demonstrated later (Sect. 4.2.3) on an example—a kinematic model of a unicycle WMR will be derived.

4.2.2 Wheels Classification

assumptions on the wheels

It is assumed that the wheels of the WMR are *nondeformable* and in *vertical position* during the motion. There are two types of rotation that can be distinguished. The first one is around the horizontal axis (this is the *rolling* of the wheel), the second one could be the rotation around the vertical axis (the *steering*). The latter however applies only to the wheels that are capable of steering. The respective axes are therefore called *rolling* *axis* and *steering axis*.

From perspective of kinematics of WMRs, the important property of a wheel is if it introduces a (nonholonomic) constraint on the motion of the robot, that means, if it limits its motion. Therefore, the wheels can be classified to two main groups,

constraint introducing which are the standard fixed and steered wheels,

without constraint which are the ball wheels, the omnidirectional (Swedish) wheels, and the stabilization points including the castors.

The respective wheel types are described hereafter.

Standard Wheels

The *standard wheels* are depicted in Fig. 4.6. *Fixed standard wheel* is in Fig. 4.6 (a), *steered standard wheel* in Fig. 4.6 (b). This type of wheels is very common, e.g., can be found on majority of cars. They have two DOF (rolling in direction they are pointing and rotation about the point of contact with the ground). They introduce a constraint which is based on fundamental assumption on the wheel motion, which is *rolling without slipping* (or *pure rolling*). By closer analysis, this assumption means that the wheel has zero lateral velocity and its forward velocity v is[5] $v = \omega r$, where $\omega[rad.s^{-1}]$ is the angular velocity and $r[m]$ is the radius of the wheel. However, it should be noted that there exists a slip that is unavoidable. It is in the case of steering the standard wheels (the point of contact slips w.r.t. to the ground). **standard wheels**

Omnidirectional Wheels

The *omnidirectional wheels* (also Swedish wheels, Mecanum wheels, Ilon wheels—invented in 1973[6] by Bengt Ilon, an engineer working that time for the company Mecanum AB, Sweden) are the standard wheels with rollers on its outer circumference. The principle of the wheel is that the circumferential rollers decomposes the angular velocity (or resulting forward velocity, respectively) of the wheel to forward component and component perpendicular to the wheel direction[7]. Thus, the resulting motion of a single omnidirectional wheel is determined by its forward velocity and the angle of the rollers. Resulting motion of the whole platform is determined by the sum of the velocities (or forces, respectively) generated by all of the wheels. **omnidirectional wheels**

The rollers can be either passive (more common due to simpler construction) or driven (less common—or rare, respectively—with more complex construction). Passive rollers **rollers**

[5]This is a theoretical value. In fact, due to external forces (resistive forces) the forward velocity is lower. This is significant mainly in case of bigger vehicles (such as cars) and the difference is caused by tyres deformation (the tyres must deform in order to generate motion force) [52].

[6]To be precise, there was an ancestor of the omnidirectional wheel—a US Pat. No. 1,305,535 from June, 3, 1919 "Vehicle wheel" by Joseph Grabowiecki, that describes a wheel that has rollers on its circumference that have their axes at right angles to the axis of the main wheel.

[7]The vector sum of these two components is normal to the roller axis of rotation.

can be kept in their positions from outside (good load carrying capacity) or centrally (i.e., the rollers are split into two parts and centrally mounted). The latter variant, also proposed by Ilon, has an advantage on inclined or uneven terrain [16].

straight-on motion

As can be easily imagined, motion in straight-on direction is energetically ineffective (energy losses occur due to the fact that the forward motion needs to be generated by balancing the sideways velocities). Two examples of solution of this problem by modification of the classical omnidirectional wheel construction can be found, e.g., in [16]. The first, easier solution, is in enabling of locking of the rollers in case of forward motion of the platform, the second one combines the possibility of change of the rollers angle with their locking.

angle of rollers

The angle of the rollers w.r.t. the wheel plane can be between 0 degrees and 180 degrees (the angles between 180 degrees and 360 degrees can be transformed to this interval by the change of direction of rotation of the wheel).

The angle of 0 degrees means that the direction of rotation of the rollers is perpendicular to the plane of the wheel. Clearly, this wheel itself can move only in the direction of the translation of the standard wheel, unless the rollers are driven. Its use is on the robots with specific configuration of these wheels.

The angle of 90 degrees means that the direction of rotation of the rollers is identical with the rotation of the wheel. This design would be degenerative, as the wheel cannot roll (neglecting the residual motion caused by imperfect mechanical characteristics), unless the rollers can be blocked.

DOFs

This type of wheels has three DOFs—translation in the longitudinal direction, translation in the direction of rolling of circumferential wheels and rotation along its vertical axis. As has been mentioned earlier, the motion of the WMR in a desired direction is ensured by appropriate mutual velocities of the wheels present on the WMR.

Ball Wheels

ball wheels

The *ball wheels* are truly omnidirectional in the sense that their motion occurs without any slip. Appropriately controlled omnidirectional wheels are kinematically equivalent to the ball wheels.

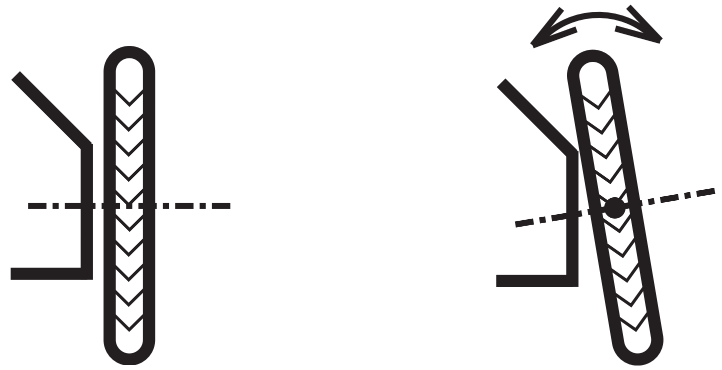
Stabilization Points

stabilization points

The *stabilization points* are used to stabilize robots so their underside is parallel to the ground (e.g., differentially-driven). They can be realized using small passive ball wheels, friction points etc. and do not affect the kinematic capabilities of the robot.

castors

The *castor wheels* (the ones that can be found, e.g., on office chairs) can be assigned to this category. Although their kinematic equations are not trivial, they do not affect the (kinematic) motion capabilities of the robot. Therefore they can be neglected when modelling a WMR. On the other hand, it should be stressed that a different situation is with creating a dynamic model. Based on the parameters of the wheels and the environment (like friction), it could be desirable to incorporate also this type of wheels



(a) Standard fixed wheel (b) Standard steered wheel

Figure 4.6: *Wheels restricting motion (top view)*

into the model.

4.2.3 Kinematic Models

Kinematics is a branch of mechanics that studies solely the geometry of motion. This means that the cause of motion is not considered—the forces and torques acting on the system are neglected. Thus, a kinematic model of a WMR represents motion of the robot body based on its configuration and motion of the robot wheels. **kinematics**

Usually there exist some limitations on possible movements of a WMR (*constraints*), that are described in Sect. 3.7. **constraints**

The kinematic models can be viewed and called differently in different branches of science. The different viewpoints are summarized in Tab 4.1.

Table 4.1: *Representation of kinematic models*

branch	**holonomic systems**	**nonholonomic systems**
mechanical engineering	kinematic equations	kinematic equations
mathematics	algebraic equations	differential equations
control engineering	static systems	dynamic systems

Kinematic models are usually used in the problems of WMR control. Compared to dynamic models, the kinematic ones are (usually) simpler and easier to create.

Validity of Kinematic Models

To *create* a kinematic model of a WMR, it is necessary to introduce the following assumptions on the wheels (recapitulated from Sect. 4.2.2): **assumptions**

- Wheels are nondeformable and their vertical axes are perpendicular to the surface which they roll on.
- Wheels are rolling without slipping and there is no lateral motion.

To *use* the kinematic model it is necessary to specify the conditions of its validity. Finding these conditions isn't a trivial task, generally it can be stated that: **usage**

- It is necessary to set the dynamic characteristics of the WMR such that the kinematic model is sufficiently exact.

This statement leads to the following discussion. In specific applications (e.g., service, household, industry) the WMRs are moving in environments along with human individuals. For safety reasons, the velocity of WMRs is usually reduced. Also, the surfaces are usually flat. If the drives are sufficiently powerful, changes in velocities (or inputs, respectively) can be considered as step ones (or a technique presented below can be employed). Therefore, in these situations the use of kinematic models is possible.

Extended Kinematic Models

Kinematic models can be extended in order to incorporate additional specifics of real WMRs, where these specific properties play an important role preventing the "basic" kinematic model to be valid. Two examples of extension principles are presented, modelling of delayed response to control commands and adding integrators to system inputs. **extensions**

Time-delayed response to controls A model that incorporates significant property of modelled WMR appeared in [7]. It brings up an idea of simulating dynamic behaviour of a WMR by means of kinematic model. In this case, a car-like WMR is used for automatic parking task and the real system has a delayed response on control signals in driving and steering velocities. Thus, this fact is simulated by (the second equation has been altered to comply with the notation used in this text):

$$\begin{aligned} v(t) &= v_{ref} + [v(t-\Delta t) - v_{ref}].e^{(\frac{-\Delta t}{\tau_v})} \\ \gamma(t) &= \gamma_{ref} + [\gamma(t-\Delta t) - \gamma_{ref}].e^{(\frac{-\Delta t}{\tau_\gamma})}, \end{aligned} \tag{4.14}$$

where v_{ref} and γ_{ref} are the reference driving velocity and curvature commands generated by the controller, and τ_v and τ_γ are the response times of the driving and steering engines of the modelled WMR. The idea of modelling approach is pretty straightforward.

Here the dynamic effects are neglected (the WMR moves slowly) but the response delay is significant, and cannot be omitted.

Extension of inputs by added integrators A kinematic model, e.g., of a simple car, allows instantaneous changes of wheel velocities. This can be realistic for slow speeds, as discussed above. Extension by adding integrators means that system inputs can be extended by adding one or more integrators. The original input variable then becomes a state variable of the added integrator and the input of the integrator is a new input to the system. This procedure can be applied repeatedly, adding as much integrators as necessary. This allows to prevent step changes in input signals, or more precisely, to ensure that the input signals for the original inputs are represented by mathematical functions of class[8] $\mathcal{C}^n$, where n is the number of added integrators.

An Example—Unicycle Kinematic Model

The process of creation of kinematic model is presented on the unicycle WMR. The steps to create it are:

- selection of state variables,
- selection of input (control) variables,
- writing appropriate equations.

reference point

The very first step, however, should be selection of a reference point. This point represents the position of the whole WMR and the desired behaviour is assigned to this point (e.g., following a planned trajectory). The usual selection is a point that lies on the wheels axis in its middle (R, Fig. 4.7). Therefore, the WMR can rotate around this point (and when the unicycle rotates, the reference point does not move), which makes the control of the WMR easier.

state variables

One possible way how to select the state variables (i.e., variables that defines uniquely the state of the WMR) is $\mathbf{q} = (x, y, \theta)$, where $x[m]$ and $y[m]$ define the position of the reference point in Cartesian coordinate system and $\theta[rad]$ is the rotation (angle) with respect to the X^+ axis, increasing anticlockwise (see Fig. 4.7).

control inputs

Control variables are usually selected as v $[m.s^{-1}]$ is forward (driving, translational) velocity and ω $[rad.s^{-1}]$ is rotational (steering) velocity of the robot. On a real robot, the typical control commands are, however, angular velocities of left and right wheel, ω_L and ω_R $[rad.s^{-1}]$, respectively, as the wheels are driven by the motors.

underactuated systems

Note: As it is easy to see, the number of control inputs is smaller than the number of controlled (state) variables. Therefore, the WMRs are underactuated.

[8] A function is said to be of class $\mathcal{C}^n$ if its derivatives up to order $n > 0$ are defined and are continuous. If $n = \infty$, then the function is called *smooth*.

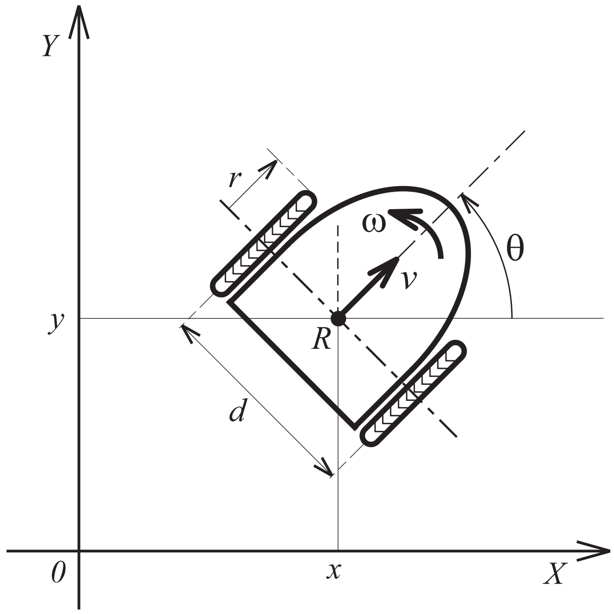


Figure 4.7: *Robot state, control and dimension variables*

There is one-to-one mapping between these velocities:

$$
\begin{aligned}
v &= \frac{r}{2}(\omega_R + \omega_L) \\
\omega &= \frac{r}{d}(\omega_R - \omega_L)
\end{aligned}
\tag{4.15}
$$

and

$$
\begin{aligned}
\omega_L &= (2v - \omega d)/2r \\
\omega_R &= (2v + \omega d)/2r.
\end{aligned}
\tag{4.16}
$$

In the above equations, d $[m]$ is the distance between the wheels (wheelbase) and r $[m]$ is the radius of the wheels (the same wheels are assumed). As seen from (4.15), the WMR translational velocity v equals to average of the translational velocities of respective wheels.

Now it is possible to write the equations which idea is to describe the changes (i.e., derivatives) of the state variables, and therefore they are also called state transition equations. It can be done easily using the Fig. 4.7: **kinematic eqns.**

$$
\begin{aligned}
\dot{x} &= v\,\cos\theta \\
\dot{y} &= v\,\sin\theta \\
\dot{\theta} &= \omega.
\end{aligned}
\tag{4.17}
$$

The kinematics of the unicycle can be then described using following matrix equation

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \mathbf{h}_1(\mathbf{q})v + \mathbf{h}_2(\mathbf{q})\omega = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega. \tag{4.18}$$

Unicycle Model in Polar Coordinates

This is a modified version of widely-used unicycle kinematic model in Cartesian coordinates, and was originally introduced in [6]. The idea is to use discontinuous transformation to overcome Brockett's theorem. **modification**

The state transition equations (4.18) are transformed via non-smooth coordinate change into polar ones. The main advantage of this transformation is increased resolution around a fixed point (origin of the coordinate system). This process (referred to as $\sigma-$process) has been used in theory of differential equations to resolve singularities of vector fields around equilibrium points [3].

Considering the system (4.18) under the following transformation [4]

$$\begin{aligned} \rho &= \sqrt{(x^2+y^2)} \\ \alpha &= -\theta + \arctan\left(\tfrac{-x}{-y}\right) \mod \left(\tfrac{\pi}{2}\right) \\ \phi &= \tfrac{\pi}{2} - \theta \end{aligned} \tag{4.19}$$

for $\alpha \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right]$, with the transformation not defined for $x = y = 0$, then the he resulting model is

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\cos\alpha & 0 \\ \sin\alpha/\rho & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{4.20}$$

with $u_1 = v, u_2 = \omega$.

An important property of this model is that it can be locally (or even globally) [4] stabilized by a state feedback control law $\mathbf{u} = \mathbf{Kq}$ with

$$\mathbf{K} = \begin{pmatrix} k_\rho & 0 & 0 \\ 0 & k_\alpha & k_\phi \end{pmatrix} \tag{4.21}$$

and $k_\rho > 0, k_\phi < 0, k_\alpha + k_\phi - k_\rho > 0$. The control law rewritten in terms of original coordinates is

$$\begin{aligned} u_1 &= v = k_\rho\sqrt{x^2+y^2} \\ u_2 &= \omega = k_\alpha \arctan\left(-\tfrac{x}{y}\right) + k_\phi\phi. \end{aligned} \tag{4.22}$$

As can be seen, the discontinuous transformation enables synthesis of the control law for exponential stabilization in a straightforward manner.

Extended Unicycle Model

The unicycle model derived above can be extended by adding integrators on its inputs (as described, e.g., in [24]). More specifically, the translational velocity input v can be extended in this way to obtain a new control input, change in speed a (this is the *acceleration* of the unicycle). The second input, steering (angular) velocity ω can be extended to obtain *angular acceleration* α. The new model equations are

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega; \;\; \dot{v} = a, \dot{\omega} = \alpha, \tag{4.23}$$

with a and α being the new inputs.

4.2.4 Dynamic Models

For the sake of completeness, dynamic models will be shortly introduced.

dynamics

Opposed to kinematics, dynamics is the branch of mechanics that studies effects of forces on the motion of a system. Here, forces, momentums, inertias and similar variables appear. Various factors influencing motion of systems (vehicles) can be described, such as friction, centrifugal forces etc. This enables modelling of skid, slip and slide of wheels, and thus recognize loss of control in case of inappropriate control commands (e.g., when "allowed" forces (velocities, respectively) are exceeded).

This category of models is required mainly in the situations, where:

- the mass of the WMR cannot be neglected (underpowered or heavy vehicles, such as cars etc.),
- modelling of force-related effects is desired (such as skid and slip, friction, tyres deformation and resulting effects, etc.).

simplification

It should be mentioned that the dynamic parameters of a WMR are not usually completely known and/or can be uncertain, time-varying etc. Therefore, some simplifications are necessary to be made when creating dynamic models. Also, complexity of a model usually increases rapidly with raising requirements on precision. Therefore, it is advisable to find appropriate *level* of simplification.

As in the case of kinematic equations, three different viewpoints are presented in Tab. 4.2.

Table 4.2: *Representation of dynamic models*

branch	**representation**
mechanical engineering	dynamic equations
mathematics	system of differential equations
control engineering	dynamic system

In case of dynamic equations, the kinematic ones are a part of the equations system. They represent the equations of constraints imposed on the system and reduce the dynamic model.

Principles of Dynamics

A principle of dynamics is a theorem that describes behaviour of a mechanical system. There are various such theorems, and each of them is called a principle, therefore there are many principles.

However, the result of any of the principles is a set of Newton's equations of motion of mass points of the described system. It should be noted that there doesn't exist a proof of neither any of the principle, nor the Newton's equations. These laws are empiric and the only possible way to check their correctness is comparison of their results with real-world observations [51]. The principles of dynamics can be divided into two groups: differential and integral ones. Differential principles are D'Alembert's, Gauss', Jourdain's, integral ones are Hamilton's, Maupertuis-Euler's, Jacobi's and Hilbert's. Hertz's principle can be formulated in both differential and integral form. According to [51], the most powerful is Gauss', the most important is Hamilton's and the simplest d'Alembert's principle. The last one is based on original d'Alembert's principle and principle of virtual work.

no proof

differential/ integral principles

The dynamic model of any system can be written in general form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V_m}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_\mathbf{d} = \mathbf{B}(\mathbf{q})\boldsymbol{\tau} - \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda}, \tag{4.24}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite inertia matrix, $\mathbf{V_m}$ is the Coriolis and centripetal matrix, $\mathbf{F}(\mathbf{q}) \in \mathbb{R}^n$ is surface friction, $\mathbf{G}(\mathbf{q})$ represents gravitational vector and τ_d is bounded unknown disturbances including unstructured unmodelled dynamics, $\mathbf{B}(\mathbf{q})$ is the input transformation matrix, $\boldsymbol{\tau}$ is the input vector of forces and/or torques, $\mathbf{A}(\mathbf{q})$ is the matrix associated with the constraints and $\boldsymbol{\lambda}$ is the vector of the constraint forces.

Note on 1^{st} Order Lagrange Equations

These equations describe movement of a constrained mass point.

Unconstrained mass point moves in the direction of resulting external force (vector sum of external forces). Any constraint that forces the mass point to move in a different manner than without it, affects the mass point by a *constraint force*. The motion of such constrained mass point is determined by the sum of external and constraint forces. The latter are not usually given directly, but the constraint equations are known, e.g., a plane (holonomic and scleronomous constraint)

constraint force

$$f(x, y, z) = 0. \tag{4.25}$$

Yet, this is not sufficient to determine the constraint forces, however (the way how the constraint works isn't known). It is common to assume that the constraint force is normal to the plane and the motion is not affected by friction (necessary for the former assumption—the constraint force is normal to the plane).

Thus, the constraint force equals to *grad f* except for a scalar coefficient λ. Finally, the motion equation of the mass point is

$$m\ddot{\mathbf{r}} = \mathbf{F} + \mathbf{P} = \mathbf{F} + grad\, f, \tag{4.26}$$

or by components

$$\begin{aligned} m\ddot{x} &= F_x + \lambda \frac{\partial f}{\partial x} \\ m\ddot{y} &= F_y + \lambda \frac{\partial f}{\partial y} \\ m\ddot{z} &= F_z + \lambda \frac{\partial f}{\partial z} \end{aligned} \tag{4.27}$$

and these equations are called 1^{st} *order Lagrange equations.*

4.3 Modelling Approaches

overview

To create a kinematic model, the basic approach is to examine the geometry of motion and compose appropriate equations (state transition equations). There have also been attempts to classify the kinematic models of the robots and to create a methodology for creating kinematic models ([33], [39], [1], [12]).

The first approach, published in [33], uses techniques for modelling of robotic manipulators (stationary robots) adapted for WMRs. It is based mainly on the use of homogenous transforms (Sect. 4.1.3). This approach has been extended in [39] by allowing inclined steering column on the WMR. A unified approach to *dynamic* modelling of robotic systems subject to holonomic, nonholonomic or both types of constraints using state space representation is presented in [55]. A numerical algorithm for implementation of the method is also developed. In [12] (also presented with minor differences in [43]), an universal methodology for creating kinematic models of WMRs is presented. It is suitable for single-body robots. This method is briefly described hereafter.

4.4 Kinematic Models and WMR Types Classification

method principle

This section discusses approaches to kinematic modelling of WMRs and classification of chassis types. When considering WMR chassis, the question on its motion capabilities arises. This leads to an idea of a method for classification of chassis types. From the kinematic point of view, the number and type of wheels attached to a WMR determine its motion capabilities. One way how to characterize these capabilities is to focus on the possibility of ICR position selection (refer to Sect. 4.1.2). The less restricted ICR positioning, the better WMR motion capabilities (maneuverability).

4.4.1 Steerability, Mobility and Maneuverability

Important note: The concept of maneuverability expresses the capability of the robot to move without violating the constraints imposed by the wheels. This means that it is not required that the motion is generated by the wheels (an "external drive" can be used (this can be imagined as moving the WMR by hand)).

A concept for classification of kinematic models of WMRs has been introduced in the literature [12]. It is presented in the following text with some refinements.

The terms *steerability*, *mobility* and *maneuverability* refer to different capabilities of a robot.

The overall maneuverability of a WMR consists of steerability and mobility. The measure of these capabilities are given by *degree of maneuverability* δ_M, *degree of mobility* δ_m and *degree of steerability* δ_s. It holds $\delta_M = \delta_m + \delta_s$.

degree of mobility

The *degree of mobility* δ_m is defined as the number of degrees of freedom that can be directly manipulated from the translational velocity inputs without reorienting the steered wheels.

This definition can be also formulated (or refined) as follows. The degree of mobility is the number of differential degrees of freedom in *local coordinate system* of the WMR that can be directly manipulated from the translational velocity inputs without reorienting the steered wheels (that means, locked in their initial position (zero angle)). For example, for a car-like WMR this means that the δ_m is judged with the steering wheels in the rest position, i.e., driving straight on only. Thus, in this case, $\delta_m = 1$. Another example could be a differentially-driven WMR. In this case, by translational velocity inputs (velocities of left and right wheel) it is possible to control translational and angular velocity of the whole WMR. Therefore, $\delta_m = 2$ in this case. To the original definition should be noted that some DOFs cannot be changed independently (such as the x and y coordinates).

degree of steerability

The *degree of steerability* δ_s represents additional degrees of freedom that are accessible from steering inputs. The action of these inputs on the posture is indirect [12]. For example, in case of car-like WMR, steering the front wheel only does not affect directly its posture (position and/or orientation). Only when the car moves (forward, backward) the orientation of the front wheel affects the motion. Orientation of the steered wheels is not directly manipulated by the steering control inputs, but it is related to them via an integral action.

degree of maneuverability

The *degree of maneuverability* δ_M is a sum $\delta_M = \delta_m + \delta_s$. It summarizes the motion capabilities of a robot. Nevertheless, two robots with the same δ_M does not necessarily have the same capabilities—they depend on the way how the degrees are divided between degrees of mobility δ_m and steerability δ_s.

As has been mentioned earlier, the maximum value of the degree of maneuverability δ_M is 3 and it remains to explore how can its value be divided between δ_m and δ_s. There are five possible nondegenerate pairs of (δ_m, δ_s) [12]. These pairs are $(3, 0)$, $(2, 0)$, $(2, 1)$ $(1, 2)$ and $(1, 1)$. The degenerate types are therefore $(1, 0)$ (this WMR can move only on a line) and $(0, k)$, where $k > 0, k \in \mathcal{Z}$ (simplifiedly, this WMR has only k undriven steered wheel(s) and therefore cannot move). The value of k could be theoretically arbitrarily high. For $k = 0$ the system would have no wheels at all and could not be considered a WMR.

4.4.2 Kinematic Model Creation—A Universal Method

An example of a universal approach to creation of kinematic models, the method from [12] will be described. It is worth reminding that it is designed for robots with single body (this means that it cannot be directly used, e.g., for tractor-trailer systems from Sect. 5.1). **usage**

The described method is a universal tool that can be applied to create also models of rather complex WMR configurations (with multiple wheels placed in different positions on the robot) by following a given procedure. However, in case of simpler chassis it may be easier to derive the model from the geometry of motion, by composing the state transition equations. **universal tool**

The method is based on the classification of wheels that can be used to construct a WMR. For each wheel type, two equations are assigned, representing two assumptions on the motion—no lateral slip (*no lateral movement condition*) and no forward slip (*rolling without slipping condition*). Then, based on the physical position of the wheels on the robot, the constraint matrices are created from constraints of the constituent wheels. The result is a matrix equation representing the constraints in implicit form. For the five basic types of WMRs (see Sect. 4.4.1), there exist pre-computed matrices for forming posture kinematic models (see [12]). However, they are created under certain assumption on the position of the reference point. This could introduce some difficulty when creating a model of a WMR with different required position of the reference point. **method principle**

Note on Differential Drive vs. Unicycle

A unicycle is a kinematic simplification of the differential drive. Their kinematic models are kinematically equivalent. However, for a unicycle, it is necessary to transform the control inputs from left and right wheel velocity (on the differential drive) to translational and angular velocity. There exists one-to-one mapping. Further analysis leads to the conclusion, that these models are not fully equivalent, as they differ in the structure of the degree of maneuverability. The differential drive has two fixed standard wheels ($\delta_m = 2, \delta_s = 0 \Rightarrow \delta_M = 2$), whereas the unicycle only one that could be considered as a steered standard wheel ($\delta_m = 1, \delta_s = 1 \Rightarrow \delta_M = 2$).

However, the unicycle should not be confused with a *single-bodied single-wheeled robot*, depicted in Fig. 4.8). This type of robot possess one standard steered wheel. Therefore, $\delta_m = 2$, $\delta_s = 1$ and $\delta_M = 2 + 1 = 3$. **single-wheeled single-body robot**

In [11], a true *single-wheeled*, gyroscopically stabilized robot is described. This WMR has only one wheel dynamically stabilized in upright position. The controller and driving mechanism is inside the wheel. The steering is performed via leaning the wheel and due **single-wheeled robot** to resulting reactive forces (torques).

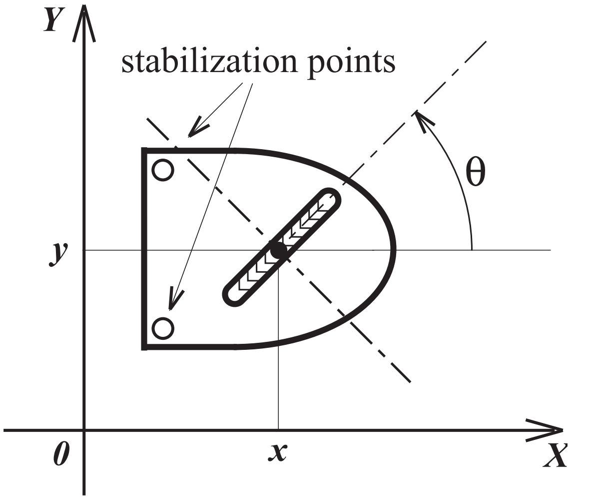


Figure 4.8: *Single-wheeled mobile robot*

Chapter 5

Control of WMRs

This chapter is focused on problems of control of wheeled mobile robots. First, configurations of the WMRs are briefly reviewed. Then, the basic control tasks are defined and overview of the control algorithms is presented.

5.1 Configurations of WMRs

There are many types of possible WMR configurations. By configuration is understood type of chassis of the robot, i.e., wheels, their type and position. **configurations**

One of the basic classifications of the WMRs can be the number of the bodies of the robot. From this point of view, there can be found two main groups— single-body and multi-body robots. The former group, as its name suggests, is represented by the robots with only one body. To the latter one belong the robots with trailers (also tractor-trailer systems).

For an example of the classification of single-body robots please refer to Sect. 4.4.

For multi-body robots, there are basic types often used in the literature on WMRs control. The following classification, as presented in [36], includes also two single-body robots (please refer to Fig. 5.1):

Unicycle (a) with three state variables (determining its position and orientation) and two control inputs (usually translational and rotational velocity).

Car-like robot (b) with four state variables and two control inputs.

N-trailer system (c) with $N + k$ state variables. The value of k depends on the tractor configuration. If it is a unicycle, then $k = 3$. In case of car-like tractor, $k = 4$.

Firetruck (d) with six state variables and three control inputs. It is a special case of the previous system, consisting of one tractor with a trailer possessing steered axle. This is a model of real firetrucks with long ladders equipped by a cab for the second driver at the end of the ladder for improved maneuverability in narrow streets. This is an example of multi-body multi-steered mobile robot. The trailer could be also a bi-steered car to fit into this group.

N-trailer system with nonzero hooking (e) meaning that each trailer is not hooked in the middle of the previous trailer rear axle. This type of chaining introduces a degree of difficulty into modelling and control.

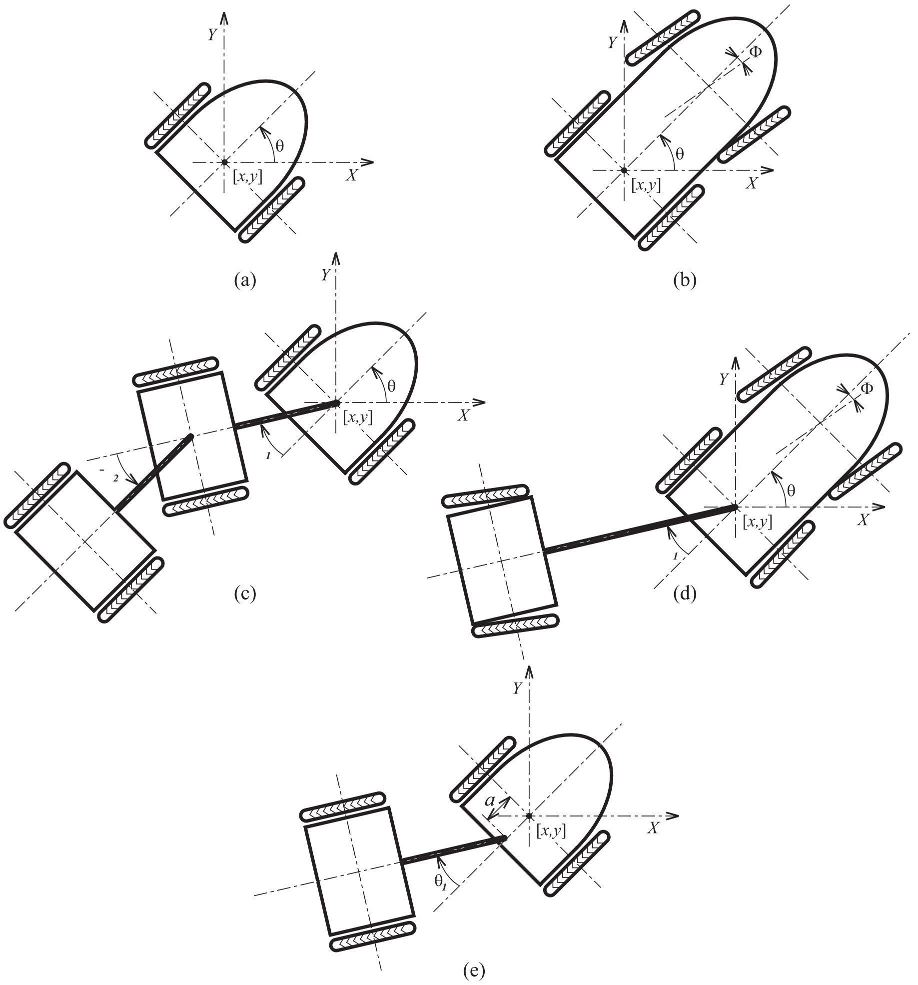


Figure 5.1: *Various configurations of WMRs*

The selection of WMR cases above is obvious, as it reflects usual configurations of vehicles used in everyday life.

conventional wheels

In the following text, the robots with *conventional wheels* will be assumed. For detailed description of various wheel types please see Sect. 4.2.2. A brief summary is included for convenience.

Here, the conventional wheel means that it rolls in the direction it is pointing (the "pure rolling without slipping condition" is satisfied)—the wheel has a permanent contact with the surface and doesn't slip or slide. This can be also expressed as a requirement on zero contact point velocity. Therefore, certain limitations on velocities arise from this requirement, such as that the wheel cannot move sideways. Also it results in the fact that vehicles equipped with conventional wheels has fewer control inputs than degrees of freedom (see above). Such systems are called *underactuated*.

omnidirectional robots

The mentioned omnidirectional (Swedish) wheels allow building the omnidirectional robots. These robots are able to move in arbitrary direction. However, this is not the only possibility how to create a chassis with this type of motion capabilities, as the independently steered and driven wheels can be used.

synchro-drive

Another type of chassis is a synchro-drive. The typical property of this type, also giving it its name, is that all wheels are steered synchronously with the same orientation (usually, there is one steering motor plus a belt connecting the wheels). Moreover, all the wheels have the common drive for spinning. It is easy to imagine, that the robot can move freely in the XY plane only but it cannot change its orientation. This type of robot is usually used for indoor research purposes.

5.2 Types of Tasks

WMR control

Controlling a WMR represents in the scope of this text moving it from one (initial) configuration (i.e., position and orientation) to another (final, goal) one, or in other words, minimization of selected error quantities (difference between the actual and desired position and angle). This can be done in more or less predictable way (transition process). The basic tasks for WMRs moving in an environment *free from obstacles* can be defined as [37]:

Moving between two postures (point-to-point motion, abbrev. PTPM) , also posture stabilization means that the robot must reach desired posture (goal configuration) from given initial configuration.

Following a given path (path following, abbreviated as PF) means that the robot must follow a path, that is a geometric curve without timing rule.

Following a given trajectory (trajectory tracking, abbrev. TT) means that the robot must reach and follow a given trajectory (i.e., path in Cartesian space with associated timing rule) with the initial position on or off the trajectory.

These three basic tasks are described in more details herein.

point to point motion

Point-to-point motion is a stabilization problem for an equilibrium point in the WMR state space. When analyzed from the viewpoint of number of inputs and outputs, the control problem is non-square, i.e., it means that the number of control inputs is smaller than the number of controlled outputs. A unicycle WMR has two control inputs, translational velocity v and *rotational* velocity ω, while it has three controlled outputs (x, y, θ). For a car-like WMR there are also two control inputs, the translational velocity v and the steering velocity ω and four controlled outputs (position in the XY plane, orientation of the robot w.r.t the X^+ axis and orientation of the steered (front) wheel w.r.t the longitudinal axis of the robot body). This causes the set-point stabilization to be a challenging task.

The error signal(s) used in the feedback controller is the difference between current and desired configuration.

Path following means following a desired geometric curve. There is no requirement on the executing velocity (the translational velocity) of the WMR during the motion, therefore it is usually determined externally. This task is often replaced by *trajectory tracking*. **path following**

Trajectory tracking means that the controller is given a geometric description of the desired trajectory in Cartesian coordinates along with the timing law. This is equivalent for the WMR to track a reference virtual vehicle. Although the desired trajectory can be split into parameterized geometric path and appropriate timing for the parameter, it is not strictly necessary. It is often simpler to specify the workspace trajectory as an evolution of chosen representative point on the robot (the reference point). Then, the task can be formulated as stabilization of two-dimensional Cartesian error to zero using both control inputs. **trajectory tracking**

When considering trajectory tracking, the task can be divided into two separate subproblems—*path planning* (creating the reasonable/feasible path) and *WMR control* (running the robot along the planned trajectory with as small deviations from it as possible). **trajectory tracking—subproblems**

5.3 Design of Control Algorithms

When designing a control algorithm, one can use, in principle, two different approaches

- linearization of WMR model,

  or

- use of non-linearized WMR equations.

Linearization of WMR equations (a nonlinear system) allows the use of theory of linear control systems. This field of research is very well treated and numerous techniques are developed. Use of nonlinearized equations is common, e.g., in reactive control (i.e., sensor-action mapping). This means that appropriate actions (motion) is generated by a controller based on sensor reading (employing, for instance, fuzzy logic, neural networks, etc.).

5.3.1 Linearization

With linearization, a nonlinear system can be converted into a linear one. There exists an extensive literature on this topic. A well-known is monography [20]. Therefore, this note about linearization presented herein is brief and simplified. **introduction**

The advantage of this approach is obvious—tools for study/control design etc. of linear systems can be used afterwards. Linearization is achieved by applying a (smooth, invertible) nonlinear coordinate transformation on the original system in order to achieve a linear system in the new (transformed) coordinates.

In general, there are two basic types of linearization that can be applied to WMRs: **linearization of WMRs**

Local linearization can be at a point or along a trajectory. In this case, the WMR is linearized near selected point (in state space) or trajectory. The disadvantage is that the linearized model is not valid for larger neighbourhood (i.e., larger deviations) of selected point/trajectory. **local linearization**

Global linearization where the target is to linearize the WMR so as the linearization is valid for the whole state space. This is the case of, e.g., dynamic feedback linearization (please refer to Sect. 5.4.4). **global linearization**

Linearization can be obtained using a *feedback linearization*, which is a technique for linearization of the original system ODEs. A subsystem inserted into the feedback ensures the desired linear behaviour of the whole system. The feedback subsystem can be *static* (time-invariant) or *dynamic* (time-varying). Feedback linearization requires access to accurate information on the system (precise measurement of the system state $\mathbf{q}(t)$ and/or its derivatives up to certain order. There are two types of feedback linearization that can be achieved for a nonlinear system. These types are *input-output* linearization and *input-state* linearization.

Input-output Linearization

is a variant with the behaviour of the whole system from input to output in a linear manner. In this case, no attention is paid to the state of the system. This could be a severe disadvantage of this approach—although the input-output behaviour can fulfil the desired expectations, the states can be unbounded or exceed the acceptable limits (e.g., physical limits of the system). **input-output linearization**

Input-state(-output) Linearization (Full State Feedback Linearization)

means a linearization w.r.t. the state(s) of the system where the linear behaviour is achieved from inputs to states. Therefore, it is not necessary to specify any output(s). **input-state linearization**

5.3.2 Nonlinear Control

Here, the control law is designed directly for the original WMR equations. This approach is common, e.g., in case of reactive obstacle avoidance, when direct mapping of sensors to actuators is used (the WMR is controlled using, e.g., translational velocity v and angular velocity ω).

5.4 Control Algorithms for WMRs—An Overview

In this section, various types of WMR control techniques for set-point stabilization and trajectory tracking are described. Due to huge volume of available literature this overview is not intended to be complete.

5.4.1 Set-point Stabilization

definition The term set-point stabilization is another name for transition of a WMR to goal posture (configuration). The usual selection of the goal configuration is the origin of global coordinate system (Cartesian, polar). By appropriate coordinate transformation, any goal posture can be transformed to origin, and therefore selection of the origin does not affect generality of developed control algorithms.

difficulty The main problem in set-point stabilization of WMRs is impossibility of their smooth static state feedback stabilizability[1], following from the fact that the system is underactuated, i.e., there are less control inputs than degrees of freedom.

control algorithms There are two main groups of control algorithms for set-point stabilization that can be further divided into subgroups:

- open-loop control algorithms:
  - *control with sinusoidal (oscillatory), polynomial and piecewise-constant control inputs*, can be found in, e.g., [48], [49];
  - *for nilpotent systems* (systems with property that all iterated Lie brackets from certain order equal to zero; examples are chained and power forms discussed below), can be found in, e.g., [23];
  - *control based on flatness of the system*;
- closed-loop control algorithms:
  - *time-varying feedback control*, which means inclusion of exogenous time-varying signal into controller, can be found in, e.g., [28];
  - *discontinuous feedback control*, which uses time invariant controller discontinuous at the origin, can be found in, e.g., [3], [5];
  - *piecewise continuous stabilization*, can be found in, e.g., [53].

In [25], a switching controller for posture stabilization of a car-like WMR has been introduced. The controller consists of three constituent controllers, all designed by Lyapunov direct method. Controllers for generic models introduced in [12] (refer to Sect. 4.4.2) based on Lyapunov functions are designed in [40]). The final state (origin of the state space) is achieved via intermediate states with piece-wise constant feedback controls between them, generated by using Lie brackets.

[1] This doesn't apply to omnidirectional WMRs, because they are not nonholonomic systems.

Chained Form Systems

In [34], a control technique for systems is a special form (*chained* form), based on sinusoidal inputs, has been introduced. The constructive conditions for existence of a feedback transformation for systems with two control inputs were given. The systems up to order $n = 4$ can always be transformed into chained form [36].

special-form systems

The article [34] was an extension of [35], which addressed use of this technique for systems in special triangular form. Since then, the class of systems in chained form has been deeply investigated and many control algorithms have been developed. This includes either techniques for conversion of the nonholonomic systems (specifically tractor-trailer WMRs) into the chained form ([44], [2]), either control algorithms (discontinuous—[42], iterative with learning process—[38], based on time-invariant manifold technique [47]). The discontinuous and time-invariant manifold based controller possess smooth and natural output trajectories. In [54], an extended chained form (i.e., chained form with integrators added on inputs, described as extended kinematic model in Sect. 4.2.3) is considered and controller for car-like WMR is developed.

control algorithms

A special subset of the chained form, so-called *power form*, is described in, e.g., [26] and [18].

power form

As can be seen, there are numerous techniques and control algorithms developed for the set-point stabilization problem. Although stability issues are considered (mostly Lyapunov stability), the quality of transient process is sometimes poor (oscillatory, erratic) with undesirable effects in potential real-world use.

Static Feedback Linearization

Static feedback linearization uses static (time-invariant) feedback controller. As has been mentioned earlier, nonholonomic WMRs are not smoothly stabilizable via static state feedback (according to Brockett's theorem).

static feedback linearization

Nevertheless, a discontinuous coordinates transformation can ensure static feedback stabilizability. This process is demonstrated in [3].

The algorithm of building static feedback can be extended to a dynamic feedback linearization that can solve the problem also for nonholonomic WMRs.

Dynamic Feedback Linearization

Dynamic feedback linearization (DFL) is globally linearizing method. It is possible to design a controller for both tasks, set-point stabilization and trajectory tracking, using this technique. It will be described later, together with an example—DFL of a unicycle.

dynamic feedback linearization

5.4.2 Trajectory Tracking

Trajectory tracking means following given reference point that moves on defined curve (*path*). There are two sub-problems in this task—*reference trajectory generation* (also feedforward) and *error cancellation*. There could exist some requirements on the trajectory in order to be feasible for given WMR (such as for WMRs with constrained controls).

Trajectory tracking is a square control problem [36], i.e., there are two control inputs ($\mathbf{u} \in \mathbb{R}^2$) for zeroing two position error signals ($\mathbf{e} \in \mathbb{R}^2$).

This problem, unlike the PTPM, can be solved by a smooth feedback, since the linearization along a nonvanishing trajectory is controllable [36].

5.4.3 Nonlinear Control

Nonlinear control uses nonlinear control theory to deal with WMRs control.

An example of this approach can be found in [46], model predictive control (MPC) method is used to develop a nonlinear controller, and afterwards to combine it with proportional controller.

5.4.4 Dynamic Feedback Linearization (Example)

One variant of dynamic feedback linearization (DFL) for wheeled mobile robots has been introduced in [37]. This technique is based on so-called dynamic extension principle. The main advantage of DFL is that its result is full input-state-output linearization. It will be demonstrated on the unicycle WMR, as this is important for the subsequent parts of this thesis. The following example is taken from [37].

dynamic extension

First, it is necessary to define an m-dimensional output of the system $\boldsymbol{\eta} = f(\mathbf{q})$. In the case of a unicycle the suitable option is $\boldsymbol{\eta} = (x, y)$. To this output, the desired behaviour can be assigned.

output definition

Then, the procedure continues by successive differentiation of the output until the input appears in nonsingular form. During this process, it can be necessary to add integrators to one or more inputs to prevent subsequent differentiation of the original inputs. This algorithm creates a state ξ of the dynamic compensator and is called *dynamic extension* algorithm. If the system is invertible from the chosen output, the differentiation ends in finite number of steps. The condition for full input-state-output linearization is that the sum of the differentiation orders is equal to the dimension of extended state space. The resulting closed-loop system is then equivalent to a set of decoupled integrators from u_i to η_i ($i = 1, \ldots, m$).

algorithm

The unicycle kinematic model is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}. \tag{5.1}$$

The process of dynamic feedback linearization starts by defining $\boldsymbol{\eta} = (x, y)^T$ as the linearizing output of the unicycle WMR (4.18). Time-differentiation of $\boldsymbol{\eta}$ results in

$$\dot{\boldsymbol{\eta}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}. \tag{5.2}$$

From this it is easy to see that only v affects $\dot{\boldsymbol{\eta}}$. However, this equation does not allow to recover ω (angular velocity). Therefore it is necessary to add an integrator to the v input. Its state is denoted ξ, and so

$$v = \xi, \dot{\xi} = a \Rightarrow \dot{\eta} = \xi \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}. \tag{5.3}$$

This defines a new input a—linear acceleration of the unicycle.

After second differentiation of $\boldsymbol{\eta}$ by time it is obtained

$$\ddot{\boldsymbol{\eta}} = \dot{\xi} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + \xi\dot{\theta} \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} = \begin{pmatrix} \cos\theta & -\xi\sin\theta \\ \sin\theta & \xi\cos\theta \end{pmatrix} \begin{pmatrix} a \\ \omega \end{pmatrix}. \tag{5.4}$$

The matrix multiplying the modified input $(a, \omega)^T$ is nonsingular for $\xi \neq 0$. Taking this into account it can be defined

$$\begin{pmatrix} a \\ \omega \end{pmatrix} = \begin{pmatrix} \cos\theta & -\xi\sin\theta \\ \sin\theta & \xi\cos\theta \end{pmatrix}^{-1} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{5.5}$$

to obtain

$$\ddot{\boldsymbol{\eta}} = \begin{pmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{u}. \tag{5.6}$$

The dynamic compensator is

$$\begin{aligned} \dot{\xi} &= u_1\cos\theta + u_2\sin\theta \\ v &= \xi \\ \omega &= \frac{u_2\cos\theta - u_1\sin\theta}{\xi}. \end{aligned} \tag{5.7}$$

In the new coordinates

$$\begin{aligned} z_1 &= x \\ z_2 &= y \\ z_3 &= \dot{x} = \xi\cos\theta \\ z_4 &= \dot{y} = \xi\sin\theta \end{aligned} \tag{5.8}$$

the system is fully linearized and described by two chains of decoupled integrators (5.6) rewritten as

$$\begin{aligned} \ddot{z}_1 &= \dot{z}_3 = u_1 \\ \ddot{z}_2 &= \dot{z}_4 = u_2. \end{aligned} \tag{5.9}$$

Structure of the system (simulation scheme) is depicted in Fig. 5.2. It should be **simulation scheme**

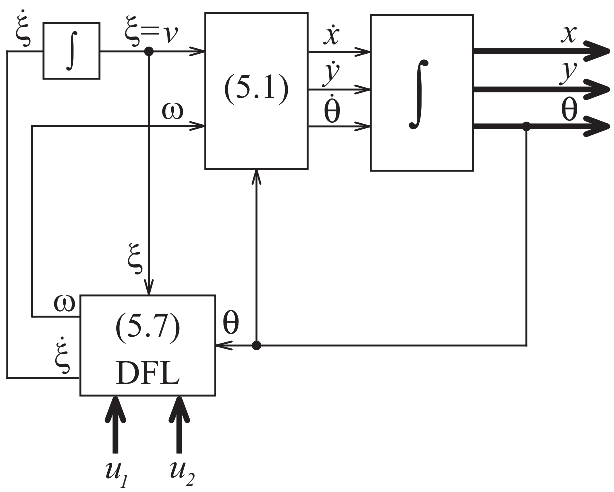


Figure 5.2: *Unicycle DFL structure*

noted that the dynamic compensator (5.6) has a singularity for $\xi = v = 0$, i.e., when the unicycle is stopped, and the singularity is structural for nonholonomic systems [37]. This is a minor drawback of the presented technique. The singularity is however present only for the steering velocity ω and can be avoided by appropriate rate of exponential convergence of the numerator such that the rate is higher than the one of the denominator. If this condition is satisfied, the numerator converges faster and therefore the angular velocity is zero before the singularity can occur.

For the linearized system, it is possible to design a PD control law

$$\begin{aligned} u_1 &= k_{p_1}(x_d - x) + k_{d_1}(\dot{x}_d - \dot{x}) + \ddot{x}_d \\ u_2 &= k_{p_2}(y_d - y) + k_{d_2}(\dot{y}_d - \dot{y}) + \ddot{y}_d, \end{aligned} \tag{5.10}$$

where $x_d, \dot{x}_d, \ddot{x}_d$ and $y_d, \dot{y}_d, \ddot{y}_d$ are the desired position, velocity and acceleration in given directions (this controller is designed for trajectory tracking). Should it be used for set-point stabilization (to the origin of the XY plane), then $x_d = \dot{x}_d = \ddot{x}_d = y_d = \dot{y}_d = \ddot{y}_d = 0$.

The derivation of conditions for exponential stabilization of the system (5.9) by the controller (5.10) are presented in Appendix D.

Chapter 6

Multi-steered Wheeled Mobile Robots

This chapter focuses on a class of WMRs with more steered conventional wheels (or axles, respectively, like on a multi-steered car). They will be denoted as multi-steered wheeled mobile robots (and abbreviated as MSWMRs further in this text).

multi-steered robots

There can be recognized two subgroups of MSWMRs, single-body and multi-body ones. The difference is the same as in case of non multi-steered WMRs. The multi-body MSWMRs refer to tractor-trailer systems where the trailers possess steered wheels.

In this chapter, the methods and theoretical background from previous part of the work will be applied primarily to examine a bi-steered WMR, also referred to as a pseudobicycle. A new control algorithm for set-point stabilization and trajectory tracking will be presented.

multi-steered cars

The multi-steered WMRs can resemble multi-steered cars that are usually denoted as four-wheel steered vehicles or four-wheel steering system (abbreviated 4 WS). These cars are in production by the car industry. In general, it is possible to find two approaches to four-wheel steering, *active* and *passive*.

active system

The first approach (active) usually employs a computer to control the rear axle, while the front one is steered by the human driver. The rear wheels are turned to opposite angle (direction) as the front wheels. This system uses the advantage of small turning radius and it is employed mainly at low speeds to increase maneuverability of the car. This is very advantageous namely for large trucks, but it is also advantageous, e.g., for parallel parking for all types of cars.

passive system

The second approach (passive) is usually used to compensate the tendency of the rear wheels to turn outside of a turn. This can reduce stability. The passive steering helps to prevent the out-turning and improves stability.

The above approaches improve stability and/or maneuverability of the car. The important property of such a car is that the steering angle of computer/passive controlled axle is a function of the actual steering angle of the human-controlled axle. The wheels/axles of a multi-steered WMR can be controlled *independently*.

Therefore, significant property of a multi-steered WMR is the fact that all its wheels are independently steerable. However, the wheels on each axle need to be steered correctly, so as the existence of the ICR is satisfied (as described in Sect. 4.1.2).

6.1 Bi-steered WMR

A kinematic simplification of a multi-steered car is also referred to as a *bi-steered WMR* (also *pseudobicycle* [43]). There are other denominations of this type of WMR that can be found in the literature, such as *two-steering-wheels mobile robot*, *bi-steerable car* [8], [19]. **bi-steered WMR**

When the bi-steered WMR is used to model a 4 WS car, then it is necessary to appropriately determine the steering angles of the 4 WS car based on the steering angles computed for bi-steered WMR. This problem is treated in Sect. 6.2.3. Using, e.g., dual Ackermann steering (one for each axle) is not possible. **bi-steered WMR vs. 4 WS**

It should be noted that another type of bi-steered WMR can be constructed, where the angle between wheels and body is not fixed, such as in case of two differentially steered robots connected by a rigid link. This modification has different behaviour (e.g., following a circular trajectory needs constant steering action for individual robots in the set). **bi-steered WMR variant**

Both wheels of the bi-steered WMR are steered. With driving, there are in principle two possibilities, like in case of a car. The first one is front-wheel driving, the second one is rear-wheel driving. When driving of both wheels is considered, then appropriate harmonization of the velocities must be guaranteed. This follows from the rolling without slipping and the no lateral movement constraints. It is a form of differential gear between the axles. When a multi-steered car is considered, then it is naturally necessary to employ the differential gear for the wheels on each axle. **bi-steered WMR properties**

6.2 Models of MSWMRs

To model a MSWMR, kinematic or dynamic model can be used (refer to Sect. 4.2). Kinematic model can use advantage of kinematic simplification (described in Sect. 4.1.5), however, it has limited validity (refer to Sect. 4.2.3).

6.2.1 Multi-steered Car

This multi-steered WMR is a single-body WMR with two independently steered axles. It will be denoted as *four-steered WMR* in the following text. As in the case of all chassis and especially the conventional car, the ICR existence condition must be satisfied. Therefore, appropriate steering has to be applied (it is discussed in Sect. 6.2.3). For modelling and control purposes, a kinematically simplified model can be conveniently used. **four-steered WMR**

6.2.2 Bi-steered WMR

A *bi-steered WMR* has two wheels connected to a rigid body and represents a model (or kinematic simplification, respectively) of a car with all steered wheels (both axles can be steered). The wheels maintain constant angles w.r.t. the body of the robot unless it is changed by appropriate control input (steering). Following from Chasles' theorem and **bi-steered WMR**

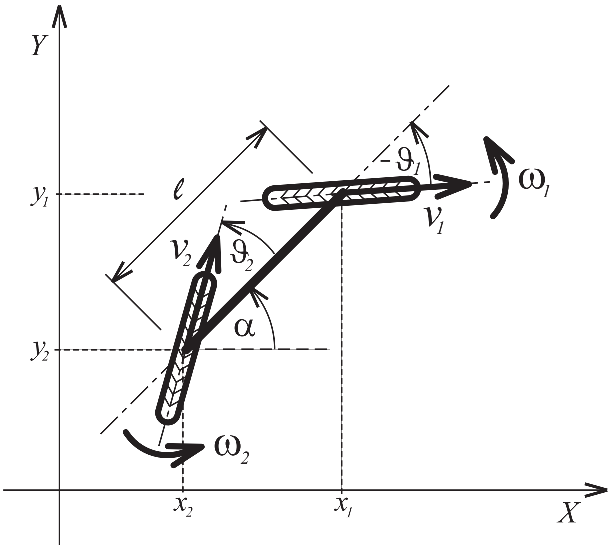


Figure 6.1: *Bi-steered WMR model*

ICR existence condition (refer to Sect. 4.1.2) and [12], this WMR is a generic model for single-body MSWMRs.

Kinematic Model

In short, to create a kinematic model, it is necessary to select the inputs and states (in the case of the model, they can be regarded as outputs) and to establish relations between them.

state variables

The state of the bi-steered WMR (as depicted in Fig. 6.1) can be determined by a vector of generalized coordinates $\mathbf{q} = (x, y, \vartheta_1, \vartheta_2, \alpha)$. The first two coordinates, x $[m]$ and y $[m]$ represent the position of the reference point in Cartesian coordinates, the angles ϑ_1, ϑ_2 $[rad]$ orientation of the wheels w.r.t. the body of the robot (relative to the X^+ axis of the local robot coordinate system) and α $[rad]$ denotes the orientation of the robot w.r.t. the X^+ axis of the global reference frame (the angle ϑ_1 in Fig. 6.1 is negative due to the fact that it is measured w.r.t. the body of the robot).

The state space of the bi-steered WMR is therefore $\mathcal{C} \in \mathbb{R}^n$ with $n = 5$.

The (control) inputs of the model are the velocities, specifically the translational (driving) velocity of the bi-steered WMR and the rotational velocities of constituent wheels $\dot{\vartheta}_1, \dot{\vartheta}_2$ $[rad.s^{-1}]$. Thus, the input space $\mathcal{U} \in \mathbb{R}^m$ has the dimension $m = 3$.

There are two possibilities for choice of the driving velocity—for the front wheel or for the rear wheel (the model is different in each case). If both wheels are supposed to be driven, then it is necessary to harmonize their velocities. The *state transition equation* for the *front-driven* bi-steered WMR is

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2+\alpha) \\ \cos\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2+\alpha) \\ 0 \\ 0 \\ (\sin\vartheta_1 - \cos\vartheta_1 \tan\vartheta_2)/\ell \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \omega_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_2. \tag{6.1}$$

The complete procedure of derivation of the above model can be found in Appendix A.

For comparison, the model of *rear-driven* bi-steered WMR is presented. The reference point is the same, and is located at the vertical axis of the rear wheel.

rear-driven bi-steered WMR

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos(\vartheta_2+\alpha) \\ \sin(\vartheta_2+\alpha) \\ 0 \\ 0 \\ (\cos\vartheta_2 \tan\vartheta_1 - \sin\vartheta_2)/\ell \end{pmatrix} v_2 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \omega_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_2. \tag{6.2}$$

As the reference point remains the same, the equations become simpler.

6.2.3 Four-steered WMR—Computation of Steering Angles

Bi-steered WMR kinematics represents also kinematics of four-steered WMR. The control algorithm for bi-steered WMR, however, determines three control input $(v_1, \vartheta_1, \vartheta_2)$, where the steering angles ϑ_1, ϑ_2 represent the orientation of the wheels w.r.t. the body of the WMR (more precisely, its longitudinal axis).

Should the control algorithm need to be used for control of the four-steered WMR (i.e., non-simplified, with all four wheels controlled), the proper steering angles of the wheels have to be computed (Sect. 4.1.2).

The position of ICR of the bi-steered WMR can be determined from angles of the wheels (Fig. 6.2). Using the sine law

$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)} \tag{6.3}$$

and

$$\alpha + \beta + \gamma = \pi, \tag{6.4}$$

it can be written

$$\begin{gathered} \left(\tfrac{\pi}{2} - \vartheta_1\right) + \left(\tfrac{\pi}{2} + \vartheta_2\right) + \vartheta_3 = \pi \\ \Downarrow \\ \vartheta_3 = \vartheta_1 - \vartheta_2 \end{gathered} \tag{6.5}$$

and

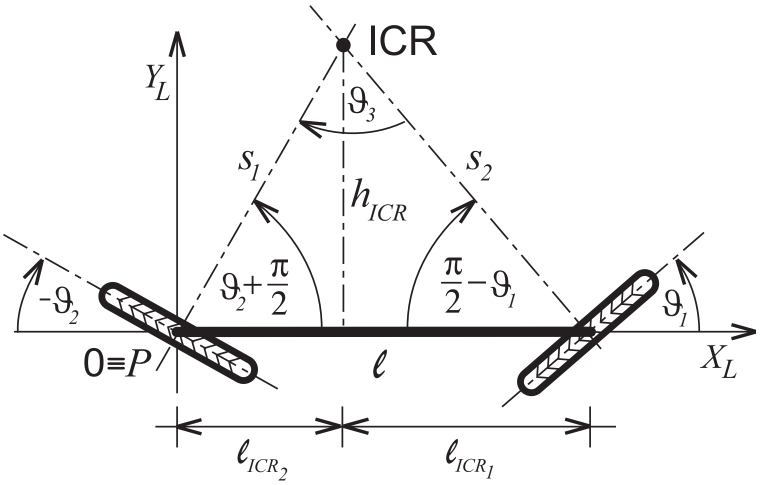


Figure 6.2: *Bi-steered WMR ICR computation*

$$\frac{s_1}{\sin\left(\frac{\pi}{2}-\vartheta_1\right)} = \frac{s_2}{\sin\left(\frac{\pi}{2}+\vartheta_2\right)} = \frac{\ell}{\sin\left(\vartheta_3\right)} \tag{6.6}$$

to obtain

$$\begin{aligned} x_{ICR_L} &= \ell_{ICR1} = \frac{-\ell\cos\vartheta_1\sin\vartheta_2}{\sin(\vartheta_1-\vartheta_2)} \\ y_{ICR_L} &= h_{ICR} = \frac{\ell\cos\vartheta_1\cos\vartheta_2}{\sin(\vartheta_1-\vartheta_2)}, \end{aligned} \tag{6.7}$$

where $x_{ICR_L} = \ell_{ICR_2}$ and $y_{ICR_L} = h_{ICR}$ are the coordinates of the ICR in *local* coordinate system $0_{X_LY_L}$ ($P_{X_LY_L}$, respectively).

Then, the following equations can be used to determine the angles of the wheels. Derivation is straightforward from geometry and will be demonstrated for the front left wheel with steering angle ϑ_{1L} or $\vartheta'_{1L} = \frac{\pi}{2} - \vartheta_{1L}$, respectively (refer to Fig. 6.3)

$$\tan\vartheta'_{1L} = \frac{h_{ICR}-d}{\ell-\ell_{ICR1}} \tag{6.8}$$

from where

$$\vartheta'_{1L} = \arctan\left(\frac{h_{ICR}-d}{\ell-\ell_{ICR1}}\right). \tag{6.9}$$

Substituting (6.7) into (6.9) after simplification yields

$$\vartheta'_{1L} = \arctan\left(\frac{-d+\cot\vartheta_1(\ell+d\tan\vartheta_2)}{\ell}\right) \tag{6.10}$$

and then $\vartheta_{1L} = \frac{\pi}{2} - \vartheta'_{1L}$.

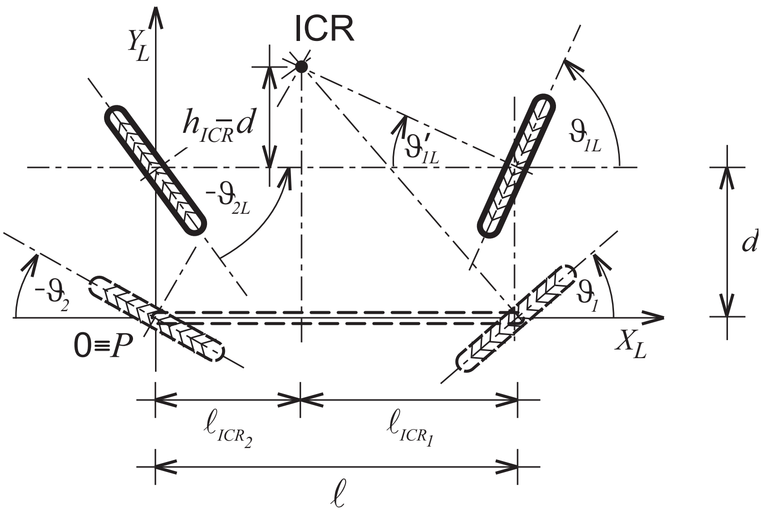


Figure 6.3: *Four-steered WMR—computation of wheel angles; displayed bi-steered WMR (dashed) and front and rear left wheel (solid)*

Similarly for the front right wheel

$$\vartheta'_{1R} = \arctan\left(\frac{d + \cot\vartheta_1(\ell - d\tan\vartheta_2)}{\ell}\right);\ \vartheta_{1R} = \frac{\pi}{2} - \vartheta'_{1R} \tag{6.11}$$

and for the rear wheels (the left rear wheel is used for demonstration)

$$\tan\vartheta'_{2L} = \frac{h_{ICR} - d}{\ell_{ICR1}} \tag{6.12}$$

from where

$$\vartheta'_{2L} = \arctan\left(\frac{h_{ICR} - d}{\ell_{ICR1}}\right) \tag{6.13}$$

and

$$\begin{aligned} \vartheta'_{2L} &= -\arctan\left(\frac{d + \cot\vartheta_2(\ell - d\tan\vartheta_1)}{\ell}\right);\ \vartheta_{2L} = \frac{\pi}{2} + \vartheta'_{2L} \\ \vartheta'_{2R} &= \arctan\left(\frac{d - \cot\vartheta_2(\ell + d\tan\vartheta_1)}{\ell}\right);\ \vartheta_{2R} = \frac{\pi}{2} + \vartheta'_{2R}. \end{aligned} \tag{6.14}$$

6.3 Control of Multi-steered WMRs—State of the Art

A literature survey on control of multi-steered (bi-steered or four-steered, respectively) WMRs follows. **literature survey**

The main areas for practical applications of control algorithms for multi-steered WMRs are the automotive industry (improvement of maneuverability) and the agriculture (design/control of autonomous, highly-maneuverable agricultural vehicles). **application areas**

A control method for bi-steered WMR can be found in [32]. Two approaches to control for *trajectory tracking* task are presented—feedback linearization and Lyapunov function-based. The used approach is based on expressing the position of the MSWMR in Frenet frame[1] and convergence to zero error (deviation from the origin of the coordinate system). The control algorithm assumes that the velocity of the reference point (vertical axle of the first wheel) is predetermined. Therefore, the error in y coordinate and angle between the robot body orientation and tangent to the trajectory is controlled (zeroed by the controller). This task, according to the definition in Sect. 5.2, should be considered as the *path following*, as the time law is not defined by the desired trajectory but an external law.

In [8], a controller for path-following based on fuzzy logic can be found. Within this paper, the bi-steered WMR is referred to as bi-steered car. The controller is designed w.r.t. the comfortable longitudinal and lateral accelerations and w.r.t. the no-slip requirement. Therefore, the maximum velocity is generated taking into account estimated tyre-surface friction coefficient. The controller is a representative of the state-action mapping controllers.

In [13], a backstepping controller is developed for trajectory tracking for four-steered WMR. A kinematic model extended with lateral slip parameters is used, in order to enable modelling of agricultural vehicles on inclined surface. The slip parameters are supposed to be indirectly estimated. The model is converted to a chained form in order to be linearized.

In [31], a controller for global tracking was designed for bi-steered WMR dynamic model.

6.4 Control of a Bi-steered WMR Using Decomposed Controller

A new method for controlling a bi-steered WMR using two controllers (one for each wheel—thus *decomposed*) is developed in this section. It represents an effective technique for either set-point stabilization and trajectory tracking tasks (please refer to Sect. 5.2).

6.4.1 Controller design

A controller is developed for the front-wheel driven bi-steered WMR in the following text. Each wheel is controlled by its own controller. The first (front, i.e., driven) one is controlled by a DFL compensator and associated PD controller (denoted only as controller in the following text for simplicity). This controller computes both translational and rotational velocities. **introduction**

The controller for the second (steered, non-driven) wheel computes the steering velocity. It is derived using the equation for desired rotational velocity and the rigid WMR

[1]Frenet frame—in 3D space, this frame consists of the tangent $\mathbf{t}$, normal $\mathbf{n}$ and binormal $\mathbf{b}$ that together form an orthonormal basis ($\mathbf{b} = \mathbf{t} \times \mathbf{n}$). Named after Jean Frédéric Frenet (1816-1900), French mathematician, astronomer and meteorologist.

body assumption. Due to this assumption, the translational velocity is inherited from the first wheel (respecting the actual orientation of the second wheel).

This assures that the computed control signals respects the physical dimensions of the bi-steered WMR (constant distance between the wheels—rigid body assumption).

steering velocities

The first controller controls the driven wheel (the front one). The process of controller design consists with the one for unicycle WMR (Sect. 5.4.4). The difference is in the fact that the orientation of the wheels of the bi-steered WMR is measured w.r.t. the longitudinal axis of the WMR (unlike the orientation of the unicycle, measured w.r.t. the $X+$ axis).

notation

In the subsequent text, the steering velocities of the first and second wheel in global coordinate system (measured w.r.t. the $X+$ axis) will be denoted as Ω_1 and Ω_2, respectively, whereas the local ones (measured w.r.t. the longitudinal WMR axis) will be denoted by ω_1 and ω_2. This notation is different from the one used for DFL control of a unicycle in Sect. 5.4.4. However, the variables ω_1 and ω_2 refer to the "real" WMR inputs in both cases (although in different coordinate systems—global for unicycle, local for bi-steered WMR).

Therefore, the steering velocity inputs Ω_1 and Ω_2 have to be transformed accordingly (i.e., re-computed from the global coordinate system to local one, by subtracting the angular velocity of the WMR body, $\dot{\alpha}$).

controller development

The bi-steered front-wheel driven WMR equations are as follows

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2+\alpha) \\ \cos\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2+\alpha) \\ 0 \\ 0 \\ (\sin\vartheta_1 - \cos\vartheta_1 \tan\vartheta_2)/\ell \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \omega_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_2. \tag{6.15}$$

The equations of the controller are

$$\begin{aligned} \dot{\xi}_1 &= u_{11}\cos\Theta_1 + u_{12}\sin\Theta_1 \\ v_1 &= \xi_1 \\ \omega_1 &= \frac{u_{12}\cos\Theta_1 - u_{11}\sin\Theta_1}{\xi_1} - \frac{\xi_1(\sin\vartheta_1 - \cos\vartheta_1\tan\vartheta_2)}{\ell}, \end{aligned} \tag{6.16}$$

where $\Theta_1 = \vartheta_1 + \alpha$ and with

$$\begin{aligned} u_{11} &= \ddot{x}_d + k_{p_{11}}(x_{1d} - x_1) + k_{d_{11}}(\dot{x}_{1d} - \dot{x}_1) \\ u_{12} &= \ddot{y}_d + k_{p_{12}}(y_{1d} - y_1) + k_{d_{12}}(\dot{y}_{1d} - \dot{y}_1). \end{aligned} \tag{6.17}$$

Controller for the second wheel can be derived from

$$\omega_2 = \frac{\ddot{y}_2\dot{x}_2 - \ddot{x}_2\dot{y}_2}{\dot{x}_2^2 + \dot{y}_2^2}, \tag{6.18}$$

that has been obtained from

$$\tan\Theta_2 = \frac{\dot{y}_2}{\dot{x}_2} \Rightarrow \Theta_2 = \arctan\left(\frac{\dot{y}_2}{\dot{x}_2}\right) \tag{6.19}$$

and

$$\Omega_2 = \dot{\Theta}_2. \tag{6.20}$$

Substituting

$$\begin{aligned} \dot{x}_2 &= \xi_2 \cos\Theta_2 \\ \dot{y}_2 &= \xi_2 \sin\Theta_2 \end{aligned} \tag{6.21}$$

into (6.18) yields

$$\Omega_2 = \frac{\ddot{y}_2 \xi_2 \cos\Theta_2 - \ddot{x}_2 \xi_2 \sin\Theta_2}{\xi_2^2} = \frac{\ddot{y}_2 \cos\Theta_2 - \ddot{x}_2 \sin\Theta_2}{\xi_2}, \tag{6.22}$$

where $\Theta_2 = \vartheta_2 + \alpha$ and

$$\xi_2 = v_2 = \xi_1 \cos\vartheta_1 \sec\vartheta_2, \tag{6.23}$$

which is the translational velocity of the second wheel computed from the velocity of the first (driven) wheel and actual steering angles of the wheels. There is a potential singularity of ξ_2 for $\vartheta_2 = \frac{\pi}{2}$.

To design stabilizing controller for $(\eta_1, \eta_2) = (x_2, y_2)$, equations $u_{21} = \ddot{x}_2$ and $u_{22} = \ddot{y}_2$ can be used with

$$\begin{aligned} u_{21} &= \ddot{x}_d + k_{p_{21}}(x_{2d} - x_2) + k_{d_{21}}(\dot{x}_{2d} - \dot{x}_2) \\ u_{22} &= \ddot{y}_d + k_{p_{22}}(y_{2d} - y_2) + k_{d_{22}}(\dot{y}_{2d} - \dot{y}_2). \end{aligned} \tag{6.24}$$

As has been mentioned earlier, the controller (6.22) computes the control inputs in global coordinate system. For conversion to local coordinate system of the bi-steered WMR, the conversion term (subtraction by $\dot{\alpha}$) has to be included, i.e.,

$$\omega_2 = \frac{u_{22}\cos\Theta_2 - u_{21}\sin\Theta_2}{\xi_2} - \frac{\xi_1(\sin\vartheta_1 - \cos\vartheta_1 \tan\vartheta_2)}{\ell}. \tag{6.25}$$

This finishes design of the controller for the steering velocity ω_2 of the second bi-steered WMR wheel. As can be seen, the designed controller is structurally consistent with the DFL controller (6.16). For ω_2, it is necessary to ensure that $\xi_2 \neq 0$, and this is true for $v_1 \neq 0$ and $\vartheta_1 \neq \frac{\pi}{2}$. The former condition is satisfied by assumptions presented in Appendix D, the latter by assumption on steering angles of the respective wheels. The structure (simulation model) of the controller is depicted in Fig. 6.4.

v_2 error minimization

The next issue that needs to be solved is minimization of v_2 error, i.e., a difference between its desired (computed) and real value. This can be done by defining additional variable η_3 as

$$\eta_3 = v_2 - v_1 \cos\vartheta_1 \sec\vartheta_2 = \xi_2 - \xi_1 \cos\vartheta_1 \sec\vartheta_2 \tag{6.26}$$

and by differentiation by time it is obtained

$$\dot{\eta}_3 = \dot{\xi}_2 - \dot{\xi}_1 \cos\vartheta_1 \sec\vartheta_2 + \xi_1\dot{\vartheta}_1 \sin\vartheta_1 \sec\vartheta_2 - \xi_1\dot{\vartheta}_2 \cos\vartheta_1 \sec\vartheta_2 \tan\vartheta_2. \tag{6.27}$$

Then it is possible to straightforwardly define a stabilizing controller

$$\dot{\eta}_3 = -k_{\eta_3}\eta_3, \tag{6.28}$$

where $k_{\eta_3} > 0$.

The proposed controller is suitable for front wheel wheel driven bi-steered WMR. The same (slightly modified) design can also be used for the rear wheel driven bi-steered WMR.

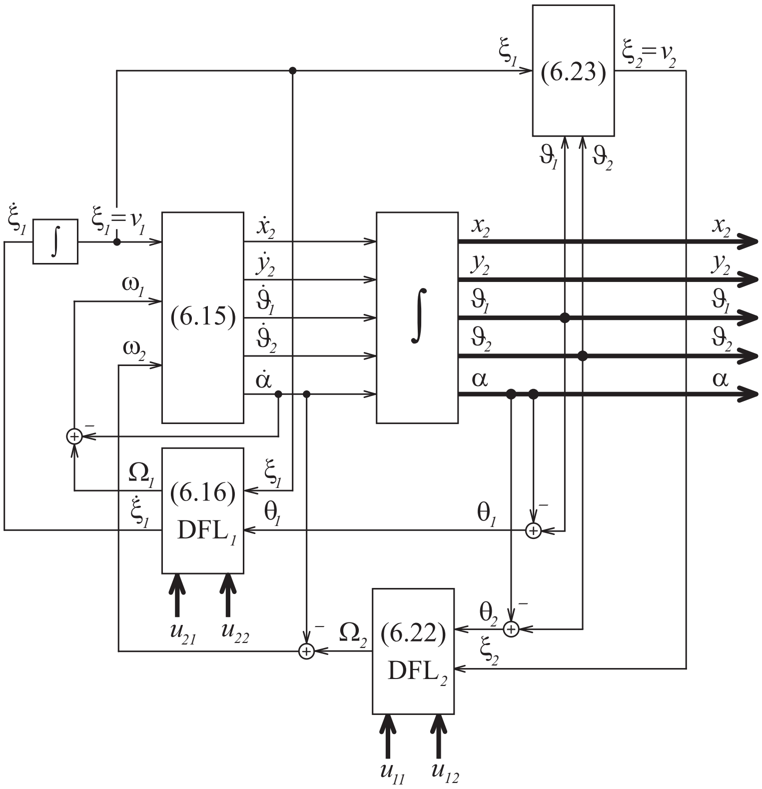


Figure 6.4: *Bi-steered WMR decomposed controller structure (without v_2 error minimization controller)*

6.4.2 Set-point Stabilization

An important part of this task is selection of the goal point (final configuration). Based on the chosen approach, it is necessary to assign to each wheel a separate desired goal point, respecting the dimensions (i.e., length) of the bi-steered WMR. It is assumed that the desired goal state of the bi-steered WMR is $(x, y, \vartheta_1, \vartheta_2, \alpha) = \mathbf{0}$, i.e., the bi-steered

WMR is "backward parked" at the origin of the coordinate system. This means that the reference point of the bi-steered WMR is at the origin of the coordinate system, the body of the robot is aligned with the X^+ axis and the wheels are at zero angle. Therefore, the goal point for the first wheel is $(\ell, 0)$ and for the second unicycle is $(0, 0)$, where ℓ is the length (distance between the wheels) of the bi-steered WMR.

Simulation Results

Fig. 6.5 demonstrates a stroboscopic view of the stabilization process. The individual postures are time-equidistant. For better readability the end is not included, because the bi-steered WMR approaches the origin by motion along the X axis.

The goal orientation of each each wheel of the bi-steered WMR is determined by appropriate setting of the constants of the controllers.

The initial conditions used for the simulation were: $x_{01} = 3.5$,$y_{01} = 2; x_{02} = 3$, $y_{02} = 1.5$, $v_{01STAB} = -0.5$; $\omega_{01} = 0$;, $v_{01TRAJ} = 2; \omega_{01} = 0$; $\Theta_{01} = \pi/6; \Theta_{02} = \pi/12$ $[m, m.s^{-1}, rad]$, with the controller constants: $k_{p11} = k_{p21} = 2; k_{d11} = k_{d21} = 3$, $k_{p21} = k_{p22} = 7; k_{d21} = k_{d22} = 12$.

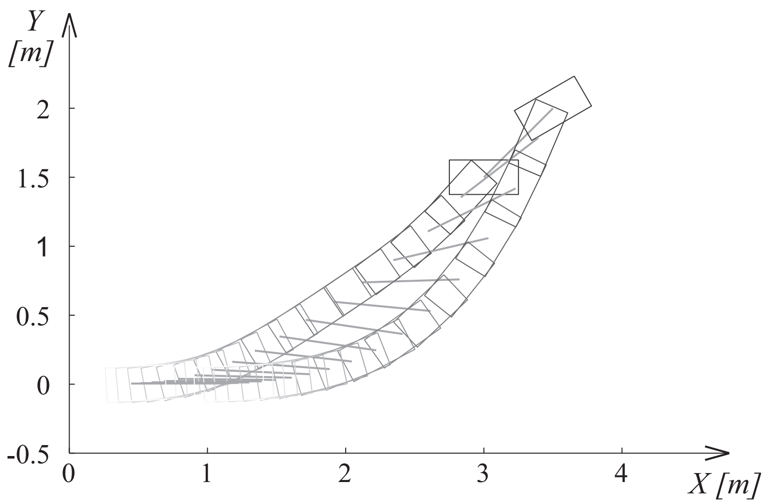


Figure 6.5: *Stroboscopic view of bi-steered WMR set-point stabilization*

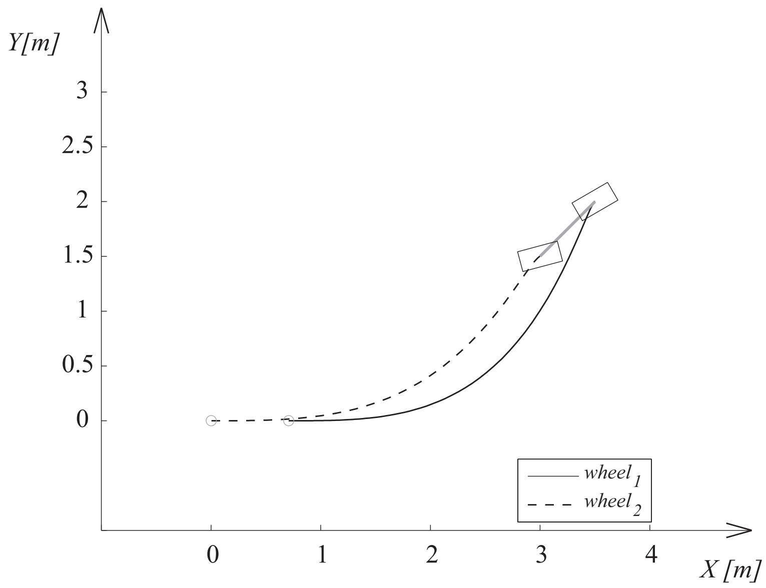


Figure 6.6: *Bi-steered WMR set-point stabilization—transition trajectories*

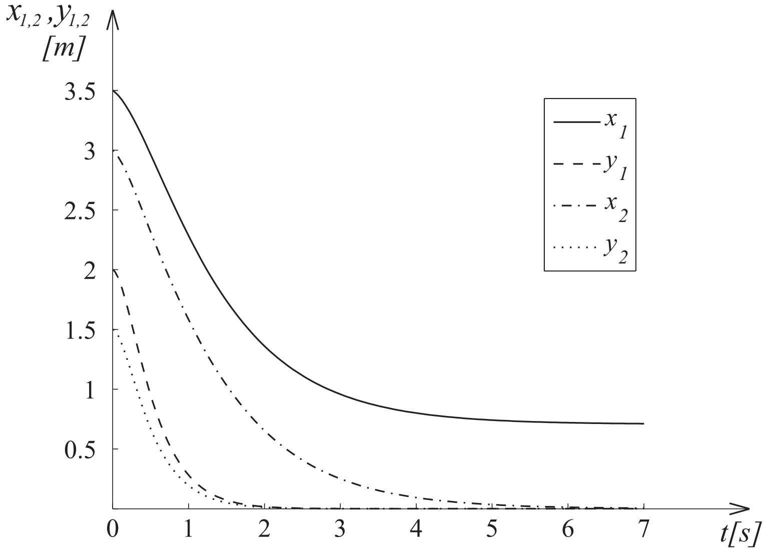


Figure 6.7: *Bi-steered WMR set-point stabilization—time development of Cartesian positions*

Fig. 6.6 presents the stabilization transition trajectories. Time evolution of the transition process (i.e., time-evolution of the position variables) is depicted in Fig. 6.7. Time evolution of the control variables (v_1, v_2, ω_1, ω_2) is depicted in Figs. 6.8 and 6.9. Time evolution of the angle variables (Θ_1, Θ_2, α) is depicted in Fig. 6.10.

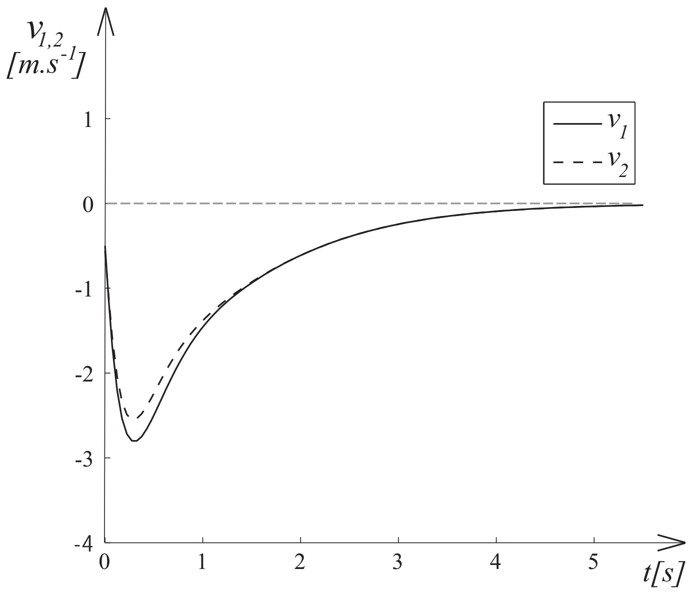


Figure 6.8: *Bi-steered WMR set-point stabilization—translational velocities*

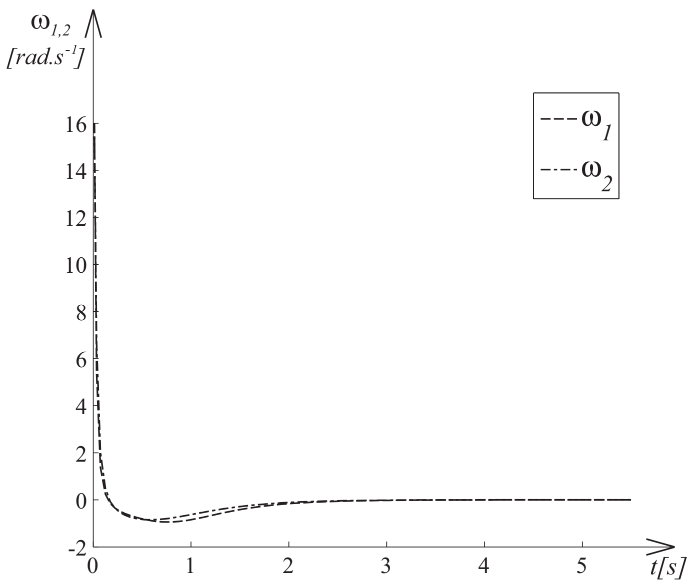


Figure 6.9: *Bi-steered WMR set-point stabilization—steering velocities of the wheels*

In Fig. 6.11, set-point stabilization from different initial configurations is presented. The initial conditions are summarized in Tab. 6.1 (physical units are omitted).

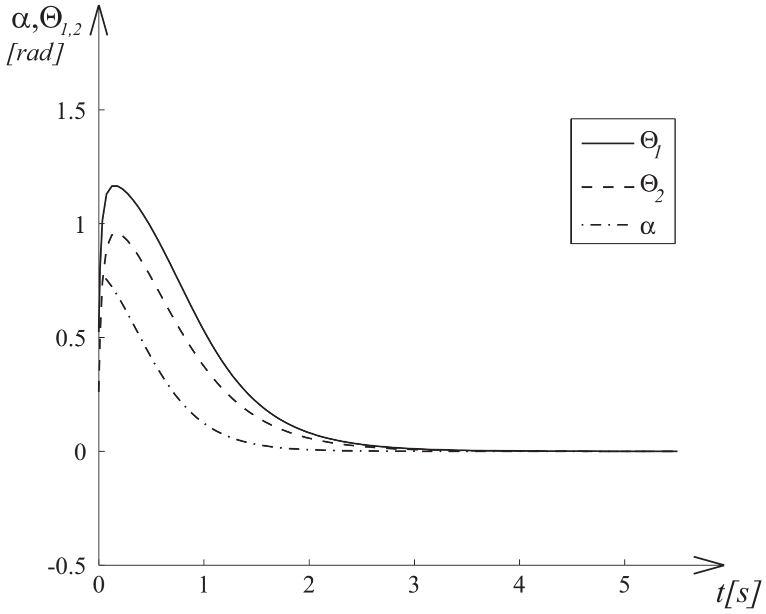


Figure 6.10: *Bi-steered WMR set-point stabilization—angle variables*

6.4.3 Trajectory Tracking

As for the set-point stabilization, it is also necessary to ensure generation of the reference points. In this case, however, the reference points are not static but move along the

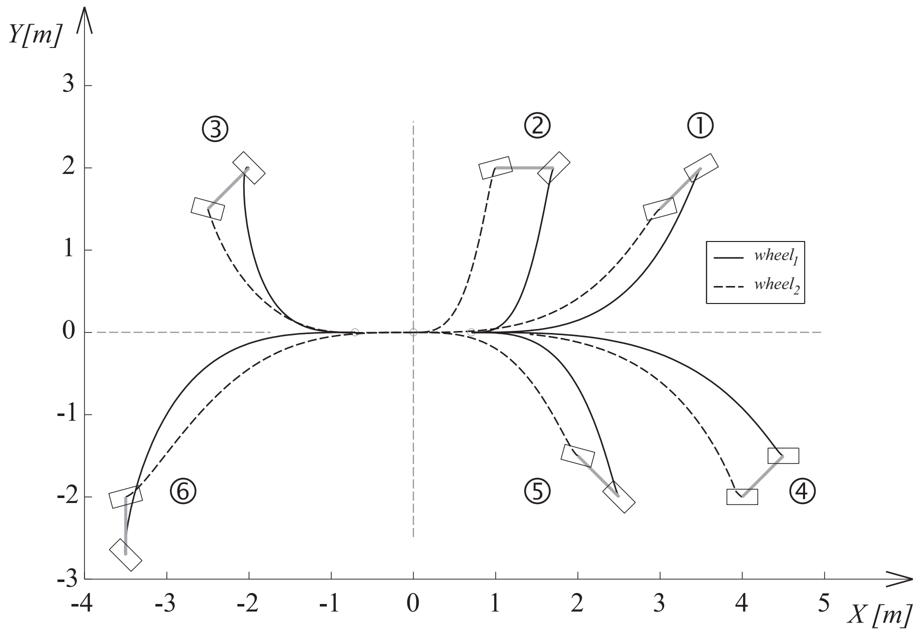


Figure 6.11: *Bi-steered WMR set-point stabilization—multiple initial configurations*

Table 6.1: *Set-point stabilization—initial conditions*

No.	Initial conditions
1	$x_{01} = 3.5, y_{01} = 2, \Theta_{01} = \frac{\pi}{4}$, $x_{02} = 3, y_{02} = 1.5, \Theta_{02} = \frac{\pi}{12}$, $v_{01} = -0.5$
2	$x_{01} = 1.7, y_{01} = 2, \Theta_{01} = \frac{\pi}{4}$, $x_{02} = 1, y_{02} = 2, \Theta_{02} = \frac{\pi}{12}$, $v_{01} = -0.5$
3	$x_{01} = -2, y_{01} = 2, \Theta_{01} = -\frac{\pi}{4}$, $x_{02} = -2.5, y_{02} = 1.5, \Theta_{02} = -\frac{\pi}{12}$, $v_{01} = -1$
4	$x_{01} = 4.5, y_{01} = -1.5, \Theta_{01} = 0$, $x_{02} = 4, y_{02} = -2, \Theta_{02} = 0$, $v_{01} = -0.5$
5	$x_{01} = 2.5, y_{01} = -2, \Theta_{01} = -\frac{\pi}{4}$, $x_{02} = 2, y_{02} = -1.5, \Theta_{02} = -\frac{\pi}{12}$, $v_{01} = -0.5$
6	$x_{01} = -3.5, y_{01} = -2.7, \Theta_{01} = -\frac{\pi}{4}$, $x_{02} = -3.5, y_{02} = -2, \Theta_{02} = \frac{\pi}{12}$, $v_{01} = -0.5$

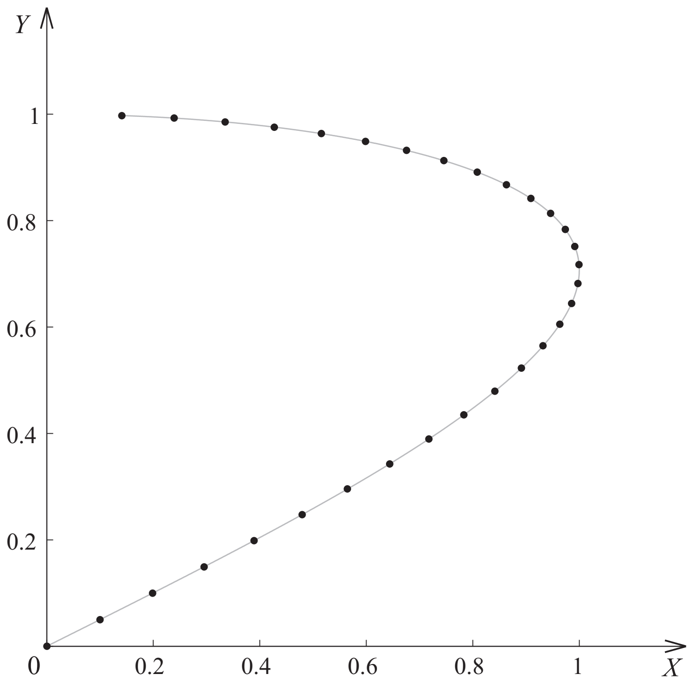


Figure 6.12: *Parametric trajectory example—time-equidistant points*

desired trajectory. The trajectory can be mathematically described (or generated) in several ways, however it is required that it is twice differentiable (of C^2 class) as for the employed controllers it is necessary to obtain desired velocity and acceleration of the individual wheels.

The problem is therefore in generation of appropriate reference points, i.e., two equidistantly spaced points belonging to a curve representing the desired trajectory. Depending on the shape of the trajectory, the constant physical distance of these two points on that trajectory does not necessarily mean constant distance in time (or parameter). Fig. 6.12 shows a parametric curve $x = sin(t/10)$, $y = sin(t/20)$ for the parameter $t \in <0, 30>$ with equidistantly spaced points for distance $t_{dist} = 1$. On the other hand, the situation is simple in case of circle or line.

The task of reference points generation can be translated into a problem of finding intersection(s) of two 2D curves—the desired trajectory and a circle, which represents the bi-steered WMR. The centre of the circle represents the front wheel of the bi-steered WMR, while the radius is equal to the distance between the bi-steered WMR wheels. In this case, the problem can arise when more than one intersection points are found (in most cases there are two). The challenge here is to determine the correct one.

In principle, one can resort to two types of solution—analytical and numerical one. Analytical solution is not a trivial task in this case.

One possible solution is to use algorithms for finding intersections of two 2D curves, known from, e.g., computer graphics, such as [29]. These algorithms are usually optimized for speed and low computational complexity, which is also advantageous in applications.

Simulation Results

Figs. 6.13, 6.14, 6.15 and 6.16 present the result of the trajectory tracking task simulation. The reference trajectory is circular with radius $r = 1\ m$. This trajectory possesses the property of constant difference in parameter for constant distance in length. The initial conditions were: $x_{01} = -1.5, y_{01} = 0.25, x_{02} = -0.25, y_{02} = 1.25, v_{01} = 1$, $\Theta_{01} = \pi/6$, $\Theta_{02} = \pi/2\ [m, m.s^{-1}, rad]$ and controller constants $k_{p11} = 12, k_{d11} = 7, k_{p21} = 12, k_{d21} = 7$ and $k_{p21} = 12, k_{d21} = 7, k_{p22} = 12, k_{d22} = 7$. These constants ensure equal rate of convergence in X and Y axes, that is desirable for trajectory tracking (results in equal error minimization rate).

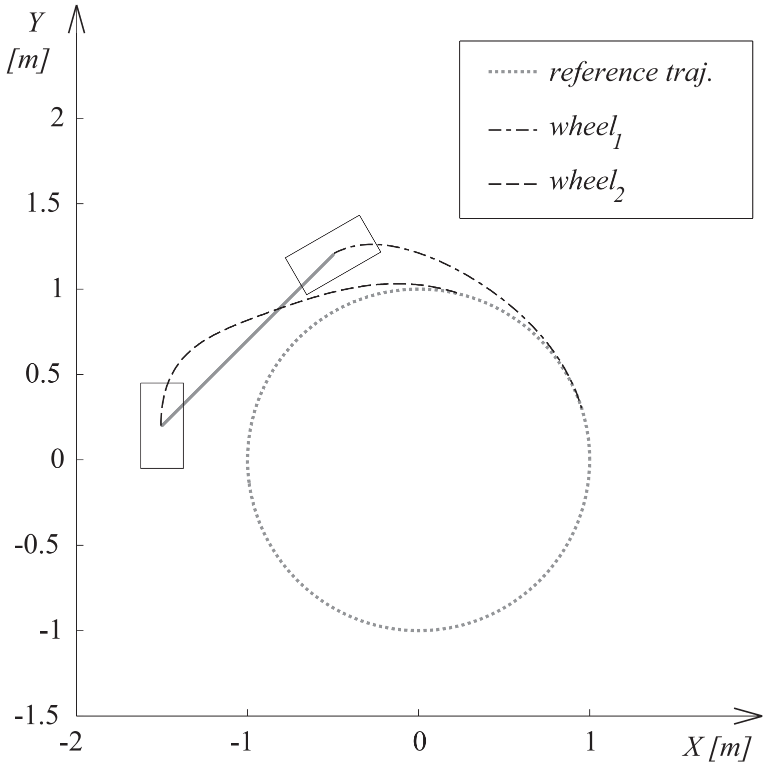


Figure 6.13: *Trajectory tracking—circular trajectory*

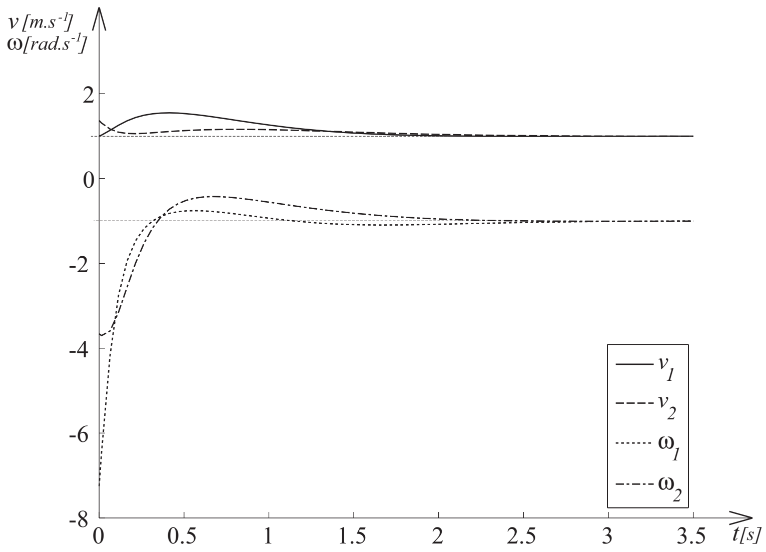


Figure 6.14: *Trajectory tracking—circular trajectory, v, ω*

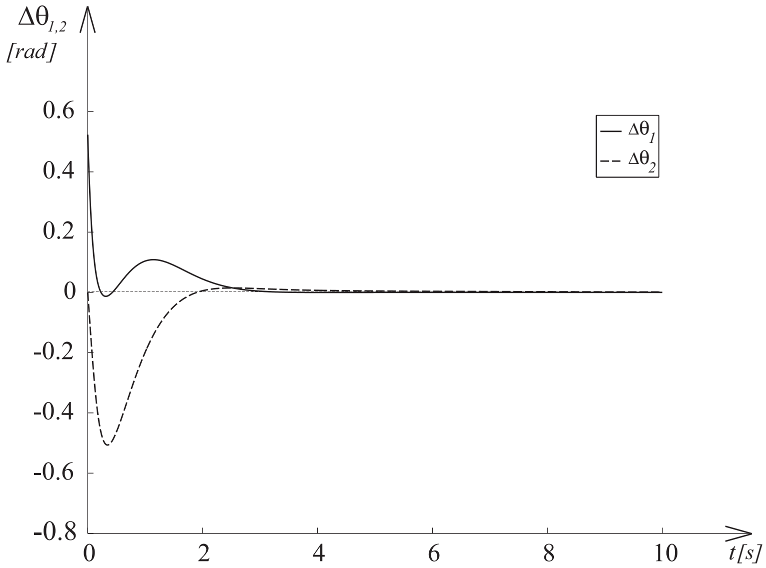


Figure 6.15: *Trajectory tracking—circular trajectory, angle errors*

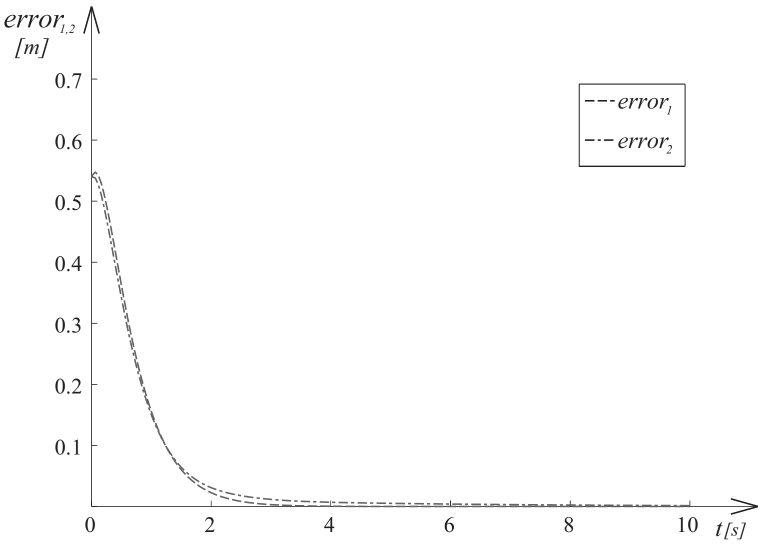


Figure 6.16: *Trajectory tracking—circular trajectory, Cartesian errors*

Fig. 6.13 shows the quality of the transition process and Fig. 6.14 time evolution of the control variables. Fig. 6.15 represents the time evolution of errors of the wheel orientation and Fig. 6.16 represents time evolution of Cartesian errors (distance from the reference points) of the first and second wheel, respectively.

6.5 Further Development

Further development would require treatment of allowed initial conditions for the bi-steered WMR and examination of the decomposed controller stability, which is not a trivial task. Next path in the area of kinematic control of multi-steered wheeled mobile robots can lead to control of vehicles with limited control inputs (such as steering angles and velocities). Further extension of the work is toward dynamic models and their control. The method presented for the bi-steered WMR can be applied (generalized) for chained types of MSWMRs (tractor-trailer MSWMR). This means that n of multi-steered trailers are hooked behind a tractor (bi-steered WMR or unicycle WMR). In this case, the control method can be easily extended for the tractor-trailer MSWMR employing n controllers for steering the trailers.

The advantage of selected approach is in the fact that it does not strongly depend on the hooking type (zero or nonzero).

Chapter 7

Conclusion

In this thesis, three main goals were outlined. The first one, to summarize and co-ordinate the problems of kinematic modelling of wheeled mobile robots. It has been addressed in Chapters 3 and 4. The second and third ones, to examine properties of multi-steered wheeled mobile robots and to develop a control scheme for the multi-steered wheeled mobile robots, have been addressed in Chapter 6.

There is a vast amount of literature on kinematic modelling of wheeled mobile robots. This work tries to cover in mathematical background as well as control-engineering viewpoint, however without claim to be neither exhaustive nor complete.

The focus in the modelling and control is on multi-steered wheeled mobile robots, represented by a bi-steered WMR (also called two-steered, bi-steered or four-wheel-steered car). This kind of carriages can be found in automotive industry to improve maneuverability at low speeds (reduced turning radius in parking maneuvers) as well as stability at high speeds. Another area is construction of autonomous agricultural vehicles and various transportation and building machines.

The control scheme developed for control of bi-steered wheeled mobile robot uses dynamic feedback linearization controller together with angular velocity controller with external velocity input (decomposed controller). This approach is suitable for the main control tasks—set-point stabilization and trajectory tracking. For the latter one, it is necessary to ensure generation of feasible trajectory. The advantage of the presented control scheme is the ability to be generalized for multi-body multi-steered wheeled mobile robots (multi-steered tractor-trailer systems).

There are many control schemes developed for control of wheeled mobile robots. Namely for the set-point stabilization task, that is very interesting and challenging from the theoretical viewpoint. Many of the control schemes possess Lyapunov stability, that is sufficient, but can be considered weak—the transition trajectories between the initial and goal positions are oscillatory and/or erratic. The control scheme presented in this work shows exponential-like convergence, with the ability of adjustment of the quality of the transition process via controller constants.

Trajectory tracking is more practical task than set-point stabilization. Since it is often possible to locally linearize the equations of wheeled mobile robot around the reference trajectory, this task is not as widely solved as the previous one. The control scheme presented in this work uses global linearization, therefore it can be used also for large initial errors (unlike the local linearization techniques).

Bibliography

[1] Alexander, J. C., Maddocks, J. H. On the kinematics of wheeled mobile robots. *The International Journal of Robotics Research*, vol. 8, no. 5, p. 15–27, 1989.

[2] Altafini, C. The general n-trailer problem: Conversion into chained form. *Proceedings of the 37th IEEE Conference on Decision and Control*, p. 3129–3130, December 1998.

[3] Astolfi, A. On the stabilization of nonholonomic systems. In *Proceedings of the 33rd Conference on Decision and Control*, p. 3481–3486. IEEE, December 1994.

[4] Astolfi, A. Exponential stabilization of a wheeled mobile robot via discontinuous control. *Journal of dynamic systems, measurement, and control*, vol. 121, no. 1, p. 121–126, 1999.

[5] Astolfi, A. Discontinuous control of nonoholonomic systems. *Systems & Control Letters*, vol. 27, p. 37–45, 1996.

[6] Badreddin, E., Mansour, M. Fuzzy-tuned state feedback control of a non-holonomic mobile robot. *IFAC World Congress*, vol. V, p. 577–580, 1993.

[7] Baturone, I., Moreno-Velo, F. J., Sánchez-Solano, S., Ollero, A. Automatic design of fuzzy controllers for car-like autonomous robots. *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, p. 447–465, 2004.

[8] Bento, L. C., Urbano, N., Mendes, A., Parent, M. Path-tracking controller of a bi-steerable cybernetic car using fuzzy logic. In *Proceedings of ICAR 2003, The 11th International Conference on Advanced Robotics*, p. 1556–1561, Coimbra, Portugal, June 2003.

[9] Brockett, R. *Asymptotic stability and feedback stabilization.*, vol. 27 of *Progress in Mathematics*, chapter Differential Geometry and Geometrical Control Theory, p. 181–191 Birkhäuser, 1983.

[10] Brockett, R. W., Millman, R. S., Susmann, H. S. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, 1983.

[11] Brown, H. B., Xu, Y. A single-wheel, gyroscopically stabilized robot. *IEEE Robotics & Automation Magazine*, p. 39–44, September 1997.

[12] Campion, G., Bastin, G., D'Andréa-Novel, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, p. 47–62, February 1996.

[13] Cariou, C., Lenain, R., Thuilot, B., Berducat, M. Automatic guidance of a four-wheel steering vehicle. In *66th LandTechnik conference*, Hohenheim (Germany), September 2008.

[14] Craig, J. J. *Introduction to Robotics.* Addison Wesley, 1996.

[15] DeLuca, A., Oriolo, G., Samson, C. *Robot Motion Planning and Control.*, chapter Feedback Control of a Nonholonomic Car-like Robot, p. 170–253 Springer Verlag, 1998 Laumond, J.,P. (Editor).

[16] Diegel, O., Badve, A., Bright, G., Potgeiter, J., Tlale, S. Improved mecanum wheel design for omni-directional robots. In *Proc. of 2002 Australian Conference on Robotics and Automation*, p. 117–121, Auckland, November 2002.

[17] Dubins, L. E. On curves of minimal length with a constraint on average curvature and with prescribed initial, and terminal positions and tangents. *American Journal of Mathematics*, vol. 79, no. 3, p. 479–516, July 1957.

[18] Godhavn, J., Egel, O. A lyapunov approach to exponential stabilization of nonholonomic systems in power form. *IEEE Transactions on Automatic Control*, vol. 42, p. 1028–1032, 1997.

[19] Hermosillo, J., Sekhavat, S. Feedback control of a bi-steerable car using flatness application to trajectory tracking. In *Proceedings of the American Control Conference*, Denver, Colorado, June 2003. IEEE, IEEE.

[20] Isidori, A. *Nonlinear Control Systems.* Springer-Verlag, New York, 1995.

[21] Kumar, V. *The Theorems of Euler and Chasles.* University of Pennsylvania, 2002 teaching materials for course MEAM520, chap. 4, online at: `www.seas.upenn.edu/~meam520/notes02/EulerChasles4.pdf` (26.3.2009).

[22] Kureš, M. Differential geometry in optimal control. Faculty of Mechanical Engineering, Brno University of Techology, Seminar text, 2008.

[23] Lafferriere, G., Sussmann, H. Motion planning for controllable systems without drift. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, p. 1148–1153, Sacramento, California, April 1991.

[24] LaValle, M. S. *Planning Algorithms.* Cambridge University Press, 2006.

[25] Lee, S., Youm, Y., Chung, W. Control of car-like robots for posture stabilization. *Proceedings of the 1990 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, p. 1745–1750, October 1999.

[26] Luo, J., Tsiotras, P. Exponentially convergent control laws for nonholonomic systems in power form. In *System & Control Letters*, p. 87–95, 1998.

[27] M'Closkey, R. T. *Exponential Stabilization of Driftless Nonlinear Control Systems.* PhD thesis, California Institute of Technology, Pasadena, CA 91125, 1995.

[28] M'Closkey, R. T., Murray, R. M. Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Trans. on Automatic Control*, vol. 42, p. 614–628, 1997.

[29] Manocha, D., Demmel, J. Algorithms for intersecting parametric and algebraic curves I: Simple intersections. *ACM Transactions on Graphics*, vol. 13, no. 1, p. 73–100, January 1994.

[30] Mason, M. *Nonholonomic Constraints.* Lecture notes, Carnegie Mellon, 2005.

[31] Mauder, M. Robust tracking control of nonholonomic dynamic systems with application to the bi-steerable mobile robot. *Automatica*, p. 2588–2592, September 2008.

[32] Micaelli, A., Samson, C. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report 2097, Institut National de Recherche en Informatique et en Automatique (INRIA), November 1993.

[33] Muir, F. P., Neuman, P. C. Kinematic modeling of wheeled mobile robots. Technical report, Carnegie-Mellon University, June 1986.

[34] Murray, R. M., Sastry, S. S. Steering nonholonomic systems in chained form. vol. 2, p. 1121–1126, Brighton, England, December 1993.

[35] Murray, R. M., Sastry, S. S. Steering nonholonomic systems using sinusoids. *Proceedings of the 29th Conference on Decision and Control*, December 1990.

[36] Oriolo, G. *Mobile Robotics: Control Problems.* Materials for Scuola CIRA di Dottorato Antonio Ruberti, 2002.

[37] Oriolo, G., De Luca, A., Vendittelli, M. WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, p. 835–852, November 2002.

[38] Oriolo, G., Panzieri, S., Ulivi, G. An iterative learning controller for nonholonomie robots. *Proceedings of the 1996 IEEE Intemational Conference on Robotics and Automation*, no. 2676-2681, April 1996.

[39] Rajagopalan, R. A generic kinematic formulation for wheeled mobile robots. *Journal of Robotic Systems*, vol. 14, no. 2, p. 71–91, December 1998.

[40] Rehman, F. U., Ahmed, M. M. Steering control algorithm for a class of wheeled mobile robots. *IET Control Theory Applications*, vol. 1, no. 4, p. 915–924, 2007.

[41] Rektorys, K. et al. *Přehled užité matematiky (in Czech).* Prometheus, 2000.

[42] Sagami, T., Sampei, M., Nakaura, S. Discontinuous controller design of the chained form system via time state control form. *Proceedings of the 45th IEEE Conference on Decision and Control*, p. 3277–3282, December 2006.

[43] Siegwart, R., Nourbakhsh, I. *Introduction to Autonomous Mobile Robots.* The MIT Press, Massachusetts Institute of Technology Cambridge.

[44] Sordalen, O. J. Conversion of the kinematics of a car with n trailers into a chained form. *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, vol. 1, p. 382–387, May 1993.

[45] Spong, M. W.and Vidyasagar, M. *Robot Dynamics and Control.* Wiley, January 1989.

[46] Sugisaka, M., Hazry, D. Predictive proportional nonlinear control for stable-target tracking of a mobile robot: an experimental study. In *Artificial Life and Robotics*. Springer–Verlag, 2006.

[47] Tayebi, A., Tadjine, M., Rachid, A. Invariant manifold approach for the stabilization of nonholonomic chained systems: Application to a mobile robot. *Nonlinead Dynamics*, vol. 24, no. 2, p. 167–181, February 2001.

[48] Tilbury, D., Murray, R. M., Sastry, S. Trajectory generation for the n-trailer problem using goursat normal form. In *Proceedings of the 32nd Conference on Decision and Control*, p. 971–977, San Antonio, Texas, December 1993.

[49] Tilbury, D., Sordalen, O. J., Bushnell, L., Sastry, S. S. A multisteering trailer system: Conversion into chained form using dynamic feedback. In *IEEE Transactions on Robotics and Automation*, vol. 11, p. 807–818, December 1995.

[50] Triggs, B. Motion planning for nonholonomic vehicles: An introduction. Survey paper presented at Seminar on Computer Vision and Robotics, Newton Institute of Mathematical Sciences, Cambridge, England, avalilable online: `http://lear.inrialpes.fr/pubs/1993/Tri93a` (26.3.2009).

[51] Trkal, V. *Mechanika hmotných bodů a tuhého tělesa (in Czech).* Nakladatelství České akademie věd, Prague, 1956.

[52] Vlk, F. *Dynamika motorových vozidel (in Czech).* Nakladatelství VLK, 2000.

[53] Canudas de Wit, C., Sordalen, O. J. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, vol. 37, no. 11, p. 1791–1797, November 1992.

[54] Xu, G., Wang, D. Exponential stabilization of extended chained forms. *Proceedings of the American Control Conference*, p. 3259–3263, June 2000.

[55] Yun, X., Sarkar, N. Unified formulation of robotic systems with holonomic and nonholonomic constraints. *IEEE Transaction on Robotics and Automation*, vol. 14, no. 4, p. 640–650, August 1998.

[56] Čelikovský, S. *Nelineární systémy (in Czech).* Nakladatelství ČVUT Praha, 2005.

Typographic Conventions

k	scalar value
$\mathbf{a}$	vector
$\mathbf{A}$	matrix
$\square^T$	transpose of $\square$, where $\square$ is a vector of matrix

Used Abbreviations

w.r.t.	with respect to
iff	if and only if
DOF	degrees of freedom
ODE	ordinary differential equation
DFL	dynamic feedback linearization
X^+/X^-	positive/negative part of the X axis (Cartesian coordinates)
0_{XYZ}	coordinate system with origin 0 and axes X, Y and Z

Used Notation

$\mathbb{R}$	real numbers
$\mathcal{Q}$	state space
$\mathcal{U}$	control space

List of Appendices

Appendix A
Bi-steered WMR Model Derivation

The reference point is located at the vertical axle of the rear wheel (its pivot point). The physical distance between the wheels is denoted ℓ [m].

The vector of generalized coordinates describing the bi-steered WMR robot is $\mathbf{q} = (x, y, \vartheta_1, \vartheta_2, \alpha)^T$, where x and y [m] denote the position of the reference point in global Cartesian coordinate system, ϑ_1 and ϑ_2 [rad] denote orientation of the wheels w.r.t. the body of the robot and α [rad] is a orientation of the robot body w.r.t. the X^+ axis in global Cartesian coordinate system. The vector of generalized velocities (evolution of generalized coordinates over time) is $\dot{\mathbf{q}} = (\dot{x}, \dot{y}, \dot{\vartheta}_1, \dot{\vartheta}_2, \dot{\alpha})^T$. The angles are normalized to α, ϑ_1, $\vartheta_2 \in (-\frac{\pi}{2}, \frac{\pi}{2})$. This also ensures that the singularities in the subsequent equations are avoided.

There are three control inputs $\mathbf{u} = (v_1, \omega_1, \omega_2)$, where v_1 is the translational velocity of the front wheel, ω_1 is the steering velocity of the front wheel and ω_2 is the steering velocity of the rear wheel.

Derivation of a model means establishing transition equations for individual generalized coordinates (i.e., the individual generalized velocities). It could be helpful to refer to Fig. A.1.

The projection of v_1 to the local X^+ axis of the robot (the forward velocity of the robot) is

$$v_{1F} = v_F = v_1 \cos \vartheta_1 \tag{A.1}$$

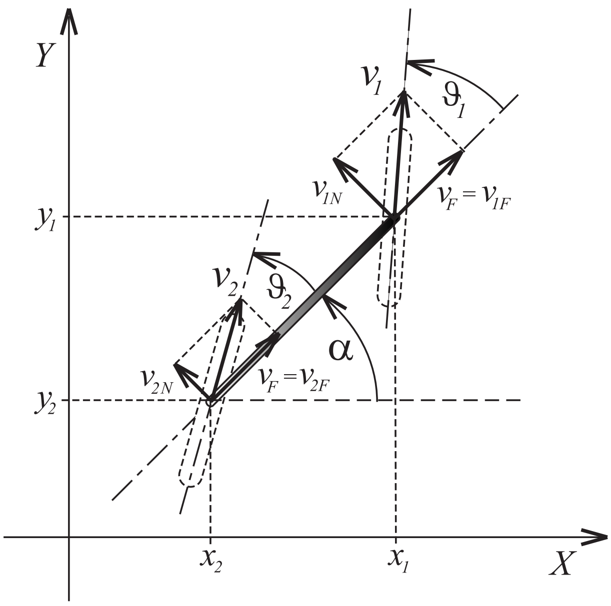


Figure A.1: *Bi-steered WMR model derivation—velocities* and the normal component (normal to the body of the robot) of v_1 is

$$v_{1N} = v_1 \sin \vartheta_1. \tag{A.2}$$

The velocity of the second (rear) wheel can be computed as

$$v_2 = \frac{v_F}{\cos \vartheta_2} = v_{1F} \sec \vartheta_2 \tag{A.3}$$

and its normal component (normal to the body of the robot) is

$$v_{2N} = v_2 \sin \vartheta_2 = v_{1F} \tan \vartheta_2. \tag{A.4}$$

Now it is possible to write the expressions for $\dot{x}$ and $\dot{y}$ as

$$\begin{aligned} \dot{x}_2 &= v_2 \cos(\vartheta_2 + \alpha) = v_{1F} \sec \vartheta_2 \cos(\vartheta_2 + \alpha) \\ \dot{y}_2 &= v_2 \sin(\vartheta_2 + \alpha) = v_{1F} \sec \vartheta_2 \sin(\vartheta_2 + \alpha) \end{aligned} \tag{A.5}$$

and further by substituting (A.1) into (A.5)

$$\begin{aligned} \dot{x}_2 &= v_1 \cos \vartheta_1 \sec \vartheta_2 \cos(\vartheta_2 + \alpha) \\ \dot{y}_2 &= v_1 \cos \vartheta_1 \sec \vartheta_2 \sin(\vartheta_2 + \alpha). \end{aligned} \tag{A.6}$$

Then it is necessary to express the evolution of α over time and it is

$$\dot{\alpha} = \frac{v_{1N} - v_{2N}}{\ell} \tag{A.7}$$

and by substituting equations (A.1), (A.2) and (A.4) into (A.7) the equation

$$\dot{\alpha} = \frac{v_1(\sin \vartheta_1 - \cos \vartheta_2 \tan \vartheta_2)}{\ell} \tag{A.8}$$

is obtained. The time-evolution of the two remaining generalized coordinates ϑ_1 and ϑ_2 is simply determined by the fact that these variables (angles of the wheels w.r.t. the body of the robot) are directly controlled by steering velocities of the wheels and thus

$$\begin{aligned} \dot{\vartheta}_1 &= \omega_1 \\ \dot{\vartheta}_2 &= \omega_2 \end{aligned}\ . \tag{A.9}$$

To conclude the model creation, the final set of equations can be formulated as

$$\begin{aligned} \dot{x}_2 &= v_1 \cos \vartheta_1 \sec \vartheta_2 \cos(\vartheta_2 + \alpha) \\ \dot{y}_2 &= v_1 \cos \vartheta_1 \sec \vartheta_2 \sin(\vartheta_2 + \alpha) \\ \dot{\vartheta}_1 &= \omega_1 \\ \vartheta_2 &= \omega_2 \\ \dot{\alpha} &= \frac{v_1(\sin \vartheta_1 - \cos \vartheta_1 \tan \vartheta_2)}{\ell}, \end{aligned} \tag{A.10}$$

which can be rewritten to

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos \vartheta_1 \sec \vartheta_2 \cos(\vartheta_2 + \alpha) \\ \cos \vartheta_1 \sec \vartheta_2 \sin(\vartheta_2 + \alpha) \\ 0 \\ 0 \\ (\sin \vartheta_1 - \cos \vartheta_1 \tan \vartheta_2)/\ell \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \omega_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_2. \tag{A.11}$$

This finalizes the creation of the kinematic model of a bi-steered WMR.

Appendix B

Frobenius' Theorem for Bi-steered WMR

The Frobenius' theorem determines if a system (or, more precisely, its distribution) is completely integrable (holonomic) or nonintegrable (nonholonomic). This means that using this theorem, it is possible to distinguish holonomic and nonholonomic systems.

If the system is smooth and its distribution is nonsingular, then the Frobenius' theorem states

A distribution is integrable if and only if it is involutive.

For more information please refer to Sect. 3.8.4.

The Frobenius' theorem will be used to determine holonomic or nonholonomic character of the bi-steered WMR.

The kinematic model of the bi-steered WMR is

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \cos\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2+\alpha) \\ \cos\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2+\alpha) \\ 0 \\ 0 \\ (\sin\vartheta_1 - \cos\vartheta_1 \tan\vartheta_2)/\ell \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \omega_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \omega_2. \tag{B.1}$$

The constituent vector fields (system vector fields) are

$$\mathbf{h}_1 = \begin{pmatrix} \cos\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2+\alpha) \\ \cos\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2+\alpha) \\ 0 \\ 0 \\ (\sin\vartheta_1 - \cos\vartheta_1 \tan\vartheta_2)/\ell \end{pmatrix}; \ \mathbf{h}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}; \ \mathbf{h}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \tag{B.2}$$

All possible Lie brackets needed for checking the integrability are $[\mathbf{h}_1, \mathbf{h}_2]$, $[\mathbf{h}_2, \mathbf{h}_3]$, $[\mathbf{h}_1, \mathbf{h}_3]$. To remind, it is necessary only to check if the matrix assembled from the system vector fields $\mathbf{H}(\mathbf{q})$ and the one extended by Lie brackets $\mathbf{H}_{ext}(\mathbf{q})$ has the same dimension as the original system matrix $\mathbf{H}(\mathbf{q})$. If it has, the system is integrable (holonomic), in opposite case it is nonholonomic.

This testing can be done in an iterative manner, by successively computing the Lie brackets. If for any of them $rank(\mathbf{H_{ext}}(\mathbf{q})) \neq rank(\mathbf{H}(\mathbf{q}))$, then the system in nonholonomic and the procedure can be stopped.

The computation can be a tedious process, as demonstrated further. The notation $\mathbf{h}_i(j)$ denotes the j^{th} component of i^{th} vector field $\mathbf{h}$. The first component of the first Lie bracket is

$$[\mathbf{h}_1,\mathbf{h}_2]_1 = \mathbf{h}_{1(1)}\frac{\partial \mathbf{h}_{2(1)}}{\partial x_2} - \mathbf{h}_{2(1)}\frac{\partial \mathbf{h}_{1(1)}}{\partial x_2} + \mathbf{h}_{1(2)}\frac{\partial \mathbf{h}_{2(1)}}{\partial y_2} - \mathbf{h}_{2(2)}\frac{\partial \mathbf{h}_{1(1)}}{\partial y_2} + \tag{B.3}$$

$$+\,\mathbf{h}_{1(3)}\frac{\partial \mathbf{h}_{2(1)}}{\partial \vartheta_1} - \mathbf{h}_{2(3)}\frac{\partial \mathbf{h}_{1(1)}}{\partial \vartheta_1} + \mathbf{h}_{1(4)}\frac{\partial \mathbf{h}_{2(1)}}{\partial \vartheta_2} - \mathbf{h}_{2(4)}\frac{\partial \mathbf{h}_{1(1)}}{\partial \vartheta_2} + \tag{B.4}$$

$$+\,\mathbf{h}_{1(5)}\frac{\partial \mathbf{h}_{2(1)}}{\partial \alpha} - \mathbf{h}_{2(5)}\frac{\partial \mathbf{h}_{1(1)}}{\partial \alpha} \tag{B.5}$$

and the only non-zero element of this computation is

$$-\mathbf{h}_{2(3)}\frac{\partial \mathbf{h}_{1(1)}}{\partial \vartheta_1} = (-1)\frac{\partial \mathbf{h}_{1(1)}}{\partial \vartheta_1} = \sin\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2+\alpha). \tag{B.6}$$

Similarly, for the second component of the first Lie bracket (the change is that the second components of $\mathbf{h_1}$ and $\mathbf{h_2}$ are used in partial derivatives)

$$[\mathbf{h}_1,\mathbf{h}_2]_2 = \mathbf{h}_{1(1)}\frac{\partial \mathbf{h}_{2(2)}}{\partial x_2} - \mathbf{h}_{2(1)}\frac{\partial \mathbf{h}_{1(2)}}{\partial x_2} + \mathbf{h}_{1(2)}\frac{\partial \mathbf{h}_{2(2)}}{\partial y_2} - \mathbf{h}_{2(2)}\frac{\partial \mathbf{h}_{1(2)}}{\partial y_2} \tag{B.7}$$

$$+\,\mathbf{h}_{1(3)}\frac{\partial \mathbf{h}_{\mathbf{2}(2)}}{\partial \vartheta_1} - \mathbf{h}_{2(3)}\frac{\partial \mathbf{h}_{1(2)}}{\partial \vartheta_1} + \mathbf{h}_{1(4)}\frac{\partial \mathbf{h}_{2(2)}}{\partial \vartheta_2} - \mathbf{h}_{2(4)}\frac{\partial \mathbf{h}_{1(2)}}{\partial \vartheta_2} + \tag{B.8}$$

$$+\,\mathbf{h}_{1(5)}\frac{\partial \mathbf{h}_{2(2)}}{\partial \alpha} - \mathbf{h}_{2(5)}\frac{\partial \mathbf{h}_{1(2)}}{\partial \alpha} \tag{B.9}$$

and the only nonzero element is

$$-\mathbf{h}_{2(3)}\frac{\partial \mathbf{h}_{1(2)}}{\partial \vartheta_1} = (-1)\frac{\partial \mathbf{h}_{1(2)}}{\partial \vartheta_1} = \sin\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2+\alpha). \tag{B.10}$$

The rest of the components are (computed in a similar way as the above ones)

$$\begin{aligned}
&[\mathbf{h}_1,\mathbf{h}_2]_3 = 0\\
&[\mathbf{h}_1,\mathbf{h}_2]_4 = 0\\
&[\mathbf{h}_1,\mathbf{h}_2]_5 = -\left(\cos\vartheta_1 + \sin\vartheta_1\tan\vartheta_2\right)/\ell.
\end{aligned} \tag{B.11}$$

The final resulting vector is

$$[\mathbf{h}_1,\mathbf{h}_2] = \begin{pmatrix} \sin\vartheta_1\sec\vartheta_2\cos(\vartheta_2+\alpha)\\ \sin\vartheta_1\sec\vartheta_2\sin(\vartheta_2+\alpha)\\ 0\\ 0\\ -\left(\cos\vartheta_1+\sin\vartheta_1\tan\vartheta_2\right)/\ell \end{pmatrix}. \tag{B.12}$$

The extended system matrix can be assembled as

$$\mathbf{H}_{ext}(\mathbf{q}) = \begin{pmatrix} \cos\vartheta_1\sec\vartheta_2\cos(\vartheta_1+\alpha) & 0 & 0 & \sin\vartheta_1\sec\vartheta_2\cos(\vartheta_2+\alpha)\\ \cos\vartheta_1\sec\vartheta_2\sin(\vartheta_1+\alpha) & 0 & 0 & \sin\vartheta_1\sec\vartheta_2\sin(\vartheta_2+\alpha)\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ (\sin\vartheta_1-\cos\vartheta_1\tan\vartheta_2)/\ell & 0 & 0 & -(\cos\vartheta_1+\sin\vartheta_1\tan\vartheta_2)/\ell \end{pmatrix} \tag{B.13}$$

and its rank is $rank(\mathbf{H}_{ext}(\mathbf{q})) = 4$ and therefore it is possible to conclude that the *bi-steered WMR is nonholonomic.*

Appendix C

P. Hall Basis for Bi-steered WMR (STLC)

A P. Hall basis can be used to determine the controllability of a driftless control-affine system using the rank of a Lie algebra associated with it (the *Lie algebra rank condition (LARC)*).

The controllability of the bi-steered WMR will be checked by generation of the P. Hall basis and then LARC.

The bi-steered WMR model consists of three system vector fields. Therefore, the Lie brackets that can be generated are

$$\begin{array}{llll} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \\ [\mathbf{h}_1, \mathbf{h}_2] & [\mathbf{h}_2, \mathbf{h}_3] & [\mathbf{h}_1, \mathbf{h}_3] & \\ [\mathbf{h}_1, [\mathbf{h}_1, \mathbf{h}_2]] & [\mathbf{h}_1, [\mathbf{h}_1, \mathbf{h}_3]] & [\mathbf{h}_2, [\mathbf{h}_1, \mathbf{h}_2]] & [\mathbf{h}_2, [\mathbf{h}_1, \mathbf{h}_3]] \\ [\mathbf{h}_2, [\mathbf{h}_2, \mathbf{h}_3]] & [\mathbf{h}_3, [\mathbf{h}_1, \mathbf{h}_2]] & [\mathbf{h}_3, [\mathbf{h}_1, \mathbf{h}_3]] & [\mathbf{h}_3, [\mathbf{h}_2, \mathbf{h}_3]]. \end{array} \tag{C.1}$$

The important question when creating a P. Hall basis is when the process should be stopped. The answer is based on the fact that the $\dim \mathcal{L}_{\mathbf{q}}(\Delta) \subseteq T_{\mathbf{q}}(Q)$, and thus the dimension of the Lie algebra cannot be greater than n, where n is the dimension of the tangent space at $\mathbf{q}$. The interpretation is that the number of possible directions of motion (independent velocity vectors) from a given state $\mathbf{q}$ cannot be greater than the dimension of the tangent space at that point. Therefore, when n linearly independent vector fields are discovered, the basis is complete (in fact it is not possible to discover more independent vector fields).

From Appendix B it is known that

$$[\mathbf{h}_1, \mathbf{h}_2] = \begin{pmatrix} \sin\vartheta_1 \sec\vartheta_2 \cos(\vartheta_2 + \alpha) \\ \sin\vartheta_1 \sec\vartheta_2 \sin(\vartheta_2 + \alpha) \\ 0 \\ 0 \\ -\left(\cos\vartheta_1 + \sin\vartheta_1 \tan\vartheta_2\right)/\ell \end{pmatrix}. \tag{C.2}$$

Similarly $[\mathbf{h}_2, \mathbf{h}_3]$ is determined

$$[\mathbf{h}_1, \mathbf{h}_2] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{C.3}$$

and also $[\mathbf{h}_1, \mathbf{h}_3]$,

$$[\mathbf{h}_1, \mathbf{h}_3] = \begin{pmatrix} \cos\vartheta_1 \sec\vartheta_2 \sin(\alpha+\vartheta_2) - \cos\vartheta_1 \cos(\alpha+\vartheta_2) \sec\vartheta_2 \tan\vartheta_2 \\ -\cos\vartheta_1 \sec\vartheta_2 \cos(\alpha+\vartheta_2) - \cos\vartheta_1 \sin(\alpha+\vartheta_2) \sec\vartheta_2 \tan\vartheta_2 \\ 0 \\ 0 \\ \cos\vartheta_1 \sec\vartheta_2 / \ell \end{pmatrix}. \tag{C.4}$$

Now it is possible to construct

$$\mathbf{H}' = \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & [\mathbf{h}_1, \mathbf{h}_2] & [\mathbf{h}_2, \mathbf{h}_3] & [\mathbf{h}_1, \mathbf{h}_3] \end{pmatrix}, \tag{C.5}$$

where $[\mathbf{h}_2, \mathbf{h}_3] = \mathbf{0}$. As $\mathrm{rank}(\mathbf{H}') = 5$, the creation process of the P. Hall basis is finished.

Recalling that $Q \in \mathbb{R}^n, n = 5$ and considering the fact that $\mathrm{rank}(\mathbf{H}') = 5$ it immediately follows from Chow-Rashevskii (see Sect. 3.9) theorem that the bi-steered WMR is nonholonomic and small-time locally controllable (STLC).

Appendix D

Conditions for DFL Controller

The conditions for dynamic feedback linearization controller will be presented in this section. The results are consistent with the ones obtained in [37], however the procedure of their derivation is included herein. The process is described for unicycle DFL controller and its modification for the bi-steered WMR is in Sect. 6.4.

Recapitulate the equations of the unicycle DFL controller

$$\begin{aligned} \dot{\xi} &= u_1 \cos\theta + u_2 \sin\theta \\ v &= \xi \\ \omega &= \frac{u_2 \cos\theta - u_1 \sin\theta}{\xi}, \end{aligned} \tag{D.1}$$

where

$$\begin{aligned} u_1 &= -k_{p_1} x - k_{d_1} \dot{x} \\ u_2 &= -k_{p_2} y - k_{d_2} \dot{y} \end{aligned} \tag{D.2}$$

and

$$\begin{aligned} \ddot{x} &= \ddot{z}_1 = u_1 \\ \ddot{y} &= \ddot{z}_2 = u_2. \end{aligned} \tag{D.3}$$

There are two requirements in order to achieve appropriate behaviour of the controller, i.e., the unicycle never stops during the transition and the steering velocity ω does not converge to infinity:

Requirement 1: $\xi(t^*) \neq 0$ for generic $t^* > 0$,

Requirement 2: θ converges to zero,

Requirement 3: $\omega(t) \nrightarrow \infty$ although $\xi \to 0$.

ad 1. This requirement means to check if there exists a time instant in which $\xi(t^*) = \sqrt{\dot{x}^2(t^*) + \dot{y}^2(t^*)}$, i.e., $\dot{x}(t^*) = \dot{y}(t^*) = 0$.

The first step is to determine expressions for $\dot{x}(t)$ and $\dot{y}(t)$. From (D.2) and (D.3) follows

$$\ddot{x} = -k_{p_1} x - k_{d_1} \dot{x} \quad \Rightarrow \quad \ddot{x} + k_{d_1} \dot{x} + k_{p_1} x = 0 \tag{D.4}$$

and

$$\ddot{y} = -k_{p_2} y - k_{d_2} \dot{y} \quad \Rightarrow \quad \ddot{y} + k_{d_2} \dot{y} + k_{p_2} y = 0 \tag{D.5}$$

which are 2^{nd} order linear homogenous ODE. Solutions of their characteristic equations (in r) are

$$\begin{aligned} r^2 + k_{d_i} r + k_{p_i} &= 0 \\ &\Downarrow \\ r_{j,k} &= \frac{-k_{d_i} \pm \sqrt{k_{d_i}^2 - 4k_{p_i}}}{2}, \end{aligned} \tag{D.6}$$

where $i = 1, 2$ and $j = 1, k = 2$ for $i = 1$ and $j = 3, k = 4$ for $i = 2$. The roots $r_{j,k} \in \mathbb{R}$ iff discriminant $D_i = \sqrt{k_{d_i}^2 - 4k_{p_i}} > 0$. The condition is fulfilled iff

$$k_{d_i}^2 - 4k_{p_i} > 0. \tag{D.7}$$

The general solutions of the original ODEs (these are the only solutions as they are homogenous ones) are

$$\begin{aligned} x(t) &= c_1 e^{r_1 t} + c_2 e^{r_2 t} \\ y(t) &= c_3 e^{r_3 t} + c_4 e^{r_4 t} \end{aligned} \tag{D.8}$$

where

$$\begin{aligned} r_1 = \lambda_{11} &= \frac{-k_{d_1} + \sqrt{(k_{d_1}^2 - 4k_{p_1})}}{2} \\ r_2 = \lambda_{12} &= \frac{-k_{d_1} - \sqrt{(k_{d_1}^2 - 4k_{p_1})}}{2} \\ r_3 = \lambda_{21} &= \frac{-k_{d_2} + \sqrt{(k_{d_2}^2 - 4k_{p_2})}}{2} \\ r_4 = \lambda_{22} &= \frac{-k_{d_2} - \sqrt{(k_{d_2}^2 - 4k_{p_2})}}{2} \end{aligned} \tag{D.9}$$

and so

$$\begin{aligned} x(t) &= c_1 e^{\lambda_{11} t} + c_2 e^{\lambda_{12} t} \\ y(t) &= c_3 e^{\lambda_{21} t} + c_4 e^{\lambda_{22} t} \end{aligned} . \tag{D.10}$$

Differentiation of (D.10) with respect to time yields

$$\begin{aligned} \dot{x}(t) &= c_1 \lambda_{11} e^{\lambda_{11} t} + c_2 \lambda_{12} e^{\lambda_{12} t} \\ \dot{y}(t) &= c_3 \lambda_{21} e^{\lambda_{21} t} + c_4 \lambda_{22} e^{\lambda_{22} t} \end{aligned} \tag{D.11}$$

Now the unknown coefficients c_1, c_2, c_3, c_4 should be determined. Setting $t = 0$ in (D.10) and (D.11) results in

$$\begin{aligned} c_1 + c_2 &= x_0 \\ c_1 \lambda_{11} + c_2 \lambda_{12} &= \dot{x}_0 \end{aligned} \tag{D.12}$$

and

$$\begin{aligned} c_3 + c_4 &= y_0 \\ c_3 \lambda_{21} + c_4 \lambda_{22} &= \dot{y}_0 \end{aligned} \tag{D.13}$$

where (x_0,y_0) represent initial coordinates and $\dot{x}_0$ and $\dot{y}_0$ initial velocities in the direction of respective axes.

Solving the sets of equations (D.12) and (D.13) results in

$$
\begin{aligned}
c_1 &= \frac{x_0\lambda_{12} - \dot{x}_0}{\lambda_{12} - \lambda_{11}} \\
c_2 &= \frac{\dot{x}_0 - x_0\lambda_{11}}{\lambda_{12} - \lambda_{11}}
\end{aligned} \tag{D.14}
$$

and

$$
\begin{aligned}
c_3 &= \frac{y_0\lambda_{22} - \dot{y}_0}{\lambda_{22} - \lambda_{21}} \\
c_4 &= \frac{\dot{y}_0 - y_0\lambda_{21}}{\lambda_{22} - \lambda_{21}}
\end{aligned} \tag{D.15}
$$

ad 2. The required final state is $(x, y, \theta) = (0, 0, 0)$. Because the controlled output is (x, y), it is necessary to ensure the convergence of θ to zero indirectly, by appropriate setting of the convergence rate for x and y controller (D.1)—(D.3). This can be done by exploiting the fact that

$$
\tan\theta = \frac{\dot{y}}{\dot{x}}, \tag{D.16}
$$

and thus required faster convergence rate of $\dot{y}$ than of $\dot{x}$.

To achieve this, exponents (eigenvalues) $\lambda_{ij} < 0$ need to be ordered as $\lambda_{22} < \lambda_{21} < \lambda_{12} < \lambda_{11}$.

According to (D.9), $\lambda_{22} < \lambda_{21}$ and $\lambda_{12} < \lambda_{11}$, therefore it rests to guarantee that $\lambda_{21} < \lambda_{12}$ by

$$
k_{d_2} - k_{d_1} > \sqrt{k_{d_1}^2 - 4k_{p_1}} + \sqrt{k_{d_2}^2 - 4k_{p_2}}, \tag{D.17}
$$

that follows from the discussed condition $\lambda_{21} - \lambda_{12} < 0$ and (D.9).

ad 3. To ensure that ω also converges to zero the sufficient and necessary condition is that the rate of convergence of its numerator is faster than its denominator. From (D.16) and $\omega = \dot{\theta}$ it can be obtained

$$
\omega = \left(\arctan\frac{\dot{y}}{\dot{x}}\right)' = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 - \dot{y}^2} \tag{D.18}
$$

and substituting (D.11) together with

$$
\begin{aligned}
\ddot{x}(t) &= c_1\lambda_{11}^2 e^{\lambda_{11}t} + c_2\lambda_{12}^2 e^{\lambda_{12}t} \\
\ddot{y}(t) &= c_3\lambda_{21}^2 e^{\lambda_{21}t} + c_4\lambda_{22}^2 e^{\lambda_{22}t}
\end{aligned} \tag{D.19}
$$

it is obtained an equation with the powers of e in numerator

$$
(\lambda_{11} + \lambda_{21})t,\ (\lambda_{12} + \lambda_{21})t,\ (\lambda_{11} + \lambda_{22})t,\ (\lambda_{12} + \lambda_{22})t
$$

and denominator

$$
2\lambda_{11}t,\ 2\lambda_{12}t,\ 2\lambda_{21}t,\ 2\lambda_{22}t,\ (\lambda_{11} + \lambda_{12})t,\ (\lambda_{21} + \lambda_{22})t.
$$

The denominator converges asymptotically as $e^{2\lambda_{11}}t$, whereas the numerator convergence is faster, due to λ_{ij} ordering. Therefore ω converges asymptotically to zero for $t \to \infty$.

Index