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

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: 338

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

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® 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.
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 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 1. The Festo Robotino®.

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].

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 ($F_i$, where $i = 1, 2, 3$), one coordinate system to each wheel (see Figure 5). In the coordinate systems ($F_i$), 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.

The axial velocity of the centre of mass of the wheels (see Figure 2).

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).

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

$$[i] \dot{\psi}_i(t) = \begin{bmatrix} \dot{\psi}_{ni}(t) & 0 & 0 \end{bmatrix}_{\text{ntoht}},$$  \hspace{1cm} (1)

where $\dot{\psi}_{ni}(t)$ 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:

$$[i] v_{pi}(t) = [\dot{n}_i(t) \ 0 \ 0]_{\text{ntoht}},$$  \hspace{1cm} (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

$$[i] \dot{S}_i(t) = [i] v_{pi} + [i] \dot{\psi}_i(t) \times [i] r_w,$$  \hspace{1cm} (3)

where $[i] r_w$ is the vectorial form of the radius of the wheels and $r$ is the radius of the wheels:

$$[i] r_w = \begin{bmatrix} 0 & 0 & r \end{bmatrix}_{\text{ntoht}},$$  \hspace{1cm} (4)

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

$$[i] \dot{S}_i(t) = [\dot{n}_i(t) \ -r\dot{\psi}_{ni}(t) \ 0]_{\text{ntoht}},$$  \hspace{1cm} (5)

Based on the assembly detailed in Figure 3, a functional outline of the idealized omnidirectional drive
The \( R_1 \) and \( R_3 \) matrices are simple rotation matrices around the \( b_1 \) axis of the coordinate systems of the respective wheels by a constant \( \beta \) angle.

Finally, the transformation between \( F_4 \) and \( F_5 \) was considered as

\[
R_G = \begin{bmatrix}
\cos(\varphi_x(t)) & -\sin(\varphi_x(t)) & 0 \\
\sin(\varphi_x(t)) & \cos(\varphi_x(t)) & 0 \\
0 & 0 & 1
\end{bmatrix},
\]

where \( \varphi_x(t) \) is the angle between \( \zeta \) and \( y \), hence the angle between \( F_4 \) and \( F_5 \). 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 \( i = 1, 2, 3 \):

\[
[5]\dot{S}_i(t) = [5]\dot{S}_i(t) + [5]\dot{\varphi}(t) \times [5]r_{SSI},
\]

The vectors \( [5]\dot{S}_i(t) \) are the velocities of the wheels described in \( F_5 \).

\( [5]\dot{S}_i(t) \) is the velocity of the robot body described in \( F_5 \), in the following form:

\[
[5]\dot{S}_i(t) = \begin{bmatrix}
\dot{x}(t) \\
\dot{y}(t) \\
\dot{z}(t)
\end{bmatrix}_{xyz},
\]

\[
[5]\dot{\varphi}(t) = \begin{bmatrix}
0 \\
0 \\
\dot{\varphi}_z(t)
\end{bmatrix}_{xyz} + R_G \sum_{i=1}^{3} R_{i[5]}[5]\dot{\varphi}_i(t),
\]

and \( [5]r_{SSI} \) are the vectors pointing from the centre of mass of the robot to the centre of mass of each wheel:

\[
[5]r_{SSI} = R_G[\begin{bmatrix}
\cos(\beta) \\
\sin(\beta) \\
0
\end{bmatrix}]^T,
\]

\[
[5]r_{SSI} = R_G[\begin{bmatrix}
-1 \\
0 \\
0
\end{bmatrix}]^T,
\]

\[
[5]r_{SSI} = R_G[\begin{bmatrix}
\cos(\beta) \\
-l \sin(\beta) \\
0
\end{bmatrix}]^T,
\]

where \( l \) is the radius of the pitch circle of the wheels.

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

\[
R_G R_i [5]\dot{S}_i(t) = [5]\dot{S}_i(t) + [5]\dot{\varphi}(t) \times [5]r_{SSI}.
\]

Solving for \( \dot{x}(t), \dot{y}(t), \dot{z}(t), \dot{n}_1(t), \dot{n}_2(t), \dot{n}_3(t) \), the three vector equations obtained by rearranging (15) to zero and expounding the coordinate velocities of the centre of mass of the robot can be expressed in the following form:
\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{y}(t) \\
\dot{\phi}_2(t)
\end{bmatrix} = \mathbf{T}_\text{robot} \begin{bmatrix}
\dot{\psi}_{n1}(t) \\
\dot{\psi}_{n2}(t) \\
\dot{\psi}_{n3}(t)
\end{bmatrix},
\] (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:

\[
\mathbf{q}(t) = \begin{bmatrix}
\psi_{n1}(t) \\
\psi_{n2}(t) \\
\psi_{n3}(t)
\end{bmatrix}^T,
\]

\[
\dot{\mathbf{q}}(t) = \begin{bmatrix}
\dot{\psi}_{n1}(t) \\
\dot{\psi}_{n2}(t) \\
\dot{\psi}_{n3}(t)
\end{bmatrix}^T.
\]

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

\[
\mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) - U(\mathbf{q}(t), \dot{\mathbf{q}}(t)),
\] (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:

\[
\frac{d}{dt} \frac{\partial \mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial \dot{\mathbf{q}}(t)} - \frac{\partial \mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t))}{\partial \mathbf{q}(t)} = \mathbf{Q}(t),
\]

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:

\[
\mathcal{L}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = T(\mathbf{q}(t), \dot{\mathbf{q}}(t)).
\]

(20)

Although, formally, both the Robotino® 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

\[
T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \frac{1}{2} m_r [\dot{\mathbf{q}}(t)]^T \mathbf{\dot{q}}(t) + \frac{1}{2} [\dot{\mathbf{q}}(t)]^T \mathbf{\dot{q}}(t),
\]

(21)

where \( \Theta_{\text{body}} \) is the inertia of the main body in \( \mathcal{F}_S \):

\[
\Theta_{\text{body}} = m_r \begin{pmatrix}
\frac{1}{12} R_1^2 & \frac{1}{12} R_1^2 & \frac{1}{12} R_1^2 \\
-\frac{1}{12} R_1^2 & \frac{1}{12} R_1^2 & \frac{1}{12} R_1^2 \\
-\frac{1}{12} R_1^2 & -\frac{1}{12} R_1^2 & \frac{1}{12} R_1^2
\end{pmatrix},
\]

(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:

\[
T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \frac{1}{2} m_r \dot{\mathbf{q}}(t)^T \dot{\mathbf{q}}(t) + \frac{1}{2} h_r^2 \mathbf{q}(t)^T \mathbf{\dot{q}}(t) + \frac{1}{4} m_r R_r^2 \dot{\mathbf{q}}(t)^T \dot{\mathbf{q}}(t).
\]

(23)

Using (16) for substituting \( \dot{x}(t) \), \( \dot{y}(t) \), \( \dot{\phi}_2(t) \) 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:

\[
\mathbf{M} \ddot{\mathbf{q}}(t) = \mathbf{Q}(t),
\]

(24)

where \( \mathbf{M} \in \mathbb{R}^{3 \times 3} \) is the mass matrix and \( \dot{\mathbf{q}}(t) = (\dot{\psi}_{n1}(t) \ \dot{\psi}_{n2}(t) \ \dot{\psi}_{n3}(t))^T \) 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]:

\[
\mathbf{Q}(t) = \frac{\partial \mathbf{P}(t)}{\partial \dot{\mathbf{q}}(t)},
\]

(25)

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

\[
\mathbf{P}(t) = \begin{bmatrix}
\tau_{n1} \\
\tau_{n2} \\
\tau_{n3}
\end{bmatrix} = \begin{bmatrix}
\mathbf{R}_1 & 0 & 0 \\
0 & \mathbf{R}_2 & \mathbf{R}_3 \\
0 & 0 & \mathbf{R}_3
\end{bmatrix} \begin{bmatrix}
\dot{\psi}_{n1}(t) \\
\dot{\psi}_{n2}(t) \\
\dot{\psi}_{n3}(t)
\end{bmatrix} + \begin{bmatrix}
\dot{\psi}_{n1}(t) \\
\dot{\psi}_{n2}(t) \\
\dot{\psi}_{n3}(t)
\end{bmatrix}
\]

(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

\[
M \begin{bmatrix} \dot{\omega}_{n1}(t) \\ \dot{\omega}_{n2}(t) \\ \dot{\omega}_{n3}(t) \end{bmatrix} = \begin{bmatrix} \tau_{n1}(t) \\ \tau_{n2}(t) \\ \tau_{n3}(t) \end{bmatrix}
\]  

(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 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

\[
\tau(t) = k_{m,t} i(t),
\]

\[
u(t) = k_{m,e} \Omega(t),
\]

(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 7. Functional outline of one of the driving blocks.](image)

![Figure 8. Linear graph of one of the driving blocks.](image)

![Figure 9. Reduced linear graph of the driving blocks of MOGI Ethon, parameters according to Table A1.](image)
For every driving block:

\[ u_i(t) = R_i i_i(t) + L_i \frac{di_i(t)}{dt} + \frac{k_m}{n_i} \Omega_m(t) \]

\[ -n_i \tau_{mi}(t) = \Theta_{mi} \frac{d\Omega_m(t)}{dt} + B_{\Sigma} \Omega_m(t) - k_{m,ni} i_i(t). \]  

(29)

Since \( \Omega_m(t) \) is the angular velocity of each wheel, it can also be expressed as \( \Omega_m(t) = \frac{\dot{\phi}_m(t)}{q_i} = \dot{\phi}_m(t) \). The torque \( \tau_{mi}(t) \) 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 \( \tau_{mi}(t) \) 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 \( B_{\Sigma} \) 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]:

\[
\dot{x}(t) = A \cdot x(t) + B \cdot u(t),
\]

\[
y(t) = C \cdot x(t) + D \cdot u(t),
\]

(30)

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

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

\[
x(t) = \begin{bmatrix} q_i^T (t) & \dot{q}_i^T (t) & i_1(t) & i_2(t) & i_3(t) \end{bmatrix}^T,
\]

\[
u(t) = \begin{bmatrix} i_{u1}(t) & u_2(t) & u_3(t) \end{bmatrix}^T,
\]

and

\[
y(t) = \begin{bmatrix} x^T (t) & \dot{z}(t) & \dot{y}(t) & \dot{w}(t) \end{bmatrix}^T,
\]

(31) (32) (33)

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

\[
A = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & A_3 & A_6 \\ 0_{3 \times 3} & A_8 & A_9 \end{bmatrix},
\]

(34)

where \( A_3, A_6 \in \mathbb{R}^{3 \times 3} \) describe the effect of the angular velocities and motor currents on the angular accelerations, respectively. The matrices \( A_8 = \text{diag}(a_{77}, a_{85}, a_{96}) \) and \( A_9 = \text{diag}(a_{77}, a_{85}, a_{99}) \) describe the effect of the angular velocities and motor currents on the derivative of the motor currents. \( A_5 \) is described in Appendix 2.

\[
B = \begin{bmatrix} 0_{6 \times 3} \\ B_2 \end{bmatrix},
\]

(35)

where \( B_2 = \text{diag}\left( \frac{1}{L_1}, \frac{1}{L_2}, \frac{1}{L_3} \right) \) describes the effect of the input voltages of the motors on the derivative of the motor currents.

\[
C = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & C_{11} & 0_{3 \times 3} \end{bmatrix},
\]

(36)

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

\[
D = 0_{12 \times 3},
\]

(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™, a graphical programming language with the Control Design and Simulation Toolkit.
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).

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
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 $M_2$ should not produce any effect on $\dot{\xi}(t)$ as $\dot{\xi}(t)$ is produced only by the driving blocks $M_1$ and $M_3$.
- The rotational coordinate velocity $\dot{\varphi}_2(t) = \omega(t)$ 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).

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, $v_{x1}(u2)$ curve is at con-
stant zero level ($u_2$ is the control voltage for M2 driving block and $v_{x\text{-}M2}$ 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() denotes $\dot{\varphi_2}(t)$ 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® 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 power-train 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™ graphical programing 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


Figure 14. Coordinate velocities of the robot.


Rowell D. State-space representation of LTI systems. Massachusetts Institute of Technology Department of Mechanical Engineering; 2004.


Appendices

Appendix 1. Nomenclature

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>[m]</td>
<td>0.294</td>
<td>Radius of the pitch circle of the wheels</td>
</tr>
<tr>
<td>( \beta )</td>
<td>[°]</td>
<td>60</td>
<td>Half angle between every two shaft of the wheels</td>
</tr>
<tr>
<td>( r )</td>
<td>[m]</td>
<td>0.051</td>
<td>Radius of the wheels</td>
</tr>
<tr>
<td>( m_r )</td>
<td>[kg]</td>
<td>40</td>
<td>Mass of the robot</td>
</tr>
<tr>
<td>( R_r )</td>
<td>[m]</td>
<td>0.270</td>
<td>Radius of the robot</td>
</tr>
<tr>
<td>( h_r )</td>
<td>[m]</td>
<td>0.800</td>
<td>Height of the robot</td>
</tr>
<tr>
<td>( m_{\text{wheel}} )</td>
<td>[kg]</td>
<td>0.238</td>
<td>Mass of the wheel</td>
</tr>
<tr>
<td>( G )</td>
<td>[kgm²]</td>
<td>9.81</td>
<td>Gravitational acceleration</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>[H]</td>
<td>0.000201</td>
<td>Inductance of the motors</td>
</tr>
<tr>
<td>( L_2 )</td>
<td></td>
<td>0.000201</td>
<td></td>
</tr>
<tr>
<td>( R_1 )</td>
<td>[Ω]</td>
<td>1.7</td>
<td>Resistance of the motors</td>
</tr>
<tr>
<td>( R_3 )</td>
<td></td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td>( k_{m,r1} )</td>
<td>[Nm/°]</td>
<td>0.0222</td>
<td>Torque constants of the motors</td>
</tr>
<tr>
<td>( k_{m,r2} )</td>
<td></td>
<td>0.0222</td>
<td></td>
</tr>
<tr>
<td>( k_{m,r3} )</td>
<td>[Nm/°]</td>
<td>0.0222</td>
<td>Speed constants of the motors</td>
</tr>
<tr>
<td>( k_{m,r4} )</td>
<td></td>
<td>0.0023</td>
<td></td>
</tr>
<tr>
<td>( n_t )</td>
<td>[1]</td>
<td>0.127</td>
<td>Gear ratio of the toothed belt</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>[1]</td>
<td>0.127</td>
<td>Gear ratio of the planetary gear (Robotino® only)</td>
</tr>
<tr>
<td>( \Theta_{\text{in}} )</td>
<td>[kgm²]</td>
<td>0.294</td>
<td>Inertia of the motor</td>
</tr>
<tr>
<td>( \Theta_{G_2} )</td>
<td></td>
<td>0.024</td>
<td>Inertia of the planetary gear</td>
</tr>
<tr>
<td>( \Theta_{\text{ci}} )</td>
<td></td>
<td>0.024</td>
<td>Resultant inertia of the driving blocks</td>
</tr>
<tr>
<td>( \Theta_{\text{ci}} )</td>
<td></td>
<td>0.024</td>
<td>( \Theta_{\text{ci}} = \Theta_{\text{in}} + \frac{1}{2} \Theta_{G_2} )</td>
</tr>
<tr>
<td>( B_{cG_2} )</td>
<td>[Nm]</td>
<td>1</td>
<td>Attenuation of the belt drive</td>
</tr>
<tr>
<td>( B_{cG_3} )</td>
<td></td>
<td>1</td>
<td>Attenuation of the planetary gear</td>
</tr>
<tr>
<td>( B_{c\bar{G}_2} )</td>
<td></td>
<td>1</td>
<td>Resultant attenuation of the driving blocks</td>
</tr>
<tr>
<td>( B_{c\bar{G}_3} )</td>
<td></td>
<td>1</td>
<td>( B_{c\bar{G}<em>3} = B</em>{cG_3} + \frac{1}{2} B_{cG_2} )</td>
</tr>
<tr>
<td>( \Omega_m )</td>
<td>[rad]</td>
<td>9.81</td>
<td>Angular velocity of the motor</td>
</tr>
<tr>
<td>( \Omega_{G_2} )</td>
<td></td>
<td>9.81</td>
<td>Angular velocity of the small wheel of the toothed belt drive of each driving block</td>
</tr>
</tbody>
</table>

Appendix 2. Description of matrices

\[
T_{\text{robot}} = \begin{bmatrix}
    t_{11} & t_{12} & t_{13} \\
    t_{21} & t_{22} & t_{23} \\
    t_{31} & t_{32} & t_{33}
\end{bmatrix}, \tag{B1}
\]

The elements of \( T_{\text{robot}} \) are detailed in the Supplementary file.

\[
M = \begin{bmatrix}
    m_{11} & m_{12} & m_{13} \\
    m_{21} & m_{22} & m_{23} \\
    m_{31} & m_{32} & m_{33}
\end{bmatrix}, \tag{B2}
\]

The elements of \( M \) are detailed in the Supplementary file.

\[
A = \begin{bmatrix}
    0_{3\times3} & I_{3\times3} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\
    a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\
    a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\
    a_{74} & 0 & 0 & a_{77} & 0 & 0
\end{bmatrix}, \tag{B3}
\]

The elements of \( A \) are detailed in the Supplementary file.

\[
C = \begin{bmatrix}
    I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
    0_{3\times3} & I_{3\times3} & 0_{3\times3} \\
    c_{104} & 0 & c_{106} \\
    0_{3\times3} & c_{114} & c_{115} & c_{116} & 0_{3\times3} \\
    c_{124} & c_{125} & c_{126}
\end{bmatrix}, \tag{B4}
\]

The elements of \( C \) are detailed in the Supplementary file.