Motion Controller Design for a Mecanum Wheeled Mobile Manipulator

Conference Paper · August 2017

DOI: 10.1109/CCTA.2017.8062502

3 authors, including:

Christof Roehrig
University of Applied Science and Arts Dortmund
111 PUBLICATIONS **820** CITATIONS
SEE PROFILE

Frank Künemund
University of Applied Science and Arts Dortmund
20 PUBLICATIONS **136** CITATIONS
SEE PROFILE

Preprint, In: Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA 2017), Kohala Coast, Hawai'i, USA, August 2017.

Motion Controller Design for a Mecanum Wheeled Mobile Manipulator

Christof Röhrig, Daniel Heß, Frank Künemund

***Abstract*— Mobile manipulators extend the workspace of manipulators by mounting them on mobile platforms. The paper presents the design of a motion controller for a mobile manipulator. The mobile manipulator consists of a mobile platform driven by Mecanum wheels and a robotic arm with 6 degrees of freedom. A Mecanum wheeled platform provides 3 degrees of freedom in motion. Such a platform is usually driven by 4 or more Mecanum wheels, which are mounted in a fixed position at the platform. If a mobile platform is driven by more than 3 wheels, it builds an over-actuated and over-sensed driving system. The paper develops a compact and easy applicable kinematic model of over-actuated Mecanum wheeled mobile platforms that includes the kinematic motion constraints of the system. The kinematic model is described by a single Jacobi matrix, which is invertible and therefore can be used in the forward and inverse kinematic model. Furthermore the paper describes the overall controller structure for our mobile manipulator *OmnMan* including the real-time synchronization of platform and manipulator. The motion controller of the platform includes a coupling controller that controls the kinematic motion constraints. Experimental results evaluates the effectiveness of the coupling controller.**

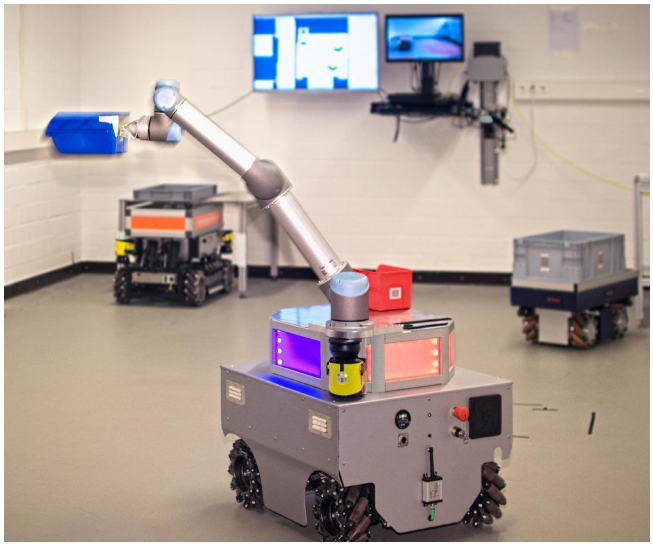

Fig. 1. Mobile manipulator *OmniMan*: UR5 robotic arm on a Mecanum wheeled mobile platform

I. Introduction

Industrial manipulators have been used in a large range of applications mainly in the production industry, but their application is limited in scenarios which need a very large working space such as aerospace manufacturing, ship building, and wind turbine manufacturing. Mobile manipulation is a solution to overcome these limitations and counts to be a key technology not only for the production industry but also for professional service robotics. Furthermore mobile manipulators find their applications in virtual or real laboratories for research and education [1], [2].

In the last years, research institutes have developed their own mobile manipulators based on commercial available mobile platforms and robotic arms. *Cody* from Georgia Institute of Technology consists of two arms from MEKA Robotics and a Mecanum wheeled Segway platform [3]. Cody was build mainly for research on service robotics in the health care domain. Another example in this area is *POLAR* (PersOnaL Assistant Robot) from Cornell University, which consists of a 7-DOF Barrett arm mounted on a Segway Omni base [4]. Other popular examples in this domain are Willow Garage's PR2 and the Fraunhofer IPA's Care-O-bot 4.

Compared to the domains of professional and domestic service, mobile manipulators for industrial applications operate in more structured environments. However, they require a higher level of operational efficiency, e.g. in terms of speed, accuracy and robustness, to be suitable for industrial applications [5]. An example of a mobile manipulator for research on industrial application is *MR ROAM* from the Southwest Research Institute (SwRI). It consist of a Motoman 7-DOF robotic arm mounted on a Vetex omnidirectional mobile platform. Other popular examples are are KUKA's *omniRob* and *Moiros* [6] as well as *ANNIE* and *LiSA* from Fraunhofer IFF, which are all based on Mecanum wheeled platforms.

A Mecanum wheel consists of a central hub with free moving rollers, which are usually mounted at $\pm 45^\circ$ angles around the hubs' periphery. The outline of the rollers is such that the projection of the wheel appears to be circular. A mobile platform driven by Mecanum wheels provides 3 degrees of freedom in motion. Usually such platforms consists of four or more wheels. A typical configuration is the four-wheeled one of the *URANUS* omnidirectional mobile robot [7]. The drive structure of a Mecanum wheeled mobile platform with four or more wheels is over-actuated, which means that actuation conflicts may occur. If two of four wheels fail, the driving system in under-actuated. For this case, Vlantis et al. develop a fault tolerant controller, which is able to control the platform if up to two wheels fail [8]. Omni wheels are similar to Mecanum wheels, they distinguish from them by the posture of the rollers, which are mounted with an angle of 90° with respect to the wheel axis. Rojas and Förster develop

This work was supported in part by the research program of the University of Applied Sciences and Arts in Dortmund (HiFF OmniMan, Project No. 04 003 39).

University of Applied Sciences and Arts in Dortmund, Intelligent Mobile Systems Lab, Otto-Hahn-Str. 23, 44227 Dortmund, Germany Web: www.imsl.fh-dortmund.de, Email: roehrig@ieee.org

a coupling matrix for a special case of an omnidirectional robot with n Omni wheels [9]. This coupling matrix is used to detect and avoid wheel slippage using a PID controller.

In this paper, a generalized kinematic model of over-actuated Mecanum wheeled mobile platforms is developed, which includes the kinematic motion constraints of the system and that is valid for Omni wheels as well. We develop a motion controller of the platform that includes a coupling controller, which controls the kinematic motion constraints. Furthermore the paper describes the overall controller structure of our mobile manipulator *OmniMan* (Fig. 1). *OmniMan* consists of an Universal Robots UR5 robotic arm mounted on Mecanum wheeled mobile platform from MIAG Fahrzeugbau GmbH (Braunschweig, Germany). The paper describes different options for interfacing a UR arm controller to a central controller PC as well as to synchronize the movements of the arm with the platform in real-time.

II. Problem Formulation

We consider the problem of motion control of a mobile manipulator, which consists of a robotic arm and a mobile platform. The mobile platform is driven by n Mecanum wheels. A Mecanum wheeled mobile platform provides any desired motion in $x-$ and $y-$direction and rotation θ around the $z-$axis, simultaneously. The mobile platform moves in 2D space, the pose of the platform (position and heading) in the world frame is defined as $\boldsymbol{x} = (x, y, \theta)^{\mathrm{T}}$ in the configuration space (C-space) $\mathcal{C}$, which is a subset of $\mathbb{R}^3$. $\mathcal{C} = \mathbb{R}^2 \times \mathcal{S}^1$ takes into account that $\theta \pm 2\pi$ yields to equivalent headings ($\theta \in [0, 2\pi)$) (see Fig. 2). The robot frame is a coordinate

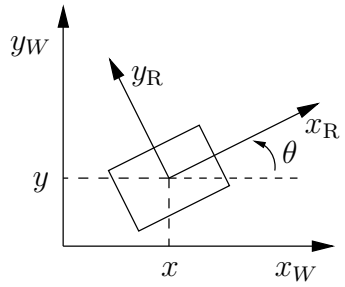

Fig. 2. Definition of pose $\boldsymbol{x} = (x, y, \theta)^{\mathrm{T}}$ and robot frame

system, which is fixed at the mobile platform. Velocities in the robot frame (F_{R}) can be transformed into the world frame (F_{W}), as a function of the heading θ:

$$\dot{\boldsymbol{x}}_{\mathrm{R}} = \boldsymbol{R}(\theta)\, \dot{\boldsymbol{x}}_{\mathrm{W}}, \quad \Rightarrow \quad \dot{\boldsymbol{x}}_{\mathrm{W}} = \boldsymbol{R}^{-1}(\theta)\, \dot{\boldsymbol{x}}_{\mathrm{R}} \tag{1}$$

$$\text{with} \quad \dot{\boldsymbol{x}}_{\mathrm{W}} = \begin{pmatrix} \dot{x}_{\mathrm{W}} \\ \dot{y}_{\mathrm{W}} \\ \dot{\theta} \end{pmatrix}, \dot{\boldsymbol{x}}_{\mathrm{R}} = \begin{pmatrix} \dot{x}_{\mathrm{R}} \\ \dot{y}_{\mathrm{R}} \\ \dot{\theta} \end{pmatrix},$$

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The kinematic model of vehicles equipped with Mecanum wheels is well known (see [7], [10]). The inverse kinematics

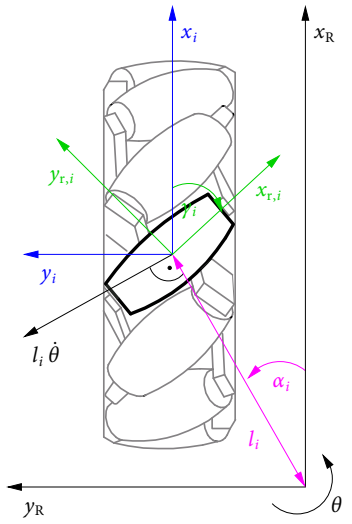

Fig. 3. Mecanum wheel with frames

of the platform can be described by linear equations in the robot frame, which can be written under the general matrix form:

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J}\dot{\boldsymbol{x}}_{\mathrm{R}}, \text{with } \boldsymbol{J} \in \mathbb{R}^{n\times 3}, \tag{2}$$

where $\dot{\boldsymbol{\varphi}} = (\dot{\varphi}_1, \dot{\varphi}_2, \ldots, \dot{\varphi}_n)^{\mathrm{T}}$ are the angular velocities of the wheels and $\boldsymbol{J}$ is a Jacobi matrix with constant parameters. For a platform equipped with $n > 3$ Mecanum wheels, the forward kinematic equations are overdetermined. The forward kinematics can be obtained by using a least square approach and applying the Moore-Penrose pseudo-inverse to $\boldsymbol{J}$:

$$\dot{\boldsymbol{x}}_{\mathrm{R}} = \boldsymbol{J}^{+}\dot{\boldsymbol{\varphi}}, \text{ with } \boldsymbol{J}^{+} = (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}} \tag{3}$$

The drive structure of a Mecanum wheeled mobile platform with 4 or more wheels is over-actuated. Solving the corresponding kinematic equations, it can be shown that $m = n-3$ kinematic motion constraints have to be met at any time in order to avoid additional wheel slippage.

In the next section, we develop a compact motion model for a mobile platform with n Mecanum wheels that considers this kinematic constraints of the system. The motion constrains are expressed in matrix form and augmented directly in the Jacobi matrix of the kinematic model. This allows sensing and controlling of the kinematic motion constraints.

III. Motion Model with Constraints for Mobile Platforms with Mecanum Wheels

For wheel i, we define the wheel frame F_i and the roller frame $F_{\mathrm{r},i}$, which are in a fixed position in the robot frame F_{R} (see Fig. 3). The position of the wheel frame F_i with respect to the robot frame is described by the 3 constant parameters: α_i, l_i and δ_i, where δ_i defines the rotation angle between F_i and the F_{R} and is usually equal to zero. γ_i defines the angle of the roller with respect to the wheel frame. $\varphi_i(t)$ drives the wheel and defines the rotation angle of the wheel around its horizontal axis of rotation. The wheel is driven in the direction of it's x_i axis. The wheel has one contact point

to the plane and is able to rotate around this point (rotation around its z_i axis).

For simplification, it is assumed that there is only one roller that has contact to the floor and that the contact point stays always in the center of the roller (and the wheel). The roller frame $F_{\mathrm{r},i}$ as well as the wheel frame F_i has its origin in this point of contact. The $x_{\mathrm{r},i}$ axis lies in the shaft of the roller. The wheel is able to move free in direction of the $y_{\mathrm{r},i}$ axis.

The movements of the platform yields to the velocities $\dot{x}_{\mathrm{R}}$, $\dot{y}_{\mathrm{R}}$, $l \cdot \dot{\theta}$ in the contact point of the wheel, which can be transformed to the roller frame $F_{\mathrm{r},i}$:

$$\dot{x}^{\mathrm{R}}_{\mathrm{r},i} = \dot{x}_{\mathrm{R}} \cos(\gamma_i + \delta_i) + \dot{y}_{\mathrm{R}} \sin(\gamma_i + \delta_i) + l_i\, \dot{\theta}\, \cos(\alpha_i + \pi/2 - \delta_i - \gamma_i) \tag{4}$$

$$\dot{y}^{\mathrm{R}}_{\mathrm{r},i} = -\dot{x}_{\mathrm{R}} \sin(\gamma_i + \delta_i) + \dot{y}_{\mathrm{R}} \cos(\gamma_i + \delta_i) + l_i\, \dot{\theta}\, \sin(\alpha_i + \pi/2 - \delta_i - \gamma_i) \tag{5}$$

where γ_i is the rotation angle between the wheel frame and the roller frame (usually $\pm 45°$) and δ_i is the angle between the wheel frame and the robot frame (usually $0°$). The rotation of the wheel drives the velocity $\dot{x}_i = r \cdot \dot{\varphi}_i$ in the point of contact, which can be transformed in x and y components of the roller frame $K_{\mathrm{r},i}$:

$$\dot{x}^{\varphi}_{\mathrm{r},i} = r\, \dot{\varphi}_i\, \cos(\gamma_i), \quad \dot{y}^{\varphi}_{\mathrm{r},i} = -r\, \dot{\varphi}_i\, \sin(\gamma_i) \tag{6}$$

Since the roller can not move in direction of its shaft, $\dot{x}^{\mathrm{R}}_{\mathrm{r},i} = \dot{x}^{\varphi}_{\mathrm{r},i}$ must be true, which leads to

$$\dot{x}_{\mathrm{R}} \cos(\gamma_i + \delta_i) + \dot{y}_{\mathrm{R}} \sin(\gamma_i + \delta_i) + l_i\, \dot{\theta}\, \cos(\alpha_i + \pi/2 - \delta_i - \gamma_i) = r\, \dot{\varphi}_i\, \cos(\gamma_i), \tag{7}$$

and finally to the inverse kinematic equation of wheel i

$$\dot{\varphi}_i = \frac{1}{r \cdot \cos(\gamma_i)} \big(\cos(\delta_i + \gamma_i),\ \sin(\delta_i + \gamma_i),\ l_i \sin(\delta_i + \gamma_i - \alpha_i) \big) \dot{\boldsymbol{x}}_{\mathrm{R}}. \tag{8}$$

For a robot equipped with n Mecanum wheels, (8) can be used to obtain the inverse kinematics of the platform as

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J} \dot{\boldsymbol{x}}_{\mathrm{R}}, \text{with } \boldsymbol{J} \in \mathbb{R}^{n \times 3} \tag{9}$$

If γ_i is set to equal zero, the Mecanum wheel becomes an Omni wheel and therefore (8) can be used to model the inverse kinematics of an Omni wheel driven platform as well.

For a robot with more than 3 Mecanum wheels, the kinematics are over-determined, which means that the wheel speed of $m = n - 3$ of the n wheels are a linear combination of the other ones. Only 3 rows of J are linear independent, m rows are a linear combination of the first 3 rows. To obtain the kinematic motion constraints, J can be split up into 2 sub-matrices:

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_3 \\ \boldsymbol{J}_m \end{pmatrix}, \text{with } \boldsymbol{J}_3 \in \mathbb{R}^{3\times 3}, \boldsymbol{J}_m \in \mathbb{R}^{m \times 3}, m = n - 3 \tag{10}$$

The velocities in the robot frame can be defined by 3 wheels:

$$\begin{pmatrix} \dot{x}_{\mathrm{R}} \\ \dot{y}_{\mathrm{R}} \\ \dot{\theta} \end{pmatrix} = \boldsymbol{J}_3^{-1} \begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{pmatrix}$$

this leads to m kinematic motion constraints:

$$\boldsymbol{J}_m \boldsymbol{J}_3^{-1} \begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{pmatrix} = \begin{pmatrix} \dot{\varphi}_4 \\ \vdots \\ \dot{\varphi}_n \end{pmatrix},$$

which can be expressed in general matrix form

$$\boldsymbol{T} \dot{\boldsymbol{\varphi}} = \boldsymbol{0}, \text{ with } \boldsymbol{T} = \left(\boldsymbol{J}_m \boldsymbol{J}_3^{-1},\ -\boldsymbol{I}_m \right), \tag{11}$$

where $\boldsymbol{I}$ denotes the identity matrix. We define a vector of angular error velocities $\dot{\boldsymbol{\epsilon}} = (\dot{\epsilon}_1 \dots \dot{\epsilon}_m)^{\mathrm{T}}$ to detect violations of the kinematic constraints:

$$\dot{\boldsymbol{\epsilon}} = \boldsymbol{T} \dot{\boldsymbol{\varphi}}, \tag{12}$$

$\boldsymbol{T}$ and $\dot{\boldsymbol{\epsilon}}$ can be used to augment the kinematics of the robot

$$\dot{\boldsymbol{x}}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}}^{-1} \dot{\boldsymbol{\varphi}}, \text{ with } \dot{\boldsymbol{x}}_{\mathrm{a}} = \begin{pmatrix} \dot{\boldsymbol{x}}_{\mathrm{R}} \\ \dot{\boldsymbol{\epsilon}} \end{pmatrix}, \boldsymbol{J}_{\mathrm{a}}^{-1} = \begin{pmatrix} \boldsymbol{J}^{+} \\ \boldsymbol{T} \end{pmatrix} \tag{13}$$

where $\boldsymbol{J}_{\mathrm{a}}$ is an invertible square matrix, which describes the augmented inverse kinematics of the system:

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J}_{\mathrm{a}} \dot{\boldsymbol{x}}_{\mathrm{a}} \tag{14}$$

The sub-vector $\dot{\boldsymbol{\epsilon}}$ in $\dot{\boldsymbol{x}}_{\mathrm{a}}$ can be used to control the kinematic constraints.

A. Mobile Platform with 4 Mecanum Wheels

A typical configuration of a Mecanum wheeled platform consists of 4 wheels (see Fig. 4). The positions of the wheels

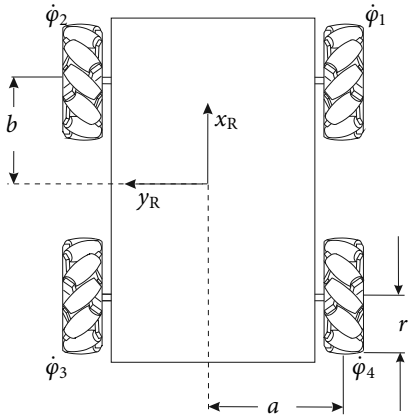

Fig. 4. Omnidirectional platform with Mecanum wheels (top view)

with respect to the robot frame can be defined as $\delta_i = 0$ and $l_i = \sqrt{a^2 + b^2}$. This leads to the configuration parameters shown in table I. These parameters in in conjunction with (8) yield to the inverse kinematics

$$\begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \\ \dot{\varphi}_4 \end{pmatrix} = \boldsymbol{J} \begin{pmatrix} \dot{x}_{\mathrm{R}} \\ \dot{y}_{\mathrm{R}} \\ \dot{\theta} \end{pmatrix}, \text{ with } \boldsymbol{J} = \frac{1}{r} \begin{pmatrix} 1 & 1 & (a+b) \\ 1 & -1 & -(a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{pmatrix} \tag{15}$$

TABLE I
CONFIGURATION OF THE PLATFORM IN FIG. 4

i	α_i	γ_i
1	$-\arctan\left(\frac{a}{b}\right)$	$+\frac{\pi}{4}$
2	$+\arctan\left(\frac{a}{b}\right)$	$-\frac{\pi}{4}$
3	$\frac{\pi}{2}+\arctan\left(\frac{a}{b}\right)$	$+\frac{\pi}{4}$
4	$-\frac{\pi}{2}-\arctan\left(\frac{a}{b}\right)$	$-\frac{\pi}{4}$

The kinematic constraint can be obtained using (11)

$$0 = \boldsymbol{T}\dot{\boldsymbol{\varphi}}, \text{ with } \boldsymbol{T} = \left(1, 1, -1, -1\right), \tag{16}$$

which leads to the augmented forward kinematics

$$\begin{aligned} \dot{\boldsymbol{x}}_\mathrm{a} = \boldsymbol{J}_\mathrm{a}^{-1}\dot{\boldsymbol{\varphi}}, \text{ with } \dot{\boldsymbol{x}}_\mathrm{a} = \begin{pmatrix} \dot{x}_\mathrm{R} \\ \dot{y}_\mathrm{R} \\ \dot{\theta} \\ \dot{\epsilon}_1 \end{pmatrix}, \dot{\boldsymbol{\varphi}} = \begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \\ \dot{\varphi}_4 \end{pmatrix}, \\ \boldsymbol{J}_\mathrm{a}^{-1} = \frac{r}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \frac{1}{a+b} & \frac{-1}{a+b} & \frac{-1}{a+b} & \frac{1}{a+b} \\ \frac{4}{r} & \frac{4}{r} & -\frac{4}{r} & -\frac{4}{r} \end{pmatrix} \end{aligned} \tag{17}$$

and the augmented inverse kinematics

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J}_\mathrm{a}\dot{\boldsymbol{x}}_\mathrm{a}, \text{ with } \boldsymbol{J}_\mathrm{a} = \frac{1}{r}\begin{pmatrix} 1 & 1 & (a+b) & 1 \\ 1 & -1 & -(a+b) & 1 \\ 1 & 1 & -(a+b) & -1 \\ 1 & -1 & (a+b) & -1 \end{pmatrix} \tag{18}$$

where r is the radius of the wheels, a and b are given by the dimension of the platform (see Fig. 4). Eqn. (17) is used in the motion controller of the platform to execute odometry and to sense the coupling error. Eqn. (18) is used in the motion controller to control the speeds in robot frame and to control the kinematic constraints.

IV. DYNAMIC MODEL FOR MOBILE PLATFORMS WITH MECANUM WHEELS

This section will briefly describe a dynamic model of Mecanum wheeled platforms including the friction force of the Mecanum wheels. The dynamics of a Mecanum wheeled platform can be described in the robot frame:

$$\boldsymbol{F}_\mathrm{R} = \boldsymbol{H}\,\ddot{\boldsymbol{x}}_\mathrm{R} = \boldsymbol{J}^{-1}\,\boldsymbol{F}_\mathrm{wheel} \tag{19}$$

with

$$\boldsymbol{F}_\mathrm{R} = \begin{pmatrix} F_\mathrm{x} \\ F_\mathrm{y} \\ M_\theta \end{pmatrix}, \quad \boldsymbol{H} = \begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{pmatrix}, \quad \boldsymbol{F}_\mathrm{wheel} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix},$$

where $\boldsymbol{F}_\mathrm{R}$ are the translational forces and the moment of force (torque) in robot frame, which accelerate the platform, m is the mass of the platform for translation, I is the moment of inertia for platform rotation around the z axis and $\boldsymbol{F}_\mathrm{wheel}$ are the tangential contact forces of the Mecanum wheels, which accelerate the platform. The acceleration in robot frame can be obtained from accelerations in world frame by differentiating (1), which leads to

$$\ddot{\boldsymbol{x}}_\mathrm{R} = \dot{\boldsymbol{R}}(\theta)\,\dot{\boldsymbol{x}}_\mathrm{W} + \boldsymbol{R}(\theta)\,\ddot{\boldsymbol{x}}_\mathrm{W} \tag{20}$$

and considers the effect of the *Coriolis* force. Each wheel i is driven by an electrical motor, which produces the moment

$$M_i = r\,F_i + M_{\mathrm{fric},i} + I_i\,\ddot{\varphi}_i \tag{21}$$

where r is the wheel radius, M_fric is the total of the moment produced by friction, I_i is the moment of inertia of the wheel and $\ddot{\varphi}_i$ is the angular acceleration of the wheel: $\ddot{\boldsymbol{\varphi}} = \boldsymbol{J}\ddot{\boldsymbol{x}}_\mathrm{R}$ The moment of motor i is produced by the armature current of the motor: $M_i = c_\phi\,I_{\mathrm{a},i}$, where c_ϕ is the motor constant. The dynamics of the motor current can be described as

$$\begin{aligned} L_\mathrm{a}\frac{\mathrm{d}I_{\mathrm{a},i}}{\mathrm{d}\,t} + R_\mathrm{a}\,I_{\mathrm{a},i} = U_{\mathrm{a},i} - U_{\mathrm{emf},i}, \\ \text{with} \quad U_{\mathrm{emf},i} = c_\phi\,n_\mathrm{gear}\,\dot{\varphi}_i \end{aligned} \tag{22}$$

where L_a is the armature inductance, R_a is the armature resistance, U_a is the armature voltage, U_emf is the voltage produced by the back electro mechanical force, n_gear is the ratio of the gear that connects motor with wheel.

The moment produced by friction M_fric is a sum of the internal friction produced by the motor and the gear, the rolling friction on the floor and the friction in the bearing of the rollers and therefore M_fric depends largely on the direction of movement. We model the moment produced by friction as a function of the platform movement

$$M_{\mathrm{fric},i} = f_i(\dot{\boldsymbol{x}}_\mathrm{R}), \tag{23}$$

which can be experimentally identified [11].

V. CONTROLLER DESIGN

As aforementioned, our mobile manipulator consists of a UR5 robotic arm and a Mecanum wheeled mobile platform. It is controlled by a control system, which is distributed on 2 control PCs. Fig. 5 shows the control architecture of the whole system. Details of the different controllers are described in the next sections.

A. Control of the UR5 Manipulator

The UR5 is a lightweight 6-DOF industrial manipulator manufactured by Universal Robots (UR). It has a weight of 18.4 kg, a reach of 85 cm and a maximal payload of 5 kg [12]. The UR5 can be controlled at three different levels: The Graphical User Interface (GUI) Level, the Script Level and the C Application Programming Interface (C-API) Level [13]. *PolyScope* is the graphical user interface (GUI) for operating the robotic arm and for creating and executing robot programs. *URScript* is the robot programming language used to control the robot at the Script Level. This programs can be saved directly on the robot controller or commands can be sent via TCP socket to the robot.

User programs that uses the C-API are executed on the UR controller and interact directly at the joint level with a cycle time of 8 ms. At this level, the UR controller can be supplied by either joint velocities or a combination of

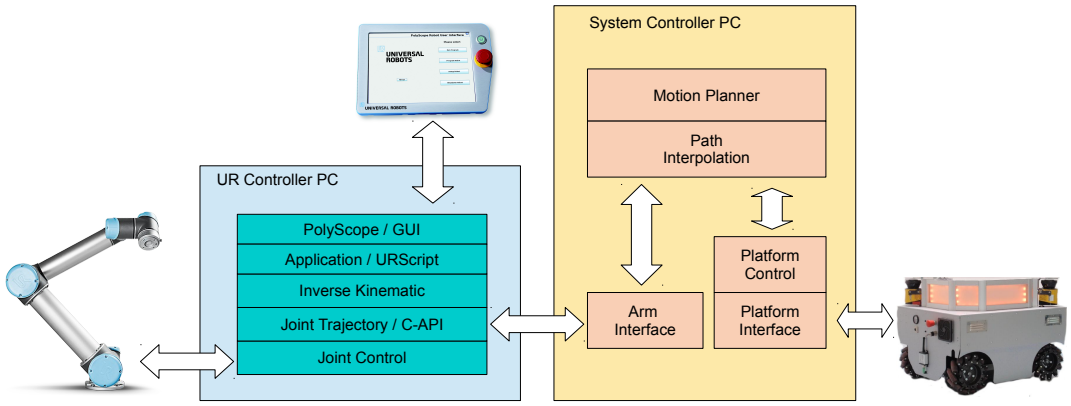

Fig. 5. Control architecture of Mecanum wheeled mobile manipulator

joint positions, joint velocities and joint accelerations. In our mobile manipulator, we use the C-API and control the UR5 at joint level. At this level, the synchronization of the platform movements with the arm movements is feasible in real-time. This enables trajectory planning for the whole system in Cartesian space and controlling the movements of platform and arm synchronously at joint level, while avoiding obstacles in real time, without leaving the trajectory.

The structure of the controllers with its interfaces is shown in Fig. 5. The robotic arm is controlled by the UR controller PC at joint level. The joint positions of the arm are generated by the path interpolation module in the system controller PC and streamed every 8 ms over a TCP socket and Ethernet to the UR controller PC. The UR controller PC controls the joints and transfers the control signals over an interface module to the motors.

B. Control of the Mecanum Wheeled Mobile Platform

The controller of the Mecanum wheeled mobile platform is integrated in the system controller PC (platform control in Fig. 5). The internal structure of this controller is shown in Fig. 6. The path interpolation generates a stream of poses $\boldsymbol{x} = (x, y, \theta)^{\mathrm{T}}$ and its derivatives $\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$ as input for the controller. The structure of the controller is derived from a classical cascade structure. It consists of a position controller, which controls the pose of the platform in world frame, a wheel controller, which controls the velocity of the wheels and a coupling controller, which controls the kinematic motion constraints.

The position controller obtains the actual pose of the platform by odometry and generates the velocities of the platform in the world frame plus feed forwarding the velocities from the path interpolation. Details of the odometry for Mecanum wheeled platforms can be found in [14]. These velocities are transformed into the robot frame and after it into reference velocities for the wheel controller. These reference velocities are calculated using the inverse kinematics of the platform (8) and therefore meet the kinematic constraints at any time. Owing to parameter differences in the control loops of the wheels or unbalanced loads, the actual wheel velocities may violate the kinematic motion constraints. These violations lead to additional wheel slippage.

Aim of the coupling controller is to change the reference velocities of the wheel controllers in such a way that the actual velocities meet the motion constraints. The coupling controller obtains the angular velocity of the coupling error $\dot{\boldsymbol{\epsilon}}$ using $\boldsymbol{J}_{\mathrm{a}}^{-1}$ from odometry (see (13)). The coupling error $\boldsymbol{\epsilon}$ is obtained by numerical integration:

$$\boldsymbol{\epsilon} = \int_0^t \dot{\boldsymbol{\epsilon}}(\tau)\mathrm{d}\tau \approx \sum \Delta\boldsymbol{\epsilon} \tag{24}$$

To correct the coupling error, a correcting variable is generated

$$\dot{\boldsymbol{\epsilon}}_{\mathrm{c}} = -\boldsymbol{k}_{\epsilon}\boldsymbol{\epsilon} \tag{25}$$

and fed back to the wheel controller using $\boldsymbol{J}_{\mathrm{a}}$ (see (14)). Owing to the discretization of the encoder signals, the coupling error velocity $\dot{\boldsymbol{\epsilon}}$ is overlaid by heavy noise. Hence, the coupling controller uses $\boldsymbol{\epsilon}$ as input, which compensate longer lasting coupling errors only and reduces the noise on $\dot{\boldsymbol{\epsilon}}_{\mathrm{c}}$.

The wheel controller controls the velocity of the wheels by feed back control using the wheel encoders plus optional calculated torque feed forward control. The desired torques of the wheels are calculated based on (21).

C. Experimental Results

In order to evaluate the effectiveness of the coupling controller several experiments are performed with some of our mobile platforms, which are shown in Fig 1. The results of one of these experiments are shown in Fig. 7. The platform moves a path with straight lines forwards, sideways and backwards. After that, it moves a circular path with two rotations back to the starting point. The red curve in the picture shows the coupling error ϵ_1. The five movements of the platform yield to a coupling error different from zero, where the individual movements can be easily distinguished. The coupling error may be caused by parameter differences between the velocity control loops of individual wheels for instance variations in friction, motor constants or amplifier gains. The coupling error leads to an additional wheel slippage, which decreases the accuracy of odometry.

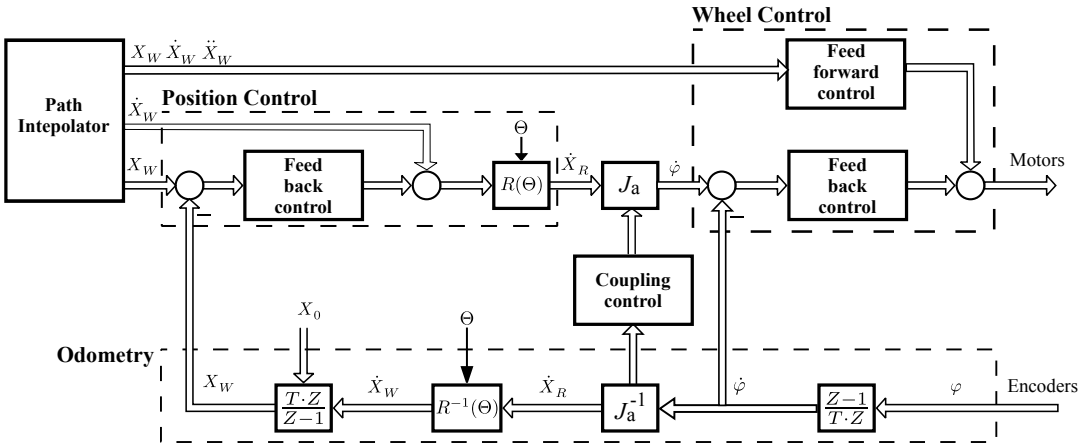

Fig. 6. Controller structure of Mecanum wheeled robot

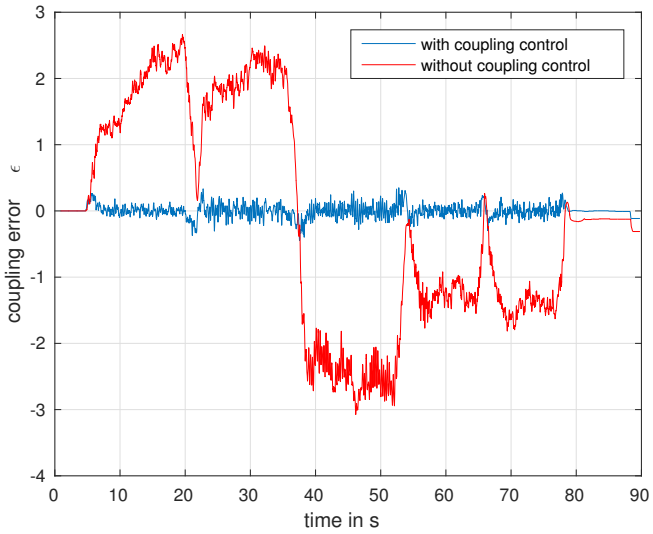

Fig. 7. Coupling error over time

The blue curve in Fig 1 shows the coupling error ϵ_1 with active coupling control ($k_c = 2$). Compared to the red curve, the coupling error is reduced nearly to zero.

VI. Conclusions

In this paper, we have developed a compact and easy applicable kinematic model of over-actuated Mecanum wheeled mobile platforms that includes the kinematic motion constraints of the system. The model is also valid for platforms driven by Omni wheels. The kinematic model is described by a single Jacobi matrix, which is invertible and therefore can be used in the forward and inverse kinematic model. The forward kinematics can be used to sense violations of the kinematic motion constraints that may be caused by wheel slippage. The inverse kinematics can be employed to control the kinematic constraints. The motion controller of the platform includes a coupling controller that controls the kinematic motion constraints using the Jacobi matrix. Experimental results have shown the effectiveness of the coupling controller.

References

[1] C. Röhrig and A. Jochheim, "The virtual lab for controlling real experiments via Internet," in *Proceedings of the 11th IEEE International Symposium on Computer-Aided Control System Design*, Hawaii, USA, Aug. 1999, pp. 279–284.

[2] R. Bischoff, U. Huggenberger, and E. Prassler, "KUKA youbot - a mobile manipulator for research and education," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1–4.

[3] C.-H. King, T. L. Chen, A. Jain, and C. C. Kemp, "Towards an assistive robot that autonomously performs bed baths for patient hygiene," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 319–324.

[4] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1021–1043, 2012.

[5] M. Hvilshøj, S. Bøgh, O. Skov Nielsen, and O. Madsen, "Autonomous industrial mobile manipulation (AIMM): past, present and future," *Industrial Robot: An International Journal*, vol. 39, no. 2, pp. 120–135, 2012.

[6] C. Sprunk, B. Lau, P. Pfaff, and W. Burgard, "An accurate and efficient navigation system for omnidirectional robots in industrial environments," *Autonomous Robots*, vol. 41, no. 2, pp. 473–493, 2017.

[7] P. F. Muir and C. P. Neuman, "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 1772–1778.

[8] P. Vlantis, C. P. Bechlioulis, G. Karras, G. Fourlas, and K. J. Kyriakopoulos, "Fault tolerant control for omni-directional mobile platforms with 4 mecanum wheels," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2395–2400.

[9] R. Rojas and A. G. Förster, "Holonomic control of a robot with an omnidirectional drive," *KI-Künstliche Intelligenz*, vol. 20, no. 2, pp. 12–17, 2006.

[10] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.

[11] F. Künemund, D. Heß, and C. Röhrig, "Energy efficient kinodynamic motion planning for holonomic AGVs in industrial applications using state lattices," in *Proceedings of the 47th International Symposium on Robotics (ISR 2016)*, Munich, Germany, Jun. 2016, pp. 459–466.

[12] *UR5 Technical Specifications*, Universal Robots A/S, Odense, Denmark, 2015, http://www.universal-robots.com/media/50588/ur5_en.pdf.

[13] *The URScript Programming Language, Version 3.1*, Universal Robots A/S, Odense, Denmark, Jan. 2015.

[14] C. Röhrig, A. Heller, D. Heß, and F. Künemund, "Global localization and position tracking of automatic guided vehicles using passive RFID technology," in *Proceedings of the joint 45th International Symposium on Robotics (ISR 2014) and the 8th German Conference on Robotics (ROBOTIK2014)*, Munich, Germany, Jun. 2014.