Dynamic modeling of a human-inspired robot based on a Newton-Euler approach

Fernando Gonçalves[1], Tiago Ribeiro[2], A. Fernando Ribeiro[2], Gil Lopes[3], and Paulo Flores[1]

[1] Department of Mechanical Engineering, University of Minho, Guimarães, Portugal
`id8699@alunos.uminho.pt` (F.G.)

[2] Department of Industrial Electronics, University of Minho, Guimarães, Portugal

[3] Department of Communication Sciences and Information Technologies,
University of Maia, Maia, Portugal

Abstract. This work deals with the modeling process of a new three dimensional human-like robot for an inverse dynamic analysis. This robot intends to be utilized by caregivers to assist persons with reduced mobility (such as the elderly). The model under analysis is composed by 24 rigid bodies: 3 to represent the robot's base and locomotion, 4 for the lower limbs and torso, 7 for each arm, and 3 for the head. The resulting multibody system has 19 degrees-of-freedom driven by 4 linear actuators and 15 revolute motors. The proposed approach was implemented using an in-house computational code, and validated against a commercial software for a general spatial motion. The outcomes achieved show that the proposed formulation is computationally effective both in terms of efficiency and accuracy. The general findings of this study are promising and useful for the mechanical design and construction of a real human-like robot prototype.

Keywords: Human-inspired robot, Multibody dynamics, Newton-Euler formulation

1 Introduction

Inverse dynamics refers to the process of determining the forces and torques applied in a body, or set of bodies, that result in a known motion (kinematics). There are three main formulations for studying a dynamics problem, all of them used in the multibody analysis of robots: Newton-Euler equations [1,2], Lagrange equations [3], and Hamilton equations [4,5]. Due to their simplicity and ease of implementation, recursive Newton-Euler algorithms are favored in the dynamic analysis of robotic systems [6].

Dynamic studies in computational environments often use one of two main methods: implementing the problem in an already developed multibody dynamics simulation software, or using computational tools to solve the equations of motion defined manually by the researcher. Simulation software are the most common resource for robotics research [7] due to: their simplicity, their convenient tools and interfaces, and their capability to tackle complex problems without requiring more in-depth studies. Implementing the equations of motion on an integrated development environment is a more laborious process, but it provides its own set of advantages. The resulting models have higher compatibility with other software, they provide more flexible solutions (giving more freedom to researchers), and they can be more thoroughly optimized for specific problems.

This study extends the authors' previous work [8] to include a complete dynamic analysis of CHARMIE, a colaborative robot for elderly care [9]. The main novelty of this work is the use of an in-house code to adapt, apply and validate the recursive algorithm of [6] for obtaining the multibody model of the considered robot. This model will play a pivotal role in optimizing the mechanical design and developing the physical prototype of the robot.

2 Description of the multibody model

The multibody model of CHARMIE is composed of 24 rigid bodies connected by 35 joints — 4 prismatic and 31 revolute — in a configuration that results in 19 degrees-of-freedom (Figure 1.b). The system is fully actuated, driven by 4 linear actuators and 15 revolute motors. Additionally, tension springs apply forces to reduce actuator loads: two identical springs are placed between bodies 3 and 4, and one between bodies 4 and 5, with 105.5; 77 mm free lenght (L_0), 1380; 1029 N/m spring rate (k), and 27.1; 15.1 N initial force (F_i) respectively.

Figure 1 labels the bodies of the multibody system, showing that 6a-12a correspond to the left arm, 6b-12b to the right arm, 6c-8c to the head, and 4 and 5 to the torso. Bodies 4_{sl} and 4_{sr} are auxiliary, used for the application of the tension spring between bodies 3 and 4. Bodies 1 to 3 correspond to the robot's base and locomotion. At this stage, the behaviour of the omnidirectional wheels and suspension system was not considered. Bodies 1 and 2 (not represented in the CAD model) are massless fictitious links that allow the robot's base to slide and rotate while traveling along the floor plane (using 2 linear actuators and 1 revolute motor). The work in [8] details the degrees-of-freedom associated with each main joint of the robot. From the CAD model, the masses and inertia of all the robot's bodies were obtained, listed on Table 1.

The robot's inverse dynamics were analysed using two methods. The first was an in-house computational code that applied an altered version of the recursive Newton-Euler algorithm presented in [10]; the in-house code was programmed in Python, using the numpy and matplotlib libraries for mathematical operations and to obtain a visual representation of the robot. The second was a commercial simulation software. The resulting computational models are shown in Figure 2.

In Figure 2.a the robot is in its starting position (zero position reference). In this configuration, all revolute actuators on the arms and head have a rotation of 0, and the robot's base is aligned with the global reference and placed in its origin. In this setup, the two linear actuators have a total length of 480 mm (lower actuator) and 400 mm (upper actuator).

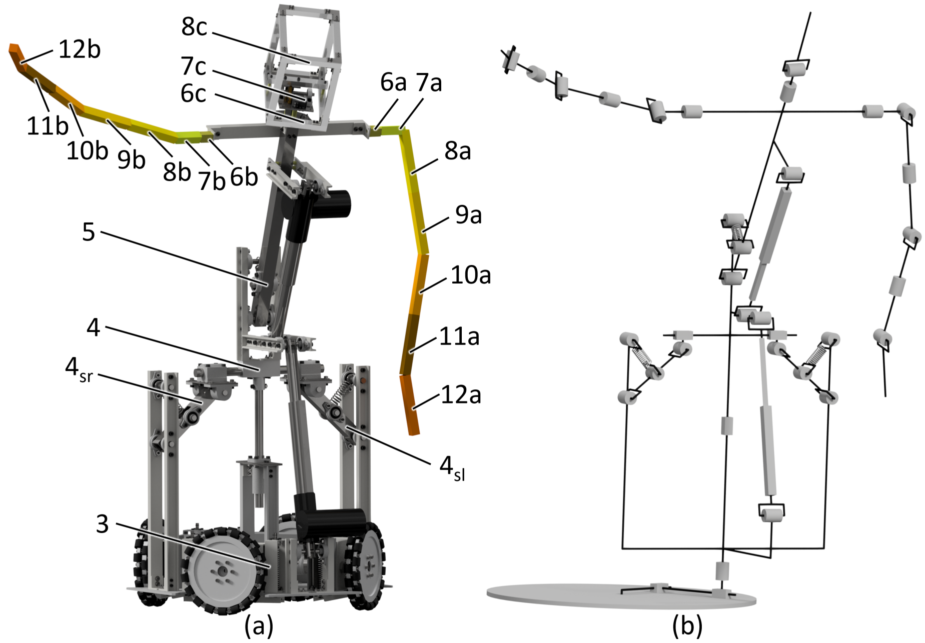

Fig. 1. The CHARMIE robot: (a) CAD model with its bodies labeled, (b) kinematic diagram

Table 1. Geometric and inertia properties of the multibody system. The origins are defined in relation to the local coordinates of the preceding link in the kinematic chain.

Body	Origin	Mass	Inertia [kg.mm^2]					
	[mm]	[kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}
1	[d$_1$, 0, 0]	0	0	0	0	0	0	0
2	[0, d$_2$, 0]	0	0	0	0	0	0	0
3	[0, 0, 0]	37.1	1187000	1304000	1610000	11450	3586	−51860
$4_{sr} = 4_{sl}$	[±245, 0, 341]	0	0	0	0	0	0	0
4	[0, 0, 290 + d$_4$]	7.5	118300	167400	94970	19.62	1.907	−7536
5	[0, 0, 434]	4.6	77230	80350	32960	2316	-229.9	4803
6a = 6b	[±200, 3, 460]	0.3	32.39	32.39	19.93	0	0	0
7a = 7b	[0, 0, 30]	0.3	102.5	102.5	19.61	0	0	0
8a = 8b	[0, 0, 61.5]	0.4	710.8	710.8	26.54	0	0	0
9a = 9b	[0, 0, 145]	0.4	710.8	710.8	26.54	0	0	0
10a = 10b	[0, 0, 145]	0.4	663.9	663.9	26.56	0	0	0
11a = 11b	[0, 0, 140]	0.4	663.9	663.9	26.56	0	0	0
12a = 12b	[0, 0, 140]	0.3	498.6	498.6	19.94	0	0	0
6c	[0, 0, 495]	0.3	145.8	132.1	246.5	0	0	4.732
7c	[0, 0, 46.3]	0.4	296.0	268.2	522.0	2.606	−0.822	6.866
8c	[14.8, −5.3, 18.3]	0.7	8548	7189	4981	31.79	143.1	−222.0

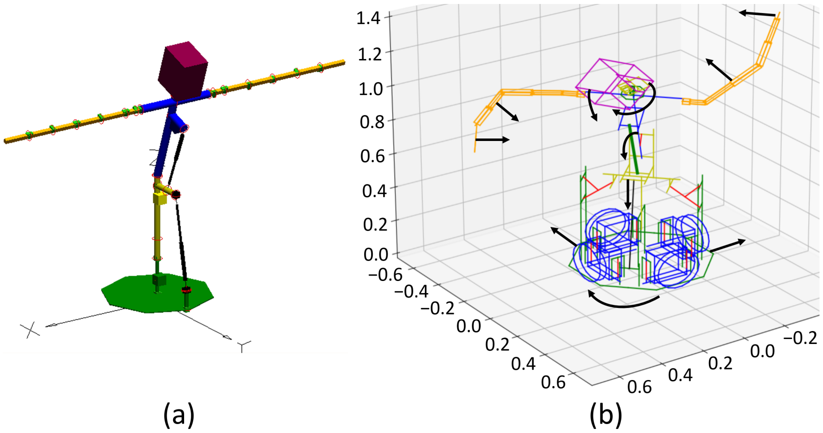

Fig. 2. Computational models of the CHARMIE robot's dynamics: (a) in a commercial multibody dynamic analysis software (frame t = 0s), (b) in the in-house developed code (frame t = 1.5s) with bold arrows representing the main directions of motion.

For comparison, the same generic motion was studied in both environments. All joints started with zero velocity. The motion lasted for 5 seconds, and was controlled via the accelerations of the 19 actuators. The accelerations were defined as: $0.032\ \mathrm{m/s^2}$ for the base x and y linear motions, $2\pi/9\ \mathrm{rad/s^2}$ for the base rotation around the z axis, $0.0032\ \mathrm{m/s^2}$ for the linear actuators on the body, and $2\pi/9\ \mathrm{rad/s^2}$ for the revolute motors on the arms and head. These accelerations were multiplied with a sign function that has a negative value for the first half of the simulation, and positive for the second half.

3 Equations of motion

The formulation of multibody system dynamics adopted in this work follows closely that of [10], in which a recursive Newton-Euler algorithm is used to derive the spatial system's equations of motion. These equations were implemented using the outputs from the forward kinematic analysis of [8] (orientations defined by rotation matrices, point coordinates, joint positions, velocities and accelerations).

The algorithm is divided into two stages. The first stage is preparatory, and converts the data from the kinematic analysis into the relevant inputs for the dynamic calculations (body linear acceleration, angular velocity, and angular acceleration). Iterations progress from the global reference (body 0) to the robotic end-effector, modeling the behaviour of each body using the calculated properties of the previous kinematic link $i-1$, and the relative motion between link i and link $i-1$. The calculation of the linear accelerations includes the Coriolis, centrifugal, and Euler accelerations for computing the fictitious forces.

The second stage uses the information from the first stage to determine the reaction forces and torques applied by each body $i-1$ on body i, progressing from the end-effectors to the robot's base. To allow tackling more complex structures (not only serial assemblies of links) the original algorithm had to be altered. The modified formulation can directly tackle any body which has only one unknown reaction force and moment vector (applied by link $i-1$). If this condition is not met, a manual analysis is required (such is the case for bodies 4 and 5 of the CHARMIE robot). To simplify the computational implementation, forces and torques applied by link i on links $i+1$ are considered positive, while forces applied by links $i-1$ on link i are negative. In the following equations, the superscript next to each variable represents the reference frame they are expressed in.

The sum of the n_a unknown $\boldsymbol{f}^i_{ja}$ forces applied by the ja links $i-1$ on link i is determined using Euler's first law of motion, written as:

$$\sum_{ja=1}^{n_a} \boldsymbol{f}^i_{ja} = m_i \ddot{\boldsymbol{p}}^i_{Ci} + \sum_{jb=1}^{n_b} \left(\boldsymbol{R}^i_{jb} \boldsymbol{f}^{jb}_{jb} \right) \tag{1}$$

where m_i is the mass of body i and $\ddot{\boldsymbol{p}}^i_{Ci}$ the linear acceleration of its center of mass. The n_b forces applied by bodies $\boldsymbol{f}^{jb}_{jb}$ in body i are determined in the previous iterations $i+1$, so they are expressed in the local coordinates of their respective bodies. Before adding these forces, they are rotated to the orientation of body i using the $\boldsymbol{R}^i_{jb}$ rotation matrices.

In order to determine the sum of the n_a reaction torques $\boldsymbol{\mu}^i_{ja}$ applied by the j_a links $i-1$ on link i, Euler's second law of motion is used:

$$\begin{aligned} \sum_{ja=1}^{n_a} \boldsymbol{\mu}^i_{ja} =& \bar{\boldsymbol{I}}^i_i \dot{\boldsymbol{\omega}}^i_i + \boldsymbol{\omega}^i_i \times (\bar{\boldsymbol{I}}^i_i \boldsymbol{\omega}^i_i) + \sum_{ja=1}^{n_a} \left(-\boldsymbol{f}^i_{ja} \times \boldsymbol{r}^i_{ja,Ci} \right) \\ &+ \sum_{jb=1}^{n_b} \left(\boldsymbol{R}^i_{jb} \boldsymbol{f}^{jb}_{jb} \times \boldsymbol{r}^i_{jb,Ci} + \boldsymbol{R}^i_{jb} \boldsymbol{\mu}^{jb}_{jb} \right) \end{aligned} \tag{2}$$

The change in angular momentum of body i is determined from the inertia matrix $\bar{\boldsymbol{I}}^i_i$, its angular velocity $\boldsymbol{\omega}^i_i$, and its angular acceleration $\dot{\boldsymbol{\omega}}^i_i$. The sum of torques considers both directly applied torques, as well as the effect from forces not aligned with the body's center of mass. This results in three known groups of torques: the sum of the n_a cross products between the $\boldsymbol{f}^i_{ja}$ forces and the respective $\boldsymbol{r}^i_{ja,Ci}$ vectors that define the displacement from the forces' point of application to the body's center of mass; the sum of the n_b cross products between the $\boldsymbol{f}^{jb}_{jb}$ forces converted to the orientation of i using the $\boldsymbol{R}^i_{jb}$ rotation matrices, and the respective $\boldsymbol{r}^i_{jb,Ci}$ vectors that define the displacement from the forces' point of application to the body's center of mass; and the sum of the $\boldsymbol{\mu}^i_{jb}$ torques applied by body i in bodies j_b with their orientations converted to that of body i using the $\boldsymbol{R}^i_{jb}$ rotation matrices.

4 Inverse dynamics for closed and overconstrained loops

The equations of motion automatically deduce the applied forces and torques if a body has only one unknown reaction force and moment vector at its base when traversing the chain from the tip to the base. However, if this condition is not met (in closed or overconstrained loops), only the sum of unknown forces is calculated, and a more detailed study must determine their distribution. This occurs for bodies 4 and 5 of CHARMIE. For simplification, these reactions were analysed based on the mechanism formed by these bodies and the tolerances of their joints.

Since this section tackles specific sections of the robot, not generalizable systems, a different nomenclature is used to facilitate reading the equations. The forces and torques are expressed using their x, y and z components, and the orientations between bodies is defined using a single angle (there is only rotation around a single axis between two consecutive bodies). Euclidean vectors associated with distances are highlighted by accenting the corresponding variable with a right arrow. Most equations used in this sections result from developing a single line of equations (1) or (2). The notation used in both sections is interchangeable using the following equations:

$$\begin{bmatrix} Fx_{jb} \\ Fy_{jb} \\ Fz_{jb} \end{bmatrix} = \boldsymbol{f}^{jb}_{jb}\boldsymbol{R}^{i}_{jb}; \quad \begin{bmatrix} \mu x_{jb} \\ \mu y_{jb} \\ \mu z_{jb} \end{bmatrix} = \boldsymbol{\mu}^{jb}_{jb}\boldsymbol{R}^{i}_{jb} \tag{3}$$

for forces and torques applied by link i on the links $i+1$, and:

$$\begin{bmatrix} Fx_{ja} \\ Fy_{ja} \\ Fz_{ja} \end{bmatrix} = \boldsymbol{f}^{i}_{ja}; \quad \begin{bmatrix} \mu x_{ja} \\ \mu y_{ja} \\ \mu z_{ja} \end{bmatrix} = \boldsymbol{\mu}^{i}_{ja} \tag{4}$$

for the forces and torques applied by links $i-1$ on link i.

Body 5 of CHARMIE (Figure 3) has three unknown forces/torques applied by: the tension spring $s5$, the revolute joint $j5$, and the linear actuator $act5$. Their distribution was determined using the following seven steps:

1. The tangential $Ft_{\mathrm{s}5}$ component of $s5$ can be determined directly from the spring initial force F_i, spring rate k and it's extension $(L_{\mathrm{s}5} - (L_0)_{\mathrm{s}5})$. This force's y and z components are determined from the spring orientation β_5. The spring does not apply any torque directly.

$$Ft_{\mathrm{s}5} = (F_i)_{\mathrm{s}5} + k_{\mathrm{s}5}(L_{\mathrm{s}5} - (L_0)_{\mathrm{s}5}) \tag{5}$$

$$Fy_{\mathrm{s}5} = Ft_{\mathrm{s}5}\sin(\beta_5); \quad Fz_{\mathrm{s}5} = Ft_{\mathrm{s}5}\cos(\beta_5) \tag{6}$$

2. Both j_5 and act_5 are revolute joints around the x axis, so they do not apply any torque on it. The force from act_5 must nullify the remaining μ_x torque around j_5; this force's component on the yz plane must be tangential to the

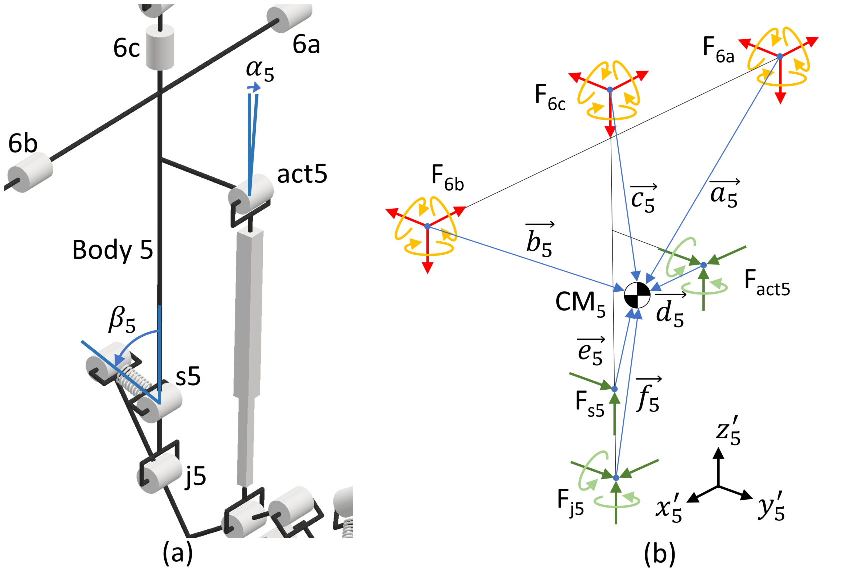

Fig. 3. Kinematic representation of body 5 of the CHARMIE robot (a) and corresponding free body diagram (b). In (b), known forces and torques are colored in red/orange, unknown forces are colored green, and displacement vectors in blue. The $x_5'y_5'z_5'$ axes represent the orientations used in all calculations of body 5

actuator's orientation. The following three equations allow determining the y and z components of F_{act5}.

$$\begin{aligned}\mu x_{\text{aux}} =& \bar{\boldsymbol{I}}_5^5 \dot{\boldsymbol{\omega}}_5^5 [1,0,0]^T + \boldsymbol{\omega}_5^5 \times (\bar{I}_5^5 \boldsymbol{\omega}_5^5)[1,0,0]^T \\ &- Fy_{\text{s5}}(\overrightarrow{e5_z} - \overrightarrow{f5_z}) + Fz_{\text{s5}}(\overrightarrow{e5_y} - \overrightarrow{f5_y}) + Fy_{\text{6a}}(\overrightarrow{a5_z} - \overrightarrow{f5_z}) \\ &- Fz_{\text{6a}}(\overrightarrow{a5_y} - \overrightarrow{f5_y}) + \mu x_{\text{6a}} + Fy_{\text{6b}}(\overrightarrow{b5_z} - \overrightarrow{f5_z}) - Fz_{\text{6b}}(\overrightarrow{b5_y} - \overrightarrow{f5_y}) \\ &+ \mu x_{\text{6b}} + Fy_{\text{6c}}(\overrightarrow{c5_z} - \overrightarrow{f5_z}) - Fz_{\text{6c}}(\overrightarrow{c5_y} - \overrightarrow{f5_y}) + \mu x_{\text{6c}}\end{aligned} \tag{7}$$

$$Ft_{\text{act5}} = \frac{\mu x_{\text{aux}}}{\sin(\alpha_5)(\overrightarrow{d5_z} - \overrightarrow{f5_z}) - \cos(\alpha_5)(\overrightarrow{d5_y} - \overrightarrow{f5_y})} \tag{8}$$

$$Fy_{\text{act5}} = Ft_{act5}\sin(\alpha_5); \quad Fz_{\text{act5}} = Ft_{\text{act5}}\cos(\alpha_5) \tag{9}$$

3. The y and z components of j_5 can now be determined with the sum of forces.

$$Fy_{\text{j5}} = m_5 \ddot{p}y_{C5}^5 + Fy_{\text{6a}} + Fy_{\text{6b}} + Fy_{\text{6c}} - Fy_{\text{act5}} - Fy_{\text{s5}} \tag{10}$$

$$Fz_{\text{j5}} = m_5 \ddot{p}z_{C5}^5 + Fz_{\text{6a}} + Fz_{\text{6b}} + Fz_{\text{6c}} - Fz_{\text{act5}} - Fz_{\text{s5}} \tag{11}$$

4. Since j_5 and act_5 are similar in terms of construction and tolerances, it was assumed that the remaining x reaction force is divided equally among them.

$$Fx_{\mathrm{j5}} = Fx_{\mathrm{act5}} = \frac{m_5 \ddot{p}x^5_{C5} + Fx_{\mathrm{6a}} + Fx_{\mathrm{6b}} + Fx_{\mathrm{6c}}}{2} \tag{12}$$

5. All unknown forces are determined.
6. The sum of unknown μ_x must be 0 at this stage. Due to the tolerance similarities between j_5 and act_5, it was estimated that each will apply half of the μ_y and μ_z torques obtained using equation (2).

$$\begin{aligned} 2\mu y_{\mathrm{j5}} = 2\mu y_{\mathrm{act5}} = {} & \bar{\boldsymbol{I}}^5_5 \dot{\boldsymbol{\omega}}^5_5 [0,1,0]^T + \boldsymbol{\omega}^5_5 \times (\bar{I}^5_5 \boldsymbol{\omega}^5_5)[0,1,0]^T \\ & - Fz_{\mathrm{s5}}\overrightarrow{e5_x} + Fx_{\mathrm{s5}}\overrightarrow{e5_z} - Fz_{\mathrm{act5}}\overrightarrow{d5_x} + Fx_{\mathrm{act5}}\overrightarrow{d5_z} \\ & - Fz_{\mathrm{j5}}\overrightarrow{f5_x} + Fx_{\mathrm{j5}}\overrightarrow{f5_z} + Fz_{\mathrm{6a}}\overrightarrow{a5_x} - Fx_{\mathrm{6a}}\overrightarrow{a5_z} + \mu y_{\mathrm{6a}} \\ & + Fz_{\mathrm{6b}}\overrightarrow{b5_x} - Fx_{\mathrm{6b}}\overrightarrow{b5_z} + \mu y_{\mathrm{6b}} + Fz_{\mathrm{6c}}\overrightarrow{c5_x} - Fx_{\mathrm{6c}}\overrightarrow{c5_z} + \mu y_{\mathrm{6c}} \end{aligned} \tag{13}$$

$$\begin{aligned} 2\mu z_{\mathrm{j5}} = 2\mu z_{\mathrm{act5}} = {} & \bar{\boldsymbol{I}}^5_5 \dot{\boldsymbol{\omega}}^5_5 [0,0,1]^T + \boldsymbol{\omega}^5_5 \times (\bar{I}^5_5 \boldsymbol{\omega}^5_5)[0,0,1]^T \\ & - Fx_{\mathrm{s5}}\overrightarrow{e5_y} + Fy_{\mathrm{s5}}\overrightarrow{e5_x} - Fx_{\mathrm{act5}}\overrightarrow{d5_y} + Fy_{\mathrm{act5}}\overrightarrow{d5_x} \\ & - Fx_{\mathrm{j5}}\overrightarrow{f5_y} + Fy_{\mathrm{j5}}\overrightarrow{f5_x} + Fx_{\mathrm{6a}}\overrightarrow{a5_y} - Fy_{\mathrm{6a}}\overrightarrow{a5_x} + \mu z_{\mathrm{6a}} \\ & + Fx_{\mathrm{6b}}\overrightarrow{b5_y} - Fy_{\mathrm{6b}}\overrightarrow{b5_x} + \mu z_{\mathrm{6b}} + Fx_{\mathrm{6c}}\overrightarrow{c5_y} - Fy_{\mathrm{6c}}\overrightarrow{c5_x} + \mu z_{\mathrm{6c}} \end{aligned} \tag{14}$$

7. The distribution of unknown torques is fully determined.

Body 4 of CHARMIE (Figure 4) has four unknown forces/torques: the ones originating from the tension springs srt_4 and slt_4, the prismatic joint j_4 along the z axis, and the linear actuator act_4. Their distribution was determined using the following nine steps:

1. The spring forces srt_4 and slt_4 required an analysis of the auxiliary bodies 4_{sr} and 4_{sl} (Figure 5). The spring force (sr_4 or sl_4) is determined directly from the spring position and orientation.

$$Ft_{\mathrm{s4}} = (F_i)_{\mathrm{s4}} + k_{\mathrm{s4}}(L_{\mathrm{s4}} - (L_0)_{\mathrm{s4}}) \tag{15}$$

$$Fx_{\mathrm{s4}} = Ft_{\mathrm{s4}}\mathrm{sin}(\gamma_4); \quad Fz_{\mathrm{s4}} = Ft_{\mathrm{s4}}\mathrm{cos}(\gamma_4) \tag{16}$$

2. Forces srt_4 and slt_4 are along the z axis, and can be calculated by using them to nullify the torque μ_y reaction on joint jr_4 or jl_4.

$$Fz_{\mathrm{srt4}} = Fz_{\mathrm{slt4}} = \frac{Fz_{\mathrm{sr4}}\overrightarrow{h4_x} - Fx_{sr4}\overrightarrow{h4_z}}{\overrightarrow{j4_x}} \tag{17}$$

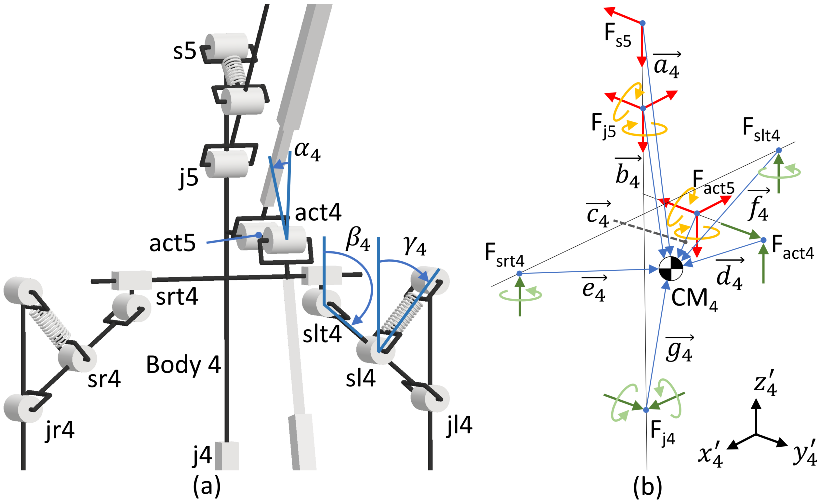

Fig. 4. Kinematic representation of body 4 of the CHARMIE robot (a) and corresponding free body diagram (b). In (b), known forces and torques are colored in red/orange, unknown forces are colored green, and displacement vectors in blue. The $x'_4y'_4z'_4$ axes represent the orientations used in all calculations of body 4.

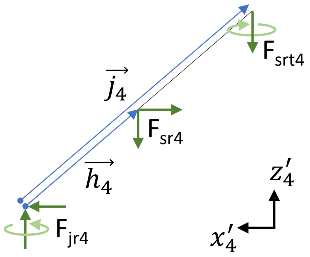

Fig. 5. Free body diagram of body 4_{sr} with unknown forces and torques colored in green, and displacement vectors in blue.

3. For modeling body 3, the joint reactions (jr_4 or jl_4) are determined with the sum of forces within these auxiliary bodies.

$$Fx_{\text{jr4}} = Fx_{\text{s4}}; \quad Fz_{\text{jr4}} = Fz_{\text{s4}} + Fz_{\text{srt4}} \tag{18}$$

4. Since act_4 is the only unknown z reaction on body 4, and its orientation in the yz plan is known, its y and z components can be determined.

$$Fz_{\text{act4}} = m_4\ddot{p}z^4_{C4} + Fz_{\text{act5}} + Fz_{\text{j5}} + Fz_{\text{s5}} - Fz_{\text{srt4}} - Fz_{\text{slt4}} \tag{19}$$

$$Fy_{\text{act4}} = Fz_{\text{act4}}\tan(\alpha 4) \tag{20}$$

5. Due to the tight tolerance of j_4, it was assumed it applies the remaining x and y reaction forces.

$$Fx_{\mathrm{j4}} = m_4 \ddot{p}x^4_{C4} + Fx_{\mathrm{act5}} + Fx_{\mathrm{j5}} \tag{21}$$

$$Fy_{\mathrm{j4}} = m_4 \ddot{p}y^4_{C4} + Fy_{\mathrm{act5}} + Fy_{\mathrm{j5}} + Fy_{\mathrm{s5}} - Fy_{\mathrm{act4}} \tag{22}$$

6. All unknown forces are determined.
7. Since joint j_4 cannot apply torques in the z axis, and joints srt_4 and slt_4 are equal with tighter tolerances then act_4, it was assumed these two joints apply half of μ_z each.

$$\begin{aligned}2\mu z_{\mathrm{srt4}} = 2\mu z_{\mathrm{slt4}} = {} & \bar{\boldsymbol{I}}^4_4 \dot{\boldsymbol{\omega}}^4_4 [0,0,1]^T + \boldsymbol{\omega}^4_4 \times (\bar{I}^4_4 \boldsymbol{\omega}^4_4)[0,0,1]^T \\ & - Fx_{\mathrm{j4}} \overrightarrow{g4}_y + Fy_{\mathrm{j4}} \overrightarrow{g4}_x + Fy_{\mathrm{act4}} \overrightarrow{d4}_x - Fy_{\mathrm{s5}} \overrightarrow{a4}_x + Fx_{\mathrm{j5}} \overrightarrow{b4}_y \\ & - Fy_{\mathrm{j5}} \overrightarrow{b4}_x + \mu z_{\mathrm{j5}} + Fx_{\mathrm{act5}} \overrightarrow{c4}_y - Fy_{\mathrm{act5}} \overrightarrow{c4}_x + \mu z_{\mathrm{act5}}\end{aligned} \tag{23}$$

8. Due to the tight tolerance of j_4, it was assumed it applies the remaining torques μ_x and μ_y.

$$\begin{aligned}\mu x_{\mathrm{j4}} = {} & \bar{\boldsymbol{I}}^4_4 \dot{\boldsymbol{\omega}}^4_4 [1,0,0]^T + \boldsymbol{\omega}^4_4 \times (\bar{I}^4_4 \boldsymbol{\omega}^4_4)[1,0,0]^T + Fz_{\mathrm{srt4}} \overrightarrow{e4}_y + Fz_{\mathrm{slt4}} \overrightarrow{e4}_y \\ & - Fy_{\mathrm{act4}} \overrightarrow{d4}_z + Fz_{\mathrm{act4}} \overrightarrow{d4}_y - Fy_{\mathrm{j4}} \overrightarrow{g4}_z + Fy_{\mathrm{s5}} \overrightarrow{a4}_z - Fz_{\mathrm{s5}} \overrightarrow{a4}_y \\ & + Fy_{\mathrm{j5}} \overrightarrow{b4}_z - Fz_{\mathrm{j5}} \overrightarrow{b4}_y + Fy_{\mathrm{act5}} \overrightarrow{c4}_z - Fz_{\mathrm{act5}} \overrightarrow{c4}_y\end{aligned} \tag{24}$$

$$\begin{aligned}\mu y_{\mathrm{j4}} = {} & \bar{\boldsymbol{I}}^4_4 \dot{\boldsymbol{\omega}}^4_4 [0,1,0]^T + \boldsymbol{\omega}^4_4 \times (\bar{I}^4_4 \boldsymbol{\omega}^4_4)[0,1,0]^T - Fz_{\mathrm{srt4}} \overrightarrow{e4}_x - Fz_{\mathrm{slt4}} \overrightarrow{e4}_x \\ & - Fz_{\mathrm{act4}} \overrightarrow{d4}_x + Fx_{\mathrm{act4}} \overrightarrow{d4}_z + Fx_{\mathrm{j4}} \overrightarrow{g4}_z + Fz_{\mathrm{s5}} \overrightarrow{a5}_x \\ & + Fz_{\mathrm{j5}} \overrightarrow{b4}_x - Fx_{\mathrm{j5}} \overrightarrow{b4}_z + \mu y_{\mathrm{j5}} + Fz_{\mathrm{act5}} \overrightarrow{c4}_x - Fx_{\mathrm{act5}} \overrightarrow{c4}_z + \mu y_{\mathrm{j5}}\end{aligned} \tag{25}$$

9. The distribution of unknown torques is fully determined.

5 Results and discussion

The two methods (in-house code and commercial software) used to analyse the inverse dynamics of CHARMIE produced similar results. In Figure 6, the reactions applied in body 3 are compared — this body was chosen since its calculations require data from all iterations of the recursive algorithm. The discontinuity at the 2.5 s mark represents the inversion of the direction of all accelerations.

Originally, in Figure 6.f, there was a systematic difference in the results. An in-depth analysis concluded this was due to the commercial software underevaluating the inertia of body 3 (this software does not allow a manual definition of inertia). To produce the near identical results shown, the inertia of body 3 was

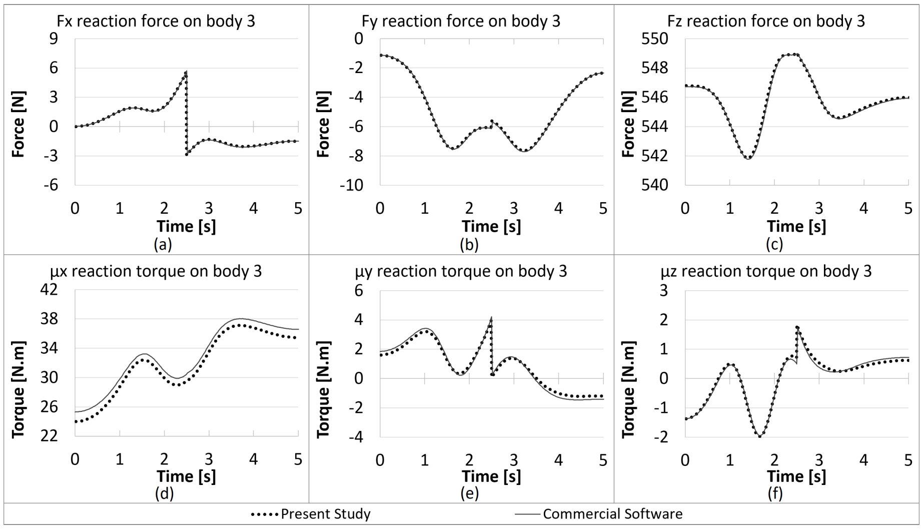

Fig. 6. Comparison of the reactions forces and torques applied by body 2 on body 3 using both considered methods for the inverse dynamic analysis of CHARMIE.

changed in the in-house code to 0. The only remaining slight variations in Figures 6.d and 6.e are user induced, caused by simplifications made to the model in the commercial software (Figure 2) to avoid redundancies. These changes slightly altered the points of application of some forces, altering the torque generated by them. The obtained results validate the developed in-house code for the multibody dynamic model of CHARMIE.

The visual interface of the aforementioned commercial software cannot be disabled, reducing its computational efficiency. For a simplified verification of the in-house algorithm's efficiency, a double pendulum was modeled both in it, and in a dynamic simulator for robotic applications (*CoppeliaSim*). Without graphical interface, both methods presented identical computational speed.

The developed code does not rely on any commercial software, allowing easier compatibility with other computational tools. This becomes especially useful for testing robotic control solutions. In the context of the CHARMIE project, this model will be used to implement and train neural networks for motion control.

One of the main advantages of the obtained model is its modularity and ease of parametrization. In an ongoing study, the robot's mechanism design is being optimized to minimize actuator loads. Six input variables (choice of springs for the robot, geometric parameters, and a variable selecting the robot's trajectory) were defined, and then automatically altered using a programming loop. This resulted in 47570 possible solutions to be compared, each simulated for 1 minute of motion with a time step of 0.1 s. The program's total run time was under 65 hours: an average computation time of 5 seconds of per case. Preliminary results show a reduction of over 70% in both maximum and average actuator loads.

Finally, it must emphasized that the outcomes of this study are playing a pivotal role in the CHARMIE project, a vital step forward for the development of the physical prototype of the robot.

Acknowledgements. This work has been supported by the Laboratory of Automation and Robotics (LAR) of University of Minho, and the ALGORITMI and CMEMS research centres. The first and second authors received funding through a doctoral scholarship from the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia) [grant numbers SFRH/BD/145993/2019 and SFRH/BD/06944/2020], with funds from the Portuguese Ministry of Science, Technology and Higher Education and the European Social Fund through the Programa Operacional do Capital Humano (POCH). This work has been supported by FCT—Fundação para a Ciência e a Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

1. W. Khalil, "Dynamic modeling of robots using newton-euler formulation," in *Informatics in Control, Automation and Robotics* (J. A. Cetto, J.-L. Ferrier, and J. Filipe, eds.), (Berlin, Heidelberg), pp. 3–20, Springer Berlin Heidelberg, 2011.
2. A. Aloulou and O. Boubaker, "A Relevant Reduction Method for Dynamic Modeling of a Seven-linked Humanoid Robot in the Three-dimensional Space," *Procedia Engineering*, vol. 41, no. Iris, pp. 1277–1284, 2012.
3. S. Y. Naing and T. Rain, "Analysis of Position and Angular Velocity of Four-Legged Robot (Mini-Bot) from Dynamic Model Using Euler-Lagrange Method," in *2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 1–4, IEEE, mar 2019.
4. V. Záda and K. Belda, "Application of Hamiltonian mechanics to control design for industrial robotic manipulators," *2017 22nd International Conference on Methods and Models in Automation and Robotics, MMAR 2017*, no. 4, pp. 390–395, 2017.
5. K. Chadaj, P. Malczyk, and J. Frączek, "A parallel Hamiltonian formulation for forward dynamics of closed-loop multibody systems," *Multibody System Dynamics*, vol. 39, pp. 51–77, jan 2017.
6. R. Featherstone, *Rigid Body Dynamics Algorithms.* Boston, MA: Springer US, jan 2008.
7. S. Ivaldi, J. Peters, V. Padois, and F. Nori, "Tools for simulating humanoid robot dynamics: A survey based on user feedback," in *2014 IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, pp. 842–849, IEEE, nov 2014.
8. F. Gonçalves, T. Ribeiro, A. F. Ribeiro, G. Lopes, and P. Flores, "A recursive algorithm for the forward kinematic analysis of robotic systems using euler angles," *Robotics*, vol. 11, no. 1, 2022.
9. T. Ribeiro, F. Gonçalves, I. S. Garcia, G. Lopes, and A. F. Ribeiro, "Charmie: A collaborative healthcare and home service and assistant robot for elderly care," *Applied Sciences*, vol. 11, no. 16, 2021.
10. B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control.* London: Springer-Verlag, 1st ed., 2009.