Autonomous Robots (2023) 47:465–481
https://doi.org/10.1007/s10514-023-10092-x

Robust inverse dynamics by evaluating Newton–Euler equations with respect to a moving reference and measuring angular acceleration

Maximilian Gießler[1] · Bernd Waltersberger[1]

Received: 5 April 2022 / Accepted: 17 January 2023 / Published online: 28 February 2023

Abstract

Maintaining stability while walking on arbitrary surfaces or dealing with external perturbations is of great interest in humanoid robotics research. Increasing the system's autonomous robustness to a variety of postural threats during locomotion is the key despite the need to evaluate noisy sensor signals. The equations of motion are the foundation of all published approaches. In contrast, we propose a more adequate evaluation of the equations of motion with respect to an arbitrary moving reference point in a non-inertial reference frame. Conceptual advantages are, e.g., getting independent of global position and velocity vectors estimated by sensor fusions or calculating the imaginary zero-moment point walking on different inclined ground surfaces. Further, we improve the calculation results by reducing noise-amplifying methods in our algorithm and using specific characteristics of physical robots. We use simulation results to compare our algorithm with established approaches and test it with experimental robot data.

Keywords Multi-body dynamics · Equation of motion · Moving reference point · Imaginary zero-moment point · Humanoid robot

1 Introduction

Humanoid robots have a significant advantage over all other types of robots. Since they are designed to resemble the human body, they are suitable for interacting in a human-made habitat. Consequently, the research field of humanoid robotics is currently gaining importance. A crucial aspect of increasing the autonomy of humanoid robots is to improve dynamic stability during locomotion in challenging settings, e.g., the robot's ability to effectively tolerate external perturbations during dynamic, omnidirectional locomotion. On the other hand, locomotion in arbitrary environments is of great relevance. In this context, the main focus is to enhance the robot's ability to cope with uneven and unknown ground surfaces of the man-made habitat. Given that stability-enhancing control algorithms are implemented, the robot's autonomy will increase. This opens up new areas of application. Some examples are household chores, care of the elderly, and use as a service worker.

The prerequisite of control algorithms for maintaining stability is the calculation of the robot's dynamics. Euler's theorem is often used in robotics as a starting point for dynamics calculation, e.g. Henze et al. (2016), Buschmann (2011), Hirukawa et al. (2006) and Kajita et al. (2014) use it. This theorem states that the absolute time derivative of the total angular momentum with respect to a fixed reference point 0 equals the total resultant moment about the same reference point ($\dot{\boldsymbol{L}}^{(0)} = \boldsymbol{M}^{(0)}$).

The approaches from Henze et al. (2016) and Buschmann (2011) require methods to determine at least the absolute position vector of one body segment. If this vector is not explicitly measurable by sensors, it is estimated by sensor fusion. E.g., Henze et al. (2016) use the kinematics, absolute position vector and the orientation provided by IMUs of all end effectors in contact. Buschmann (2011) uses the odometry. In general, sensor fusions or odometry can be sensitive to errors, because of signal noise or undefined contact impacts the accuracy. Further, the proposed sensor fusion from Henze et al. (2016) requires the assumption of rigid contact of at least one body segment. Likewise, a stand-alone calculation of absolute position vectors from the kinematic chains of the

✉ Maximilian Gießler
maximilian.giessler@hs-offenburg.de

Bernd Waltersberger
bernd.waltersberger@hs-offenburg.de

[1] Department of Mechanical and Process Engineering, University of Applied Sciences Offenburg, 77652 Offenburg, Germany

robot is not feasible. Kajita et al. (2014) calculate the angular momentum $\boldsymbol{L}^{(0)}$ and propose to use a numerical differentiation to determine its time derivative. To calculate $\boldsymbol{L}^{(0)}$ both the absolute position and linear velocity vector are required. Due to the mentioned aspects, we regard the requirement of the absolute position or linear velocity vector as the main problem of these published approaches. This especially applies to the need to handle undefined contacts and noisy sensor signals on real robots.

Also, a human being has no sense organs to directly perceive its global position or absolute velocity with respect to a fixed spatial point. Further, it does not know the position vector of its center of mass (COM). In conclusion, human control algorithms for stabilizing locomotion can operate without this information.

Considering all mentioned aspects, we propose an approach that does not require any information about the global position, velocity, or COM vector of the robot. We achieve this by evaluating the Newton–Euler equations of motion (EOM) with respect to an arbitrary moving reference point in a non-inertial reference frame (NIF). The dynamics calculation is then independent of absolute position vectors of the robot. Similar approaches using a moving reference point are common in multi-body dynamics, as seen e.g. in Wittenburg (1977).

Furthermore, one goal is to provide an approach that is usable on real robots without filtering the sensor signal. Therefore, we propose to measure the angular acceleration vector via angular velocity and linear acceleration measurements to reduce the number of noise-amplifying numerical differentiation. So this work contributes to the practical use of evaluating the EOMs on physical robots with noisy sensor signals.

For this work, we choose the imaginary zero-moment point (IZMP) to evaluate the results of the EOM calculation. We find the IZMP gives additional valuable insight into the stability evaluation of gait behaviors compared to the sole evaluation of the position of the center of pressure (COP) measured with force/torque sensors. Especially for gait behaviors with possible external perturbations, calculating the distance of the IZMP from the support region of the robot's feet or as well from the COP could give additional quantitative information as to the impact of the perturbation or control strategies.

In the following Sect. 2, we review the state of the art of referring control algorithms and stability criteria and point out where we see further contributions achieved through our approach. In Sect. 3, we explain the mathematical foundations. In Sect. 4, we discuss the methods used for providing the data and sensor signals required for the core algorithm (seen in Sect. 3). On the one hand, we describe a way to measure the total kinematic state, especially the angular acceleration of the root segment. On the other hand, we show the calculation of the remaining kinematic quantities for the following kinematic chain using the relative derivative and the kinematic state of all joints.

Moreover, we use simulation and experimental results of the humanoid robot *Sweaty* to show our approach's capabilities applied on the physical robots. Section 5 illustrates the outcomes. In the article's final parts, we discuss the differences and impacts on physical robots of our approach compared to Henze et al. (2016), Buschmann (2011) and Kajita et al. (2014) and give an outlook to future work.

2 Previous work

Several approaches to describe the stability of humanoid robots during locomotion have been published. Mostly, their aim was to determine an indicator assessing stability in dynamic situations quantitatively. One of the most discussed and applied stability criteria uses the notion of the zero-moment point (ZMP). If the ZMP lies within the support region, the current dynamic equilibrium of the robot can be assumed stable in the sense that the robot cannot tip over an edge of the support region. The ZMP is the spatial point in the ground surface plane where the components of the moment vector resulting from the reaction force parallel to this plane are zero. Vukobratovic et al. (1970) and Vukobratović and Stepanenko (1972) introduced this concept to the field of robotics. Their approach can be regarded as a generalization of the classical static tilting stability concept for a rigid body, e.g. Mehrtens (1887), where tipping over edges of the support region due to external forces is prevented when the resultant of the externally applied forces and the body's weight applied to the effective point of action intersects the support region. Algorithms for calculating the ZMP are shown by, e.g., Kajita et al. (2014). The work of Sardain and Bessonnet (2004) shows that the ZMP coincides with the COP under the assumption that the soles are in flat contact with one single surface and the robot is free of external perturbation forces and moments. If the theoretically calculated ZMP differs from the COP then this is obviously caused by perturbation forces or moments, since such perturbations are not considered in the kinematical ZMP calculation. Vukobratovic et al. (2001) hence proposed the name IZMP for this configuration of the ZMP, to which we adhere in our work.

Several other concepts exist to quantify stability using a point in space. For example, there is the foot-rotation-indicator by Goswami (1999a, 1999b), the feasible-solution-of-wrench by Saida et al. (2003), the zero-rate-of-change angular momentum by Goswami and Kallem (2004) or the capture point (CP) by Pratt et al. (2006). Furthermore, methods were developed to modulate contact state and control the ground reaction forces. Hirukawa et al. (2006), Anitescu and Potra (1997), Pang and Trinkle (2000) propose numerical methods to search for feasible solutions and, based on their characteristics, predictions are made about the stability proposed.

All the mentioned proposals use the EOMs as a foundation. The level of detail of the mechanical model is essential for the accuracy. Different approaches have been established over the last decades. On the one hand, there are conceptual models such as the linear inverted pendulum (LIP), e.g., shown in Kajita et al. (2010). The use case is to implement simplified control algorithms in combination with stability criteria. For example, Englsberger et al. (2011) present a bipedal walking control approach based on the LIP and the CP dynamics. Mesesan et al. (2021) introduced a trajectory adaptation for push and stumble recovery based on the CP. Kamioka et al. (2018) are optimizing the ZMP and footsteps based on the CP for a LIP. Also, Wang et al. (2018) and InJoon et al. (2020) use the concepts of LIP and ZMP to stabilize the gait of humanoid robots. On the other hand, modeling robots as rigid multi-body models has become established, see, e.g., in Vijaykumar Shah et al. (2013). Featherstone (2008) shows the floating-base model as a kind of rigid multi-body model and it is defined most generally.

A further distinctive criterion is the calculation method of the time derivatives of the EOMs. Either numerical (Kajita et al., 2014) or analytical derivatives of the equations are proposed. For the numerical derivatives of the linear and angular momentum, the knowledge of the linear velocity of at least one segment is required. Wang et al. (2017) or Kimura et al. (2018) show approaches to estimate the linear velocity vector. As mentioned before, mostly the reference point for the balance of angular momentum is referenced to a fixed point in the inertial frame (WF) or the immaterial COM. Kajita et al. (2014) propose to calculate the rate of change of angular momentum by applying a numerical time derivative. Hirukawa et al. (2006) use the rate of change of angular momentum without specifying an explicit equation. Only the equation of angular momentum is stated explicitly. Further, Kajita et al. (2003) also propose a method to control the total linear and angular momentum.

The proposed analytical derivative methods require also the absolute position vector of the root segment from a spatially fixed reference point as an input. Henze et al. (2016) use a dynamics calculation based on a fixed reference point in a WF. They estimate the absolute position vector of the root segment by sensor fusion. This estimation requires information about the kinematics, absolute position vectors and orientations of all end effectors in contact. In their approach, they calculate the absolute position vector of the root segment from each end effector in rigid contact. The redundant results are combined in a sensor fusion to estimate global position vectors.

Buschmann (2011) describes a method to introduce a new reference frame for every gait cycle. Each swing leg touchdown introduces a new reference frame. It is placed below the stance foot. This avoids numerical inaccuracies through large distances from the initial reference point. Finally, Buschmann (2011) also uses a dynamics calculation based on a fixed reference point in the current reference frame.

In contrast, we propose an algorithm to evaluate the EOMs using Newton–Euler equations for a rigid multi-body system related to an arbitrary moving reference point in a NIF. To maintain generality, we use the floating base model. We use the rigid body theory typically known from the field of multibody dynamics to derive the EOMs. We further use the so-called relative derivative to evaluate the EOMs referenced in a NIF as well as to calculate all kinematic quantities for the multi-body segments.

In combination, this will allow us to carry the reference point and the NIF along with arbitrary locomotion of the robot. E.g., the reference point can be an arbitrary spatial point or fixed to any material or immaterial point of the robot. Thus, for different types of control algorithms, an adapted reference point can be used. Also, we will show that our algorithm requires no absolute position and linear velocity vector of the root segment. Thus, our approach omits the measurement or estimation by sensor fusion to determine those.

To evaluate the effects when applying these approaches on physical robots, we will simulate the influence of realistic sensor noise behavior. We will also show the impact of feasible sensor and actuator controller frequencies. In addition, we will demonstrate a way to measure the angular acceleration of the root segment using gyroscope and linear acceleration data. We will compare this method against the numerical calculation from the angular velocity signal.

Furthermore, using a moving reference point as the origin of a moving evaluation frame allows following the contour of the ground's surface. IMUs located in the feet provide information about the inclination of the surface during the stance phase. A transition for changing surface inclinations is no longer necessary for calculating, e.g., the IZMP.

Besides combining the mentioned approaches with our work, we intend to fully outline the computation procedure of the dynamics and the IZMP of a robot omitting noise-amplifying numerical differentiation for critical terms in the procedure and without estimations of global vectors. For an easy application, the backgrounds and the algorithm we are going to present are fully adapted to use the *Kinematics and Dynamics Library* (KDL) and to integrate into the *Robot Operating System 2* (ROS2).

3 Backgrounds

At first, we introduce in Fig. 1 the frames and vectors. NIF is the frame, in which the EOMs are evaluated. Its origin rep-

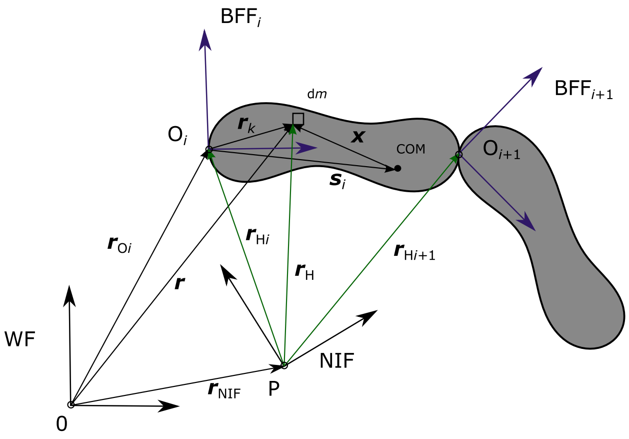
BFF$_i$
dm
BFF$_{i+1}$
O$_i$
$\boldsymbol{r}_k$
$\boldsymbol{x}$
COM
O$_{i+1}$
$\boldsymbol{s}_i$
$\boldsymbol{r}_{\mathrm{O}i}$
$\boldsymbol{r}_{\mathrm{H}i}$
$\boldsymbol{r}$
$\boldsymbol{r}_{\mathrm{H}}$
$\boldsymbol{r}_{\mathrm{H}i+1}$
WF
NIF
$\boldsymbol{r}_{\mathrm{NIF}}$
P
0

Fig. 1 Definition of the vectors and frames used to describe the kinematic description of a rigid multi-body system

resents the arbitrary moving reference point for the angular momentum balance. Each rigid-body part is given a body-fixed frame (BFF). The origin of a BFF is located at the center of the joint axis considering the mechanical joint structure. The equivalent transformations used in the following are based on the rigid body theory used in multi-body dynamics. The vector from the origin of the NIF to the corresponding BFF of an arbitrary segment i is called $\boldsymbol{r}_{\mathrm{H}_i}$. The vector from an arbitrarily positioned inertial frame to the origin of a BFF is denoted as $\boldsymbol{r}_{\mathrm{O}_i}$. The vector from the origin of the BFF to any mass element dm is declared as $\boldsymbol{r}_k$. The vector $\boldsymbol{s}_i$ points from the origin of the BFF to the segment COM and $\boldsymbol{x}$ points from the COM to an arbitrary mass element dm. The definitions of these vectors are given by

$$\boldsymbol{r} = \boldsymbol{r}_{\mathrm{O}_i} + \boldsymbol{r}_k \tag{1}$$

$$\boldsymbol{r}_{\mathrm{O}_i} = \boldsymbol{r}_{\mathrm{NIF}} + \boldsymbol{r}_{\mathrm{H}_i} \tag{2}$$

$$\boldsymbol{r}_{\mathrm{COM}_i} = \boldsymbol{r}_{\mathrm{O}_i} + \boldsymbol{s}_i\,. \tag{3}$$

Here, $\boldsymbol{r}_{\mathrm{NIF}}$ is the position vector pointing from the origin of a arbitrary WF to the origin of NIF. The absolute time derivative of an arbitrary vector $\boldsymbol{p} = \sum_\beta p_{i\beta}\boldsymbol{e}_{i\beta}$ is calculated according to the basic equation of kinematics as follows

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} &= \sum_{\beta=x,y,z}\frac{\mathrm{d}p_{i\beta}}{\mathrm{d}t}\boldsymbol{e}_{i\beta} + \sum_{\beta=x,y,z}\frac{\mathrm{d}\boldsymbol{e}_{i\beta}}{\mathrm{d}t}p_{i\beta}\\ &= \frac{{}^{i}\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} + \boldsymbol{\omega}_{i0}\times\boldsymbol{p}.\end{aligned} \tag{4}$$

Here, $p_{i\beta}$ is the coordinate in direction β of $\boldsymbol{p}$ and $\boldsymbol{e}_{i\beta}$ is the unit vector of direction β. The suffix i indicates an arbitrary frame in which the vector is referenced and $\boldsymbol{\omega}_{i0}$ indicates the frame's angular velocity. The term $\frac{{}^{i}\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}$ represents the so called relative derivative. This term can be interpreted as the vector's rate of change that one would observe if one were fixed to the moving reference frame. The term $\boldsymbol{\omega}_{i0}\times\boldsymbol{p}$ describes the rate of change of the vector due to the rotation of frame i.

For the implemented EOMs, Newton's second law and angular momentum theorem of Euler, are taken as starting point. Hence

$$\dot{\boldsymbol{P}} = \boldsymbol{F} \tag{5}$$

$$\dot{\boldsymbol{L}}^{(0)} = \boldsymbol{M}^{(0)}. \tag{6}$$

apply. Here, $\boldsymbol{P} = m\boldsymbol{v}_{\mathrm{COM}}$ is the linear and $\boldsymbol{L}^{(0)}$ the angular momentum, which we discuss in more detail in the following section. $\boldsymbol{F}$ is the total external force and $\boldsymbol{M}^{(0)}$ the total moment with respect to the origin of WF applied on the body. Due to the linearity of the mathematical operators used, the balance of momentum and angular momentum for the whole robot can be calculated as the sum of the single rigid-body segments.

In the following, the velocity $\boldsymbol{v}_{\mathrm{X}}$ is the first and the acceleration $\boldsymbol{a}_{\mathrm{X}}$ is the second total time derivative of $\boldsymbol{r}_{\mathrm{X}}$. In this context, X represents all the following different subscripts used, e.g. O$_i$ or COM$_i$. Substituting the second total time derivative of Eq. 3 into Eq. 5 and by taking into account the law of the COM, results in

$$\boldsymbol{F}_i = m_i\left(\boldsymbol{a}_{\mathrm{O}_i} + \dot{\boldsymbol{\omega}}_{i0}\times\boldsymbol{s}_i + \boldsymbol{\omega}_{i0}\times(\boldsymbol{\omega}_{i0}\times\boldsymbol{s}_i)\right). \tag{7}$$

For $\boldsymbol{L_i}^{(0)}$, we start from the following equation

$$\begin{aligned}\boldsymbol{L}_i{}^{(0)} = m_i\big(&\boldsymbol{r}_{\mathrm{O}_i}\times\left(\boldsymbol{v}_{\mathrm{O}_i} + \boldsymbol{\omega}_{i0}\times\boldsymbol{s}_i\right)\\ &+\boldsymbol{s}_i\times\boldsymbol{v}_{\mathrm{O}_i}\big) + \boldsymbol{\Theta}^{(\mathrm{O}_i)}\boldsymbol{\omega}_{i0},\end{aligned} \tag{8}$$

proposed by Wittenburg (1977), who therefrom derived a formulation with respect to an arbitrary moving reference point. However, our approach differs so far that we will derive a formulation that assumes an arbitrary moving reference point for the moments while using another arbitrary reference point, e.g., pivot points for the inertia tensors. The latter is a common practice in the generic description of input data, e.g., in *C++* libraries.

Form now, i indicates the reference of the quantities to the ith body of a multi-body system. In Eq. 8, $\boldsymbol{\Theta}^{(\mathrm{O}_i)}$ is the inertia tensor of the ith body with respect to the reference point O$_i$. We substitute Eq. 2 and the total time derivative of Eq. 3 into Eq. 8. Further, we take the total time derivative of Eq. 8. This gives

$$\begin{aligned}\dot{\boldsymbol{L}}_i{}^{(0)} &= m_i\big(\left(\boldsymbol{r}_{\mathrm{NIF}} + \boldsymbol{r}_{\mathrm{H}_i}\right)\times\boldsymbol{a}_{\mathrm{COM}_i}\\ &\quad+\boldsymbol{s}_i\times\boldsymbol{a}_{\mathrm{O}_i}\big)\\ &\quad+\boldsymbol{\Theta}^{(\mathrm{O}_i)}\dot{\boldsymbol{\omega}}_{i0} + \boldsymbol{\omega}_{i0}\times\boldsymbol{\Theta}^{(\mathrm{O}_i)}\boldsymbol{\omega}_{i0}\\ &= \boldsymbol{M}_i^{(0)}.\end{aligned} \tag{9}$$

We see all absolute velocity terms cancel out.

Next, we subtract the term $\boldsymbol{r}_{\mathrm{NIF}} \times m_i \boldsymbol{a}_{\mathrm{COM}_i}$ from Eq. 9. Further, for the moment on body i and by using Eq. 5

$$\begin{aligned} \boldsymbol{M}_i^{(\mathrm{P})} &= \boldsymbol{M}_i^{(0)} - \boldsymbol{r}_{\mathrm{NIF}} \times \boldsymbol{F}_i \\ &= \boldsymbol{M}_i^{(0)} - \boldsymbol{r}_{\mathrm{NIF}} \times m_i \boldsymbol{a}_{\mathrm{COM}_i} \end{aligned} \tag{10}$$

applies in general. Here, $\boldsymbol{M}_i^{(\mathrm{P})}$ describes the total moment applied on body i with respect to the arbitrary moving reference point P. We get finally

$$\begin{aligned} \boldsymbol{M}_i^{(\mathrm{P})} = m_i (&\boldsymbol{r}_{\mathrm{H}_i} \times \boldsymbol{a}_{\mathrm{O}_i} + \boldsymbol{r}_{\mathrm{H}_i} \times (\dot{\boldsymbol{\omega}}_{i0} \times \boldsymbol{s}_i) \\ &+ \boldsymbol{r}_{\mathrm{H}_i} \times (\boldsymbol{\omega}_{i0} \times (\boldsymbol{\omega}_{i0} \times \boldsymbol{s}_i)) + \boldsymbol{s}_i \times \boldsymbol{a}_{\mathrm{O}_i}) \\ &+ \boldsymbol{\Theta}^{(\mathrm{O}_i)} \dot{\boldsymbol{\omega}}_{i0} + \boldsymbol{\omega}_{i0} \times \boldsymbol{\Theta}^{(\mathrm{O}_i)} \boldsymbol{\omega}_{i0}. \end{aligned} \tag{11}$$

Equation 11 describes the change in angular momentum for an arbitrary moving reference point. From Eq. 11, we see that all absolute position and velocity vectors cancel out. The Eqs. 7 and 11 describe the Newton–Euler EOMs for an arbitrary moving reference point in a NIF.

4 Methods for a robust and efficient evaluation of the inverse dynamics

With our approach, we intend to provide an effective calculation of the inverse dynamics. We use this to calculate the IZMP exemplarily. We believe the IZMP provides an additional benefit in evaluating gait situations under external perturbations compared to a COP measured using 6-axis force/torque sensors. For example, the distance of the IZMP from the tipping edge or the measured COP provides a metric for the needed response to stabilize the robot in the presence of external perturbations. We set the goal that our approach is efficient and works stand-alone with noisy and unfiltered inertia sensor signals. Other inaccuracies or disturbances in the evaluation of the EOMs are, e.g., measurement errors of encoder inadequacies in the mechanical design of the robot, the underlying rigid body hypothesis, or modeling errors of the inertia data. However, these types of disturbance are not part of this work. By sensor fusion with 6-axis force/torque sensors and combined with an adapted Kalman filter, our contribution is extendable to become less sensitive to the mentioned inaccuracies in the model assumption.

4.1 EOMs for a multi-body system

To describe the dynamics of a multi-body system, we sum up the Eqs. 7 and 11 for all n rigid bodies. In this sum, the reaction forces and moments between two body segments i and $i+1$ cancel out, due to Newton's third law. The external forces and moments, e.g., caused by the gravity force or the ground reaction force remain. This results in

$$\begin{aligned} \boldsymbol{F}_{\mathrm{ext}} = \sum_{i=1}^{n} (m_i (&\boldsymbol{a}_{\mathrm{O}_i} + \dot{\boldsymbol{\omega}}_{i0} \times \boldsymbol{s}_i \\ &+ \boldsymbol{\omega}_{i0} \times (\boldsymbol{\omega}_{i0} \times \boldsymbol{s}_i))) \end{aligned} \tag{12}$$

$$\begin{aligned} \boldsymbol{M}_{\mathrm{ext}}^{(\mathrm{P})} = \sum_{i=1}^{n} \Big(m_i (&\boldsymbol{r}_{\mathrm{H}_i} \times \boldsymbol{a}_{\mathrm{O}_i} + \boldsymbol{r}_{\mathrm{H}_i} \times (\dot{\boldsymbol{\omega}}_{i0} \times \boldsymbol{s}_i) \\ &+ \boldsymbol{r}_{\mathrm{H}_i} \times (\boldsymbol{\omega}_{i0} \times (\boldsymbol{\omega}_{i0} \times \boldsymbol{s}_i)) + \boldsymbol{s}_i \times \boldsymbol{a}_{\mathrm{O}_i}) \\ &+ \boldsymbol{\Theta}_i^{(\mathrm{O}_i)} \dot{\boldsymbol{\omega}}_{i0} + \boldsymbol{\omega}_{i0} \times \boldsymbol{\Theta}_i^{(\mathrm{O}_i)} \boldsymbol{\omega}_{i0} \Big). \end{aligned} \tag{13}$$

The algorithm is recursively designed and thus generic. The mechanical structure and the physical properties of each rigid body segment of the robot can be provided via a *Unified Robot Description Format* file.

For the implementation in the code, we switch from vector to coordinate matrix notation. E.g., the representation of Eq. 13 is given by

$$\begin{aligned} {}^{\mathrm{NIF}}\boldsymbol{M}_{\mathrm{ext}}^{(\mathrm{P})} = \sum_{i=1}^{n} \Big(m_i\, &\boldsymbol{R}^{\mathrm{NIF}\,i} ({}^{i}\boldsymbol{r}_{\mathrm{H}_i} \times {}^{i}\boldsymbol{a}_{\mathrm{O}_i} \\ &+ {}^{i}\boldsymbol{r}_{\mathrm{H}_i} \times \left({}^{i}\dot{\boldsymbol{\omega}}_{i0} \times {}^{i}\boldsymbol{s}_i\right) \\ &+ {}^{i}\boldsymbol{r}_{\mathrm{H}_i} \times ({}^{i}\boldsymbol{\omega}_{i0} \times ({}^{i}\boldsymbol{\omega}_{i0} \times {}^{i}\boldsymbol{s}_i)) \\ &+ {}^{i}\boldsymbol{s}_i \times {}^{i}\boldsymbol{a}_{\mathrm{O}_i}) + {}^{i}\boldsymbol{\Theta}_i{}^{(\mathrm{O})i}\, {}^{i}\dot{\boldsymbol{\omega}}_{i0} \\ &+ {}^{i}\boldsymbol{\omega}_{i0} \times {}^{i}\boldsymbol{\Theta}_i{}^{(\mathrm{O})\,i}\, \boldsymbol{\omega}_{i0} \Big). \end{aligned} \tag{14}$$

Here, $\boldsymbol{R}^{\mathrm{NIF}\,i}$ is a 3 × 3 rotation matrix that transforms the coordinates with respect to the frame i to the coordinates with respect to the frame NIF. If a symbol is superscripted on the left side by i, this means that the referring vector or tensor is expressed as its 3 × 1 (vector) or 3 × 3 (tensor) coordinate matrix with respect to frame i. This notation is adopted from Woernle (2016). Each summand of the right-hand side of Eq. 14 is a 3 × 1 coordinate matrix. The number of n rigid bodies corresponds to the number of summands.

4.2 Analytically derived EOMs

We link the arbitrary moving reference point and the origin of the NIF to a material point of the robot. This point coincides with the origin of the root segment, called pelvis. As seen in Fig. 2, the x-axis of the NIF is defined as pointing always parallel to the sagittal plane of the robot and the z-axis points parallel to the normal vector of the ground surface. Also, it is feasible to place the NIF on the ground surface. E.g., the origin could coincide with the moving immaterial point of

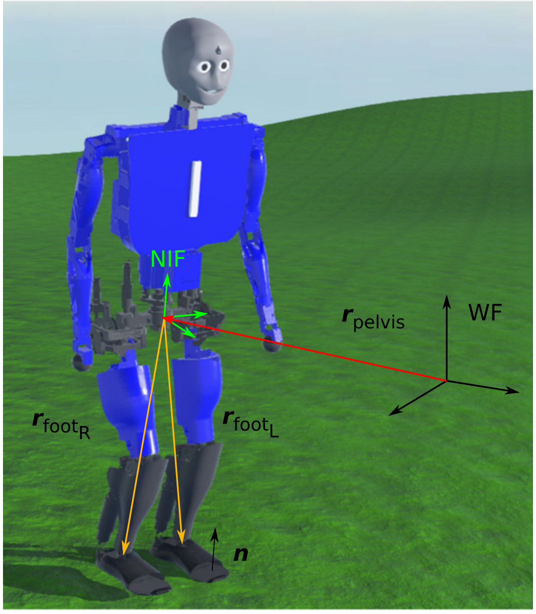

Fig. 2 Visualization of the vectors and frames WF and NIF in the mentioned application related to the robot *Sweaty*. The vector $\boldsymbol{r}_{\text{pelvis}}$ describes the absolute position of the root segment with respect to the origin of the WF. $\boldsymbol{r}_{\text{foot}_\text{R}}$ and $\boldsymbol{r}_{\text{foot}_\text{L}}$ indicate the relative position vectors from the origin of the NIF to origin of each foot

the vertical projection of the origin of the BFF from the root segment onto the ground surface.

On the other hand, the approaches based on $\dot{\boldsymbol{L}}^{(0)} = \boldsymbol{M}^{(0)}$ evaluate the EOMs with respect to a fixed spatial point. Figure 2 displays the difference between the mentioned (Henze et al., 2016; Buschmann, 2011; Kajita et al., 2014) and our approach. These approaches use an absolute position vector pointing from the fixed reference point to at least one material point of the robot, e.g., $\boldsymbol{r}_{\text{pelvis}}$. With kinematics, it is then feasible to calculate the absolute position vectors of all segments with respect to the fixed reference point. Since our chosen reference point coincides with one material point, no absolute position vector is needed. Thus we can use relative vectors, e.g., $\boldsymbol{r}_{\text{foot}_\text{R}}$ and $\boldsymbol{r}_{\text{foot}_\text{L}}$ pointing from the reference point to the origin of each foot.

4.3 Relative derivative of angular velocity

We use the relative derivative to link the joint kinematics with the total angular velocity $\boldsymbol{\omega}_{i0}$ and angular acceleration $\dot{\boldsymbol{\omega}}_{i0}$. Using the law of time derivative (Eq. 4), the following relationship generally applies

$$\dot{\boldsymbol{\omega}}_{i0} = \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{i0}}{\mathrm{d}t} + \boldsymbol{\omega}_{i0} \times \boldsymbol{\omega}_{i0} = \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{i0}}{\mathrm{d}t}. \tag{15}$$

Due to the addition formula of angular velocities, for Eq. 15 follows

$$\begin{aligned}\frac{{}_i\mathrm{d}\boldsymbol{\omega}_{i0}}{\mathrm{d}t} &= \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{(i-1)0}}{\mathrm{d}t} + \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{i(i-1)}}{\mathrm{d}t} \\ &:= \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{(i-1)0}}{\mathrm{d}t} + \ddot{q}_i\boldsymbol{n}.\end{aligned} \tag{16}$$

Here, $\ddot{q}_i\boldsymbol{n}$ is the joint acceleration vector. The vector $\boldsymbol{n}$ describes the unit vector of the joint axis. In general, $\dot{\boldsymbol{\omega}}_{(i-1)0}$ can be calculated by

$$\dot{\boldsymbol{\omega}}_{(i-1)0} = \frac{{}_i\mathrm{d}\boldsymbol{\omega}_{(i-1)0}}{\mathrm{d}t} + \boldsymbol{\omega}_{i0} \times \boldsymbol{\omega}_{(i-1)0}, \tag{17}$$

using Eq. 4. Equation 17 can be rearranged accordingly

$$\begin{aligned}\frac{{}_i\mathrm{d}\boldsymbol{\omega}_{(i-1)0}}{\mathrm{d}t} &= \dot{\boldsymbol{\omega}}_{(i-1)0} - \boldsymbol{\omega}_{i0} \times \boldsymbol{\omega}_{(i-1)0} \\ &= \dot{\boldsymbol{\omega}}_{(i-1)0} \\ &\quad - (\boldsymbol{\omega}_{(i-1)0} + \boldsymbol{\omega}_{i(i-1)}) \times \boldsymbol{\omega}_{(i-1)0} \\ &= \dot{\boldsymbol{\omega}}_{(i-1)0} - \dot{q}_i\boldsymbol{n} \times \boldsymbol{\omega}_{(i-1)0}.\end{aligned} \tag{18}$$

Here, $\dot{q}_i$ is the joint velocity of body i compared to $i-1$. For $\dot{\boldsymbol{\omega}}_{i0}$ finally follows

$$\dot{\boldsymbol{\omega}}_{i0} = \dot{\boldsymbol{\omega}}_{(i-1)0} + \boldsymbol{\omega}_{(i-1)0} \times \dot{q}_i\boldsymbol{n} + \ddot{q}_i\boldsymbol{n}. \tag{19}$$

The matrix ${}^i\dot{\boldsymbol{\omega}}_{i0}$ is given by,

$${}^i\dot{\boldsymbol{\omega}}_{i0} = {}^i\dot{\boldsymbol{\omega}}_{(i-1)0} + {}^i\boldsymbol{\omega}_{(i-1)0} \times \dot{q}_i^i\boldsymbol{n} + \ddot{q}_i^i\boldsymbol{n}. \tag{20}$$

4.4 IZMP calculation

To calculate the IZMP, the EOMs will be evaluated with measured sensor and actuator information at discrete points in time. Due to the time derivatives being available as measured values, we can interpret the coupled nonlinear differential equations as algebraic equations. We formulate all equations presented in Sect. 3 as continuous-time equations. They can be evaluated without adjustments at any discrete instant in time t_k.

For the calculation of the IZMP, we use Eq. 13. We assume the ground reaction force $\boldsymbol{F}_{\text{gr}}$ and the gravity force $m\boldsymbol{g}$ act externally on the robot. Then the following applies to the sum of all applied external moments $\boldsymbol{M}_{\text{ext}}^{(\text{P})}$,

$$\boldsymbol{M}_{\text{ext}}^{(\text{P})} = \boldsymbol{M}^{(\text{IZMP})} + \boldsymbol{r}_{\text{IZMP}} \times \boldsymbol{F}_{\text{gr}} + \boldsymbol{r}_{\text{COM}} \times m\boldsymbol{g} \tag{21}$$

Here, $\boldsymbol{M}^{(\text{IZMP})}$ is the resultant vector of the moment at point $\boldsymbol{r}_{\text{IZMP}}$. Per the definition of the IZMP, the components of $\boldsymbol{M}^{(\text{IZMP})}$ are zero for the horizontal plane. When we substitute Eq. 5 into 21, the ground reaction force can be replaced.

Further, substituting this into Eq. 13, the IZMP calculation equation is given by

$$\begin{aligned}&\boldsymbol{M}^{(\mathrm{IZMP})}+\boldsymbol{r}_{\mathrm{IZMP}}\times(\dot{\boldsymbol{P}}-m\boldsymbol{g})+\boldsymbol{r}_{\mathrm{COM}}\times m\boldsymbol{g}\\&=\sum_{i=1}^{n}\Bigg(m_i(\boldsymbol{r}_{\mathrm{H}_i}\times\boldsymbol{a}_{\mathrm{O}_i}+\boldsymbol{r}_{\mathrm{H}_i}\times(\dot{\boldsymbol{\omega}}_{i0}\times\boldsymbol{s}_i)\\&\quad+\boldsymbol{r}_{\mathrm{H}_i}\times(\boldsymbol{\omega}_{i0}\times(\boldsymbol{\omega}_{i0}\times\boldsymbol{s}_i))+\boldsymbol{s}_i\times\boldsymbol{a}_{\mathrm{O}_i})\\&\quad+\boldsymbol{\Theta}_i^{(\mathrm{O}_i)}\dot{\boldsymbol{\omega}}_{i0}+\boldsymbol{\omega}_{i0}\times\boldsymbol{\Theta}_i^{(\mathrm{O}_i)}\boldsymbol{\omega}_{i0}\Bigg).\end{aligned}\tag{22}$$

This system of the coordinate equations from Eq. 22 is underdetermined due to the skew-symmetric property of the cross product. To obtain a single unique solution of the equation system, we calculate the IZMP coordinates upon the assumption that the IZMP is located in the ground surface plane. Therefore, the z-coordinate of the IZMP is set equal to that of the reference point in relation to the ground surface plane.

4.4.1 Required inputs

By analyzing Eq. 13 and regarding the underlying floating-base model, the required input signals can be identified. From Eq. 13 it follows the evaluation of the dynamics requires the position, angular velocity, linear and angular acceleration vector. For all segments of the multi-body system except the root segment, these vectors can be calculated through sensor measurements of the actuators and the total angular velocities $\boldsymbol{\omega}_{(i-1)0}$ and $\dot{\boldsymbol{\omega}}_{(i-1)0}$ of the previous segment known from the recursive loop. As shown, these vectors are functions of the joint angle q_i, joint velocity $\dot{q}_i$, and joint acceleration $\ddot{q}_i$. The commonly used actuator encoder in robotics delivers signals for q_i. We calculate $\dot{q}_i$ and $\ddot{q}_i$ with numerical derivatives using the actuator controller. Therefore, we can use the increment Δt corresponding to the frequency of the actuator controller. Common frequencies of actuator controllers used in robotics are in the range of kilohertz. Therefore, we name this procedure actuator coordinate differentiation (ACD). Due to the option of choosing an arbitrary moving reference point in a non-inertial frame and the analytical way of deriving the EOMs, we name the proposed method MR-AD.

Considering the root segment, these vectors must be provided from sensor data or estimations. Since our chosen reference point coincides with a material point of the root segment, we can use a relative body-fixed position vector. The angular velocity is measured by a gyroscope and the linear acceleration vector by an accelerometer.

In perspective of the authors, direct angular acceleration measurements are not commonly used in robotics. One approach is to calculate the angular acceleration with a numerical derivative of $\boldsymbol{\omega}_{10}$. In general, numerical derivatives are noise-amplifying. So the accuracy depends on the noise behavior of the angular velocity signal and on the increment Δt used.

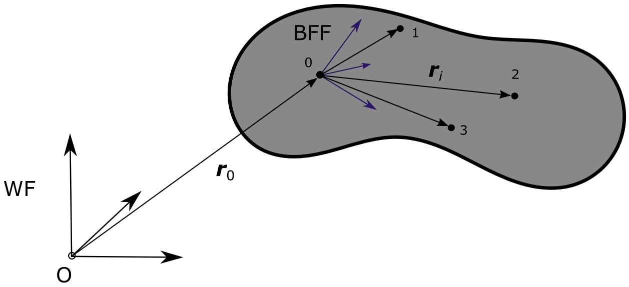

Fig. 3 Location of the IMU and the accelerometers on a rigid body to calculate the angular acceleration. $\boldsymbol{r}_0$ describes the position vector of the IMU. $\boldsymbol{r}_i$ are feasible relative position vectors of the accelerometers with respect to the IMU. $\boldsymbol{r}_i$ are linearly independent to each other

For our approach, we propose an algebraic method to calculate the angular acceleration vector, similar to the proposed methods, e.g., from Padgaonkar et al. (1975). Therefore, we place three additional accelerometers on the root segment and use the rigid body assumption. Figure 3 shows the schematic setup. With this setup, we measure the linear acceleration vector of specific spatial points and the angular velocity vector of the rigid body.

In general, the locations of the sensors on the rigid body must satisfy the condition that the connection vectors from the IMU to the accelerometers are linearly independent. This prevents the sensors from lying in a spatial plane. If the sensors are placed with orthogonal connection vectors to each other, then uniaxial accelerometers can be used.

For the acceleration of an arbitrary fixed point on a rigid body applies in general,

$$\boldsymbol{a}_i=\boldsymbol{a}_0+\dot{\boldsymbol{\omega}}\times\boldsymbol{r}_i+\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}_i).\tag{23}$$

Eq. 23 can be rearranged as follows,

$$\dot{\boldsymbol{\omega}}\times\boldsymbol{r}_i=\boldsymbol{a}_i-\boldsymbol{a}_0-\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}_i).\tag{24}$$

To separate the term $\dot{\boldsymbol{\omega}}$, the cyclic permutation of the vector triple product is used. For this purpose, the scalar product of Eq. 24 and the connection vector $\boldsymbol{r}_j$ is formed, as follows

$$(\dot{\boldsymbol{\omega}}\times\boldsymbol{r}_i)\cdot\boldsymbol{r}_j=(\boldsymbol{a}_i-\boldsymbol{a}_0-\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}_i))\cdot\boldsymbol{r}_j.\tag{25}$$

Applying the cyclic permutation Eq. 25 gives

$$(\boldsymbol{r}_i\times\boldsymbol{r}_j)\cdot\dot{\boldsymbol{\omega}}=(\boldsymbol{a}_i-\boldsymbol{a}_0-\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}_i))\cdot\boldsymbol{r}_j.\tag{26}$$

Eq. 26 leads to three scalar equations. Each equation contains one coefficient from the angular acceleration vector $\dot{\boldsymbol{\omega}}$. For $i=2,\ j=3$; $i=3,\ j=1$ and $i=1,\ j=2$ the following applies,

$$\dot{\omega}_x = ((\boldsymbol{a}_2 - \boldsymbol{a}_0 - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_2)) \cdot \boldsymbol{r}_3)\frac{1}{r_2 r_3} \tag{27}$$
$$= \frac{a_{2z}}{r_2} - \frac{a_{0z}}{r_2} - \omega_y\omega_z$$
$$\dot{\omega}_y = ((\boldsymbol{a}_3 - \boldsymbol{a}_0 - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_3)) \cdot \boldsymbol{r}_1)\frac{1}{r_3 r_1} \tag{28}$$
$$= \frac{a_{3x}}{r_3} - \frac{a_{0x}}{r_3} - \omega_z\omega_x$$
$$\dot{\omega}_z = ((\boldsymbol{a}_1 - \boldsymbol{a}_0 - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_1)) \cdot \boldsymbol{r}_2)\frac{1}{r_1 r_2} \tag{29}$$
$$= \frac{a_{1y}}{r_1} - \frac{a_{0y}}{r_1} - \omega_x\omega_y.$$

4.4.2 Momentum based dynamics calculation

To put the generated results in perspective, the momentum-based differentiation method (IR-MD) for calculating the linear and angular momentum balance proposed by Kajita et al. is also implemented. The implementation strictly follows the procedure given in Kajita et al. (2014). Due to the numerical derivative calculation, only the equations for momentum and angular momentum are used. To determine the rate of change of these quantities in time, a numerical time derivative based on the differential quotient is used. Since the calculated momentum and angular momentum vectors are subtracted from each other in two different instants in time, they must be referenced in the time-constant WF. The purpose of IR-MD is to reduce the complexity of the required equations. Also, any dependencies on the acceleration states of the segments and actuators are omitted. Kajita et al. use the floating base model for the mentioned algorithm. Consequently, the MD method requires q_i and $\dot{q}_i$ and for the root segment the absolute position, the absolute linear and the angular velocity vector must also be specified. The increment Δt of the numerical differentiation is linked to the frequency of the field bus in physical robot systems.

4.5 Physical robot *Sweaty* and simulation model

Sweaty, shown in Fig. 4 (left), is 1.65 m tall and weighs 27.6 kg. The humanoid robot has 39 degrees of freedom (DOF). Spindle motors actuate the DOFs of the lower body and torso.

Rod-and-lever systems translate the linear motion of the spindle nut into rotational motion. The mechanical structure of the rod-and-lever system leads to a variable gear ratio, which results in variable torque and angular velocity characteristics depending on the joint angle. This design also minimizes the joint backlash due to the high gear ratio of the spindles. The *Xsens MTi-20 VRU* IMU is mounted on the hip of *Sweaty*. Two CAN-BUS lines provide communication between actuator controllers and the control unit. The CAN-BUS is clocked with 125 Hz. *Sweaty* was designed to interact autonomously with the human-made habitat. Each sensor used has an equivalent to the human senses. *Sweaty* has no sensors like GPS or linear velocity measurement installed. The global position and linear velocity must be estimated.

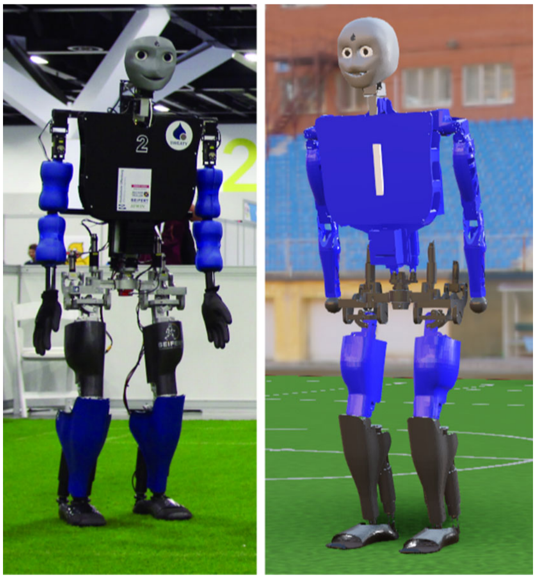

Fig. 4 The real humanoid robot *Sweaty* (left) and the simulation model (right), which was designed to interact autonomously with the human-made habitat

Figure 4 (right) shows the simulation model of *Sweaty*. This model has 31 DOF. The DOFs of both hands are not implemented. *Webots 2021b* from the company *Cyberbotics Ltd.* is used as the simulation tool. In the simulation, however, all DOFs are simulated as rotational motors. This modification is made due to the closed kinematic chains of the rod-and-lever systems leading to a decrease in simulation performance.

Sweaty's software architecture is based on ROS2, where information is exchanged in a network according to the publisher and subscriber principle. Figure 5 shows the ROS2 nodes and their structural connections used to generate the results for this work.

The interface between ROS2 and the simulator is the node *Webots-Controller-Client*. Via this interface, the node *Motion* sends the desired motion pattern to the simulator. In the simulator, these data are processed and the physics engine simulates the dynamics for a fixed time step. The simulation results for joint angles, sensor data, and the current time are made available by the simulator to the ROS2 network via the node *Webots-Controller-Client*. *Webots 2021b* does not provide information about the angular velocity $\dot{q}_i$ of the actuators. Therefore, the required input signals, e.g., angular velocities $\dot{q}_i$ or the angular accelerations $\ddot{q}_i$ of the actuators are calculated through a differential quotient with the simulation time step Δt. The *TF-Framework* of ROS2 makes it

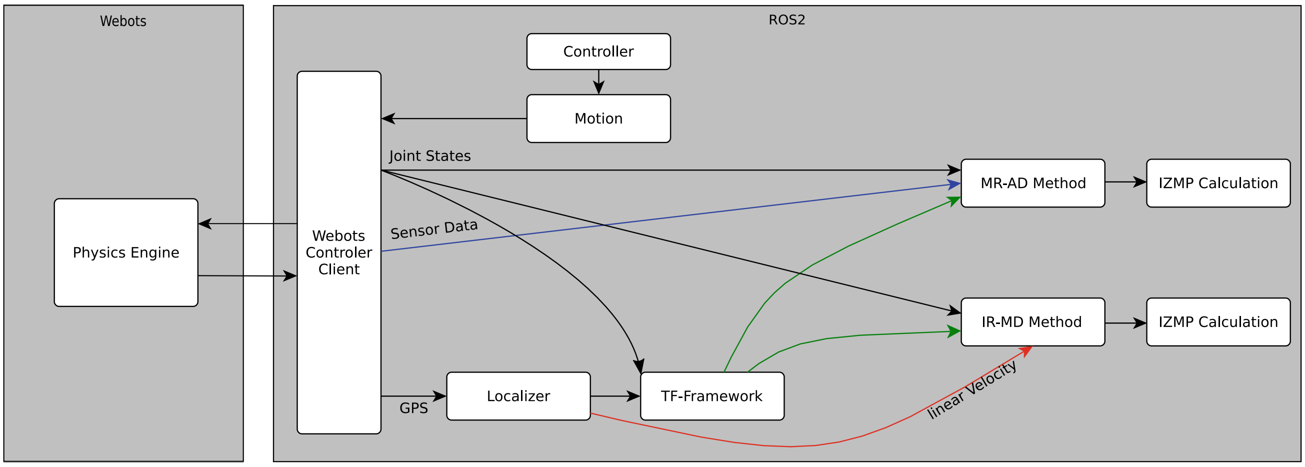

Fig. 5 Nodegraph of the setup to generate the simulation results

possible to request arbitrary coordinate transformations by using the joint angle information and other sensor data.

Especially for this work, a GPS and an IMU are located in the root segment. The IMU combines a gyroscope and an accelerometer. It delivers the signals for angular velocity, linear acceleration, and the orientation of the root segment. The GPS signal is subscribed by the node *Localizer* and published as the transformation from WF to the root segment in the *TF-Framework*. Furthermore, this node takes a time derivative of the GPS signal by using the differential quotient to provide the linear velocity vector. Both mentioned calculation methods are integrated parallel as separate nodes. This ensures compatibility, as the calculations access the same input signals in each time step. Both calculation methods publish their results, which are subscribed by the node *IZMP Calculation*. The calculated coordinates of the IZMP are transformed into a common evaluation frame via the *TF-Framework* and are recorded in *ROS2 bag* files.

Table 1 List of the main abbreviation

Abbreviation	Meaning
ACD	Actuator controller differentiation[a]
BFF	Body-fixed frame
IR-MD	Inertial reference—momentum derivative[b]
MR-AD	Moving reference—analytically derived[c]
NIF	Non-inertial frame
WF	Inertial frame

[a] Method to calculate the time derivatives of the actuator position signal q_i obtaining $\dot{q}_i$ and $\ddot{q}_i$

[b] Dynamics evaluated by numerical differentiation of momentum based equations for a fixed reference point in inertial frame

[c] Dynamics evaluated by analytically derived EOMs for an arbitrary moving reference point in non-inertial frame

4.6 Evaluation methods

We verify our MR-AD approach, compare it to IR-MD and show its general scopes using simulation results. Further, we test our approach on the physical robot *Sweaty*. Table 1 reviews all of the necessary main abbreviations for the following sections.

Due to *Sweaty* having no GPS or sensors for measuring the linear velocity installed, the global position and the linear velocity vector are not available during the execution of the experiment. An estimation by our localization and odometry does not provide usable results. Therefore, we focus on MR-AD for the evaluation of the EOMs on the real robot. For the comparison in simulation, the mentioned approaches perform different test cases and the obtained results will be compared quantitatively through the IZMP coordinates. We choose the BFF of the root segment as the common evaluation frame, which corresponds to the introduced NIF (see Sect. 4.2). A further quantitative evaluation is the relative error Δ. It is defined as $\Delta = \sum_{k=0}^{n} \frac{|f_{\text{proc}}(t_k) - f_{\text{ref}}(t_k)|}{|f_{\text{ref}}(t_k)|} \cdot 100\% \cdot \frac{1}{n}$. Here, $f_{\text{proc}}(t_k)$ represents the value at the time point t_k of the investigated procedure's signal. Similarly, $f_{\text{ref}}(t_k)$ represents the reference signal for the comparison at the time point t_k.

To analyze the applicability to physical robots in real-world applications, we add white noise behavior to the ideal sensor signals to simulate typical noisy signals $\hat{\boldsymbol{s}}(t)$ from physical sensors. White noise behavior is added to linear velocity, angular velocity, and linear acceleration signals. It generally applies $\hat{\boldsymbol{s}}(t) = \boldsymbol{s}(t) + \boldsymbol{n}(t)$. Here, $\boldsymbol{s}(t)$ is the original ideal signal and $\boldsymbol{n}(t)$ is the noise. $\boldsymbol{n}(t)$ is generated normally distributed depending on the standard deviation σ and the mean μ of the physical sensor. We use $\sigma = \nu \cdot \sqrt{f}$ to calculate the standard deviation for the gyroscope and accelerometer as mentioned in the data sheet of *Xsens MTi-20 VRU*. ν is the noise density and f is the sample rate. The mean μ is set to zero.

Sweaty does not have a sensor to measure linear velocity. Therefore, we use the same standard deviation σ for the noise term of the linear velocity signal that applies to the accelerometer. The following equation applies to calculate the noise for each coordinate direction, $\hat{s}_\beta(t_k) = s_\beta(t_k) + n_\beta(t_k) = s_\beta(t_k) + g(\sigma, \mu)$. Here, t_k indicates the discrete point in time.

For this work, we add noise for our approach to $\boldsymbol{a}_i$ for i in range $i = 1, 2, 3, 4$ and $\boldsymbol{\omega}$ of the root segment. It applies $a_{i\beta} = \hat{s}_{a\beta}$ and $\omega_\beta = \hat{s}_{\omega\beta}$. Here, β indicates the coordinate axis in rage $\beta = x, y, z$. For IR-MD, we add noise just to the velocity $\boldsymbol{v}$ of the root segment, in the same way, it applies to $\boldsymbol{a}_i$. Thus for IR-MD, we expect the influence of noise as underestimated since velocity measurements are usually indirect measurements based on filtered acceleration and camera position data.

In general, real robot systems offer the possibility to provide higher update frequencies of sensor and actuator signals compared to the field bus frequency. To analyze the impacts of this characteristic in calculating the dynamics, the simulation runs with a time step of 0.002 s (500 Hz). However, the dynamics calculation runs with 0.008 s (125 Hz). Thus all sensor and actuator signals are available in 500 Hz frequency for processing. Consequently, the frequency of the simulation set to 500 Hz can be interpreted as the actuator controller frequency, and the dynamics calculation frequency set to 125 Hz as the field bus frequency of a physical robot.

5 Results

In the following sections, we show the results of the EOM evaluation by the IZMP. Each line diagram listed in Table 2 contains curve developments for the MR-AD and IR-MD method. The adjustable parameters are the ACD and field bus frequencies and the signal noise that can be switched on or off. If signal noise is present, it is applied, as explained in Sect. 4.6. On the one hand, the column *ACD/Bus* of Table 2 contains the chosen ACD and field bus frequencies for the simulated or real robot. On the other hand in the column *Noise* specifies the present signal noise for each test case.

Likewise, the adjustable parameters for all bar diagrams are the ACD and field bus frequencies and the angular acceleration vector determination approach. Table 3 contains the chosen parameters and properties of each bar diagram. It displays in the column *ACD/Bus* the ACD and field bus frequencies used. In the column *Ang. acc. calc.* it specifies the underlying method of angular acceleration vector estimation for each setup. Further, the signal noise is turned on for each setup (see Sect. 4.6).

If the column *ACD/Field bus* of Tables 2 and 3 contain only one value, this value represents the frequency of both ACD and field bus.

Table 2 Parameters and properties of the line diagrams

Figure	ACD/bus (Hz)	Noise
Fig. 6	125	None
Fig. 7	62.5, 125, 200, 250	None
Fig. 8	125	IMU[a], GPS[b]
Fig. 12	1000/125	Real sensors
Fig. 11	125	None

If the column *ACD/Bus* contains one value, it belongs to both ACD and field bus frequency

[a] Noisy accelerometer and gyroscope signals

[b] Noisy velocity signal from *Webots*

Table 3 Parameters and properties of the bar diagrams

Setup	Method	ACD/bus (Hz)	Ang. acc. calc
A	IR-MD	125	Diff. quotient[b]
B	MR-AD	125	Algebraic[a]
C	IR-MD	500/125	Diff. quotient[b]
D	MR-AD	500/125	Algebraic[a]
E	MR-AD	500/125	Diff. quotient[b]

If the column *ACD/Bus* contains only one value, it belongs to both ACD and field bus frequency

[a] One IMU and three accelerometers are mounted on the root segment to measure $\dot{\boldsymbol{\omega}}$

[b] Differential quotient is used

5.1 Initial conditions for the simulation

At first, we compare the results of MR-AD and IR-MD under ideal conditions. For this purpose, the robot walks in an arbitrary direction starting from the origin of the WF at a gait speed of 0.3m/s. Optimal signal qualities from the simulation are provided for each calculation method. Figure 6 shows exemplary curves of the x- and y-coordinate from the IZMP.

The gait phase of the robot can be divided into two sub-step phases as shown in Fig. 6. The jump in both curves describes the switch of the support foot and thus the transition from sub-step one into two. From the data, it follows that a sub-step takes 0.448 s, and the entire step takes 0.896 s. In the development of the x-coordinate, the two sub-steps can be identified via the consecutively similar curve sections. The sub-step phase begins with the heel strike and ends with the toe-off. The x-coordinate starts after the peaks of the heel strike at approx. -0.08 m and increases in the development to approx. 0.1 m until the toe-off event. The x-coordinate, related to the foot, moves from the rear area to the front area during the duration of a sub-step.

The y-coordinate starts with negative values. The development of the curve shows a relatively constant range between the time frame 0.1 s and 0.25 s. In the further development of the sub-step, the value of the y-coordinate moves towards the zero line. Here, the robot tilts towards the swing leg.

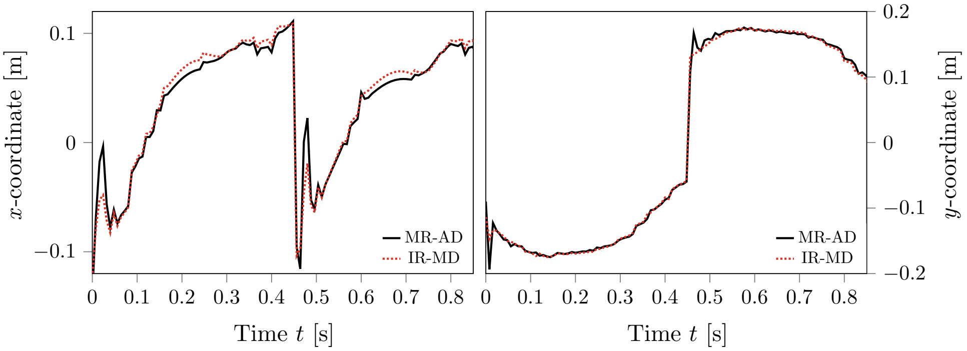

Fig. 6 Comparison of the calculated IZMP x- (left) and y-coordinate (right) of MR-AD and IR-MD under ideal conditions in simulation. The calculations are executed in parallel for the same step phase. The gait speed is0.3 m/s

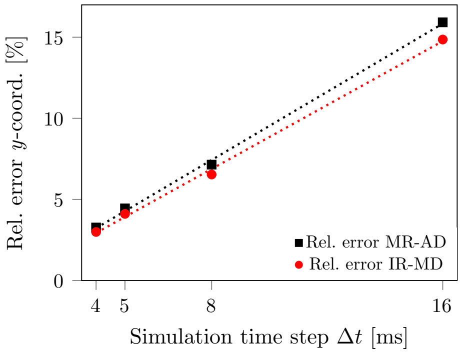

Fig. 7 Convergence behavior for the IZMP y-coordinates of MR-AD and IR-MD for decreasing simulation time steps and ideal signals. The reference curve for the corresponding method is generated with a time step of 0.002 s. The gait speed of 0.3 m/s are the same for each case

The development of the y-coordinate curve indicates that the analyzed step starts with the right leg as the support leg.

In Fig. 7 the convergence behavior of the IR-MD and MR-AD methods for the y-coordinate of IZMP are compared at different simulation time steps. The tested frequencies are 250 Hz, 200 Hz, 125 Hz and 62.5 Hz. All curves were generated with ideal signals. The reference curves have been determined with the corresponding method at a frequency of 500 Hz. Figure 7 shows a linear convergence behavior for both methods with respect to the simulation time step Δt.

5.2 Simulated robot applications

The output signals of physical sensors are generally noisy. To simulate the qualitative impact of this noise, noise behavior is added as explained in 4.6. Figure 8 shows the white noise effects of linear velocity, angular velocity, and linear acceleration signals on the x- and y-coordinates of the IZMP. For comparison, the coordinate curves calculated by MR-AD and IR-MD with ideal input signals are also shown in Fig 8. All displayed curves are generated with a simulation and dynamics calculation frequency of 125 Hz. Figure 8 shows MR-AD and IR-MD do not provide similar curves compared to Fig. 6. The noise behavior of the sensors has a marginal impact on the MR-AD method compared to the curves with ideal sensor signals. In contrast, noise behavior has a significant impact on the IR-MD method.

In contrast to Fig. 8, Fig. 9 shows the relative errors of the IZMP y-coordinate compared between MR-AD and IR-MD. Setup B and D use MR-AD and Setup A and C use IR-MD. Further, Setup A and B use a frequency of 125 Hz for ACD and field bus. In contrast, Setup C and D use sensor and actuator controller frequencies of 500 Hz and a modeled field bus frequency of 125 Hz. For IR-MD increased sensor and actuator controller frequencies impact the accuracy negatively. The relative error increases from 65.2 Hz to 112.4%. For MR-AD the contrary is observed. Here, increased sensor and actuator controller frequencies increase the accuracy. The relative error decreases from 16.1 to 4.4%.

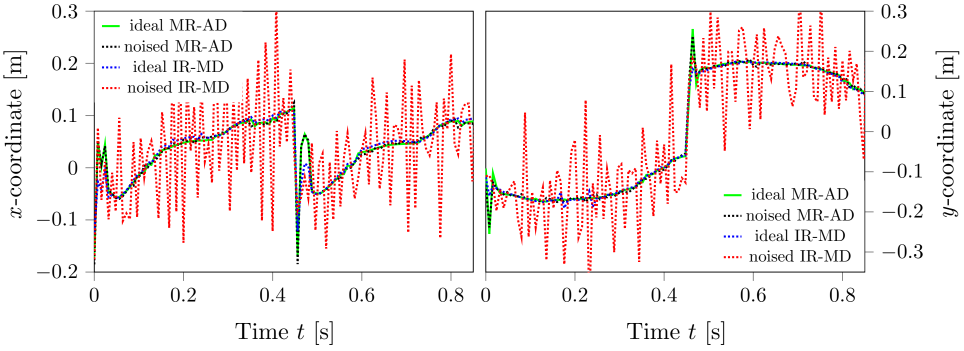

Fig. 8 Comparison between the results of the IZMP x- and y-coordinate of MR-AD and IR-MD with noised linear velocity, angular velocity and linear acceleration signals at a simulation time step of 0.008 s (corresponds to the field bus frequency of the real robot *Sweaty*). The gait speed is 0.3 m/s

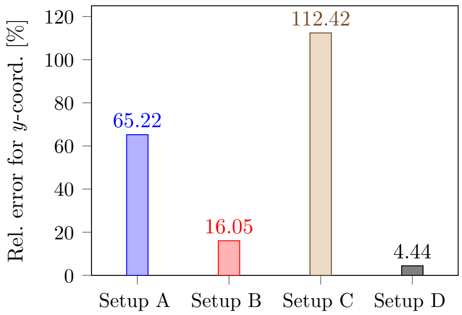

Fig. 9 Relative error comparison between MR-AD (Setup B and D) and IR-MD (Setup A and C). Setup A and B use a frequency of 125 Hz for ACD and field bus. Setup C and D use 500 Hz for ACD and 125 Hz for field bus. Setup B and D use the method for measuring of angular acceleration vector of the root segment. The relative error is calculated compared to the corresponding method with ideal signals and a frequency of 500 Hz for ACD and field bus

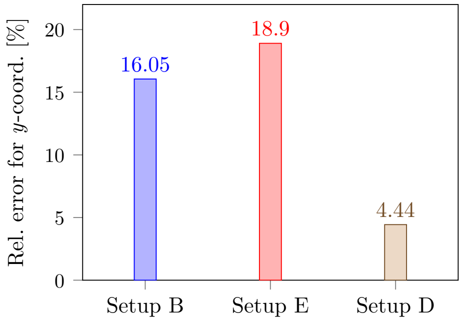

Fig. 10 Comparison between the different calculation methods. Setup B, D and E use MR-AD. Setup B uses a frequency of 125 Hz for ACD and field bus. Setup D and E use 500 Hz for ACD and 125 Hz for field bus. Setup E uses a differential quotient to calculate the angular acceleration vector of the root segment. Setup B and D use the method for measuring of angular acceleration vector of the root segment. The relative error is calculated compared to the corresponding method with ideal signals and a frequency of 500 Hz for ACD and field bus

Figure 10 compares the impact of the estimation of the angular acceleration on the IZMP y-coordinate. To analyze just the impact of the method providing the angular acceleration vector of the root segment, Setup B, D and E use the MR-AD. Setup E and B use 500 Hz for ACD and 125 Hz for the field bus frequency. Setup E uses a differential quotient of the measured angular velocity and Setup B and D use the angular acceleration measurement method to determine this vector of the root segment. The relative error is calculated compared to the corresponding method with the same angular acceleration calculation method, ideal input signals and a frequency of 500 Hz for ACD and field bus. The numerical differentiation leads to a higher relative error (18.9%) compared to the method using angular acceleration measurement (4.4%) under equal test case settings. Setup E has also a higher relative error compared to Setup B using the ACD frequency of 125 Hz and the angular acceleration measurement (16.1%), shown in Fig. 9.

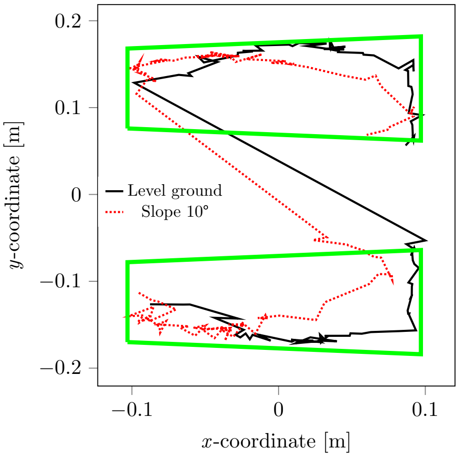

Fig. 11 The IZMP x- and y-coordinate calculated by MR-AD for the xy-plane and referenced in the evaluation frame while walking on a level surface and slope. The gait speed is 0.15 m/s. The slope angle is set to 10°. The support polygon for each foot is represented with the shown polygons

Next, we analyze the impact of walking onto a slope, especially for the IZMP calculation. The robot starts on level ground (slope angle 0°) and walks up an inclined plane with a fixed slope angle of 10° without stopping. For this purpose, the gait speed is 0.15 m/s.

Figure 11 shows the IZMP coordinates referenced in the evaluation frame and displayed in the xy-plane of the ground surface for one exemplary gait cycle. Here we use the time t as the parameter for the xy-plot. In Fig. 11 the solid curve indicates a gait cycle on the level surface, the dotted curve indicates a gait cycle walking on a slope (slope angle 10°). The polygons in Fig. 11 represent the support polygon of each foot referenced in the evaluation frame. Both support polygons are calculated with the kinematics in the respective stance phase. Due to the support polygons being represented in the NIF, both support polygons have the same x-coordinates. For the gait cycle on the level surface as well as for the gait cycle on the slope, it can be seen that IZMP curves are located inside the support polygon.

5.3 Real robot application

Figure 12 shows the development of the x- and y-coordinates of the IZMP calculate on the physical robot *Sweaty*. The actuator controllers return the actuator positions with a frequency of 1 kHz and the CAN-BUS frequency is set to 125 Hz. A differential quotient calculates the angular acceleration of the root segment. The robot walks on flat ground with a gait speed of 0.15 m/s. In comparison to the developments of the x- and y-coordinates of MR-AD shown in Fig. 8, these curves have a smoother transition in the change of the stance foot. The x-coordinate curve shows a larger slope followed by a relatively constant profile at about 0.1 m, compared with

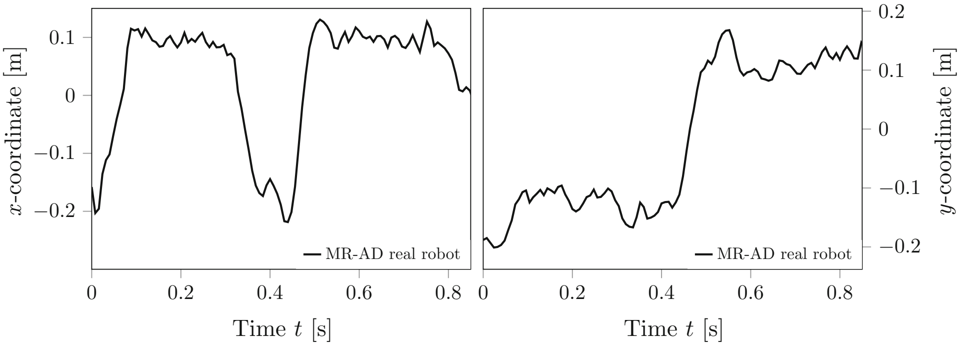

Fig. 12 IZMP x- and y-coordinate of MR-AD calculated on the physical robot *Sweaty*. The gait speed is 0.15 m/s

Fig. 8. Overall, the y-coordinate curve shows a similar but more inconstant development as seen in Fig. 8.

6 Discussion

Before we discuss our main findings, we start with the interpretation of the simulation characteristics. The convergence analysis (Fig. 7) shows for AD and MD methods linear convergence with respect to the simulation time step Δt. The simulator *Webots 2021b* uses a fixed time step for the physics engine. This time step is specified in milliseconds and can be set to an integer. Therefore, simulations are limited at $\Delta t = 0.001$ s (1000 Hz). With this time step, the simulation delivers no usable results simulating the robot *Sweaty* due to impaired communication between the *Webots-Controller-Client* and the other ROS2 nodes. Regarding this, we set simulation results generated with a time step of 0.002 s as the reference. However, due to the linear convergence, we consider it appropriate to define these curves as reference curves. Further, for the theoretical comparisons of all simulation results, we omit measurement inaccuracies of encoders and joint backlash as signal noise for q_i to exclusively analyze the influence of white noise of inertial sensors and velocity estimations.

Our first main findings are about the characteristics of the MD method. Generally, this approach reduces the complexity of the EOMs since only total linear and angular momentum have to be calculated. Consequently, IR-MD gets rid of joint acceleration $\ddot{q}_i$, linear and angular acceleration vectors of the root segment. However, it requires the linear velocity vector of the root segment. Likewise, taking numerical time derivatives, the calculation requires quantities of at least two discrete points in time. For MR-AD, the calculation requires quantities of one instant in time and no linear velocity vectors. Instead, the complexity of the calculation increases. These differences in calculating time derivatives cause the deviations in the development of IZMP coordinate curves shown in Fig. 6).

As far as we consider realistic noise behavior of physical sensor signals, e.g., for the linear velocity or the position vector, the results of the MD method are no longer usable (see Figs. 8, 9). This is caused by the dependency on the quantities from different instances in time. Due to this characteristic, the white noise behavior of real sensors has a direct impact on the results. By using discrete sensor signal values, the dynamics calculation inevitably includes at least two different noise terms. Due to the randomness of the underlying normal distribution of the noise, the numerical time derivatives could determine physically unrealistic gradients of the momentum. Figures 8 and 9 exactly show this behavior.

Further, measuring linear velocity is impractical since a direct measurement method is not feasible. Thus, the analytically obtained derivatives have an obvious advantage as far as noisy signals are concerned. Consequently, they are significantly better suited for calculating the dynamics of physical robotic systems.

Second, for physical robots, it is commonly feasible to operate sensors and actuator controllers with significantly higher sample rates than the field bus. A numerical derivative can calculate the joint velocity $\dot{q}_i$ and joint acceleration $\ddot{q}_i$ using the joint position q_i provided by the encoder. Figure 9 shows the impact of higher sample rates. Due to the smaller time step, both methods benefit theoretically by the increased accuracy of the calculated $\dot{q}_i$ and $\ddot{q}_i$ (see. Figure 7). Additionally, the measurement error of incremental encoders does not scale proportionally with the square root of the sampling frequency compared to, e.g., inertial sensors used (see Sect. 4.6).

However, the numerical time derivative of IR-MD calculates even larger gradients for the momentum in terms of magnitude caused by the combination of the noise-amplifying numerical differentiation and the white noise in the sensor signals. Figure 9 shows finally that the influence of the larger inaccuracies due to the white noise in the sensor signals in combination with the numerical time derivatives dominates. The relative error increases significantly compared to the ideal reference. In contrast, MR-AD benefits through higher sample rates for sensors and actuator controllers due to increased accuracy of $\dot{q}_i$ and $\ddot{q}_i$. It is further significantly less impacted through white noise than the MD method, as seen in Fig. 8.

Based on the results shown in Fig. 9, we propose to execute this calculation directly in the actuator controllers. In general, the frequency of the field bus from robots depends mostly on the total number of motors and sensors connected to the field bus. Typically frequencies of field buses (e.g., *CAN* or *EtherCAT*) for robots with more than 30 degrees of freedom are in the range of 100–1000 Hz.

For physical robots, the higher frequency of the actuator controllers is an advantage regarding the accuracy of $\dot{q}_i$ and $\ddot{q}_i$. The errors of the numerical derivative approximation of $\dot{q}_i$ and $\ddot{q}_i$ decrease due to smaller Δt. However, numerical differentiation with Δt tending to zero generally amplifies signal noise. Therefore, e.g., implemented filtering methods in the actuator controller have to compensate for the noise-amplifying effects. Through the significantly higher actuator controller frequency, we can design filtering methods with adapted delays so that they have no effects on the dynamics calculations executed in field bus frequency. Consequently, in this case we do not use the Δt of the field bus frequency in any calculation step. Therefore, it has no impact on the accuracy of the dynamics calculation.

Third, in contrast to the MD method, analytical derivative EOMs generally require the angular acceleration vector of the root segment. In this work, we compare two different approaches to assess angular acceleration. On the one hand, a numerical derivative of the angular velocity signal, measured by IMU, is used. On the other hand, we present a measurement through three additional uniaxial accelerometers.

For calculating the angular acceleration of the root segment, a noise-amplifying numerical differentiation has the same disadvantages as mentioned before. The increment Δt used depends on the sampling rate of the sensor. Noise in the estimated angular velocity signal combined with high sampling rates could result in unrealistic angular acceleration approximations. Figure 10 confirms this. Low sampling rates would lead to inaccuracy for angular acceleration due to the increasing Δt between two measured quantities. A decreasing Δt through higher sampling rates increases the noise-amplifying effects of the differential quotient used.

Since the calculation method from three additional accelerometers does not involve numerical differentiation, the accuracy gets independent of Δt or the physical sensor update frequency. The disadvantages previously described do not apply. However, this includes four additional noisy signals in the calculation, which affects the measurement of the angular acceleration. Equations 27, 28, and 29 show that the components of angular acceleration depend on the distance between the accelerometers. For example, if the distance r_2 tends to zero, the measured values a_{2z} and a_{0z} converge. The noise behavior is then the dominant difference between these measurements. Thus $\dot{\omega}_x$ would be mostly influenced by the noise.

Here, we want to discuss the use of three additional accelerometers to determine all joint velocities $\dot{q}_i$ and acceleration $\ddot{q}_i$ of the robot. The above mentioned arguments against numerical differentiation apply analogously. One consequence would be to omit numerical differentiation entirely when calculating joint kinematics. Further, it is thus independent of the actuator controller and field bus frequencies. A disadvantage is the increasing number of installed sensors and the transmitted data.

Fourth, MR-AD and all analytical derivative methods are based on the same basic physical laws. Provided that consistent inputs are given for the calculation, the results must be identical. Compared to our MR-AD, the mentioned analytical derivative methods require the absolute position vector of the root segment from a spatially fixed reference point as an input.

Henze et al. (2016) estimate the absolute position vector of the root segment by sensor fusion. Therefore, this kind of estimation requires the kinematics as well as the absolute position vector and orientation of each end effector in contact. One assumption required for the proposed sensor fusion is a rigid contact of at least one body segment.

Despite the redundant calculation and the introduced averaging method in the approach of Henze et al. (2016), this sensor fusion always contains inaccuracies due to non-present rigid contact and white noise of the IMU sensor signals. Also, the continuous determination of the absolute position vectors of all end effectors contains inaccuracies, especially on unknown, uneven terrain and through undefined contacts. This kind of estimation for the absolute position vector of the root segment is, in our view, error-prone.

The approach of Buschmann (2011) avoids numerical inaccuracies through large distances from the initial reference point to the robot by shifting the reference frame depending on the gait cycle. In general, this reference frame does not coincide with the BFF of the stance foot and the origin is fixed. This means the position vector of the COM cannot be calculated only through kinematics on physical robots. Buschmann does not propose an estimation of the global position vector of the COM, since the work aims to simulate robots.

In contrast, our approach leads to high flexibility in choosing the reference point and the reference frame for dynamics calculations. The reference point can be any material point, e.g., the origin of the root segment or any immaterial point. The state of motion of this point compared to the other material points of the robot is irrelevant. This leads to the omission of estimating absolute position vectors. We made this effort to avoid all estimations since they can cause inaccuracies in the results and for saving computational capacity. E.g., if we choose the origin of the root segments BFF as the moving reference point, we can calculate relative position vectors of arbitrary material points through kinematics without inaccuracies due to uneven ground or undefined contact. Consequently, conditions with no rigid body contact of at least one segment or in flight phases do not disturb our used EOM evaluation approach.

Furthermore, the spatial distance between the evaluation frame and all BFFs stays always in a similar order of magnitude. Similarly, the recommendation of Featherstone (2008) for the distance of one to two radii of gyration is continuously satisfied. This reduces numerical inaccuracy. We further use an algorithm for calculating the relative derivative from numerically available values, which allows us to derive physical quantities in moving frames. Specifically, we use it to calculate the relative time derivative of the angular velocity for all body segments. With this, we evaluate the dynamics for each body segment in its BFF. This is an efficient way to evaluate the EOMs according to Featherstone (2008).

Fifth, a further advantage of using our approach is seen for the IZMP calculation walking, e.g., on slopes. In general, one axis of the evaluation frame must point parallel to the normal vector of the ground surface for calculating the IZMP. This is required to resolve the under-determination of the system of coordinate equations from Eq. 22 through determining the z-coordinate. This demand cannot be fulfilled by the compared approaches. Consequently, they do not provide usable results without further adjustments in calculating the IZMP on slopes. For valid results, the compared approaches would require a transition of the inertial frame. Likewise, the equation to calculate the IZMP, as, e.g., shown by Kajita et al. (2014), also has to be adapted to the rotated gravity vector.

For evaluating the EOMs in a MR, this condition for the IZMP calculation remains fulfilled. Here, the evaluation frame rotates in a suitable orientation to cover the slope to ensure that the required axis parallelism remains fulfilled. IMUs placed in the feet measure the orientation of the surface during the stance phase in each gait cycle.

The comparison of the resulting IZMP coordinates for the level ground and the slope shows that they are located within the support polygons as seen in Fig. 11. Our approach evaluates the locomotion on the slope as stable.

Consequently, with our approach, the reference point is adjustable to the implemented control algorithm. E.g., if we use an IZMP-based control algorithm, the moving reference could coincide with the surface plane to be independent of friction forces. Or, if we control the ground reaction forces, it may be efficient to place the reference point into the moving root segment. The calculated quantities can thus directly trigger a reaction without additional spatial transformation.

Sixth, if we compare the IZMP coordinates of the physical robot shown in Fig. 12 with the simulation results shown in Fig. 6, we recognize generally similar curve developments. The main differing characteristics of the coordinate curves from physical robot to simulation are, on the one hand, the more inconstant development and the noticeably smoother transition between the change of the stance foot. The joint backlash present and measurement errors of the encoders on physical robots, the elasticities, and manufacturing and assembly inaccuracies mainly cause these two characteristics. In conclusion, despite the mechanical inadequacies and white noise of all sensor signals, MR-AD is suitable to calculate the IZMP also for physical robots. If the proposed method for measuring angular acceleration vectors is applied to all segments of the robot, we expect that the results of dynamics calculation on real robots will be further improved.

Further, Fig. 12 shows the justification that omitting the measurement errors of the encoders and the joint backlash is sensible in the theoretical comparisons of the applied methods. The impact of the measurement errors of the encoders, the joint backlash, and other mechanical inaccuracies on the IZMP curves is much lower than the impact of the numerical differentiation of the momentum equations with noisy linear and angular velocity signals.

As we mentioned previously, there are disturbances in the dynamics calculation in general. These include inaccuracies in the mechanical model compared to the real robot, e.g., due to deviating mounting positions of the joint axes, inertia data or the underlying rigid body hypothesis, to name a few. To reduce the impact of inaccuracies, we propose combining a sensor fusion from our approach with a 6-axis force/torque sensor and adding a Kalman filter. A limitation of the sensor fusion is that it only provides valid data for a bipedal walking without external perturbations applied to the robot.

7 Conclusion and future work

From the perspective of the authors, the combination of evaluating the EOMs formulated through the Newton–Euler equations with respect to a moving reference point in a NIF and measuring the angular acceleration vector of the root segment contributes to an efficient and robust evaluation of the inverse dynamics in robotics. The entire proposed approach produces comparably similar results as the previous approaches evaluating the EOMs with respect to a fixed point in the inertial frame or the center of mass for a humanoid robot, with the main benefit that the proposed approach is less prone to noise than the previous approaches.

We confirm that in theory and by comparing it with the mentioned approaches using simulation results. In theory, we show the conceptual advantages of our approach. E.g., it gets rid on the dependency of any global position vector. Further, we show the flexibility of choosing the reference point and the evaluation frame concerning the preferences of the control algorithm used.

The comparison of simulation results clearly shows further advantages for the application to physical robots. On the one hand, we reduce the quantity of noise-amplifying numerical differentiation. On the other hand, with our approach, we can calculate the IZMP for the gait from the level to the inclined surface. We also show that our algorithm calculates usable results applied to physical robots.

It is also worth mentioning that this approach is not limited to bipedal robots or humanoid robots in general. Due to the generic structure, the EOMs can be calculated for multi-legged robots or human beings. The adaption to KDL and the integration in ROS2 of this algorithm support straightforward usability.

Next, we will equip all segments of *Sweaty* with one sensor unit combining an IMU and three additional uniaxial accelerometers to eliminate all numerical differentiation and use our approach to evaluate the stability of the robot *Sweaty*. As a second step, we will setup up the sensor fusion with the self-developed 6-axis force/torque sensor and we will try to implement a Kalman filter. Further, we will try to use the sensor unit also to evaluate the stability of human beings.

Finally, we will use this approach as the basis of a control algorithm to stabilize both static situations and dynamic locomotion. We want to align this controller specifically to the control paradigms of human beings, taking benefit of the identified characteristics of our approach.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors have no relevant financial or non-financial interests to disclose. The project *Sweaty* of the University of Applied Sciences Offenburg is a non-profit project. The objective is to contribute to the research field of humanoid robotics. Regional companies sponsor equipment. However, there is no influence on or exchange of research results.

References

Anitescu, M., & Potra, F. A. (1997). Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics, 14*(3), 231–247. https://doi.org/10.1023/A:1008292328909

Buschmann, T. (2011). *Simulation and control of biped walking robots* (1 Aufl ed.). München: Verl. Dr. Hut.

Englsberger, J. , Ott, C. , Roa, M.A. , Albu-Schaffer, A. , & Hirzinger, G. (2011). Bipedal walking control based on Capture Point dynamics. In *2011 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4420–4427). CAIEEE. https://doi.org/10.1109/IROS.2011.6094435

Featherstone, R. (2008). Dynamics of rigid body systems. R. Featherstone (Ed.), Rigid body dynamics algorithms (pp. 39–64). Springer. https://doi.org/10.1007/978-1-4899-7560-7_3

Goswami, A. (1999a). Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots. In *Proceedings 1999 IEEE international conference on robotics and automation* (Cat. No.99CH36288C) (vol. 1, pp. 47–52). USAIEEE. https://doi.org/10.1109/ROBOT.1999.769929

Goswami, A. (1999b). Postural stability of biped robots and the foot-rotation indicator (FRI) point. *The International Journal of Robotics Research, 18*(6), 523–533. https://doi.org/10.1177/02783649922066376

Goswami, A. , & Kallem, V. (2004). Rate of change of angular momentum and balance maintenance of biped robots. In IEEE international conference on robotics and automation, 2004 proceedings (ICRA'04). 2004 (vol.4, pp. 3785–3790). USAIEEE. https://doi.org/10.1109/ROBOT.2004.1308858

Henze, B., Roa, M. A., & Ott, C. (2016). Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *The International Journal of Robotics Research, 35*(12), 1522–1543. https://doi.org/10.1177/0278364916653815

Hirukawa, H. , Hattori, S. , Harada, K. , Kajita, S. , Kaneko, K. , Kanehiro, F., & Morisawa, M. (2006). A universal stability criterion of the foot contact of legged robots—Adios ZMP. In *Proceedings 2006 IEEE international conference on robotics and automation, 2006 (ICRA 2006)* (pp. 1976–1983). USAIEEE. https://doi.org/10.1109/ROBOT.2006.1641995

InJoon, M. , DongHa, Y. , MinSung, A. , & Jeakweon, H. (2020). Body trajectory generation using quadratic programming in bipedal robots. In *2020 20th international conference on control, automation and systems (ICCAS)* (pp. 251–257). IEEE. https://doi.org/10.23919/ICCAS50221.2020.9268204

Kajita, S., Hirukawa, H., Harada, K., & Yokoi, K. (2014). *Introduction to humanoid robotics* (vol. 101). Springer. https://doi.org/10.1007/978-3-642-54536-8

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003)* (cat. no.03ch37453) (vol. 2, pp. 1644–1650). https://doi.org/10.1109/IROS.2003.1248880

Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., & Yokoi, K. (2010). Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4489–4496). IEEE. https://doi.org/10.1109/IROS.2010.5651082

Kamioka, T., Kaneko, H., Takenaka, T., & Yoshiike, T. (2018). Simultaneous optimization of ZMP and footsteps based on the analytical solution of divergent component of motion. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 1763–1770). IEEE. https://doi.org/10.1109/ICRA.2018.8460572

Kimura, K. , Nozawa, S., Mizohana, H., Okada, K., & Inaba, M. (2018). Riding and speed governing for parallel two-wheeled scooter based on sequential online learning control by humanoid robot. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1–9). IEEE. https://doi.org/10.1109/IROS.2018.8593685

Mehrtens, G. (1887). *Technische Mechanik fester, flüssiger und luftförmiger Körper*(vol. 3). Kommissions-Verlag v. E. Toeche.

Mesesan, G., Englsberger, J., & Ott, C. (2021). Online DCM trajectory adaptation for push and stumble recovery during humanoid locomotion. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 12780–12786). IEEE. https://doi.org/10.1109/ICRA48506.2021.9560808

Padgaonkar, A. J., Krieger, K. W., & King, A. I. (1975). Measurement of angular acceleration of a rigid body using linear accelerometers. *Journal of Applied Mechanics, 42*(3), 552–556. https://doi.org/10.1115/1.3423640

Pang, J. , & Trinkle, J. (2000). Stability characterizations of fixtured rigid bodies with coulomb friction. In *Proceedings 2000 icra. millennium conference. IEEE international conference on robotics and automation* (vol. 1, pp. 361–368). IEEE. https://doi.org/10.1109/ROBOT.2000.844083

Pratt, J., Carff, J., Drakunov, S., & Goswami, A. (2006). Capture Point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS international conference on humanoid robots* (pp. 200–207). University of Genova, IEEE. https://doi.org/10.1109/ICHR.2006.321385

Saida, T., Yokokohji, Y., & Yoshikawa, T. (2003). FSW (feasible solution of wrench) for multi-legged robots. In *2003 IEEE international conference on robotics and automation* (Cat. No.03CH37422) (vol. 3, pp. 3815–3820). IEEE. https://doi.org/10.1109/ROBOT.2003.1242182

Sardain, P., & Bessonnet, G. (2004). Forces acting on a biped robot. Center of pressure–Zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 34*(5), 630–637. https://doi.org/10.1109/TSMCA.2004.832811

Vijaykumar Shah, S., Kumar Saha, S., & Kumar Dutt, J. (2013). *Dynamics of tree-type robotic systems* (vol. 62). Springer.

Vukobratovic, M., Borovac, B., & Šurdilović, D. (2001). Zero-moment point—Proper interpretation and new applications. In *IEEE-RAS international conference on Humanodi robots* (pp. 237–244). IEEE.

Vukobratovic, M., Frank, A. A., & Juricic, D. (1970). On the stability of biped locomotion. *IEEE Transactions on Biomedical EngineeringBME, 17*(1), 25–36. https://doi.org/10.1109/TBME.1970.4502681

Vukobratović, M., & Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical Biosciences, 15*(1–2), 1–37. https://doi.org/10.1016/0025-5564(72)90061-2

Wang, H. , Tian, Z., Hu, W., & Zhao, M. (2018). Human-like ZMP generator and walking stabilizer based on divergent component of motion. In *2018 IEEE-RAS 18th international conference on humanoid robots (Humanoids)* (pp. 1–9). IEEE. https://doi.org/10.1109/HUMANOIDS.2018.8624926

Wang, P., Liu, J., Zha, F., Guo, W., Wang, X., Li, M., & Sun, L. (2017). A velocity estimation algorithm for legged robot. *Advances in Mechanical Engineering, 9*(12), 1687–8140. https://doi.org/10.1177/1687814017732736

Wittenburg, J. (1977). *Dynamics of systems of rigid bodies*. Vieweg+Teubner Verlag. https://doi.org/10.1007/978-3-322-90942-8

Woernle, C. (2016). *Mehrkörpersysteme: eine Einführung in die Kinematik und Dynamik von Systemen starrer Körper* (2., erweiterte Auflage ed.). Springer Vieweg.

Maximilian Gießler received his B.Eng. in 2018 and M.Sc. in 2020 in Mechanical Engineering from the Offenburg University of Applied Sciences. He is currently a Ph.D. student and employed as a scientific assistant at the Offenburg University of Applied Sciences. His research interests include multibody dynamics and humanoid robotics, with an emphasis on bipedal locomotion and stability theory.

Bernd Waltersberger has been a professor at the Faculty of Mechanical and Process Engineering at the Offenburg University of Applied Sciences since 2010. He received his diploma in mechanical engineering in 2002 from the University of Karlsruhe (now K.I.T) and was awarded his doctorate there in 2007 for a thesis on structural mechanics. He subsequently worked as a development engineer in the automotive field and is co-inventor on 7 patent applications. His research interests are in the dynamic interaction of machine elements and multibody dynamics, especially in the area of vibration engineering and stability theory.