

# Modelling and simulation of the drivetrain of an omnidirectional mobile robot

Levente Raj & András Czmerk

To cite this article: Levente Raj & András Czmerk (2017) Modelling and simulation of the drivetrain of an omnidirectional mobile robot, *Automatika*, 58:2, 232-243, DOI: [10.1080/00051144.2017.1391612](https://doi.org/10.1080/00051144.2017.1391612)

To link to this article: <https://doi.org/10.1080/00051144.2017.1391612>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



[View supplementary material](#)



Published online: 20 Nov 2017.



[Submit your article to this journal](#)



Article views: 1903



[View related articles](#)



[View Crossmark data](#)



# Modelling and simulation of the drivetrain of an omnidirectional mobile robot

Levente Raj \* and András Czmerk

Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics, Budapest, Hungary

## ABSTRACT

The main contribution of this paper is to present the state-space model of an omnidirectional drivetrain (Kiwi drive) for mobile robots. Holonomy or omnidirectional movement of mobile robots is important in applications where navigation through tight space is required. Besides the already available literature regarding either dynamic or kinematic models, this paper presents the complex analysis of a mechatronic system and shows a coherent model that takes into account both the kinematics and the dynamics of a coupled electromechanical system. The model was developed based on two holonomic mobile robots: the Festo Robotino<sup>®</sup> and the MOGI Ethon. A simulation software developed by the authors is also presented for numerical evaluation of the model. The simulation results qualitatively confirm the model by fulfilling the empirical requirements. Both the model and the simulation software are expandable in order to improve the accuracy of the model describing actual mobile robots. Compared to other models which are based only on the kinematics of the Kiwi drivetrain, in accordance to one's technical instinct, the presented model shows that there is a cross effect between the motors of the drivetrain, where the motors affect each other.

## ARTICLE HISTORY

Received 20 June 2016  
Accepted 29 August 2017

## KEYWORDS

Kiwi drivetrain;  
omnidirectional mobile  
robot; simulation; state-space  
model

## 1. Introduction

Omnidirectional movement of mobile robots is widely used in the field of mobile robotics. Different approaches have been made through the last few years. The possible drivetrains were analysed and compared to each other in the literature several times [1,2].

In this paper, a mathematical model is presented for an omnidirectional, holonomic mobile robot equipped with the so-called Kiwi drivetrain based on three omnidirectional wheels. The omnidirectional wheel is similar to the Mecanum wheel [3,4], but the difference between the two wheels is that the omnidirectional wheel has a number of free rollers on the circumference of the wheel and the axes of the rollers are tangentially arranged to the wheel, while the axes of the rollers of a Mecanum wheel have non-zero angle with the tangent of the wheel. (see Figures 1 and 2).

The suggested model takes into account both the kinematics and the dynamics of the drivetrain; furthermore, it contains the dynamics of the DC motors in state-space representation.

The intention of creating this model is to further improve the literature of the omnidirectional mobile robots. Dynamic models from different drivetrains already exist. The most commonly used drivetrain is the differential drive which is a so-called quasi-holonomic drivetrain. This drivetrain was already modelled and analysed in many studies; thus, it has the widest literature

ranging from kinematics, through dynamics and even covering the control of vehicles equipped with differential drive [6–9]. The four Mecanum wheel-based drivetrain is also modelled and explained in several publications. It is also widely used in different applications [10–13]. Another quasi-holonomic drivetrain is the active-caster drivetrain, in which a number of casters are driven individually by motors and the orientation of the casters can be also adjusted electrically [14,15].

It is quite common to find models of the Kiwi drive where only the kinematics of the drivetrain is considered, while the dynamics is neglected [16–19]. In Liu et al. [20], a dynamical model was mentioned; however, the focus was kept for path planning and path control. In their publication, the equations of the current rates of the DC motors were neglected. At that state-of-art, neglecting the current rates was appropriate; however, nowadays, hardware capabilities enable us to create robotic applications based on more detailed models and so to consider the equations of the current rates.

The subjects of our investigation are two omnidirectional mobile robots, equipped with Kiwi drivetrain. The Festo Robotino<sup>®</sup> and the Ethon, developed by the Department of Mechatronics, Optics and Mechanical Engineering Informatics (MOGI) of the Budapest University of Technology and Economics (BME) were chosen for this purpose. The next section gives a brief description of these robots.



Figure 1. The Festo Robotino®.

### 1.1. The Festo Robotino®

The Festo Robotino® is an omnidirectional mobile robot equipped with three omni-wheels and designed for educational use (see Figure 1).

In this paper, only the drivetrain and the mechanical assembly of the Robotino® will be analysed. A complete description of the robot can be found in [21].

In Figure 2, one driving block of the Robotino® is shown. This block consists of a 24 V DC motor, an incremental encoder, a toothed belt, a gear unit and finally an all-way-roller also called omnidirectional wheel or omni-wheel. The wheel is a common wheel equipped with a number of free-running rollers, fitted on the circumference of the wheel.

Figure 3 shows the arrangement of the above mentioned driving blocks. There are three driving blocks in the Robotino®. All of them are equally spaced at 120° around the vertical axle crossing the centre of mass of

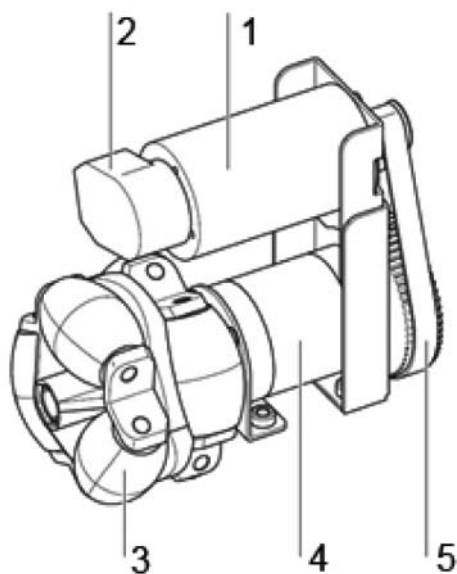


Figure 2. Driving block of the Robotino®. (1) Motor, (2) Incremental encoder, (3) Omni-wheel, (4) Gear unit, (5) Toothed belt [5].

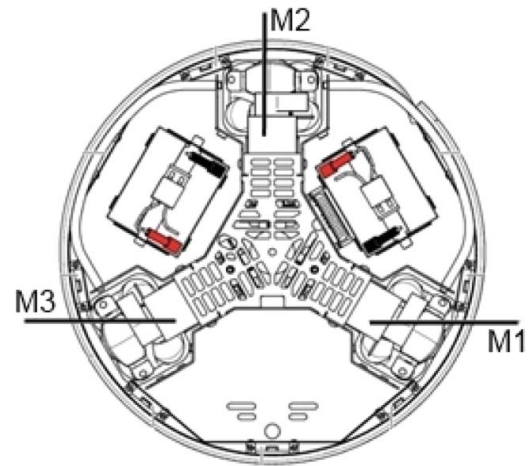


Figure 3. Drivetrain assembly of the Robotino®, where the driving blocks are noted as M1, M2 and M3 [5].

the robot. In Figure 3, the driving blocks are noted as M1, M2 and M3.

### 1.2. The MOGI Ethon

The latest development from the MOGI of the BME is Ethon, the ethologically inspired service robot (see Figure 4). The Ethon has an omnidirectional drivetrain, with three independent DC servo motors and associated powertrain elements. This drivetrain enables the robot to move freely on a planar surface and also to rotate around its vertical axis. The motors are controlled individually by custom-developed embedded hardware and software. The control algorithms of the robot are implemented on a PC. The personal computer (PC) is located on the robot, thus making the autonomous movement of the robot possible [16].



Figure 4. MOGI Ethon.

The MOGI Ethon is used in etho-robotics research where ethologically inspired mobile robots are created and continuously investigated by researchers from robotics, mechatronics, biology and ethology [22].

## 2. Development of the state-space model

In this section, the state-space model of the holonomic drive is pursued. Although the kinematic model of the Kiwi drivetrain can be found in the literature, in order to build a coherent model including kinematics, dynamics and the electrical part (DC motors), a new kinematic model was developed based on unified notation.

In the first approximation, some assumptions were made regarding the mechanical construction of the robot. Regarding the driving blocks, only the wheel speeds and inertias are taken into account. All masses and inertias of the elements in the driving blocks are considered as parts of the mass and inertia of the wheel. The gear ratio between the motor and the wheel is also neglected.

The omni-wheel itself is considered as one rigid body with the ability to produce tractive effort without slipping on the ground. This corresponds to the rolling constraint in case of a traditional wheel. On the other hand, the centre point of the wheel could have an axial velocity which is produced by the other two wheels. The axial velocity of the centre of mass of the wheels exists because of the free-running rollers. The rolling direction of the rollers is parallel to the shaft of the wheels (see Figure 2).

It is assumed that the axes of the shafts of each wheel cross each other at one point, and one point only, and this point is the centre of mass of the robot (see Figure 3).

### 2.1. Kinematics of the omnidirectional drivetrain

Former paper of the authors includes the kinematics of the omnidirectional drivetrain and can be found in [21], only the main steps are highlighted in this section. The following assumptions were made in order to derive the relationship between the wheel speeds and the coordinate velocities of the robot body:

- Each wheel contacts the ground at a single point, at a given time.
- All wheels roll on the ground without slipping.

First three Cartesian coordinate systems were fixed to the wheels ( $\mathcal{F}_i$ , where  $i = 1, 2, 3$ ), one coordinate system to each wheel (see Figure 5). In the coordinate systems ( $\mathcal{F}_i$ ), the normal axis  $n_i$  is parallel to the shaft of the wheels, the binormal direction  $b_i$  is parallel to the vertical direction and the tangential direction  $t_i$  is perpendicular to both the above-mentioned directions.

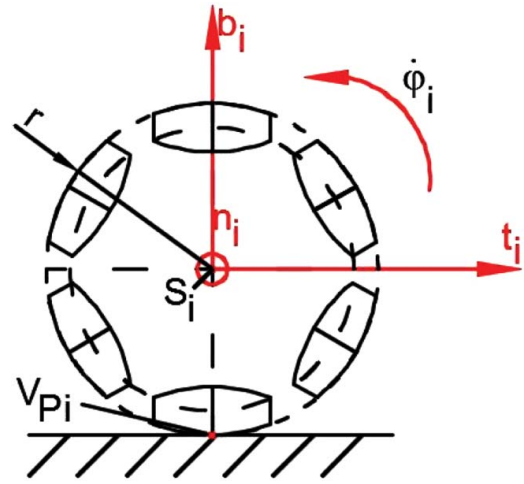


Figure 5. Side view of an idealized omni-wheel, showing the coordinate system  $\mathcal{F}_i\{S_i; n_i, t_i, b_i\}$  attached to the wheel.

The lower index  $i$  represents the driving blocks. The upper index  ${}^{[i]}$  denotes a coordinate system in which the variable is considered.

Using the notation of Figure 5, the angular velocities of the wheels can be expressed as

$${}^{[i]}\dot{\varphi}_i(t) = [\dot{\varphi}_{ni}(t) \ 0 \ 0]_{n_i t_i b_i}^T, \quad (1)$$

where  $\dot{\varphi}_{ni}(t)$  denotes the angular velocity for each wheel in the normal direction in their own coordinate systems, produced by their corresponding motors.

Since there is no sliding in any direction, the following expression can be written for the velocity of the contact point:

$${}^{[i]}\mathbf{v}_{P_i}(t) = [\dot{n}_i(t) \ 0 \ 0]_{n_i t_i b_i}^T, \quad (2)$$

where  $\dot{n}_i(t)$  is a constraint velocity, produced by the other two wheels, looking from the coordinate system of a given wheel.

Using the velocity transformation inside the rigid body, the translational velocity of the centre point of each wheel can be expressed as

$${}^{[i]}\dot{\mathbf{S}}_i(t) = {}^{[i]}\mathbf{v}_{P_i} + {}^{[i]}\dot{\varphi}_i(t) \times {}^{[i]}\mathbf{r}_w, \quad (3)$$

where  ${}^{[i]}\mathbf{r}_w$  is the vectorial form of the radius of the wheels and  $r$  is the radius of the wheels:

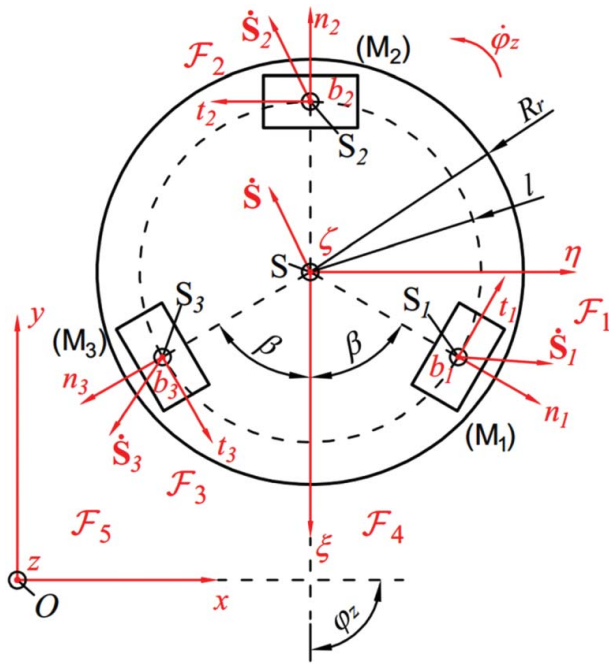
$${}^{[i]}\mathbf{r}_w = [0 \ 0 \ r]_{n_i t_i b_i}^T. \quad (4)$$

Applying (3) to each wheel, the velocity of the centre of mass of each wheel has the following form:

$${}^{[i]}\dot{\mathbf{S}}_i(t) = [\dot{n}_i(t) \ -r\dot{\varphi}_{ni}(t) \ 0]_{n_i t_i b_i}^T \quad (5)$$

Based on the assembly detailed in Figure 3, a functional outline of the idealized omnidirectional drive





**Figure 6.** Top view of the idealized omnidirectional drive system, showing the coordinate systems attached to the wheels ( $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ ), the robot coordinate system ( $\mathcal{F}_4$ ) and the global coordinate system ( $\mathcal{F}_5$ ).

system can be created as shown in Figure 6. A Cartesian coordinate system,  $\mathcal{F}_4\{S; \xi, \eta, \zeta\}$  is considered as the robot coordinate system that is attached to the centre of mass of the robot body. Another Cartesian coordinate system,  $\mathcal{F}_5\{O; x, y, z\}$  is introduced as the global coordinate system that is fixed in space. The goal is to develop the relationship between the wheel speeds and the coordinate velocities of the robot body in  $\mathcal{F}_5$ , because  $\mathcal{F}_5$  is the reference coordinate system.

In Figure 6, the coordinate systems attached to the wheels are also shown. The black rectangles in Figure 6 represent the driving blocks.

The robot coordinate system ( $\mathcal{F}_4$ ) is fixed in such a way that  $\zeta$  maintains equal angular spacing between  $n_1$  and  $n_3$ . Let  $\beta$  be the angle between  $\zeta$  and  $n_1$  and also  $\zeta$  and  $n_3$ . Since the driving blocks are equally spaced by  $120^\circ$  in both robots,  $\beta$  is constant  $60^\circ$ ; however,  $\beta$  is kept as a constructional parameter in the model that enables the model to handle other symmetrical constructions.

Using the notations of Figure 6, transformation matrices can be considered to transform vectors from one coordinate system to another. Essentially the transformation matrices between ( $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ ) and  $\mathcal{F}_4$  are given as

$$\mathbf{R}_1 = \mathbf{R}_3^T = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$\mathbf{R}_2 = \text{diag}(-1, -1, 1), \quad (7)$$

The  $\mathbf{R}_1$  and  $\mathbf{R}_3$  matrices are simple rotation matrices around the  $b_i$  axis of the coordinate systems of the respective wheels by a constant  $\beta$  angle.

Finally, the transformation between  $\mathcal{F}_4$  and  $\mathcal{F}_5$  was considered as

$$\mathbf{R}_G = \begin{bmatrix} \cos(\varphi_z(t)) & -\sin(\varphi_z(t)) & 0 \\ \sin(\varphi_z(t)) & \cos(\varphi_z(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

where  $\varphi_z(t)$  is the angle between  $\xi$  and  $x$ , hence the angle between  $\mathcal{F}_4$  and  $\mathcal{F}_5$ . Here, it is noted that  $\zeta$  and  $z$  are parallel and pointing to the same direction.

In order to get the desired relationship between the coordinate velocities of the robot and the angular velocities of the wheels, the following equations must be solved for  $i = 1, 2, 3$ :

$${}^{[5]}\dot{\mathbf{S}}_i(t) = {}^{[5]}\dot{\mathbf{S}}(t) + {}^{[5]}\dot{\varphi}(t) \times {}^{[5]}\mathbf{r}_{SSi}. \quad (9)$$

The vectors  ${}^{[5]}\dot{\mathbf{S}}_i(t)$  are the velocities of the wheels described in  $\mathcal{F}_5$ .

${}^{[5]}\dot{\mathbf{S}}(t)$  is the velocity of the robot body described in  $\mathcal{F}_5$ , in the following form:

$${}^{[5]}\dot{\mathbf{S}}(t) = [\dot{x}(t) \quad \dot{y}(t) \quad 0]_{xyz}^T. \quad (10)$$

${}^{[5]}\dot{\varphi}(t)$  is the angular velocity of the robot body in  $\mathcal{F}_5$ :

$${}^{[5]}\dot{\varphi}(t) = \begin{bmatrix} 0 \\ 0 \\ \dot{\varphi}_z(t) \end{bmatrix}_{xyz} + \mathbf{R}_G \sum_{i=1}^3 \mathbf{R}_i^{[i]}\dot{\varphi}_i(t), \quad (11)$$

and  ${}^{[5]}\mathbf{r}_{SSi}$  are the vectors pointing from the centre of mass of the robot to the centre of mass of each wheel:

$${}^{[5]}\mathbf{r}_{SS1} = \mathbf{R}_G [l \cos(\beta) \quad l \sin(\beta) \quad 0]^T, \quad (12)$$

$${}^{[5]}\mathbf{r}_{SS2} = \mathbf{R}_G [-l \quad 0 \quad 0]^T, \quad (13)$$

$${}^{[5]}\mathbf{r}_{SS3} = \mathbf{R}_G [l \cos(\beta) \quad -l \sin(\beta) \quad 0]^T, \quad (14)$$

where  $l$  is the radius of the pitch circle of the wheels.

Combining (9) to (14) yields for  $i = 1, 2, 3$ :

$$\mathbf{R}_G \mathbf{R}_i^{[i]}\dot{\mathbf{S}}_i(t) = {}^{[5]}\dot{\mathbf{S}}(t) + {}^{[5]}\dot{\varphi}(t) \times {}^{[5]}\mathbf{r}_{SSi}. \quad (15)$$

Solving for  $\dot{x}(t), \dot{y}(t), \dot{\varphi}_z(t), \dot{n}_1(t), \dot{n}_2(t), \dot{n}_3(t)$ , the three vector equations obtained by rearranging (15) to zero and expanding the coordinate velocities of the centre of mass of the robot can be expressed in the following form:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\varphi}_z(t) \end{bmatrix} = \mathbf{T}_{\text{robot}} \begin{bmatrix} \dot{\varphi}_{n1}(t) \\ \dot{\varphi}_{n2}(t) \\ \dot{\varphi}_{n3}(t) \end{bmatrix}. \quad (16)$$

Here,  $\mathbf{T}_{\text{robot}}$  is the pursued relationship between the angular velocities of the wheels and the coordinate velocities of the robot body.  $\mathbf{T}_{\text{robot}}$  is described in [Appendix 2](#).

## 2.2. Dynamics of the omnidirectional drivetrain

Former paper of the authors includes the dynamics of the omnidirectional drivetrain and can be found in [21]. In this paper, a different aim is given compared to [21]. In this section, the relationship between the wheel torques and the angular acceleration of the wheels is pursued. The Lagrangian approach is used to derive the equations of motion of the omnidirectional drivetrain. In order to derive the equations of motion, first the general coordinates and general coordinate velocities are determined:

$$\begin{aligned} \mathbf{q}(t) &= [\varphi_{n1}(t) \quad \varphi_{n2}(t) \quad \varphi_{n3}(t)]^T, \\ \dot{\mathbf{q}}(t) &= [\dot{\varphi}_{n1}(t) \quad \dot{\varphi}_{n2}(t) \quad \dot{\varphi}_{n3}(t)]^T. \end{aligned} \quad (17)$$

The Lagrangian approach to derive the equations of motion is based on the definition of the Lagrange function:

$$\mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) - U(\mathbf{q}(t), \dot{\mathbf{q}}(t)), \quad (18)$$

where  $T(\mathbf{q}(t), \dot{\mathbf{q}}(t))$  is the total kinetic energy, and  $U(\mathbf{q}(t), \dot{\mathbf{q}}(t))$  is the potential energy of the dynamic system. The variable  $\mathbf{q}(t)$  denotes the configuration of the system, which is the set of variables that can uniquely describe the position of the system at a given time. For an unconstrained system, Lagrange's equations of motion are derived from the Lagrange function as follows:

$$\frac{d}{dt} \frac{\partial \mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial \dot{\mathbf{q}}(t)} - \frac{\partial \mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial \mathbf{q}(t)} = \mathbf{Q}(t), \quad (19)$$

where  $\mathbf{Q}(t)$  is the vector of the generalized forces acting on the system.

Only the case in which the robot is moving freely on a flat surface is considered in this article. Thus, the potential energy is constant and expounding (19), the Lagrange's equations of motion will not include any portion of the potential energy function. Thus, let the Lagrange function be constructed solely from the kinetic energy terms:

$$\mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = T(\mathbf{q}(t), \dot{\mathbf{q}}(t)). \quad (20)$$

Although, formally, both the Robotino<sup>®</sup> and the MOGI Ethon consist of four rigid bodies, i.e. three driving blocks and a main body, only the kinetic energy of the main body is considered in this chapter.

The properties of the driving blocks will be detailed in [Subsection 2.4](#). The kinetic energy of the main body is

$$\begin{aligned} T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) &= \frac{1}{2} m_r {}^{[5]} \dot{\mathbf{S}}^2(t) \\ &+ \frac{1}{2} {}^{[5]} \dot{\varphi}^T(t) {}^{[5]} \Theta_{\text{body}} {}^{[5]} \dot{\varphi}(t), \end{aligned} \quad (21)$$

where  ${}^{[5]} \Theta_{\text{body}}$  is the inertia of the main body in  $\mathcal{F}_5$ :

$${}^{[5]} \Theta_{\text{body}} = m_r \cdot \text{diag} \left( \frac{h_r^2 + 3R_r^2}{12}, \frac{h_r^2 + 3R_r^2}{12}, \frac{R_r^2}{2} \right), \quad (22)$$

and  $m_r$  is the mass,  $h_r$  is the height and  $R_r$  is the radius of the robot.

Substituting (10), (11) and (22) into (21), the following yields for the kinetic energy of the robot:

$$\begin{aligned} T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) &= \frac{1}{2} m_r \dot{x}^2(t) \\ &+ \frac{1}{2} m_r \dot{y}^2(t) + \frac{1}{4} m_r R_r^2 \dot{\varphi}_z^2(t). \end{aligned} \quad (23)$$

Using (16) for substituting  $\dot{x}(t)$ ,  $\dot{y}(t)$ ,  $\dot{\varphi}_z(t)$  with the  $\dot{\mathbf{q}}(t)$  general coordinate velocities and performing the derivatives of (19), the Lagrange's equations of the second kind takes the form:

$$\mathbf{M} \ddot{\mathbf{q}}(t) = \mathbf{Q}(t), \quad (24)$$

where  $\mathbf{M} \in \mathbb{R}^{3 \times 3}$  is the mass matrix and  $\ddot{\mathbf{q}}(t) = (\ddot{\varphi}_{n1}(t) \quad \ddot{\varphi}_{n2}(t) \quad \ddot{\varphi}_{n3}(t))^T$  is the vector of the angular acceleration of the wheels, i.e. the general coordinate accelerations.  $\mathbf{M}$  is described in [Appendix 2](#). Furthermore,  $\mathbf{Q}(t)$  is the vector of the generalized forces, which are acting on the system, and can be produced using the principle of the virtual work [23,24]:

$$\mathbf{Q}(t) = \frac{\partial P(t)}{\partial \dot{\mathbf{q}}(t)}, \quad (25)$$

where  $P(t)$  is the power of the driving blocks in  $\mathcal{F}_5$ , i.e. the power of the DC motors, which are attached to the wheels and can be expressed as

$$\begin{aligned} P(t) &= \mathbf{R}_G \mathbf{R}_1 \begin{bmatrix} \tau_{n1}(t) \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{R}_G \mathbf{R}_1 \begin{bmatrix} \dot{\varphi}_{n1}(t) \\ 0 \\ 0 \end{bmatrix} + \\ &\mathbf{R}_G \mathbf{R}_2 \begin{bmatrix} \tau_{n2}(t) \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{R}_G \mathbf{R}_2 \begin{bmatrix} \dot{\varphi}_{n2}(t) \\ 0 \\ 0 \end{bmatrix} + \\ &\mathbf{R}_G \mathbf{R}_3 \begin{bmatrix} \tau_{n3}(t) \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{R}_G \mathbf{R}_3 \begin{bmatrix} \dot{\varphi}_{n3}(t) \\ 0 \\ 0 \end{bmatrix} \\ &= \sum_{i=1}^3 \tau_{ni}(t) \dot{\varphi}_{ni}(t), \end{aligned} \quad (26)$$

and  $\tau_{n1}$ ,  $\tau_{n2}$ ,  $\tau_{n3}$  are the torques of the wheels.

Combining (24), (25) and (26), the desired relationship is obtained as

$$\mathbf{M} \begin{bmatrix} \ddot{\varphi}_{n1}(t) \\ \ddot{\varphi}_{n2}(t) \\ \ddot{\varphi}_{n3}(t) \end{bmatrix} = \begin{bmatrix} \tau_{n1}(t) \\ \tau_{n2}(t) \\ \tau_{n3}(t) \end{bmatrix} \quad (27)$$

### 2.3. Modelling the driving blocks

After modelling the mechanical part of the drivetrain with the Lagrange’s method, the DC motors and coupled parts must be also modelled, because the DC motors are the power inputs of the system. The drive systems including the DC motors have been modelled together in the interest of unified view, however the motor model is well known in the literature [25,26].

The linear graph modelling method was used to model the dynamics of the driving blocks [25,27,28]. This method is based on the structural analysis of the physical entity, which is modelled. Figure 7 is a so-called functional outline, which is an extension of the assembly of the driving block (Figure 2), that shows all relevant parameters, inputs and outputs of the driving block. Parameters include passive elements and transducers also. Both the models of “MOGI Ethon” and the “Festo Robotino®” mobile robots are presented in this section. The two robots share the same omnidirectional drivetrain: one DC motor, one toothed belt drive and the omnidirectional wheel in each driving block.

The Festo Robotino® has one more stage attached between the belt drive and the wheel, namely a planetary gear. Based on Figure 7, the linear graph of the two robots can be produced as in Figure 8. Parameters of the graph are summarized and explained in Appendix 1. The flexibility of the toothed belt drive is not modelled here, i.e. the toothed belt is considered as an

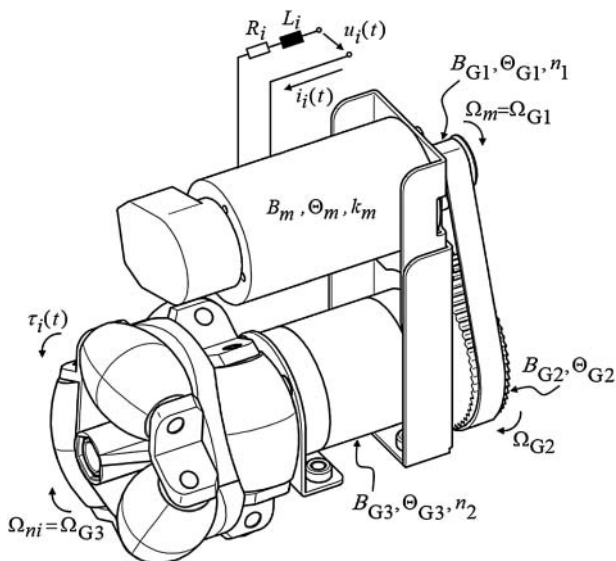


Figure 7. Functional outline of one of the driving blocks.

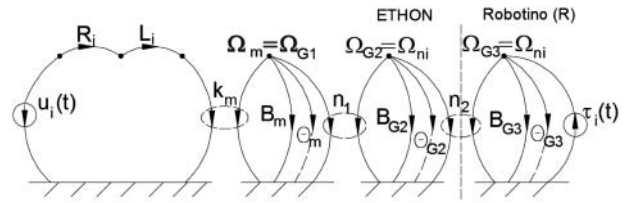


Figure 8. Linear graph of one of the driving blocks.

infinitely stiff, or a rigid body. It is considered that the force necessary for flexible deforming of the belt is significantly larger than the special force responsible for the torque causing slip between the wheels and the ground. In this case, the assumptions of Subsection 2.1 and thus the whole model would become invalid. Therefore, this case will need further investigation, which lies outside the scope of this paper. Further consequence is that the vibration tendency of the toothed belt drive is not relevant here. If the belt still vibrates, then it is considered as a manufacturing defect and the drivetrain should be reassembled with adequate tension applied on the belt.

Hereinafter, only the MOGI Ethon will be kept in focus. In case of MOGI Ethon, the linear graph can be reduced to one electrical part and one mechanical part (Figure 9).

Based on the reduced graph and with the use of the equations of the DC motor as an electromagnetic transducer described in [25] and in [28], the differential equations, which describe the dynamics of the driving blocks can be formulated.

In Figure 7, the constant  $k_m$  represents the relationship between the mechanical and electrical sides of the DC motor and symbolizes two different numbers, the relationship can be expanded as

$$\begin{aligned} \tau(t) &= k_{m,t}i(t), \\ u(t) &= k_{m,e}\Omega(t), \end{aligned} \quad (28)$$

where  $\tau(t)$  is the torque,  $i(t)$  is the current,  $u(t)$  is the voltage,  $\Omega(t)$  is the angular velocity of the motor,  $k_{m,t}$  is the torque constant and  $k_{m,e}$  is the speed constant (see Appendix 1).

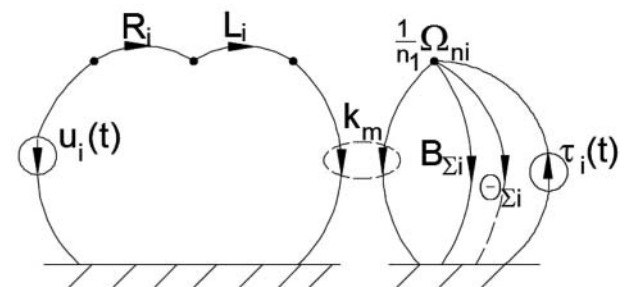


Figure 9. Reduced linear graph of the driving blocks of MOGI Ethon, parameters according to Table A1.

For every driving block:

$$\begin{aligned} u_i(t) &= R_i i_i(t) + L_i \frac{di_i(t)}{dt} + \frac{k_{m,ei}}{n_1} \Omega_{ni}(t) \\ -n_1 \tau_{ni}(t) &= \Theta_{\Sigma i} \frac{d\Omega_{ni}(t)}{dt} + B_{\Sigma i} \Omega_{ni}(t) - k_{m,ti} i_i(t). \end{aligned} \quad (29)$$

Since  $\Omega_{ni}(t)$  is the angular velocity of each wheel, it can also be expressed as  $\Omega_{ni}(t) = \frac{d\phi_{ni}(t)}{dt} = \dot{\phi}_{ni}(t)$ . The torque  $\tau_{ni}(t)$  is the active torque, which is trying to block the movement of each wheel; on the other hand, it is the torque that is required to accelerate the omnidirectional drivetrain in (27). Substituting  $\tau_{ni}(t)$  for  $i = 1, 2, 3$  from (27) into (29), the angular acceleration of the wheels and the current rates of the motors can be expressed.

The  $B_{\Sigma i}$  parameters represent the dissipation of the driving blocks. Currently viscous dampers are used for modelling the mechanical friction in the driving blocks. In case of necessity, the parameters of these viscous dampers can be substituted by special friction models and describing functions (e.g. Széll et al. [29]).

#### 2.4. The state-space model of the omnidirectional drivetrain

In this section, the state-space model is used for describing the complete electromechanical drive system including mechanical and electrical components since the state-space model is the basis of digital simulation. This formalism is widely used in control engineering since it is the basis of advanced control theory (e.g. state feedback, model-based control, predictive control etc.)

Considering the general form of the state-space model of a linear, time-invariant dynamic system [25,26,30]:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t), \end{aligned} \quad (30)$$

where  $\mathbf{x}(t)$  is a vector of the time-dependent state variables,  $\dot{\mathbf{x}}(t)$  is the first derivative of  $\mathbf{x}(t)$ .  $\mathbf{u}(t)$  is a vector of the system inputs.  $\mathbf{y}(t)$  is a vector composed from the required output variables of the system, i.e. output vector.  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  are constant coefficient matrices.

The state variables, the system inputs and outputs were chosen as

$$\mathbf{x}(t) = [\mathbf{q}^T(t) \quad \dot{\mathbf{q}}^T(t) \quad i_1(t) \quad i_2(t) \quad i_3(t)]^T, \quad (31)$$

$$\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad u_3(t)]^T, \quad (32)$$

and

$$\mathbf{y}(t) = [\mathbf{x}^T(t) \quad \dot{\xi}(t) \quad \dot{\eta}(t) \quad \dot{\phi}_z(t)]^T, \quad (33)$$

where  $\dot{\xi}(t)$  and  $\dot{\eta}(t)$  are the so-called coordinate velocities and  $\dot{\phi}_z(t)$  is the angular velocity of the robot body described in  $\mathcal{F}_4$ . Thus, the state-space equation of the omnidirectional drivetrain can be written in the form of (30), with the following coefficient matrices.

$\mathbf{A}$  is called the system matrix, or the state transition matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_5 & \mathbf{A}_6 \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_8 & \mathbf{A}_9 \end{bmatrix}, \quad (34)$$

where  $\mathbf{A}_5, \mathbf{A}_6 \in \mathbb{R}^{3 \times 3}$  describe the effect of the angular velocities and motor currents on the angular accelerations, respectively. The matrices  $\mathbf{A}_8 = \text{diag}(a_{74}, a_{85}, a_{96})$  and  $\mathbf{A}_9 = \text{diag}(a_{77}, a_{88}, a_{99})$  describe the effect of the angular velocities and motor currents on the derivative of the motor currents.  $\mathbf{A}$  is described in Appendix 2.

$\mathbf{B}$  is the input matrix:

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ \mathbf{B}_2 \end{bmatrix}, \quad (35)$$

where  $\mathbf{B}_2 = \text{diag}(\frac{1}{L_1}, \frac{1}{L_2}, \frac{1}{L_3})$  describes the effect of the input voltages of the motors on the derivative of the motor currents.

$\mathbf{C}$  is the output matrix:

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{C}_{11} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad (36)$$

where  $\mathbf{C}_{11} \in \mathbb{R}^{3 \times 3}$  describes the effect of the angular velocities on the system outputs.  $\mathbf{C}$  is described in Appendix 2.

$\mathbf{D}$  matrix describes the effect of the input voltages of the motors on the system outputs:

$$\mathbf{D} = \mathbf{0}_{12 \times 3} \quad (37)$$

### 3. Simulation

In this section, a simulation software developed by the authors is presented. This software is used to simulate the dynamics of the complete omnidirectional drivetrain. The simulation software of the complete omnidirectional drivetrain serves two purposes. First, it is used to validate and later to experiment with the model. Second, the simulation results and future measurements taken from the compatible on-board control system of the robot will be easily comparable using this software. The software was developed in the National Instruments LabVIEW<sup>TM</sup>, a graphical programming language with the Control Design and Simulation Toolkit



[31]. A complete description of the Control Design and Simulation Toolkit can be found in [32].

The architecture of the simulation software is a simple state machine, with a user interface event-handling structure. The software starts with an initialization phase, where all inputs and outputs are initialized to their default values. Parameters and their values (see Appendix 1) are stored in a comma-separated spreadsheet file, which is read during initialization. The software can handle several different parameter files. Switching between parameter files can be done in runtime via a selector. After initialization, the software switches to a continuously running loop, where the user event-handling structure is operated.

The user interface is ordered by pages, where each page displays different simulation results of the system. The first and starting page is the *Model construction*, where basic model-related details are presented, i.e. the model parameters and numerical values, the parameter file selector, states, inputs and outputs of the system and the numerical representation of the model. The representation of the model is also adjustable; the user can switch between state-space, transfer function or even zero-pole-gain representation.

The software waits until the *Start* button is pressed and then creates the numerical state-space model based on Subsection 2.4 (see Figure 10).

The controllability and observability matrices are calculated automatically and boolean displays informing the user whether or not the system is controllable and observable. The zero-pole map of the system is also generated automatically, along with the indication of the number of zeros and poles and stability of the system.

Once the simulation software is running and the model is created, the user can choose what the software should calculate. Actual calculations take place once the user clicks on one of the labels of the page selector. The software is able to calculate the dynamic responses of the system in the time domain on the *Dynamic*

*Responses* page, both step and impulse responses can be calculated and presented. LabVIEW™ Control Design and Simulation Toolkit calculates the step and impulse responses of a multiple input–multiple output dynamic system (MIMO) in such way that the step or impulse excitation is applied only on one input and the others are kept at zero, for all inputs separately. Therefore, the effect of each input on each state and on each output can be presented on separate graphs.

On the *Frequency Domain* page, the Bode plot (magnitude and phase diagrams) and the Nyquist plot (root locus) of the system are calculated and presented for all states and outputs.

Up to this point, the state-space model presented and evaluated in the previous sections was a continuous model. On the *Sampled Model* page, the model can be transformed into a discrete model. The user can choose between different sampling methods, i.e. zero-order-hold, Tustin approximation or Z-transformation among others, and the sampling time is also adjustable. The coefficients of the discrete model are immediately recalculated if the sampling method or the sampling time is changed by the user. The discrete model will be used to design and study the effect of different discrete controllers.

The last page of the simulation software is labelled *Continuous Simulation*. On this page, arbitrary excitations can be given onto the inputs of the model and the responses are calculated immediately. This is made by a separate, continuously running task in the background, using the so-called producer–consumer architecture.

This architecture takes advantage of the multi-core processing, where at least two independent tasks are running in parallel and data between the tasks is sent using inter-process communication techniques. LabVIEW™ supports the multi-core processing by offering a great variety of libraries and functions. The producer task (the user interface event handler) produces data (i.e. input values) and commands (i.e. start or stop) for the consumer task, where the received commands and data are processed (i.e. the simulation is running with new parameters). The continuous simulation can be stopped and restarted at will. If the continuous simulation is stopped, the task is not terminated; it just becomes idle and waits for further user interaction (i.e. restart or termination).

$$\begin{aligned} \frac{dx}{dt} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -33,2259 & -2,72844 & -2,72844 & 0,737616 & 0,0605713 & 0,0605713 & 0 \\ 0 & 0 & -2,72844 & -33,2259 & -2,72844 & 0,0605713 & 0,737616 & 0,0605713 & 0 \\ 0 & 0 & -2,72844 & -2,72844 & -33,2259 & 0,0605713 & 0,0605713 & 0,737616 & 0 \\ 0 & 0 & -862,023 & 0 & 0 & -8457,71 & 0 & 0 & 0 \\ 0 & 0 & 0 & -862,023 & 0 & 0 & -8457,71 & 0 & 0 \\ 0 & 0 & 0 & 0 & -862,023 & 0 & 0 & -8457,71 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4975,12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4975,12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4975,12 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} u(t) \\ \\ v(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u(t) \\ \\ & \begin{bmatrix} 0 & 0 & -0,0294449 & 0 & 0,0294449 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,017 & -0,034 & 0,017 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,0578231 & 0,0578231 & 0,0578231 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Figure 10. The numerical representation of the state-space model.

### 3.1. Simulation results

Simulations were done with nominal parameters (see Appendix 1). Distance and mass properties were adopted from the CAD model of the MOGI Ethon and the parameters of the motors were adopted from the datasheet. Although there could be differences between the CAD model and the actual sizes, furthermore, the

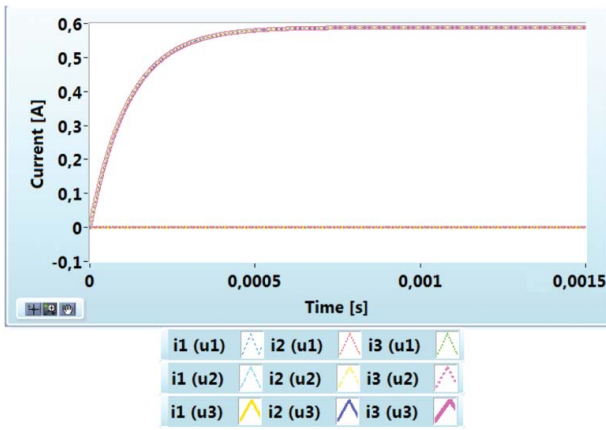


Figure 11. Step response graph of the motor currents.

motors could differ from each other and from the datasheet also.

The aim of this simulation is to confirm the model. Further refinements will be needed to relate the model to the actual robot. This stands especially for the dissipation. There were no measurements carried out to validate the model quantitatively yet. The model is validated qualitatively based on the following empirical considerations and requirements:

- In  $\mathcal{F}_4$ , the driving block M2 should not produce any effect on  $\dot{\xi}(t)$  as  $\dot{\xi}(t)$  is produced only by the driving blocks M1 and M3.
- The rotational coordinate velocity  $\dot{\varphi}_z(t) = \omega(t)$  is produced with equal proportions by all driving blocks.

Basic results indicate that the developed model is controllable and observable.

Simulation results in the time domain, in case of a step response, confirm the two orders of magnitude difference between the time constants of the mechanical and electrical parts (see Figures 11 and 12).

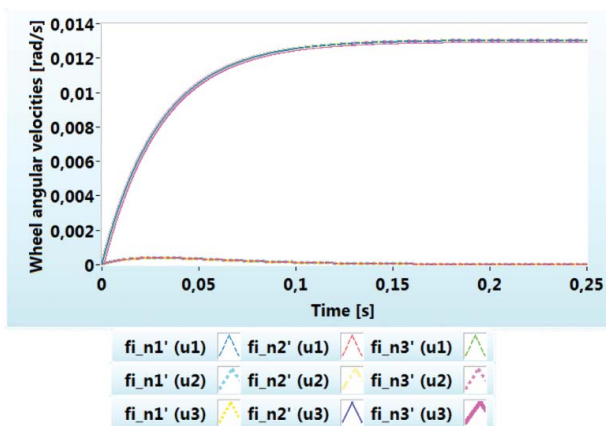


Figure 12. Step response graph of the angular velocities of the wheels.

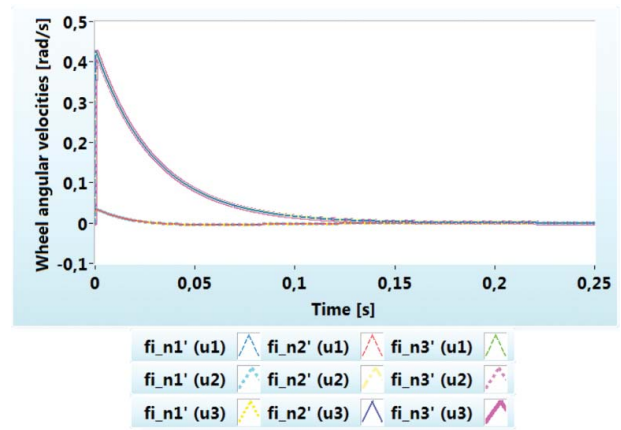


Figure 13. Impulse response graph of the angular velocities of the wheels.

This difference is commonly known and often results in neglecting the electrical part of the DC motor model. In this paper, the electrical parts of the DC motors will be kept for further research.

In case of impulse response simulation, the results show the same difference between the time constants of the mechanical and electrical parts. A so-called cross effect can also be observed (Figure 13), where the angular velocity of one motor changes while the corresponding control voltage is kept at zero level. This effect is caused by the omnidirectional drivetrain itself. The magnitude of this cross effect is about 10% of the main effects (angular velocity change caused by the corresponding control voltage change), that seems negligible, but a precise motion control system needs to be designed to calculate this effect. This effect can also be observed on Figure 12, in case of step response simulation.

Simulating the system with equal corresponding parameters of each driving block (see Appendix 1) leads to a completely symmetric drivetrain. In Figures 12 and 13, the graphs of the main effects run together, and the graphs of the cross effects run together also. Although the drivetrain is symmetric (neglecting the manufacturing inaccuracies), measuring the actual motors may lead to different numeric values of the corresponding parameters; thus, the model would become asymmetric.

Therefore, the use-case where the motors are individually controlled by a position or speed controller with identical control parameters is not the best practice, not even when the controllers are tuned individually.

Finally, the outputs of the system can be seen in Figure 14, where the coordinate velocities are shown in case of step response simulation.

The coordinate velocities are defined in  $\mathcal{F}_4$  coordinate system (see Figure 5). Figure 14 demonstrates the following results. In Figure 14,  $v_{xi}(u2)$  curve is at con-

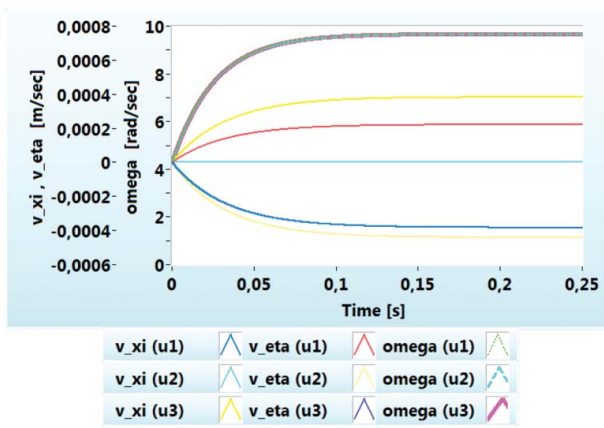


Figure 14. Coordinate velocities of the robot.

stant zero level ( $u_2$  is the control voltage for M2 driving block and  $v_{xi}(t)$  denotes  $\dot{\zeta}(t)$ ). Hence,  $\dot{\zeta}(t)$  is produced by the driving blocks M1 and M3 only. In Figure 14, curves named  $\omega(u_1)$ ,  $\omega(u_2)$ ,  $\omega(u_3)$  overlap each other completely; thus, the rotational coordinate velocity is produced in equal proportions by all driving blocks ( $\omega(t)$  denotes  $\dot{\varphi}_z(t)$  in the simulation). Thus, the simulation confirms the model qualitatively based on the empirical considerations and requirements defined earlier is this subsection.

#### 4. Conclusions

In this paper, the development of the state-space model of an omnidirectional drivetrain (Kiwi drive) for mobile robots was presented. Two holonomic mobile robots were used as examples, the Festo Robotino<sup>®</sup> and the MOGI Ethon. Detailed mathematical description was given with a coherent system of symbols for both the kinematics and dynamics of the omnidirectional drivetrain. Results from the kinematics and dynamics were then used to formulate the state-space model including the DC motors and associated powertrain elements.

The model is particularly useful when the mobile robot is rolling on a flat surface. The model includes basic dissipative elements, which can be replaced with more accurate dissipation models if needed.

Self-developed simulation software was also presented for numerical evaluation of the developed model. The simulation software was developed in LabVIEW<sup>™</sup> graphical programming language, and uses built-in functions to simulate the model. The impact of the change of each model parameter can be evaluated with this software. The software gives a good base to evaluate different control algorithms.

Basic simulation results confirm the model, but further evaluation is needed to improve the accuracy of the model describing actual mobile robots. Measurements for quantitative verification of the model are in progress.

The long-term aim is to give new, more precise motion control systems for mobile robots using omnidirectional drivetrains. Future plans are to implement and evaluate different control algorithms based on the model in real-time software environment.

With a real-time control system, the electric torque of the motors could be controlled via current control; thus, current consumption could be optimized resulting in maximization of battery lifetime of the robots.

Final results can also be used in higher education and research in the field of mobile robotics in order to develop professional precision control systems for position control of robots.

#### Acknowledgments

The authors would like to express their gratitude to Dr Antal Huba, Dr Csaba Budai and Ágnes Urbin of the Department of Mechatronics, Optics and Engineering Informatics of the Budapest University of Technology and Economics, for their contribution to this paper.


#### Disclosure statement


No potential conflict of interest was reported by the authors.

#### Funding

This work was supported by the National Research, Development and Innovation Office – NKFIH [grant number K-120501].

#### ORCID

Levente Raj  <http://orcid.org/0000-0002-7043-3878>

András Czmerk  <http://orcid.org/0000-0003-2032-7687>

#### References

- [1] El-Shenawy A, Wellenreuther A, Baumgart AS, et al. Comparing different holonomic mobile robots. IEEE International Conference on Systems, Man and Cybernetics, 2007 Oct 7–10; Montreal, QC; Montreal (QC): ISIC; 2007.
- [2] Batlle J, Barjau A. Holonomy in mobile robots. *Robotics Auton Syst.* 2008;57:433–440.
- [3] Gfrerrer A. Geometry and kinematics of the Mecanum wheel. *Comput Aided Geom Des.* 2008;25:784–791.
- [4] Saha SK, Angeles J, Darcovich J. The design of kinematically isotropic rolling robots with omnidirectional wheels. *Mech Mach Theory.* 1995;30(8):1127–1137.
- [5] Weber R-C, Bellenberg M. Robotino<sup>®</sup> manual. Esslingen (Germany): Festo Didactic GmbH & Co. KG; 2010.
- [6] Li L, Jiang S, Dai F, et al. Dynamic model and balance control of two-wheeled robot with non-holonomic constraints. *Proceeding of the 11th World Congress on Intelligent Control and Automation*; 2014 Jun 27–30; Shenyang, China. New York (NY): IEEE; 2014.
- [7] Martins FN, Celeste WC, Carelli R, et al. An adaptive dynamic controller for autonomous mobile robot

- trajectory tracking. *Control Eng Pract.* 2008;16:1354–1363.
- [8] Kumbasar T, Hagnas H. A Type-2 fuzzy cascade control architecture for mobile robots. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics; Manchester, United Kingdom. New York (NY): IEEE; 2013. p. 3226–3231.
- [9] Safar MJA, Watanabe K, Maeyama S, et al. Tip-over stability control for a holonomic omnidirectional mobile robot with active dual-wheel caster assemblies using SGCMG. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2013 Nov 3–7; Tokyo, Japan. New York (NY): IEEE; 2013.
- [10] Kuenemund F, Heß D, Wißing M, et al. Online kinodynamic motion planning for omnidirectional automatic guided vehicles. 2014 13th International Conference on Control, Automation, Robotics & Vision Marina Bay Sands; 2014 Dec 10–12; Singapore. 2014. (ICARCV 2014) Fr43.3, New York (NY): IEEE; 2014.
- [11] Martínez-García EA, Córdoba RT, Martínez-Villafane A, et al. Directional fields algebraic non-linear solution equations for mobile robot planning. *Appl Math Model.* 2014;38(21–22):5298–5314.
- [12] Purwin O, D’Andrea R. Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics Autonom Syst.* 2006;54:13–22.
- [13] Hashemi E, Jadidi MG, Jadidi NG. Model-based PI-fuzzy control of four-wheeled omni-directional mobile robots. *Robotics Autonom Syst.* 2011;59:930–942.
- [14] Wada M. Virtual link model for redundantly actuated holonomic omnidirectional mobile robots. Proceedings of the 2006 IEEE International Conference on Robotics and Automation; 2006 May 15–19; Orlando, FL. New York (NY): IEEE; 2006.
- [15] Wada M, Torii R. Cooperative transportation of a single object by omnidirectional robots using potential method. 2013 16th International Conference on Advanced Robotics (ICAR); 2013 Nov 24–29; Montevideo, Uruguay. New York (NY): IEEE; 2013.
- [16] Tajti F, Szayer G, Kovács B, et al. Mobile robot performance analysis for indoor robotics. *Period Polytech Civil Eng.* 2015;59(2):123–131.
- [17] Inthiam J, Deelertpaiboon C. Self-localization and navigation of holonomic mobile robot using omni-directional wheel. TENCON 2014 – 2014 IEEE Region 10 Conference; 2014 Oct 22–25; Bangkok, Thailand. New York (NY): IEEE; 2014.
- [18] Cristancho CA, Florez CA, Rodriguez HL. Holonomic platform and six DOF manipulator robot through virtual. Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC); 2011 Oct 1–4; Bogotá, Colombia; New York (NY): IEEE; 2011.
- [19] Kim KS, Kwok AS, Thomas GC, et al. Fully omnidirectional compliance in mobile robots via drive-torque sensor feedback. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014); 2014 Sep 14–18; Chicago, IL; New York (NY): IEEE; 2014.
- [20] Liu Y, Zhu JJ, Williams RL II, et al. Omni-directional mobile robot controller based on trajectory linearization. *Robotics Autonom Syst.* 2008;56:461–479.
- [21] Raj L, Devecseri V, Fekete RT. State space model of an omni-directional holonomic mobile robot. XV International PhD Workshop OWD 2013; 2013 Oct 19–22; Wisła, Poland. Conference Archives PTETiS, Vol. 33, p. 472–477.
- [22] Szabó C, Róka A, Faragó T, et al. Building a human-dog interaction inspired emotional engine model. IECON 2012 – 38th Annual Conference on IEEE Industrial Electronics Society; 2012 Oct 25–28; Montreal, Canada. New York (NY): IEEE; 2012.
- [23] Béda G, Bezák A. Kinematika és dinamika. Budapest: Tankönyvkiadó; 1991. Hungarian.
- [24] Béda G, Stépán G. Analitikus mechanika [Analytical mechanics]. Budapest: Tankönyvkiadó; 1989. Hungarian.
- [25] Huba A, Aradi P, Czmerk A, et al. Mechatronikai berendezések tervezése [Designing of mechatronic equipment]. Budapest: Budapest University of Technology and Economics, Faculty of Mechanical Engineering, Department of Mechatronics, Optics and Mechanical Engineering Informatics; 2014. Hungarian.
- [26] Lantos B. Irányítási rendszerek elmélete és tervezése I. [Theory and design of control systems I]. Budapest: Akadémiai kiadó; 2001. Hungarian.
- [27] Rowell D. Linear graph modeling: one-port elements. Cambridge (MA): Massachusetts Institute of Technology Department of Mechanical Engineering; 2004.
- [28] Rowell D. Linear graph modeling: two-port energy transducing elements. Cambridge (MA): Massachusetts Institute of Technology Department of Mechanical Engineering; 2004.
- [29] Széll K, Czmerk A, Korondi P. Friction with hysteresis loop modeled by tensor product. *Automatika.* 2014;55(4):463–473.
- [30] Rowell D. State-space representation of LTI systems. Massachusetts Institute of Technology Department of Mechanical Engineering; 2004.
- [31] National Instruments Corporation. LabVIEW (TM) control design user manual. Austin (TX): National Instruments Corporation; 2009.
- [32] National Instruments Corporation. LabVIEW (TM) control design and simulation module algorithm references. Austin (TX): National Instruments Corporation; 2009.



## Appendices

### Appendix 1. Nomenclature

Table A1. List of parameters.

Parameter	Unit	Value	Description
$l$	[m]	0.294	Radius of the pitch circle of the wheels
$\beta$	[°]	60	Half angle between every two shaft of the wheels
$r$	[m]	0.051	Radius of the wheels
$m_r$	[kg]	40	Mass of the robot
$R_r$	[m]	0.270	Radius of the robot
$h_r$	[m]	0.800	Height of the robot
$m_{\text{wheel}}$	[kg]	0.238	Mass of the wheel
$g$	$\frac{m}{s^2}$	9.81	Gravitational acceleration
$L_1$	[H]	0.000201	Inductance of the motors
$L_2$		0.000201	
$L_3$		0.000201	
$R_1$	[Ω]	1.7	Resistance of the motors
$R_2$		1.7	
$R_3$		1.7	
$k_{m,t1}$	$\frac{Nm}{A}$	0.0222	Torque constants of the motors
$k_{m,t2}$		0.0222	
$k_{m,t3}$		0.0222	
$k_{m,e1}$	$\frac{V}{rpm}$	0.0023	Speed constants of the motors
$k_{m,e2}$		0.0023	
$k_{m,e3}$		0.0023	
$n_1$	[1]	0.127	Gear ratio of the toothed belt
$n_2$			Gear ratio of the planetary gear (Robotino® only)
$\Theta_m$	[kgm <sup>2</sup> ]		
Inertia of the motor			
$\Theta_{G2}$			Inertia of the belt drive
$\Theta_{G3}$			Inertia of the planetary gear
$\Theta_{\Sigma1}$	0.024		Resultant inertia of the driving blocks
$\Theta_{\Sigma2}$	0.024		$\Theta_{\Sigma i} = \Theta_m + \frac{1}{n_i^2} \Theta_{G2}$
$\Theta_{\Sigma3}$	0.024		
$B_m$	$\frac{kgm^2}{s}$		Attenuation of the motor
$B_{G2}$			Attenuation of the belt drive
$B_{G3}$			Attenuation of the planetary gear
$B_{\Sigma1}$	1		Resultant attenuation of the driving blocks
$B_{\Sigma2}$	1		$B_{\Sigma i} = B_m + \frac{1}{n_i^2} B_{G2}$
$B_{\Sigma3}$	1		
$\Omega_m$	$\frac{rad}{s}$		Angular velocity of the motor
$\Omega_{Gi}$			Angular velocity of the small wheel of the toothed belt drive of each driving block

### Appendix 2. Description of matrices

$$\mathbf{T}_{\text{robot}} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}, \quad (\text{B1})$$

The elements of  $\mathbf{T}_{\text{robot}}$  are detailed in the *Supplementary file*.

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}, \quad (\text{B2})$$

The elements of  $\mathbf{M}$  are detailed in the *Supplementary file*.

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} & \\ a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & \\ \mathbf{0}_{3 \times 3} & a_{54} & a_{55} & a_{56} & a_{56} & a_{58} & a_{59} \\ a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & \\ a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} & \\ a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & a_{\text{den}} & \\ a_{74} & 0 & 0 & a_{77} & 0 & 0 & \\ \mathbf{0}_{3 \times 3} & 0 & a_{85} & 0 & 0 & a_{88} & 0 \\ 0 & 0 & a_{96} & 0 & 0 & a_{99} & \end{bmatrix}, \quad (\text{B3})$$

The elements of  $\mathbf{A}$  are detailed in the *Supplementary file*.

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & c_{104} & 0 & c_{106} \\ \mathbf{0}_{3 \times 3} & c_{114} & c_{115} & c_{116} & \mathbf{0}_{3 \times 3} \\ & c_{124} & c_{125} & c_{126} & \end{bmatrix}, \quad (\text{B4})$$

The elements of  $\mathbf{C}$  are detailed in the *Supplementary file*.