Chapter 11
Automated Kinematics Reasoning for Wheeled Mobile Robots

Michael Hofbaur, Christoph Gruber, and Mathias Brandstötter

Abstract Control schemes for wheeled mobile robots typically assume a specific mobility capability of a drive and implicitly use the drive's kinematics within its control procedures. This makes it difficult to deal with faults in the drive and to handle drives with diverse geometry and functionality that might even change during operation of a robot. As a consequence, we propose a model-based control scheme that builds upon an automated analysis of a robotic drive and on an on-line deduction of the drive's kinematics. We achieve this functionality through (1) the introduction of steering-angle independent, generalized variants of the rolling and sliding constraints for wheeled mobile robots and (2) the corresponding reformulation of kinematic analysis. This leads to a computationally efficient algorithm that deduces the (inverse) kinematics of a drive for its mode of operation or failure. Fault tolerant and robust behavior, however, is only one aspect of our control architecture. On-line kinematics analysis enables us to easily handle robots that change in geometry or functionality such as self-configuring modular robot systems and teams of cooperative robots.

11.1 Introduction

Wheeled mobile robots utilize a variety of application specific drives with different geometries and functionalities such as differential drives, drives with steered standard wheels or omni-directional drives with mecanum wheels, for example. This implies drive-specific kinematic capabilities. A controller uses the drive's kinematics to translate the requested robot motion into the appropriate actuation of the drive's wheels in terms of their rotational velocity and, whenever applicable, their steering angles.

M. Hofbaur (✉) • C. Gruber • M. Brandstötter
UMIT, Institute of Automation and Control Engineering, E. Wallnöfer Center 1, Hall in Tirol 6060, Austria
e-mail: michael.hofbaur@umit.at

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*, DOI 10.1007/978-3-7091-1289-2_11, © Springer-Verlag Wien 2013

Once in operation, the control law and also higher-level control layers, such as the path-planner, will use the drive's (inverse) kinematics implicitly through the implemented control algorithms. It is thus often impossible for such controllers to adapt their control laws once the kinematics of the drive changes significantly. Such a situation can occur in the case of a fault in the drive, for example, an impaired steering actuator. Another situation would be due to a geometry or functional adaption of the drive, for example, a drive that widens its base to increase stability or decreases its wheel distance to move through confined spaces [4]. Another realistic scenarios with implications to a drive's kinematics could be the adaption of the robot's number of wheels to deal with varying payload situations [8] or multi-robot transport [2]. Of course, one can solve such operational situations through specifically designed controllers that can account for pre-defined operational modes. However, it would be desirable to handle such situations in a more general and generic way.

To overcome these difficulties we propose a *model-programmed* control scheme that integrates control-design and control-execution within one intelligent solver for control. More precisely, the controller performs kinematic analysis for the drive on the basis of a *drive-model* at run-time (e.g. whenever the drive changes in geometry and functionality). As a side effect, we obtain a controller that is always aware of the drive's kinematic capabilities and that can perform re-configurations in the drive to automatically compensate for or recover from fault situations and notify a higher-level controller to re-plan the desired robot path according to the currently valid kinematic capabilities of the drive.

The key component of our control architecture is a *kinematics reasoning engine* that operates within the stringent time-constraints of real-time drive control. It thus implements the kinematic analysis and control design procedures as computationally efficient algorithm.

11.1.1 Background and Related Research

Kinematics in robot drives is a well-studied field. A detailed analysis of the underlying mechanism can be found in text-books (e.g. [10]) and various publications (e.g. [1, 3]). In the latter reference, Campion et al. provide an unified framework that classifies robot drives according to their degree of mobility and degree of steerability and provides general procedures for the analysis of the robot-kinematics. However, almost all papers describe analysis and design procedures that are applied at the design stage of a robot and thus, assume functionally reasonable drives with dedicated topology. We intend to extend this scope to handle faults in the drive and drives with adaptive geometry/functionality. Faults, in particular, change the kinematics significantly and it is easily possible that one ends up with a degenerated robot with impaired mobility. Nevertheless it is important for an autonomous robot to cope with such situations as well. One can also find applications of robots with varying geometry or dynamically adaptive

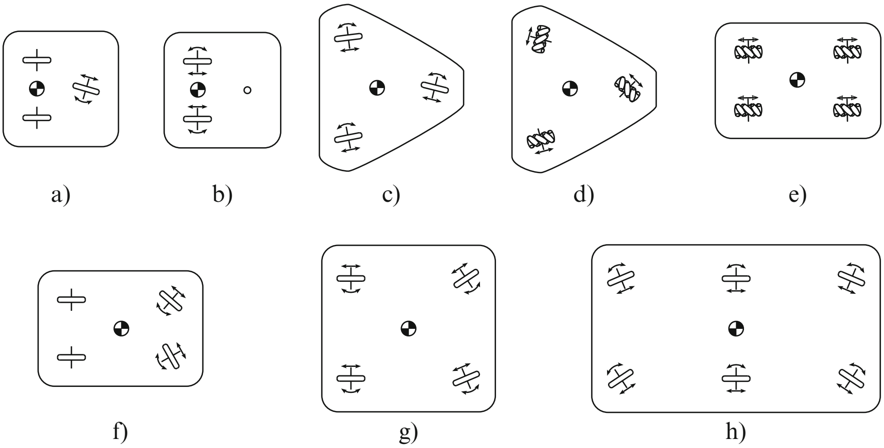

Fig. 11.1 Wheeled mobile robot examples

number of wheels (e.g. see [4, 8]). However, these systems typically apply specialized drive controllers that handle few operational conditions. Our approach goes beyond this scope. We intend to provide a generic model-programmed control scheme in the spirit of [11] that can be applied to a wide spectrum of robot drives and thus, automatically include special situations such as anticipated and non-anticipated faults, re-configurable drives and multi-robot control scenarios.

11.2 Preliminaries

Wheeled mobile robots (WMR) provide mobility and load capabilities through various wheel arrangements as shown in Fig. 11.1. Many approaches, documented in literature, e.g. [3, 7], classify robots in terms of the wheels (steered-, unsteered-, mecanum-wheels) and their geometric arrangements and derive the kinematics accordingly. The basis for the robot's kinematics are the so called *rolling* and *sliding* conditions for the robot's wheels (e.g. see [3]) that we summarize in the following:

Focusing on the movement of the robot and neglecting the robot's pose within a global reference frame we can directly consider the robot with respect to a robot-body fixed reference frame $\Sigma_R : \{O_R; x_R, y_R\}$. The robot's velocity with respect to an inertial frame expressed in the this robot-body frame Σ_R is a vector

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \tag{11.1}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal velocities and $\dot{\theta}$ denotes the angular velocity.

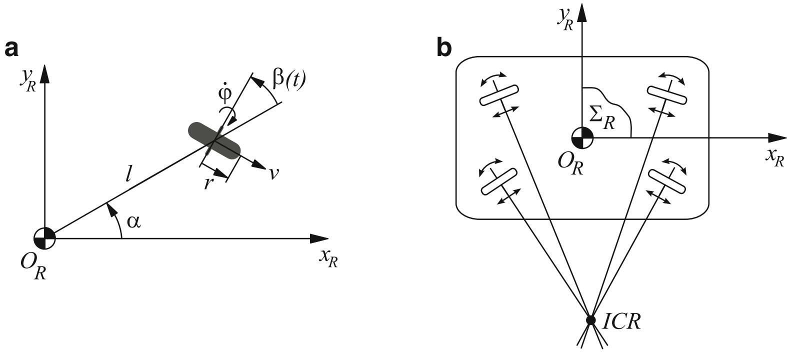

Fig. 11.2 Robot wheel geometry and body-fixed reference frame Σ_R

Let us define the drive through its model that encodes the geometric alignment of the robot's wheels. We will use $\mathcal{W}$ to denote the set of n wheels in a drive ($n = |\mathcal{W}|$) and specify the wheel alignment in terms of the wheel contact point expressed by the polar coordinates l_i and α_i relative to Σ_R and the wheel radius r_i as shown in Fig. 11.2a for a wheel. The kinematics of a drive defines the possible movements for a drive and relates $\dot{\xi}$ to the rotational velocities $\dot{\varphi}_i$ and steering angles β_i of the wheels $W_i \in \mathcal{W}$. The basis for a drive's kinematics is given through the *rolling* and *sliding* constraints. We introduce them for a centered orientable (steered) standard wheel W_i. The rolling constraint

$$r_i \, \dot{\varphi}_i = [\sin(\alpha_i + \beta_i) - \cos(\alpha_i + \beta_i) \quad -l_i \; \cos(\beta_i)] \, \dot{\xi} \tag{11.2}$$

$$= \mathbf{j}_i^T(\beta_i)\dot{\xi} \tag{11.3}$$

encodes the velocity v *perpendicular to the wheel axis* and relates the rotational velocity $\dot{\varphi}_i$ of the wheel with the robot's velocity $\dot{\xi}$. The sliding constraint

$$0 = [\cos(\alpha_i + \beta_i) \quad \sin(\alpha_i + \beta_i) \quad l_i \sin(\beta_i)] \, \dot{\xi} \tag{11.4}$$

$$= \mathbf{c}_i^T(\beta_i)\dot{\xi} \tag{11.5}$$

assures that there is no motion *perpendicular the wheel plane*. The rolling and sliding constraints of fixed oriented standard wheels are conceptually the same, except that β_i is fixed and not time-dependent.

To derive the rolling and sliding constraints for the complete drive, one combines the individual row vectors $\mathbf{j}_i^T$ and $\mathbf{c}_i^T$ to form the matrices

$$\mathbf{J}_1(\beta) = \begin{bmatrix} \mathbf{j}_1^T(\beta_1) \\ \vdots \\ \mathbf{j}_n^T(\beta_n) \end{bmatrix}, \; \mathbf{C}_1(\beta) = \begin{bmatrix} \mathbf{c}_1^T(\beta_1) \\ \vdots \\ \mathbf{c}_n^T(\beta_n) \end{bmatrix}. \tag{11.6}$$

This leads to the overall rolling and sliding constraints for the robot

$$\mathbf{J}_1(\beta)\dot{\xi} = \mathbf{J}_2\dot{\varphi} \tag{11.7}$$

$$\mathbf{C}_1(\beta)\dot{\xi} = \mathbf{0}, \tag{11.8}$$

where $\mathbf{J}_2 = \mathrm{diag}([r_i])$ combines the wheel radii r_i into a constant diagonal matrix.

Of course, we allow a mixture of wheels with n_c centered orientable (steerable) standard wheels and n_f fixed oriented standard wheels[1] so that $n = n_c + n_f$. In that sense, we can group wheels accordingly and obtain

$$\mathbf{J}_1(\beta_s) = \begin{bmatrix} \mathbf{J}_{1c}(\beta_s) \\ \mathbf{J}_{1f} \end{bmatrix}, \mathbf{C}_1(\beta_s) = \begin{bmatrix} \mathbf{C}_{1c}(\beta_s) \\ \mathbf{C}_{1f} \end{bmatrix}. \tag{11.9}$$

The vector $\beta_s = [\beta_1, \ldots, \beta_{n_c}]^T$ combines all (time varying) steering angle parameters $\beta_i(t)$ for the n_c centered orientable wheels.

The sliding constraint $\mathbf{C}_1(\beta_s)\dot{\xi} = \mathbf{0}$ is satisfied if $\dot{\xi} \in \ker(\mathbf{C}_1(\beta_s))$. If we consider two centered orientable standard wheels, then $1 \le \mathrm{rank}\,(\mathbf{C}_1(\beta_s)) \le 2$. If we add more wheels, then the rank of $\mathbf{C}_1(\beta_s)$ can increase up to 3, meaning that no motion might be possible! To avoid that, one has to choose the steering angles β_s so that $\mathrm{rank}(\mathbf{C}_1(\beta_s)) \le 2$. This demand on the rank of $\mathbf{C}_1(\beta_s)$ corresponds to the existence of a common intersection of all wheel axles at a finite distance from the robot center for $\mathrm{rank}(\mathbf{C}_1(\beta_s)) = 2$ or at infinity for $\mathrm{rank}(\mathbf{C}_1(\beta_s)) = 1$. In a perfectly controlled and actuated robot, this intersection defines the *instantaneous center of rotation (ICR)* of the robot-body motion (see Fig. 11.2b).

It is the task of the robot's low-level *kinematics* controller to maintain the steering angles and rotational speeds synchronously such that all wheel axles intersect and the sliding and rolling constraints define the appropriate ICR for the desired motion $\dot{\xi}$. However, this will only work if the path-planner or any other high-level control authority restricts itself to motion set-points $\dot{\xi}$ that are possible according to the drive's kinematics.

[1] It is straight forward to include mecanum wheels as well. The associated details are given in the Appendix.

11.3 Kinematics Reasoning

Our goal is to provide a generally applicable mechanism for the low level controller so that it can handle a large variety of drives with different geometric wheel alignments and wheel types. I.e. we are seeking for a computationally efficient algorithm that derives the kinematics or better the inverse kinematics from a model of the drive.

Definition 11.1. We define the model of a robot drive in terms of the tuple $\langle \Sigma_R, \mathcal{W} \rangle$ that specifies the robot-body fixed reference frame and the set of wheel models $\mathcal{W} = \{W_1, \ldots, W_n\}$. Each wheel model defines a tuple $W_i = \langle T_i, G_i, M_i \rangle$ where T_i details the wheel type {standard-fixed, standard-steered, mecanum}, G_i specifies the wheel's geometry within Σ_R in terms of the polar coordinates $\{l_i, \alpha_i, \beta_i\}$, the wheel radius r_i and γ_i for mecanum wheels. Finally, M_i captures the set of operational and fault modes for a specific wheel, e.g. {actuated, non-actuated, blocked-steering, blocked-rotation, ...}.

We allow the model to evolve over time. This includes a dynamically changing set of wheels and adaptations in wheel geometry. Furthermore, in our control architecture we use an additional functional unit that estimates the mode of operation or failure for each wheel and its associated actuators. Mode-estimation and fault detection, however goes beyond the scope of this paper and we refer the interested reader to [9].

Given the drive model and a mode specification that captures the operational/ fault condition for a drive at a specific time point, we are interested to analyze a drive beyond the classification in terms of its degree of mobility and degree of steerability. In fact, it is our goal to explicitly compute the *space B of admissible and controllable motions* that fully describes the mobility capabilities and drive characteristics of a wheeled mobile robot.

We use this explicit knowledge in two ways. Firstly, we provide this information to higher-level control so that a path-planner automatically takes the kinematic constraints of a drive into account. Secondly, we enable our low-level controller to check, whether a drive command from the path-planner is admissible for the drive in its current mode of operation/failure. This can be done by evaluating the condition:

$$\dot{\xi} \in B \, . \tag{11.10}$$

A violation of (11.10) indicates an incompatible drive command that would infringe the kinematic constraints and thus, cannot be executed as desired by the low level controller. We therefore propose a control architecture that extends the low-level drive controller with an additional *kinematics reasoning unit* which (a) analyzes the kinematics of a drive during on-line operation of the robot, (b) provides the inverse kinematics for the low level controller and (c) checks drive commands on the basis of the condition (11.10). An invalid drive command is thus captured and the kinematics reasoning unit can (d) either stop the robot and notify the path-planner about the failure to execute a desired drive command or (e) compute a re-configuration that maintains the operation of the robot to some extend in that it overrules the path-planner and deduces an alternative set-point $\dot{\xi}^*$ within B.

The basis for this operation of the robot drive controller is not a hard wired inverse kinematics model that provides the basis to compute the set-points for the wheel's steering angles β_i and angular velocities $\omega_i = \dot{\varphi}_i$ but a general procedure that derives the inverse kinematics from the drive geometry and its present operational/fault condition. We achieve this functionality through introducing a generalized form of the rolling and sliding conditions that we call *qualitative rolling and sliding constraints*.

11.3.1 Kinematic Analysis with Qualitative Rolling and Sliding Constraints

The key algorithm for the kinematics reasoning procedure is the deduction of B. The kinematics of robot drives are typically described in terms of the *rolling* and *sliding constraints* given above in (11.7) and (11.8) where the matrices $\mathbf{J}_1$ and $\mathbf{C}_1$ depend upon the (time-varying) steering angles $\beta_s = [\beta_1, \ldots, \beta_{n_s}]^T$. In our *kinematics reasoning unit*, however, we want to decide, whether a velocity command $\dot{\xi}$ is admissible and controllable or not – *independently* from the steering angles of the wheels in the drive. Therefore, we take a more detailed look at the meaning of steering angle-independency for rolling and sliding constraints exemplary for a drive with n_c standard centered orientable (steered) wheels and n_f fixed oriented standard wheels.

Admissible movements for a robot are well understood and documented in literature [3, 7, 10]. Given the sliding constraint

$$\mathbf{C}_1(\beta_s)\dot{\xi} = \mathbf{0} \tag{11.11}$$

we deduce the admissible movements through null-space analysis of $\mathbf{C}_1$, i.e.

$$\ker(\mathbf{C}_1(\beta_s)) \subseteq \mathbb{R}^3. \tag{11.12}$$

The typical approach to obtain a steering angle independent analysis is to focus on the robot's unsteered wheels, as one can always select appropriate steering angles β_s so that the steered wheels do not impose additional constraints.[2] We generalize this notion and provide *qualitative constraints* that dynamically capture the kinematics of a drive at its specific operational condition. Each wheel $W_i \in \mathcal{W}$ contributes a mode-specific *qualitative sliding constraint* $\mathbf{c}_{q,i}^T$ for the combined matrix

[2] Of course, this only holds for steered wheels with unconstrained steering angles.

$$\mathbf{C}_q = \begin{bmatrix} \mathbf{c}_{q,1}^T \\ \vdots \\ \mathbf{c}_{q,n}^T \end{bmatrix} . \tag{11.13}$$

For example, a standard wheel with blocked steering (at angle β_i) contributes the usual sliding constraint

$$\mathbf{c}_{q,i}^T := \left[\cos(\alpha_i + \beta_i) \quad \sin(\alpha_i + \beta_i) \quad l_i \ \sin(\beta_i)\right] . \tag{11.14}$$

A wheel with blocked steering and blocked rotation provides an additional constraint perpendicular to the wheel's axis (i.e. in rolling direction) so that we use

$$\mathbf{c}_{q,i}^T := \begin{bmatrix} \cos(\alpha_i + \beta_i) & \sin(\alpha_i + \beta_i) & l_i \ \sin(\beta_i) \\ \cos(\alpha_i + \beta_i + \frac{\pi}{2}) & \sin(\alpha_i + \beta_i + \frac{\pi}{2}) & l_i \ \sin(\beta_i + \frac{\pi}{2}) \end{bmatrix} . \tag{11.15}$$

A wheel with blocked rotation and operational steering leads to an analogous, steering angle independent constraint

$$\mathbf{c}_{q,i}^T := \begin{bmatrix} \cos(\alpha_i) & \sin(\alpha_i) & l_i \ \sin(0) \\ \cos(\alpha_i + \frac{\pi}{2}) & \sin(\alpha_i + \frac{\pi}{2}) & l_i \ \sin(\frac{\pi}{2}) \end{bmatrix} \tag{11.16}$$

that prevents all movements of the robot except a rotation at the wheel's contact point. A fully operational centered orientable steered wheel, on the other hand, does not impose any constraint as noted above. We express this fact by using a null-vector

$$\mathbf{c}_{q,i}^T := [0 \quad 0 \quad 0] \tag{11.17}$$

for the qualitative sliding constraint matrix $\mathbf{C}_q$.

Combining the mode dependent qualitative sliding constraints for all wheels $W_i, i = 1, \ldots, n$, provides $\mathbf{C}_q$ which generalizes $\mathbf{C}_{1f}$ that specifies the sliding constraints for fixed orientation wheels. This allows us to obtain the *space of admissible motions Z* for a specific configuration and mode of operation/failure of the drive through the null-space analysis

$$Z = \ker(\mathbf{C}_q) \subseteq \mathbb{R}^3 . \tag{11.18}$$

We now intend to provide an analogous analysis for the *space S of controllable movements*, i.e. the velocities $\dot{\xi} \in S$ that one can actuate through the motorized wheels in the drive. The implication of wheel actuation, i.e. $\dot{\varphi}_i \neq 0$ is captured through the rolling constraint

$$\mathbf{J}_1(\beta_s)\dot{\xi} = \mathbf{J}_2\dot{\varphi} . \tag{11.19}$$

A null-space analysis of $\mathbf{J}_1(\beta_s)$

$$\dot{\xi} \in \bar{S}_1\,(\beta_s) = \ker(\mathbf{J}_1(\beta_s))\,, \tag{11.20}$$

however, provides the *complement* $\bar{S}_1$ to the space of controllable movements.

Let us analyze this fact with one single wheel W_i at geometric position $\langle l_i, \alpha_i \rangle$ as shown in Fig. 11.2a to capture this implication in detail. Recall, that the wheel defines the following row

$$\mathbf{j}^T_{1,i}(\beta_i) = [\sin(\alpha_i + \beta_i) - \cos(\alpha_i + \beta_i) - l_i \cos(\beta_i)] \tag{11.21}$$

in the matrix $\mathbf{J}_1(\beta_s)$ of the rolling constraint. The null-space of $\mathbf{j}^T_{1,i}(\beta_s)$ has dimension 2. This can be argued by simple linear algebra: Let $\dot{\xi} \in \mathcal{X} = \mathbb{R}^3$. Then, (11.19) is a linear map $f(\dot{\xi}) : \mathcal{X} \to \mathbb{R}$ in the single wheel case. Thus,

$$\dim(\ker(f)) + \dim(\mathrm{im}(f)) = \dim(\mathcal{X}) = 3 \tag{11.22}$$

holds. The question of interest is now: How does the steering angle β_i influence the null-space of $\mathbf{j}^T_{1,i}(\beta_s)$? For this purpose, we re-write the rolling constraint in a factorized form that follows from the addition theorems of trigonometry

$$\mathbf{j}^T_{\beta,i}(\beta_i)\mathbf{j}^T_{q,i}\dot{\xi} = r_1\,\dot{\varphi}_i \tag{11.23}$$

where

$$\mathbf{j}^T_{\beta,i}(\beta_i) = [\,\cos(\beta_\mathrm{i})\ \sin(\beta_\mathrm{i})\,] \tag{11.24}$$

and

$$\mathbf{j}^T_{q,i} = \begin{bmatrix} \sin(\alpha_\mathrm{i}) & -\cos(\alpha_\mathrm{i}) & -l_i \\ \cos(\alpha_\mathrm{i}) & \sin(\alpha_\mathrm{i}) & 0 \end{bmatrix}. \tag{11.25}$$

Note that, until now, we have not made any changes to the rolling constraint. As mentioned before are we interested in the following question: Which motion set-points $\dot{\xi}$ are not controllable for *any* steering angle β? The answer is found in the factorized rolling constraint (11.23), which emphasizes:

$$\begin{aligned} \bar{S} = \ker(\mathbf{j}^T_{q,i}) &\subset \ker(\mathbf{j}^T_{\beta,i}(\beta_i)\,\mathbf{j}^T_{q,i}) \\ &\subset \ker(\mathbf{j}^T_{1,i}(\beta_i)) \\ &\subset \bar{S}_1(\beta_i)\,. \end{aligned} \tag{11.26}$$

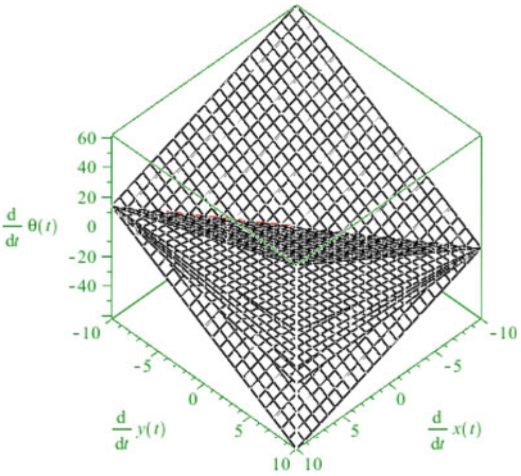

Fig 11.3 Planes $\bar{S}_1(\beta)$ at different steering angles of one single wheel at $\alpha = \pi/4$ and $l = 1$. The rotation axis of the planes defines $\bar{S}$

Since $\mathbf{j}_{q,i}^T$ does not depend on β_i, its null-space $\ker(\mathbf{j}_{q,i}^T)$ delivers exactly the motion set-points,[3] which are non-controllable through wheel W_i for any steering angle β_i. Note that $(\ker(\mathbf{j}_{q,i}^T)) = 1$. This means, that the space of qualitatively non-controllable velocities $\bar{S}$ has one dimension less than $\bar{S}_1(\beta_i)$. Thereby, the qualitative rolling constraint is less restricting than the effective rolling constraint, as it allows the steering angle to take the appropriate value. This corresponds to the apparent gain of a degree of freedom for the choice of controllable motion set-points.

If one thinks of $\mathcal{X}$ as three-dimensional space, the base vectors of $\bar{S}_1(\beta_i)$ span a plane (cf. 11.20). Any general velocity $\dot{\xi}$ can be depicted into components parallel to the plane and orthogonal to it. The parallel components violate the rolling constraint. Hence, all velocities having a non-zero component parallel to the plane are not controllable. Consistently, the set of controllable velocities is a straight line through the origin, which is orthogonal to the plane.

The set $\bar{S}$ has one degree of freedom – which corresponds to a straight line – and lies in $\bar{S}_1(\beta)$, independent of the steering angle β. Hence, with varying steering angle, the plane $\bar{S}_1(\beta)$ rotates around an axis $\bar{S}$ through the origin (see Fig. 11.3).

How can this result be interpreted, especially when there is more than one wheel? Again, we stack all qualitative rolling constraints to form $\mathbf{J}_q$

[3] More detailed analysis shows, that these are those motion set-points, which place the ICR in the wheel contact point.

$$\mathbf{J}_q = \begin{bmatrix} \mathbf{j}_{q,1}^T \\ \vdots \\ \mathbf{j}_{q,n}^T \end{bmatrix} \tag{11.27}$$

and analyze the null-space of $\mathbf{J}_q$ to obtain those motion commands, that cannot be actuated independently from the choice of the steering angles. However, this does not mean that velocity set-points that are not in this null-space can be driven! The non-controllable motions must be understood in a way similar to "non-controllable due to steering-angle independent under-actuation". Let us give the following examples: Consider a robot with:

1. One fixed standard wheel: $\ker(\mathbf{J}_q)$ will contain all motions that are not parallel to the rolling plane of the wheel.
2. One centered orientable standard wheel: $\ker(\mathbf{J}_q)$ will contain only those motions, that place the ICR in the contact point of the wheel.
3. One fixed standard wheel and one centered orientable wheel: $\ker(\mathbf{J}_q)$ will always be empty. The steered wheel is always able to turn the robot about the contact point of the fixed oriented wheel into the proper direction and execute a requested motion command.
4. Two centered orientable standard wheels: $\ker(\mathbf{J}_q)$ will always be empty.
5. Three fixed wheels, aligned in a way that they block any motion: $\ker(\mathbf{J}_q)$ will always be empty, meaning that all velocities are controllable – although no velocity can be driven! The disability of executing any motion will be handled by the sliding constraint.
6. Three fixed wheels, co-aligned in a way so that the rolling planes of the wheels are parallel: $\ker(\mathbf{J}_q)$ will contain all motions that are not parallel to the rolling planes of the wheels.

The specification of $\mathbf{j}_{q,i}^T$ in (11.4) describes a centered orientable standard wheel in the *nominal* case. A wheel with blocked steering simply acts as an unsteered wheel so that $\mathbf{j}_{q,i}^T$ becomes $\mathbf{j}_{q,i}^T = \mathbf{j}_{1,i}^T(\beta_i)$. The situation for a drive with impaired rotational actuation (e.g. a freely spinning wheel or a wheel with blocked rotation) implies that the wheel W_i cannot contribute to the robot's motorization. We express this fact through the null-vector $\mathbf{j}_{q,i}^T = [0 \;\; 0 \;\; 0]$. Table 11.1 summarizes the variants of $\mathbf{j}_{q,i}^T$ for operational and fault conditions together with the qualitative sliding constraints introduced above.

Summing up, our approach for kinematics reasoning proceeds as follows:

1. At the current time-point t_k, use the mode estimate $m(t_k)$ (provided through an estimation/diagnosis unit) and the currently valid drive's model $\mathcal{M}(t_k)$ to deduce the matrices of the rolling and sliding constraints $\{\mathbf{C}_q, \mathbf{J}_q, \mathbf{C}_1(\beta_s), \mathbf{J}_1(\beta_s), \mathbf{J}_2\}$.
2. Compute the *space of the admissible motions* Z through

$$Z = \ker(\mathbf{C}_q) \subseteq \mathbb{R}^3 \,. \tag{11.28}$$

Table 11.1 Qualitative rolling and sliding constraints for a standard wheel

Mode of operation	$\mathbf{j}_q^T$	$\mathbf{c}_q^T$
OK: actuated rotation and operational steering	$\begin{bmatrix} \sin(\alpha) & \cos(\alpha) \\ -\cos(\alpha) & \sin(\alpha) \\ -l & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$
Fault 1: actuated rotation and blocked steering	$\begin{bmatrix} \sin(\alpha+\beta) \\ -\cos(\alpha+\beta) \\ -l\cdot\cos\beta \end{bmatrix}^T$	$\begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \\ l\cdot\sin\beta \end{bmatrix}^T$
Fault 2: freely spinning wheel and blocked steering	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$	$\begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \\ l\cdot\sin\beta \end{bmatrix}^T$
Fault 3: blocked rotation and operational steering	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$	$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \\ 0 & l \end{bmatrix}^T$
Fault 4: freely spinning wheel and operational steering	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$

3. Compute the *space of non-controllable motions* $\bar{S}$ through

$$\bar{S} = \ker(\mathbf{J}_q) \subseteq \mathbb{R}^3\,. \tag{11.29}$$

4. Whenever the two spaces intersect, i.e. $Z \cap \bar{S} \neq \emptyset$, we have to refine the admissible velocities Z to exclude those movements that cannot be actuated through the robot's wheels. By computing the complement of $\bar{S}$

$$S = \ker(\bar{\mathbf{S}}^T) \subseteq \mathbb{R}^3\,, \tag{11.30}$$

where $\bar{\mathbf{S}}$ denotes the *matrix of basis vectors* for $\bar{S}$, we obtain the *controllable* velocities so that, finally, the intersection

$$Z \cap S =: B \tag{11.31}$$

defines the space of *admissible and controllable* velocities for a given mode of operation or failure of the robot.

5. Validate a drive command through checking

$$\dot{\xi} \in B\,,$$

provide $\{\mathbf{C}_q, \mathbf{J}_q, Z, \bar{S}, S, B\}$ to higher-level control and compute the steering angles β_s and rotational speeds $\dot{\varphi}$ for all steered and/or actuated wheels on the basis of $\{\mathbf{C}_1(\beta_s), \mathbf{J}_1(\beta_s), \mathbf{J}_2\}$.

The introduction of the *qualitative sliding* and, in particular, the *qualitative rolling constraints* enables us to re-formulate the well-known argument on null-space analysis of $\mathbf{C}_1$ (e.g. see [3, 10]) into an efficient algorithmic form that allows on-line drive-space and control/motorization analysis. The computations rely on several null-space computations (e.g. via singular-value decomposition) and the vector-space intersection (11.31) (e.g. with the Zassenhaus algorithm [12]). Illustrative examples for drive space computations can be found in [5]. All operations can be implemented efficiently so that one can utilize the reasoning concept directly within the drive's control loop and thus, reactively adapt the drive's control mechanism to the kinematics for the onset of operational modes and faults in the drive.

11.3.2 Applications

Our kinematics reasoning capability provides adaptivity and a sense of *self-awareness* to the low level robot drive controller. I.e. it can validate and execute velocity set-points that are commanded through a higher-level controller for an adaptive robot drive. This is particularly important for autonomously dealing with faults in the drive.

For example, in [5] we demonstrated an efficient way to recover from a faulty steering actuator through geometric reasoning on the basis of the admissible and controllable velocity space B . A blocked steering in an omni-directional robot leads to the loss of one-degree of freedom, so that we cannot choose the robot's velocity $\dot{\xi} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ arbitrarily. B degenerates to a two-dimensional subspace, i.e. a plane as shown in Fig. 11.4. Using this knowledge, on can easily compute a feasible drive command, e.g. with adapted rotational speed $\dot{\Theta} \rightarrow \dot{\Theta}^*$ through projecting the desired velocity $\dot{\xi}$ onto the plane as shown in Fig. 11.4. This re-configuration procedure maintains the longitudinal velocities $\dot{x}$ and $\dot{y}$ and thus, maintains the robot's movement along the planned path.

Another aspect of self-awareness can be drawn from the qualitative rolling constraint matrices/vectors $\mathbf{j}_{q,i}^T$. As we mentioned earlier, the null-spaces of $\mathbf{j}_{q,i}^T$ indicate critical velocities that put the ICR at wheel contact points. They capture the *singularities* of the drive. Providing this information to the path-planner, it is possible to actively avoid these critical velocities during operation of the drive. Furthermore, we were able to show that the qualitative rolling constraint matrices ($\mathbf{j}_{q,i}^T$) provide valuable information for robot-drive diagnosis as they allow one to formulate a kinematics-aware diagnosis and mode estimation procedure [9].

We use our control concept to deal with an onset of drive configurations which we can form with our modular robot platform [6] that allows us to configure

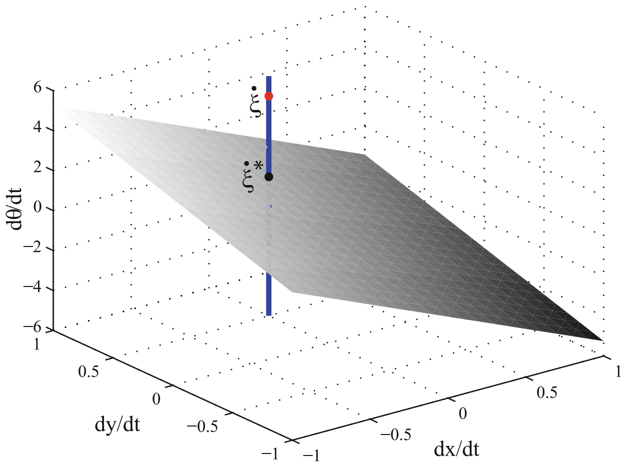

Fig. 11.4 Drive command adaption $\dot{\xi} \to \dot{\xi}^*$ to maintain the longitudinal velocities $\dot{x}, \dot{y}$ (cf. [5])

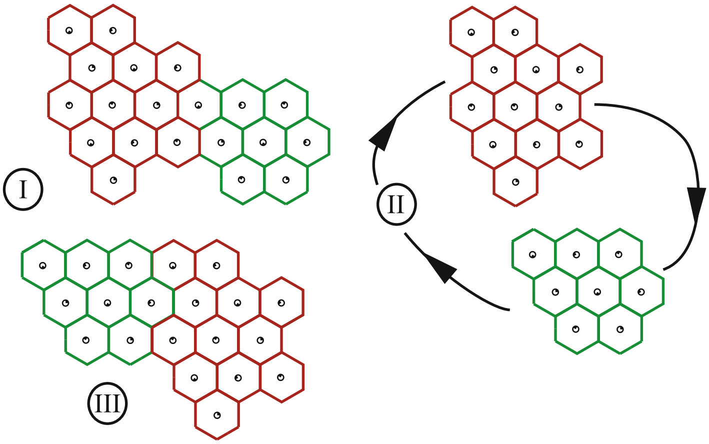

Fig. 11.5 Reconfigurable robot drive built from hexagonal drive modules

a large variety of robot drives and, in particular, re-configurable robots from hexagonal drive modules. Figure 11.5 shows such a compound robot-drive and its reconfiguration schematically. Controlling such an adaptive drive with a potentially large number of wheels is surely non-trivial. Our model-based approach with kinematic reasoning as its key component, however, allows us to deal with time-variant robot topologies and functionalities so that we can straightforwardly handle such complex robot-drives.

11.4 Conclusions and Outlook

Traditional kinematics analysis techniques are well suited for off-line robot drive design. Our intention, however, was to build a generic robot controller that can be applied to a wide variety of robot drives. Therefore, we presented a scheme for automated reasoning that analyzes and deduces the kinematics of a robot drive. Our algorithmic solution builds upon qualitative sliding and rolling constraints. These constraints represent steering-angle independent constraints that allow us to efficiently analyze a robot drive in terms of its admissible and controllable movements as well as its singularities. An efficient algorithmic formulation enables us to perform kinematics reasoning within the control-loop during the run-time of the robot. As a consequence, we can directly deal with drives that adapt in terms of geometry and functionality, as well as with typical faults that, otherwise, significantly change the kinematic behavior of the drive. Performing kinematics reasoning within the controller of the robot drive leads to a sense of self-awareness. This enables interesting new perspectives for robot control, such as intelligent, kinematics-aware path planning and alternative approaches for re-configurable robots and coordinated control of multi-robot systems.

To achieve these goals, we are currently working on a re-formulation of kinematics reasoning as distributed algorithm. This functionality, together with an autonomous modeling capability will enable us to deal with complex robot structures even more easily. Another line of research is the integration of kinematics reasoning with higher-level drive control. For example, we intend to use the drive's self-awareness to deduce appropriate mobility classifications that enables a hybrid control algorithm to schedule, or even deduce on-line, the appropriate control strategy to maintain an autonomous and robust behavior of a robot.

Acknowledgement This work has been funded through the Austrian Science Fund (FWF) under grant no. P20041-N15.

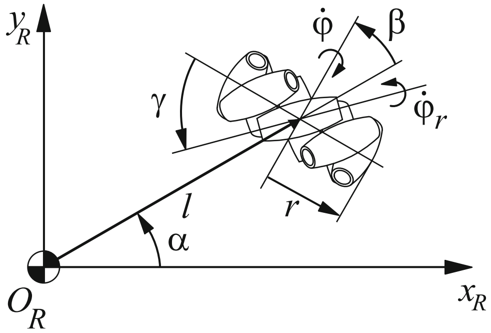

Fig. 11.6 Mecanum wheel geometry

Appendix

Qualitative Rolling and Sliding Constraints for an Omni-Directional Mecanum Wheel

It is straight forward to provide qualitative sliding constraints for Mecanum wheels as well. If one considers a mecanum wheel with the geometry as shown in Fig. 11.6, one obtains the standard rolling and sliding constraints as

$$\begin{aligned} r\dot{\varphi}\cos(\gamma) &= [\sin(\alpha+\beta+\gamma) \; -\cos(\alpha+\beta+\gamma) \quad -l\cos(\beta+\gamma)\,]\dot{\xi} \\ &= \mathbf{j}^T\dot{\xi}\,, \end{aligned}$$

$$\begin{aligned} 0 &= [\,\cos(\alpha+\beta+\gamma) \quad \sin(\alpha+\beta+\gamma) \quad l\sin(\beta+\gamma)\,]\;\dot{\xi} - r\dot{\varphi}\sin(\gamma) - r_r\,\dot{\varphi}_r \\ &= \mathbf{c}^T\dot{\xi}\,. \end{aligned}$$

Note that a fully operational mecanum wheel (actuated or freely spinning) does not impose any constraints on the robot's movement as the rotation of the individual rolls ensures the sliding constraint 1.32. We express this fact in terms of a qualitative constraint with $\mathbf{c}_q^T := [\,0 \quad 0 \quad 0\,]$. The rolling constraint is time-invariant as mecanum wheels are typically used without active steering. Therefore, we obtain $\mathbf{j}_q^T := [\,\sin(\alpha+\beta+\gamma) \quad -\cos(\alpha+\beta+\gamma) \quad -l\cos(\beta+\gamma)\,]$. However, a blocked mecanum wheel exhibits a behavior, where the individual rolls act like non-actuated standard wheels at $\beta+\gamma+\frac{\pi}{2}$. The according qualitative sliding constraint is thus

$$\mathbf{c}_q^T := \left[\,\cos(\alpha+\beta+\gamma+\tfrac{\pi}{2}) \quad \sin(\alpha+\beta+\gamma+\tfrac{\pi}{2}) \quad l\sin(\beta+\gamma+\tfrac{\pi}{2})\,\right]\,.$$

We summarize the matrix entries $\mathbf{c}_q^T$ and $\mathbf{j}_q^T$ for the qualitative constraint matrices $\mathbf{C}_q$ and $\mathbf{J}_q$ in Table 11.2.

Table 11.2 Qualitative rolling and sliding constraints for an omni-directional mecanum wheel

Mode of operation	$\mathbf{j}_{\mathbf{q}}^{\mathbf{T}}$	$\mathbf{c}_{\mathbf{q}}^{\mathbf{T}}$
OK: actuated rotation	$\begin{bmatrix} \sin(\alpha+\beta+\gamma) \\ -\cos(\alpha+\beta+\gamma) \\ -l\cdot\cos(\beta+\gamma) \end{bmatrix}^T$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$
Fault 1: blocked rotation	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$	$\begin{bmatrix} \cos\left(\alpha+\beta+\gamma+\frac{\pi}{2}\right) \\ \sin\left(\alpha+\beta+\gamma+\frac{\pi}{2}\right) \\ l\cdot\sin\left(\beta+\gamma+\frac{\pi}{2}\right) \end{bmatrix}^T$
Fault 2: freely spinning	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T$

References

1. Alexander J, Maddocks J (1989) On the kinematics of wheeled mobile robots. Int J Robot Res 8:15–27
2. Bhatt R, Tang C, Krovi V (2009) Formation optimization for a fleet of wheeled mobile robots – a geometric approach. Robot Auton Syst 57:102–120
3. Campion G, Bastin G, D'Andréa-Novel B (1996) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. IEEE Trans Robot Autom 12:47–62
4. Robuffo-Giordano P, Fuchs M, Albu-Schäffer A, Hirzinger G (2009) On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs. In: Proceedings of the IEEE international conference on robotics and automation (ICRA09), pp 4080–4087
5. Hofbaur M, Brandstötter M, Schörghuber C, Steinbauer G (2010) On-line kinematics reasoning for reconfigurable robot drives. In: Proceedings of the IEEE international conference on robotics and automation (ICRA10), pp 5441–5446
6. Hofbaur M, Brandstötter M, Jantscher S, Schörghuber C (2010) Modular reconfigurable robot drives. In: Proceedings of the international conference on robotics and automation and mechatronics (RAM 2010), pp 150–155
7. Kim W, Lee S, Yi B (2002) Mobility analysis of planar mobile robots. In: Proceedings of the IEEE international conference on robotics and automation (ICRA02), pp 2861–2867
8. Nüchter E, Bahr E, Lohmüller H (2008) Omnidirektionales Fahrzeug, Fahrmodul und mobiler Industrieroboter. German Patent Office, Offenlegungsschrift DE 10 2007 016 662 A1
9. Rienmüller T, Gruber C, Brandstötter M, Hofbaur M (2011) Odometry-based fault diagnosis for pseudo-omnidirectional wheeled mobile robots. In: Proceedings of the 22nd international-workshop on principles of diagnosis (DX11), Murnau, Germany, pp 29–35
10. Siegwart R, Nourbakhsh I (2004) Introduction to autonomous mobile robots. MIT Press, Cambridge
11. Williams B, Ingham M, Chung S, Elliott P, Hofbaur M, Sullivan G (2003) Model-based programming of fault-aware systems. AI Mag 24(4):61–75
12. Zassenhaus H (1948) Über einen Algorithmus zur Bestimmung der Raumgruppen. Commentarii Mathematici Helvetici 21:117–141