Hindawi
Journal of Robotics
Volume 2022, Article ID 8715161, 14 pages
https://doi.org/10.1155/2022/8715161

Research Article

Online Trajectory Tracking Control Based on the Explicit Form of the Equations of Motion for Serial Manipulator Using the New Formulation

Otman El Baji, Nabil Ben Said Amrani, and Driss Sarsri

Laboratory of Innovative Technologies, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University, B. P. 1818, Tangier, Morocco

Correspondence should be addressed to Otman El Baji; otman.elbaji@etu.uae.ac.ma

Received 27 September 2022; Revised 27 October 2022; Accepted 1 November 2022; Published 21 November 2022

Academic Editor: L. Fortuna

In this article, the novel approach to equations of motion for serial manipulators developed in literature by Bertrand and Bruneau (2013) is extended to make it usable for manipulators with general joints (i.e., prismatic and/or rotational). In this method, the dynamic model is explicitly and directly obtained from the structural parameters of the manipulator and matrix algebra without intermediate heavy calculations such as energy derivation. The correctness and efficiency of the described method are demonstrated through simulation of the dynamical equations of a 5-DOF SCARA robot. The simulation results obtained using the new formulation were compared with those derived by Kane's method, Lagrange–Euler formulation, and GIM (generalized inertia matrix)-Christoffel's algorithm, which proves the efficiency and correctness of the presented model. It was concluded that the new formulation requires less CPU time to generate explicit closed-form inverse dynamics. Finally, to illustrate the power of the new formulation in real-time control, a trajectory tracking control for the SCARA manipulator based on the numeric and symbolic computation of the inverse dynamic is established, and it is shown that the numeric and symbolic approaches based on our method are equivalent. As a consequence, the applicability of the new formulation in real-time model-based control is proved.

1. Introduction

The dynamics of industrial manipulators are highly nonlinear. The precise, explicit, closed-form dynamics of robots play an important role in their design, simulation, and control. There are two categories of dynamics modeling: direct and inverse dynamics. Inverse dynamics modeling is used to calculate the required actuator generalized forces in order to achieve the planned trajectory and is also known as model-based control, whereas the direct dynamics model determines the end effector motion for a given actuator force/torque. The inverse dynamics are related to the control and design of the manipulator, whereas the direct dynamics are associated with simulation. As regards robot control, the dynamic model is the most important part to design controllers and determine controllers' parameters; inverse dynamics are always employed to calculate the generalized forces as feed-forward data in order to follow the desired trajectory, which is called computed torque control or internal model control. The trajectory tracking performance of most commercial robots is limited because they are controlled by ignoring coupled dynamics and using a simple single-joint feedback controller, which is an imperfect model [1–4].

Several techniques have been employed to derive the dynamics model of manipulators: the Newton–Euler method [5–7], Lagrange–Euler method [4, 8–11], Kane's method [12, 13], and Gibbs–Appell equations [14].

Using the Newton–Euler approach, the acceleration of each isolated rigid body and each reaction force and moment between the links should be determined, whereas the workless constraint moments and forces are not used to control the manipulator, which results in a large number of equations and leads to a high computational cost. The alternative one, the recursive Newton–Euler method [6, 7], gives the more efficient algorithm; however, it is not applied to the control design of the manipulator and does not give better insight into the form of the equations. For this purpose, a set of explicit closed-form equations of motion must be determined.

In Lagrange's formulation, the manipulator is considered an entire system instead of isolating each link; all workless constraint forces and moments can be automatically eliminated in the beginning. However, its disadvantage is the calculation of complex derivatives of Lagrangians, which always leads to a huge number of intermediate and complicated computations. Therefore, the Lagrange–Euler equations are the most adequate form for controllers' design and simulation of manipulators. Unfortunately, this method is consuming more computational time for online control of manipulators because the procedure involves laborious manipulation of energy expressions. So, the controller design of a manipulator was usually based on neglecting the centrifugal and Coriolis terms of the dynamic model.

Kane's method is based on analytic mechanics and vector mechanics. To obtain the generalized forces by using Kane's equation, the active forces and inertia forces are multiplied, respectively, by the intermediate vector entities such as partial angular speeds and partial speeds. It has the benefit of automatically eliminating workless forces and moments without applying the principle of virtual work. The introduction of intermediate variables results in a bit of laborious calculation.

The Gibbs–Appell's formulation is formulated from the Gibbs function, which is considered the kinetic energy of acceleration. The dynamics model is derived by differentiating the Gibbs function with respect to the generalized velocities.

However, none of the previous formalisms cited above offers a simple and straightforward combination between the structural parameters of the manipulators and the dynamical equations. In fact, the dynamical equations are usually established based on recursive algorithms or require complicated calculations, such as the differentiation of energies. Therefore, this kind of model cannot directly provide a clear description and comprehension of the dynamical equations of a multibody system in order to exploit it efficiently and accurately in the robot control industrial architecture. Consequently, the dynamic equations have not been clarified yet, and the complete comprehension of the dynamics of a given multibody system has not been reached. One way to improve this point is to revisit the dynamical formalisms.

In addition, the motion of manipulators with variable or unknown loads may slow down or speed up, and the manipulators are not capable of tracking the planned trajectory closely. In order to control a manipulator with an unknown or variable load that follows a desired trajectory, it is necessary to calculate the generalized forces required to move all its joints repeatedly and accurately at an acceptable time step (between 10 ms and 1 ms).

In general, the control algorithm involves the calculation of the suitable input generalized forces employing the measured values of generalized speeds and coordinates and the calculated values of generalized accelerations, which are normally computed from dynamic formulations cited in literature. The execution time of the dynamics model is one of the main problems because none of the aforementioned formulations can be exploited numerically in real-time control. The challenging task for real-time control is that the dynamic model must be computed very frequently at every integration step. Consequently, the feasibility of online implementation is highly dependent on the dynamics formulation, which affects the computation cost.

A novel formulation that satisfies all of the above requirements has been proposed by [15]. In this novel formulation, the dynamical equations were obtained by considering only the structural parameters and matrix algebra. This method is simple and straightforward, as the terms of the inertia matrix and coriolis/centrifugal matrix can be computed simultaneously and independently of the generalized velocities and accelerations. But it was developed especially for a single kinematic chain with only rotational joints. Using this formulation, a dynamical approach was developed for trajectory planning of a Mikron UCP710 multiaxis machining system and a KUKA anthropomorphic robot [16].

Therefore, we aim with the new formulation to obtain a precise and efficient inverse dynamic for both symbolic and numeric approaches without any simplification of the dynamical equations for the development of the explicit form of the equations of motion in the active joint space.

The contributions of this paper: the first objective is to improve the previous formalism [15] to make it usable for serial manipulators with general joints (i.e., prismatic and/or rotational). Furthermore, computation efficiency is an important issue in the derivation of inverse dynamics; however, the computational cost of this new formulation with respect to other approaches has not been compared yet, which is the second objective. The final objective is to demonstrate the feasibility of online control for manipulators based on the new formulation.

This study is organized as follows: Section 2 introduces the system and notation used in the new formalism. Section 3 presents the generalized new formalism for serial manipulators with general joints. In Section 4, the kinematic and dynamic properties of the 5-DOF SCARA manipulator based on this formalism are presented using the symbolic computer algebra software SYMPY. Section 5 discusses the simulation results. The future work and conclusions are highlighted in the last section.

2. Notations

This section reviews briefly some of the notations used for the new formalism.

2.1. General Presentation. The considered multibody system is shown in Figure 1. It is a serial manipulator composed of N rigid links and N single-degree-of-freedom joints of a general type, either revolute or prismatic. The base frame is numbered as a link 0. The joint i being the connection between link $i-1$ and link i. The generalized actuated force associated with joint i is denoted by τ_i.

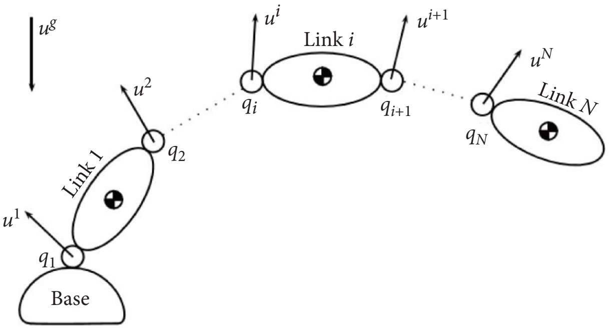


Figure 1: General scheme of the considered serial manipulator [15].

2.2. Notations. The notations and their descriptions are given in Table 1.

Table 1: Notations for the new formulation.

Symbols	Designations
S_i	The i-th body or link
$\sum_i \cup_{k=i}^{N} S_i$	The i-th subsystem
q	The N-dimensional generalized coordinates.
$\dot{q}$, $\ddot{q}$	The N-dimensional velocities and accelerations vectors of the joints.
ε_i, $\bar{\varepsilon}_i = (1 - \varepsilon_i)$	The i-th joint type, ε_i is zero if the i-th joint type is a revolute, ε_i is one if the i-th joint type is a prismatic.
o_i	The i-th joint center.
c_i	The centeroid of the link S_i
$c_{ij} = o_i c_j$	The position vector of c_j relative to joint center o_i
u_i	The i-th joint axis.
u_g	The downward unit vector specifying the gravity direction.
B_i	The i-th base that is attached to body S_i
$\mathfrak{R}_i(o_i, \mathrm{B}_i)$	The frame attached to body S_i
m_i	The mass of body S_i.
I_i	The 3×3 inertia tensor associated with body S_i defined in c_i
$J_i = -I_i + (1/2)\mathrm{Tr}(I_i)I$	The 3×3 pseudoinertia tensor associated with body S_i defined in c_i
$\boldsymbol{I}$	The 3×3 identity matrix
$\tau = [\tau_1 \ldots\ldots \tau_N]^T$	The N-vector of the joint generalized forces.
R_{ij}	The rotation matrix of base B_j with respect to base B_i

3. New Formulation of a Serial Manipulator with General Joints

The objective of this section is to extend the existing work presented in [15], which is developed especially for serial manipulators with only revolute joint to cover those with revolute or prismatic joints.

3.1. Accelerations of the Body S_k in the Case of General Joints. The angular velocity $\omega_{k/0}$ of the body S_k with respect to the base is expressed as follows:

$$\omega_{k/0} = \sum_{j=1}^{k} \bar{\varepsilon}_j \omega_{j/j-1} = \sum_{j=1}^{k} \bar{\varepsilon}_j \dot{q}_j u^j. \tag{1}$$

Its dual matrix can be expressed as

$$\widehat{\omega}_{k/0} = \sum_{j=1}^{k} \bar{\varepsilon}_j \widehat{\omega}_{j/j-1} = \sum_{j=1}^{k} \bar{\varepsilon}_j \dot{q}_j \widehat{u}^j, \tag{2}$$

where the term $\widehat{x}$ denoted the dual tensor associated with a vector $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$, which is defined by

$$\mathrm{dual}(x) = \widehat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{3}$$

The linear velocity $V_{o_j}^{j-1}$ and the linear acceleration $\Gamma_{o_j}^{j-1}$ of body S_j with respect to frame $\mathfrak{R}_{j-1}$ are expressed, respectively, by

$$\begin{aligned} V_{o_j}^{j-1} &= \varepsilon_j \dot{q}_j u_j, \\ \Gamma_{o_j}^{j-1} &= \varepsilon_j \ddot{q}_j u_j. \end{aligned} \tag{4}$$

As shown in [17] and by using equations (1) and (2), the angular acceleration $a_{k/0}$ of S_k relative to $\mathfrak{R}_0$ for the general joints can be derived as follows:

$$a_{k/0} = \sum_{j=i+1}^{k} \varepsilon_j \left(\ddot{q}_j u_j + \dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n \widehat{u}_n u_j \right). \tag{5}$$

As shown in [15], the dual matrix $\widehat{a}_{k/0}$ can be established as follows:

$$\widehat{\alpha}_{k/0} = \sum_{j=1}^{k} \bar{\varepsilon}_j \left[\ddot{q}_j \widehat{u}^j + \dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n \left(\widehat{u}^n \widehat{u}^j - \widehat{u}^j \widehat{u}^n \right) \right]. \tag{6}$$

The expression of the acceleration Γ_{c_k} of the mass center c_k can be derived as follows:

$$\Gamma_{c_k} = \Gamma_{o_k} + a_{k/0} \wedge o_k c_k + \omega_{k/0} \wedge \left(\omega_{k/0} \wedge o_k c_k \right), \tag{7}$$

$$\Gamma_{o_k} = \sum_{j=1}^{k} \varepsilon_j \left(\Gamma_{o_j}^{j-1} + 2\omega_{j-1/0} \wedge V_{o_j}^{j-1} \right) + \sum_{j=1}^{k-1} \bar{\varepsilon}_j \left(a_{j/0} \wedge o_j o_{j+1} + \omega_{j/0} \wedge \left(\omega_{j/0} \wedge o_j o_{j+1} \right) \right). \tag{8}$$

Consequently, including equations (8) in equation (7), we have

$$\Gamma_{c_k} = a_{k/0}o_kc_k + \omega_{k/0}\wedge\left(\omega_{k/0}\wedge o_kc_k\right) + \sum_{j=1}^{k} \varepsilon_j\left(\Gamma_{o_j}^{j-1} + 2\omega_{j-1/0}\wedge V_{o_j}^{j-1}\right) + \sum_{j=1}^{k-1} \overline{\varepsilon}_j\left(a_{j/0}\wedge o_jo_{j+1} + \omega_{j/0}\wedge\left(\omega_{j/0}\wedge o_jo_{j+1}\right)\right). \tag{9}$$

By using the relation between cross product and dual tensor, which is defined by $x\wedge y = \widehat{x}\cdot y = -\widehat{y}\cdot x$, we can write

$$\Gamma_{c_k} = (\widehat{a}_{k/0} + \widehat{\omega}_{k/0}\widehat{\omega}_{k/0})o_kc_k + \sum_{j=1}^{k} \varepsilon_j\left(\Gamma_{o_j}^{j-1} + 2\widehat{\omega}_{j-1/0}V_{o_j}^{j-1}\right) + \sum_{j=1}^{k-1} \overline{\varepsilon}_j\left(\widehat{a}_{j/0} + \widehat{\omega}_{j/0}\widehat{\omega}_{j/0}\right)o_jo_{j+1}. \tag{10}$$

As demonstrated in [15] and taking into account (2) and (6), the vectors $\sum_{j=1}^{k-1}\overline{\varepsilon}_j\,(\widehat{a}_{j/0} + \widehat{\omega}_{j/0}\widehat{\omega}_{j/0})o_jo_{j+1}$ and $(\widehat{a}_{k/0} + \widehat{\omega}_{k/0}\,\widehat{\omega}_{k/0})o_kc_k$ can be expressed by

$$\sum_{j=1}^{k-1} \overline{\varepsilon}_j\left(\widehat{a}_{j/0}\widehat{\omega}_{j/0}\widehat{\omega}_{j/0}\right)o_jo_{j+1} = \sum_{j=1}^{k} \overline{\varepsilon}_j\left(\ddot{q}_j\widehat{u}_j + \dot{q}_j^2\widehat{u}_j\widehat{u}_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\widehat{u}_j\right)o_jo_k, \tag{11}$$

$$(\widehat{a}_{k/0} + \widehat{\omega}_{k/0}\widehat{\omega}_{k/0})o_kc_k = \sum_{j=1}^{k} \overline{\varepsilon}_j\left(\ddot{q}_j\widehat{u}_j + \dot{q}_j^2\widehat{u}_j\widehat{u}_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\widehat{u}_j\right)\cdot o_kc_k. \tag{12}$$

By using (2) and (4), we get

$$\sum_{j=1}^{k} \varepsilon_j\left(\Gamma_{o_j}^{j-1} + 2\widehat{\omega}_{j-1/0}V_{o_j}^{j-1}\right) = \sum_{j=1}^{k} \varepsilon_j\left(\ddot{q}_ju_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_nu_j\right). \tag{13}$$

Lastly, based on equations (11)–(13), the acceleration Γ_{c_k} can be derived as follows:

$$\begin{aligned}
\Gamma_{c_k} &= \sum_{j=1}^{k}\left[\varepsilon_j\left(\ddot{q}_ju_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_nu_j\right) + \overline{\varepsilon}_j\left(\ddot{q}_j\widehat{u}_j + \dot{q}_j^2\widehat{u}_j\widehat{u}_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\widehat{u}_j\right)c_{\mathrm{jk}}\right]\\
&= \sum_{j=1}^{k}\left[\varepsilon\mathrm{j}\left(\ddot{q}_jI + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\right) - \overline{\varepsilon}_j\left(\ddot{q}_jI + \dot{q}_j^2\widehat{u}_j + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\right)\widehat{c}_{\mathrm{jk}}\right]u_j\\
&= \sum_{j=1}^{k}\left[\left(\varepsilon_jI - \overline{\varepsilon}_j\widehat{c}_{\mathrm{jk}}\right)\ddot{q}_j - \overline{\varepsilon}_j\dot{q}_j^2\widehat{u}_j\widehat{c}_{\mathrm{jk}} + 2\dot{q}_j\sum_{n=1}^{j-1}\overline{\varepsilon}_n\dot{q}_n\widehat{u}_n\left(\varepsilon_jI - \overline{\varepsilon}_j\widehat{c}_{\mathrm{jk}}\right)\right]u_j.
\end{aligned} \tag{14}$$

3.2. The Dynamic Generalized Forces. The dynamic generalized forces denoted by $F_d(\Sigma_i)$ can be expressed by

$$F_d(\Sigma_i) = \bar{\varepsilon}_i \delta_{\Sigma_i}(o_i) + \varepsilon_i R_d(\Sigma_i) = \sum_{k=i}^{N} \bar{\varepsilon}_i \delta_{S_k}(o_i) + \varepsilon_i R_d(S_k), \quad \forall i \in [1, N], \tag{15}$$

where $\delta_{o_i}(S_k)$ is the torque vector generated by the dynamic forces applied about o_i to body S_k, and $R_d(S_k)$ is the vector of the force due to the dynamic forces applied to the bodyS_k.

$$F_d(\Sigma_i) = \sum_{k=i}^{N} \bar{\varepsilon}_i \left(\bar{\varepsilon}_k \delta_{S_k}(c_i) + m_k \hat{c}_{ik} \cdot \Gamma_{c_k} \right) + \varepsilon_i R_d(S_k), \tag{16}$$

where $R_d(S_k)$ and $\delta_{S_k}(c_k)$ are described by the Newton–Euler equations as follows:

$$F_d(\Sigma_i) = \sum_{k=i}^{N} \bar{\varepsilon}_i \left[(\mathrm{I}_k \cdot \alpha_{k/0} + \hat{\omega}_{k/0}(\mathrm{I}_k \cdot \omega_{k/0})) + m_k \hat{c}_{ik} \cdot \Gamma_{c_k} \right] + \varepsilon_i m_k \Gamma_{c_k}. \tag{17}$$

As shown in [15], by considering equations (1), (2), and (5), and by using equation (14), equation (17) becomes

$$\begin{aligned} F_d(\Sigma_i) = \sum_{k=i}^{N} \Bigg\{ & \bar{\varepsilon}_i \bar{\varepsilon}_k \sum_{j=1}^{k} \bar{\varepsilon}_j \left[\ddot{q}_j I_k + \dot{q}_j^2 \hat{u}_j \mathrm{I}_k + 2\dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n dual(J_k u_n) \right] u_j \\ & + m_k (\bar{\varepsilon}_i \hat{c}_{\mathrm{ik}} + \varepsilon_i I) \sum_{j=1}^{k} \left[\left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \ddot{q}_j - \bar{\varepsilon}_j \dot{q}_j^2 \hat{u}_j \hat{c}_{\mathrm{jk}} + 2\dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n \hat{u}_n \left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \right] u_j \\ = \sum_{k=i}^{N} \sum_{j=1}^{k} \Bigg\{ & \bar{\varepsilon}_i \bar{\varepsilon}_k \bar{\varepsilon}_j \left[\ddot{q}_j I_k + \dot{q}_j^2 \hat{u}_j \mathrm{I}_k + 2\dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n dual(J_k u_n) \right] \\ & + m_k (\bar{\varepsilon}_i \hat{c}_{\mathrm{ik}} + \varepsilon_i I) \left[\left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \ddot{q}_j - \bar{\varepsilon}_j \dot{q}_j^2 \hat{u}_j \hat{c}_{\mathrm{jk}} + 2\dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \dot{q}_n \hat{u}_n \left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \right] \Bigg\} u_j. \end{aligned} \tag{18}$$

3.3. Generalized Force Due to the Environment Forces. The environmental generalized forces applied to the *i*-th DoF denoted by $F_e(\Sigma^i)$, can be denoted by

$$F_e(\Sigma_i) = \bar{\varepsilon}_i N_{o_i}(\Sigma_i) + \varepsilon_i R_e(\Sigma_i) = \sum_{k=i}^{N} \bar{\varepsilon}_i N_{o_i}(S_k) + \varepsilon_i R_e(S_k), \tag{19}$$

where $N_{o_i}(S_k)$ are the total external torques applied about o_i on bodyS_k, and $R_e(S_k)$ are the total external forces applied to bodyS_k.

Since the external torque is applied about c_k, the vector $F_e(\Sigma^i)$ can be written as

$$\begin{aligned} F_e(\Sigma_i) &= \sum_{k=i}^{N} \bar{\varepsilon}_i \left[M_k + c_{ik} \wedge \left(F_k - m_k g \cdot u_g \right) \right] + \varepsilon_i \left[F_k - m_k g \cdot u_g \right] \\ &= \sum_{k=i}^{N} \bar{\varepsilon}_i \left[M_k + \hat{c}_{ik} \left(F_k - m_k g \cdot u_g \right) \right] + \varepsilon_i \left[F_k - m_k g \cdot u_g \right], \end{aligned} \tag{20}$$

where F_k is the external force other than gravity applied on bodyS_k, and M_k is the external torque other than gravity applied on the body S_k expressed aboutc_k.

3.4. The Generalized New Formulation of Inverse Dynamics. The dynamic equations can be established by

$$\tau_i = (F_d(\Sigma_i) + F_e(\Sigma_i)) u_i, \quad \forall i \in [1, N]. \tag{21}$$

By using (18), (20), and (21), the generalized new formulation of the inverse dynamics for a serial manipulator with general joints can be derived as

$$\begin{aligned} \tau_i = \sum_{k=i}^{N} \Bigg\{ \bar{\varepsilon}_i \Bigg\{ & \sum_{j=1}^{k} \Big[\ddot{q}_j \left(\bar{\varepsilon}_j \left(\bar{\varepsilon}_k I_k - m_k \hat{c}_{\mathrm{ik}} \hat{c}_{\mathrm{jk}} \right) + \varepsilon_j m_k \hat{c}_{\mathrm{ik}} \right) \\ & + \dot{q}_j^2 \bar{\varepsilon}_j \left(\bar{\varepsilon}_k \hat{u}_j \mathrm{I}_k - m_k \hat{c}_{\mathrm{ik}} \hat{u}_j \hat{c}_{\mathrm{jk}} \right) \\ & + 2\dot{q}_j \sum_{n=1}^{j-1} \bar{\varepsilon}_n \left(\bar{\varepsilon}_j \left(\bar{\varepsilon}_k du\ al(J_k u_n) - m_k \hat{c}_{\mathrm{ik}} \hat{u}_j \hat{c}_{\mathrm{jk}} \right) \right. \\ & \left. + \varepsilon_j m_k \hat{c}_{\mathrm{ik}} \hat{u}_n \right) \dot{q}_n \Big] u_j + \left(M_k + \hat{c}_{\mathrm{ik}} \left(F_k - m_k g u_g \right) \right) \Big\} \\ \varepsilon_i \Bigg\{ & \sum_{\mathrm{j}=1}^{k} \Big[m_k \left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \ddot{q}_j - \bar{\varepsilon}_j m_k \hat{u}_j \hat{c}_{\mathrm{jk}} \dot{q}_j^2 \\ & + 2 m_k \dot{q}_j \sum_{\mathrm{n}=1}^{\mathrm{j}-1} \dot{q}_n \bar{\varepsilon}_n \hat{u}_n \left(\varepsilon_j I - \bar{\varepsilon}_j \hat{c}_{\mathrm{jk}} \right) \Big] u_j + \left(F_k - m_k g u_g \right) \Bigg\} \Bigg\} u^i. \end{aligned} \tag{22}$$

The above expression for τ_i is very simple to obtain than appears at first sight because the coefficients ε_i and $\overline{\varepsilon}_i$ cannot both be nonzero simultaneously.

The inverse dynamics can be established in the following equivalent form:

$$\tau_i = \sum_{k=i}^{N} \sum_{j=1}^{k} \left(H_{\mathrm{ijk}} \ddot{q}_j + \beta_{\mathrm{ijk}} \dot{q}^2 + 2\dot{q}_j \sum_{n=1}^{j-1} \zeta_{\mathrm{ijnk}} \dot{q}_n \right) + G_{\mathrm{ik}} + Q_{\mathrm{ik}}. \tag{23}$$

The coefficients of the generalized inertia matrix H_{ijk}, the coefficients relative to centrifugal forces β_{ijk}, the coefficients relative to Coriolis forces ζ_{ijnk}, the gravity forces G_{ik}, and the external forces Q_{ik} can be expressed in function of the system's structural parameters independently of the generalized velocities and accelerations $(\dot{q}, \ddot{q})$:

$$H_{\mathrm{ijk}} = \left\{ \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_j \left(I_k - m_k \widehat{c}_{ik} \widehat{c}_{jk} \right) + \varepsilon_j m_k \widehat{c}_{ik} \right) + \varepsilon_i \left(\varepsilon_j m_k I - \overline{\varepsilon}_j m_k \widehat{c}_{jk} \right) \right] u_j \right\} u_i, \tag{24}$$

$$\beta_{ijk} = \left\{ \overline{\varepsilon}_j \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_k \widehat{u}_j \mathrm{I}_k - m_k \widehat{c}_{ik} \widehat{u}_j \widehat{c}_{jk} \right) - \varepsilon_i m_k \widehat{u}_j \widehat{c}_{jk} \right] u_j \right\} u_i, \tag{25}$$

$$\zeta_{\mathrm{ijnk}} = \left\{ \overline{\varepsilon}_n \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_j \left(\overline{\varepsilon}_k \mathrm{dual} \left(J_k u_n \right) - m_k \widehat{c}_{ik} \widehat{u}_n \widehat{c}_{jk} \right) + \varepsilon_j m_k \widehat{c}_{ik} \widehat{u}_n \right) + \varepsilon_i \widehat{u}_n \left(\varepsilon_j m_k I - \overline{\varepsilon}_j m_k \widehat{c}_{jk} \right) \right] u_j \right\} u_i, \tag{26}$$

$$G_{ik} = -m_k g \left(\overline{\varepsilon}_i \widehat{c}_{ik} u_g + \varepsilon_i u_g \right) u_i, \tag{27}$$

$$Q_{ik} = \left[\overline{\varepsilon}_i \left(M_k + \widehat{c}_{ik} F_k \right) + \varepsilon_i F_k \right] u_i. \tag{28}$$

The advantages of this dynamic model are as follows.

This formulation uses 3 × 3 matrices and 3 × 1 vectors to describe the dynamics of the system, which can be exploited to reduce the computational complexity of the closed-form equations of motion.

This formulation significantly improves the understanding of the equations of motion of a specific system:

> It provides the ability to separate the dynamic forces with respect to its mass and inertia and with respect to its type of joint (translational or rotational)
>
> It may be employed to accurately identify the influence of a body on each joint force. Furthermore, the later operation can be accomplished with each part of the equations, such as the inertia, gravity, centrifugal, Coriolis, and external force terms, of each body S_k applied to each joint.

For applications, in order to use this method, it is necessary to consider the attached bases of the different matrices and vectors by using the rotation matrices between the bases. As a consequence, equations (24) through (28) should be written as follows:

$$\begin{aligned}
H_{\mathrm{ijk}} &= \left\{ \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_j \left(\overline{\varepsilon}_k R_{ik} I_k R_{kj} - m_k \widehat{c}_{ik} R_{ij} \widehat{c}_{jk} \right) + \varepsilon_j m_k \widehat{c}_{ik} R_{ij} \right) + \varepsilon_i R_{ij} \left(\varepsilon_j m_k I - \overline{\varepsilon}_j m_k \widehat{c}_{jk} \right) \right] u_j \right\} u_i, \\
\beta_{\mathrm{ijk}} &= \left\{ \overline{\varepsilon}_j \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_k R_{ij} \widehat{u}_j R_{jk} I_k R_{kj} - m_k \widehat{c}_{ik} R_{ij} \widehat{u}_j \widehat{c}_{jk} \right) - \varepsilon_i m_k R_{ij} \widehat{u}_j \widehat{c}_{jk} \right] u_j \right\} u_i, \\
\zeta_{\mathrm{ijnk}} &= \left\{ \overline{\varepsilon}_n \left[\overline{\varepsilon}_i \left(\overline{\varepsilon}_j \left(\overline{\varepsilon}_k R_{ik} \mathrm{dual} \left(J_k R_{kn} u_n \right) R_{kj} - m_k \widehat{c}_{ik} R_{in} \widehat{u}_n R_{nj} \widehat{c}_{jk} \right) \right. \right. \right. \\
&\quad \left. \left. \left. + \varepsilon_j m_k \widehat{c}_{ik} R_{in} \widehat{u}_n R_{nj} + \varepsilon_i R_{in} \widehat{u}_n R_{nj} \left(\varepsilon_j m_k I - \overline{\varepsilon}_j m_k \widehat{c}_{jk} \right) \right] u_j \right) \right\} u_i, \\
G_{ik} &= -m_k g \left(\overline{\varepsilon}_i \widehat{c}_{ik} R_{i0} u_g + \varepsilon_i R_{i0} u_g \right) u_i, \\
Q_{ik} &= \overline{\varepsilon}_i \left[\left(R_{ik} M_k + \widehat{c}_{ik} R_{ik} F^k \right) + \varepsilon_i R_{ik} F_k \right] u_i.
\end{aligned} \tag{29}$$

3.5. Matrix Form of the Equations of Motion. The dynamic equations of motion (23) can be established in a more closed form vector-matrix notation as follows:

$$\tau = M(q)\ddot{q} + \dot{q}^T C(q) \dot{q} + G(q) + Q, \tag{30}$$

where $\tau \in R^N$ is the generalized forces vector; $q, \dot{q}, \ddot{q} \in R^N$ are the joint positions, velocities, and accelerations vectors; $M(q) \in R^{N \times N}$ is the generalized inertia matrix of the manipulator; $C \in R^{N \times N \times N}$ is the centrifugal/Coriolis matrix; $G(q) \in R^N$ is the gravity force vector; and $Q(q) \in R^N$ is the vector of external forces. In this formulation, the inertia and Coriolis/centrifugal matrices of the manipulator as well as the gravitational force is only dependent on the configuration of the manipulator q.

The generalized inertia matrix $M(q)$ can be denoted by

$$M(q)=\begin{bmatrix} M_{11} & M_{12} & M_{13} & \dots & M_{1N} \\ M_{21} & M_{22} & M_{23} & \dots & M_{2N} \\ M_{31} & M_{32} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ M_{N1} & M_{N2} & \dots & \dots & M_{NN} \end{bmatrix}=\begin{bmatrix} \sum\limits_{j=1}^{N} H_{11j} & \sum\limits_{j=2}^{N} H_{12j} & \sum\limits_{j=3}^{N} H_{13j} & \dots & \sum\limits_{j=N}^{N} H_{1Nj} \\ \sum\limits_{j=2}^{N} H_{21j} & \sum\limits_{j=2}^{N} H_{22j} & \sum\limits_{j=3}^{N} H_{23j} & \dots & \sum\limits_{j=N}^{N} H_{2Nj} \\ \sum\limits_{j=3}^{N} H_{31j} & \sum\limits_{j=3}^{N} H_{32j} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \sum\limits_{j=N}^{N} H_{N1j} & \sum\limits_{j=N}^{N} H_{N2j} & \dots & \dots & \sum\limits_{j=N}^{N} H_{NNj} \end{bmatrix}. \quad (31)$$

The centrifugal/Coriolis $N \times N \times N$ matrix C can be denoted by

$$C=\left[\, C_1 \;\; C_2 \;\; \dots \;\; C_N \,\right]^T, \quad (32)$$

where C_i is a $N \times N$ matrix denoted as

$$C_i=\begin{bmatrix} B_{i1} & C_{i21} & C_{i31} & \dots & C_{iN1} \\ C_{i21} & B_{i2} & C_{i32} & \dots & C_{iN2} \\ C_{i31} & C_{i32} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ C_{iN1} & C_{iN2} & \dots & \dots & B_{iN} \end{bmatrix}=\begin{bmatrix} \sum\limits_{j=\max(i,1)}^{N} \beta_{i1j} & \sum\limits_{j=\max(i,2)}^{N} \zeta_{i21j} & \sum\limits_{j=\max(i,3)}^{N} \zeta_{i31j} & \dots & \sum\limits_{j=\max(i,N)}^{N} \zeta_{iN1j} \\ \sum\limits_{j=\max(i,2)}^{N} \zeta_{i21j} & \sum\limits_{j=\max(i,2)}^{N} \beta_{i2j} & \sum\limits_{j=\max(i,3)}^{N} \zeta_{i32j} & \dots & \sum\limits_{j=\max(i,N)}^{N} \zeta_{iN2j} \\ \sum\limits_{j=\max(i,3)}^{N} \zeta_{i31j} & \sum\limits_{j=\max(i,3)}^{N} \zeta_{i32j} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \sum\limits_{j=\max(i,N)}^{N} \zeta_{iN1j} & \sum\limits_{j=\max(i,N)}^{N} \zeta_{iN2j} & \dots & \dots & \sum\limits_{j=\max(i,N)}^{N} \beta_{iNj} \end{bmatrix}. \quad (33)$$

4. Application Example: 5-DOF SCARA Manipulator

4.1. Presentation of the Manipulator. The proposed method has been employed for the 5-DOF SCARA robot (Figure 2) [10].

$q=\left[\, q_1 \;\; q_2 \;\; q_3 \;\; q_4 \;\; q_5 \,\right]^T$: the generalized coordinates for revolute joints $i=[2,3,4]$, $q_i=\theta_i$ and for prismatic joints $i=[1,5]$, $q_i=d_i$.

l_1, l_2, l_3, l_4, l_5: the lengths of the five links

l_{c2}, l_{c3}, l_{c4}: the lengths from the origins to the centers of mass of the second, third, and fourth links

m_1, $m_2 m_3 m_4$, m_5: the masses of the links

I_2, I_3, I_4: the inertia matrix of the second, third, and fourth links

4.2. Kinematics. To write the transformations for each link, the Denavit–Hartenberg method [10, 17] has been used, whose parameters are mentioned in Table 2.

The general transformation matrix between the $i-1$th link and the i-th link can be given by

$$T_{i-1,i}=\begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

The general rotation matrix between the $i-1$th link and the i-th link can be given by

$$R_{i-1,i}=\begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix}. \quad (35)$$

Substitute the DH parameters from Table 2 in equation (34). Then, the homogenous transformation matrix between the end-effector and the base can be expressed by

$$T=\begin{bmatrix} c_{234} & s_{234} & 0 & l_2 c_2 + l_3 c_{23} + l_4 c_{234} \\ s_{234} & -c_{234} & 0 & l_2 s_2 + l_3 s_{23} + l_4 s_{234} \\ 0 & 0 & -1 & l_1 + d_1 - l_5 - d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (36)$$

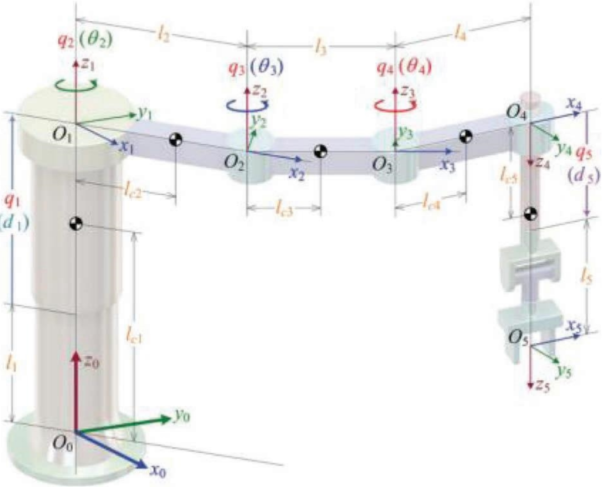


Figure 2: Scheme of the 5-DOF SCARA robot.

Table 2: Parameters of Denavit–Hartenberg.

Joint i	θ_i	d_i	a_i	α_i
1	0°	$l_1 + d_1$	0	0°
2	θ_2	0	l_2	0°
3	θ_3	0	l_3	0°
4	θ_4	0	l_4	180°
5	0°	$l_5 + d_5$	0	0°

From equation (36), we can get the end-effecter coordinates:

$$\begin{cases} x = l_2 c_2 + l_3 c_{23} + l_4 c_{234}, \\ y = l_2 s_2 + l_3 s_{23} + l_4 s_{234}, \\ z = l_1 + d_1 - l_5 - d_5. \end{cases} \tag{37}$$

For simplicity, the condition $\varphi = \theta_2 + \theta_3 + \theta_4$ is set. Where φ is known, then the inverse kinematic model is given by

$$\begin{aligned} c_3 &= \frac{\left[\left(x - l_4 c_\varphi\right)^2 + \left(y - l_4 s_\varphi\right)^2 - l_2^2 - l_3^2\right]}{\left(2 l_2 l_3\right)}, \\ s_3 &= \pm\sqrt{1 - c_3^2}, \\ \theta_3 &= \arctan 2\left(s_3, c_3\right), \\ s_2 &= \frac{\left[\left(y - l_4 s_\varphi\right)\left(l_2 + l_3 c_3\right) - l_3 s_3\left(x - l_4 c_\varphi\right)\right]}{\left(l_2^2 + l_3^2 + 2 l_2 l_3 c_3\right)}, \\ c_2 &= \frac{\left[\left(x - l_4 c_\varphi\right)\left(l_2 + l_3 c_3\right) - l_3 s_3\left(y - l_4 s_\varphi\right)\right]}{\left(l_2^2 + l_3^2 + 2 l_2 l_3 c_3\right)}, \\ \theta_2 &= \arctan 2\left(s_2, c_2\right), \\ \theta_4 &= \varphi - \theta_3 - \theta_2, \\ d_5 &= l_1 + d_1 - l_5 - z, \end{aligned} \tag{38}$$

where z and $d1$ are known.

4.3. The Inverse Dynamics of the 5-DOF SCARA Manipulator.

The inverse dynamic model of the SCARA manipulator when the external forces are omitted can be expressed in a compact form as

$$\tau = M(q)\ddot{q} + \dot{q}^T C(q)\dot{q} + G(q). \tag{39}$$

Here, $M(q) \in R^{5\times5}$ is the generalized mass matrix; $C \in R^{5\times5\times5}$ is the centrifugal/Coriolis matrix; $G(q) \in R^{5\times1}$ is the gravitational force; $q, \dot{q}, \ddot{q} \in R^{5\times1}$ are the generalized coordinates, speeds, and accelerations, respectively; $\tau \in R^{5\times1}$ is the generalized forces vector.

To the opposite of the aforementioned methods, the main advantage of the new formalism is to derive the dynamics model directly from the physical parameters and configuration of the manipulator without complex computations, such as the differentiation of energies, and without recursive algorithms. Practically, it is only needed to introduce the physical parameters of the system and to apply matrix algebra.

This procedure is implemented in a symbolic computer software algebra called SYMPY based on the Python language, as shown in Figure 3.

In the next section, a comparative study in terms of computational cost between our approach and other formulations is made to illustrate the power of the new formulation.

5. Results and Discussion

In the previous section, the dynamic description of a 5-DOF SCARA manipulator is obtained by the generalized new formulation. However, the efficiency of this method with respect to other existing formulations has not been determined to date. The objective of this section is to validate the presented method, then compare it with other approaches in terms of dynamic formulation complexity, and finally prove its applicability for online control.

$$\begin{bmatrix}
-gm_1 - gm_2 - gm_3 - gm_4 - gm_5 + m_5\dot{v}_5 - (m_1 + m_2 + m_3 + m_4 + m_5)\,\dot{v}_1 + F_1 \\
l_2 l_3 m_4(\omega_2 + \omega_3)^2 \sin(\theta_3) - l_2 l_3 m_4 \omega_2^2 \sin(\theta_3) + l_2 l_3 m_5(\omega_2 + \omega_3)^2 \sin(\theta_3) - l_2 l_3 m_5 \omega_2^2 \sin(\theta_3) + l_2 l_4 m_5(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_3 + \theta_4) \\
- l_2 l_4 m_5 \omega_2^2 \sin(\theta_3 + \theta_4) + l_2 l_{c3} m_3(\omega_2 + \omega_3)^2 \sin(\theta_3) - l_2 l_{c3} m_3 \omega_2^2 \sin(\theta_3) + l_2 l_{c4} m_4(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_3 + \theta_4) - l_2 l_{c4} m_4 \omega_2^2 \sin(\theta_3 + \theta_4) \\
- l_3 l_4 m_5(\omega_2 + \omega_3)^2 \sin(\theta_4) + l_3 l_4 m_5(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_4) - l_3 l_{c4} m_4(\omega_2 + \omega_3)^2 \sin(\theta_4) + l_3 l_{c4} m_4(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_4) \\
- (I_{4z} + l_4 m_5 (l_2 \cos(\theta_3 + \theta_4) + l_3 \cos(\theta_4) + l_4) + l_{c4} m_4 (l_2 \cos(\theta_3 + \theta_4) + l_3 \cos(\theta_4) + l_{c4}))\,\dot{\omega}_4 \\
- \left(I_{3z} + I_{4z} + l_{c3} m_3 (l_2 \cos(\theta_3) + l_{c3}) + m_4 \left(l_2 l_3 \cos(\theta_3) + l_2 l_{c4} \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_{c4} \cos(\theta_4) + l_{c4}^2\right)\right. \\
\left. + m_5 \left(l_2 l_3 \cos(\theta_3) + l_2 l_4 \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_4 \cos(\theta_4) + l_4^2\right)\right) \dot{\omega}_3 \\
- \left(I_{2z} + I_{3z} + I_{4z} + l_{c2}^2 m_2 + m_3 \left(l_2^2 + 2 l_2 l_{c3} \cos(\theta_3) + l_{c3}^2\right) + m_4 \left(l_2^2 + 2 l_2 l_3 \cos(\theta_3) + 2 l_2 l_{c4} \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_{c4} \cos(\theta_4) + l_{c4}^2\right)\right. \\
\left. + m_5 \left(l_2^2 + 2 l_2 l_3 \cos(\theta_3) + 2 l_2 l_4 \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_4 \cos(\theta_4) + l_4^2\right)\right) \dot{\omega}_2 + T_2 \\
- l_2 l_3 m_4 \omega_2^2 \sin(\theta_3) - l_2 l_3 m_5 \omega_2^2 \sin(\theta_3) - l_2 l_4 m_5 \omega_2^2 \sin(\theta_3 + \theta_4) - l_2 l_{c3} m_3 \omega_2^2 \sin(\theta_3) - l_2 l_{c4} m_4 \omega_2^2 \sin(\theta_3 + \theta_4) - l_3 l_4 m_5(\omega_2 + \omega_3)^2 \sin(\theta_4) \\
+ l_3 l_4 m_5(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_4) - l_3 l_{c4} m_4(\omega_2 + \omega_3)^2 \sin(\theta_4) + l_3 l_{c4} m_4(-\omega_2 - \omega_3 - \omega_4)^2 \sin(\theta_4) \\
- (I_{4z} + l_4 m_5 (l_3 \cos(\theta_4) + l_4) + l_{c4} m_4 (l_3 \cos(\theta_4) + l_{c4}))\,\dot{\omega}_4 \\
- \left(I_{3z} + I_{4z} + l_{c3}^2 m_3 + m_4 \left(l_3^2 + 2 l_3 l_{c4} \cos(\theta_4) + l_{c4}^2\right) + m_5 \left(l_3^2 + 2 l_3 l_4 \cos(\theta_4) + l_4^2\right)\right) \dot{\omega}_3 \\
- \left(I_{3z} + I_{4z} + l_{c3} m_3 (l_2 \cos(\theta_3) + l_{c3}) + m_4 \left(l_2 l_3 \cos(\theta_3) + l_2 l_{c4} \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_{c4} \cos(\theta_4) + l_{c4}^2\right)\right. \\
\left. + m_5 \left(l_2 l_3 \cos(\theta_3) + l_2 l_4 \cos(\theta_3 + \theta_4) + l_3^2 + 2 l_3 l_4 \cos(\theta_4) + l_4^2\right)\right) \dot{\omega}_2 + T_3 \\
- l_2 l_4 m_5 \omega_2^2 \sin(\theta_3 + \theta_4) - l_2 l_{c4} m_4 \omega_2^2 \sin(\theta_3 + \theta_4) - l_3 l_4 m_5(\omega_2 + \omega_3)^2 \sin(\theta_4) - l_3 l_{c4} m_4(\omega_2 + \omega_3)^2 \sin(\theta_4) - \left(I_{4z} + l_4^2 m_5 + l_{c4}^2 m_4\right) \dot{\omega}_4 \\
- (I_{4z} + l_4 m_5 (l_3 \cos(\theta_4) + l_4) + l_{c4} m_4 (l_3 \cos(\theta_4) + l_{c4}))\,\dot{\omega}_3 \\
- (I_{4z} + l_4 m_5 (l_2 \cos(\theta_3 + \theta_4) + l_3 \cos(\theta_4) + l_4) + l_{c4} m_4 (l_2 \cos(\theta_3 + \theta_4) + l_3 \cos(\theta_4) + l_{c4}))\,\dot{\omega}_2 + T_4 \\
gm_5 + m_5 \dot{v}_1 - m_5 \dot{v}_5 + F_5
\end{bmatrix}$$

Figure 3: Equations of motion for the 5-DOF SCARA manipulator.

5.1. Model Validation. The direct dynamics are employed to validate the present model. For the direct dynamic problem, the generalized active forces are given. The objective is to determine the end-effector trajectory or motion. To prove the correctness of the dynamical model, the simulation results in terms of the joint profile obtained by our approach and those derived by existing methods like Kane's method are compared with each other. Figure 4 shows the results calculated using the two methods. The results are identical, which means they produce the same output, which validates the correctness of the presented model.

5.2. Computational Cost. In order to prove the power of this new formulation with respect to other existing methods, we also employed the Lagrange–Euler method, Kane's method, and GIM (generalized inertia matrix)-Christoffel's algorithm to calculate the same equations of motion of the 5-DOF SCARA manipulator. The computer algebra software SYMPY is used to compute symbolically the dynamics model using these four methods. The algorithm is executed on a computer equipped with an Intel Core i3-6006U CPU. Table 3 shows the comparison of simulation results in terms of computational cost obtained by the aforementioned methods and our method.

The results confirm clearly the power of the new formalism for efficient generation of the explicit form of equations of motion. By using more advanced computer architecture, we can have the minimum CPU time. As the different terms of the dynamic equations can be computed simultaneously, the present method is more appropriate for parallel algorithms. Consequently, the robot model can be implemented for real-time control.

It is more desirable to get precise and efficient explicit forms of equations of motion because of their important role in model-based control. In Section 5.3, we will illustrate that the dynamics model obtained from our method could be implemented in real-time control.

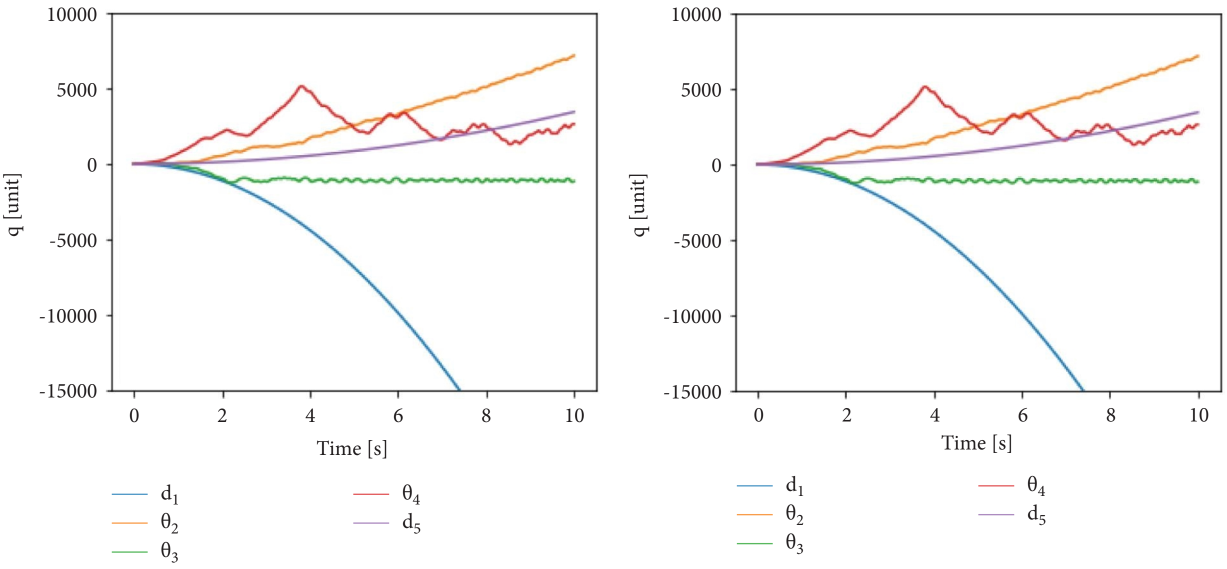
5.3. Computed Torque Control CTC Based on the New Formulation. In the previous section, a comparative study in terms of CPU time between the different algorithms was achieved, and it was concluded that the new formulation is the more efficient one in providing the explicit form of equations of motion. In this subsection and based on the SCARA manipulator, we will demonstrate that the fully numeric computation and the symbolic computation of the inverse dynamic for online control based on the new formulation are equivalent.

The desired trajectory to follow denoted by $X_d = [\, x_d \;\; y_d \;\; z_d \;\; \varphi_d \,]$ is defined as

$$\begin{cases}
x_d(t) = 0,1.\ \cos\left(\dfrac{t}{2}\right)\cos(t) + 0,2, \\[2ex]
y_d(t) = 0,1.\ \sin\left(\dfrac{t}{2}\right)\cos(t) + 0,2, \\[2ex]
z_d(t) = 0,2, \\[2ex]
\varphi(t) = \dfrac{\pi}{3}.
\end{cases} \tag{40}$$

To track the desired trajectory, the model-based control scheme chosen in the present work is the computed-torque control as shown in Figure 5.

To illustrate the power of the new formulation in online computation of equations of motion, the numeric and symbolic computation of the inverse dynamic of the SCARA manipulator are elaborated and compared using the SymPy and NumPy libraries of the Python language. In the numerical



(a)

(b)

Figure 4: Joints profile computed by (a) the proposed method and (b) Kane's method.

Table 3: Computational cost.

Formulations	Estimated CPU time (s)
Kane's method	46
Lagrange formulation	32
GIM-Christoffel's algorithm	7
New formulation	3

methods, a numerical value is related to each variable that is added in model construction, and the equations of motion are repeated for each integration step. In symbolic methods, a symbol is associated with each variable, which represents their dependence on system parameters and states. Unfortunately, the generation of symbolic equations is an extremely complex problem for complicated mechanisms.

The control law is defined by

$$\tau = M\left(\ddot{q}_d + Kp \cdot e + K\ d \cdot \dot{e}\right) + \dot{q}^T C\dot{q} + G, \tag{41}$$

where $e = q_d - q \in R^n$ is the error vector in the joint space and is the difference between the desired values of the joint coordinates and the actual values, and $Kp = \text{diag }(kp)$, $Kd = \text{diag }(kd)$.

To simulate the manipulator motion, the direct dynamic is used, and then, equation (39) can be written as

$$\ddot{q} = M^{-1}\left(\tau - \dot{q}^T C\dot{q} - G\right), \tag{42}$$

where τ represents the control torque vector. For a given value of generalized control input τ, the resulting motion can be obtained by the numerical integration of the generalized accelerations. As for the 5-DOF SCARA manipulator, we can get the discrete state space model as follows:

$$\begin{aligned} q(k+1) &= q(k) + \dot{q}(k) \cdot \mathrm{d}t + \frac{1}{2}\ddot{q}\,(k) \cdot \mathrm{d}t^2, \\ \dot{q}(k+1) &= \dot{q}(k) + \ddot{q}(k) \cdot \mathrm{d}t, \end{aligned} \tag{43}$$

where the integration step is chosen to be $\mathrm{d}t = 1\,\text{ms}$, the initial conditions of the system are given as $q(0) = [0.1, -0.8, 1.9, -0.1, 0.284]^T$, $\dot{q}(0) = [0.1, 0.4, 0, -0.5, 0.1]^T$, the values of the CTC control gains are selected through numerical simulation as follows: $kp = 200$, $k_d = 10$. The numerical values of different parameters are given in Table 4. Finally, a Python program is developed to simulate the end-effector motion during the task using the developed algorithm.

After developing the manipulator model and the simulation environment and establishing the control laws to be used, a test trajectory in space was determined to make the manipulator model follow it.

The control simulation algorithms are executed by a Python program on the same computer mentioned before. Function "time" in Python gives the estimated run time of the control cycle of each algorithm; for the fully numeric algorithm, it is 0.016 s. Whilst, for the code generated from symbolic computation, the computational cost of the control cycle is about 0.0175 s.

It is shown that the computational costs of the numeric and symbolic approaches based on the new formulation are equivalent, which is not the cases when using the traditional methods. It can be concluded that the numeric approach based on the new formulation is more suitable for online control. In addition, the present method is more suitable for parallel programming as the different terms of the equations of motion can be computed separately and simultaneously.

By considering only the revolute joints, Figure 6 shows the desired and actual motions in the workspace. The actuation torques are shown in Figure 7. The trajectory tracking error in joint space is shown in Figure 8. The desired and actual trajectories in joint space are shown in Figure 9. These results validate the actuator torques computed by the developed model.

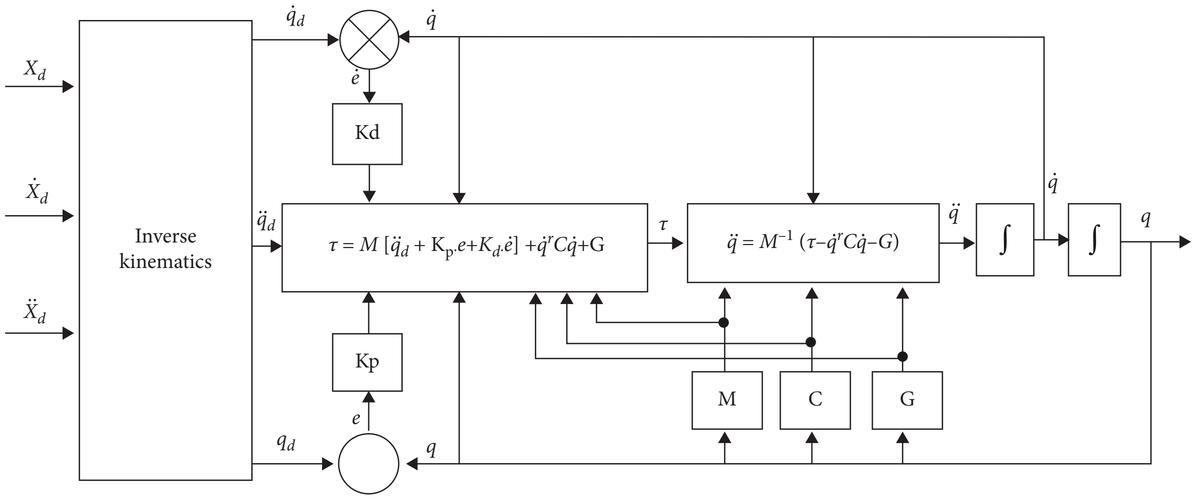


Figure 5: Computed-torque control scheme for trajectory tracking control of the manipulator.

Table 4: Parameters of the SCARA manipulator.

Parameter	Value (kg)	Parameter	Value (m)	Parameter	Value (m)	Parameter	Value (kgm^2)
m_1	1.228	l_1	0.524	l_{c1}	—	I_{zz1}	—
m_2	1.023	l_2	0.2	l_{c2}	0.229	I_{zz2}	0.0058
m_3	1.023	l_3	0.2	l_{c3}	0.229	I_{zz3}	0.0058
m_4	1.023	l_4	0.2	l_{c4}	0.229	I_{zz4}	0.0058
m_5	0.5114	l_5	0.14	l_{c5}	-	I_{zz5}	—

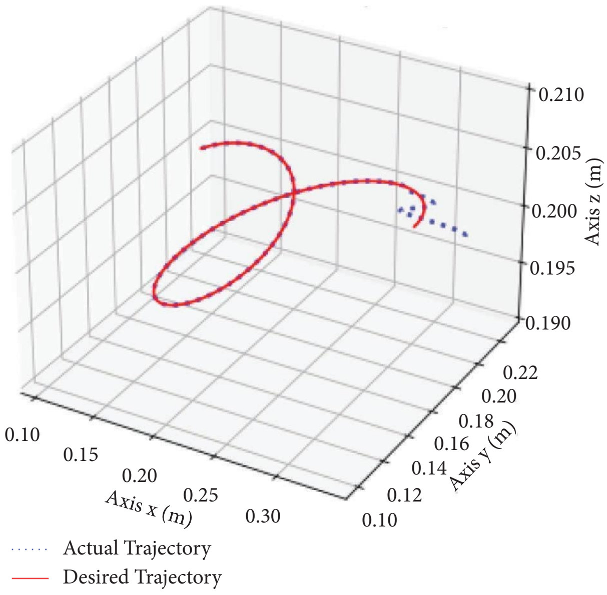


Figure 6: Desired and actual trajectories in the workspace.

Imperfections are present in the robot model, such as friction in the joints and unknown external perturbations and uncertainties in the parameters. So, it is more important to take these effects into account in order to obtain precise and perfect explicit closed-form dynamics in the actuation space because of their crucial role for model-based control; the more perfect the model is, the better the performances of its model-based controllers are [18]. However, when the dynamic model is employed in model-based control, the neglecting of any part of the dynamic equations will affect the control performance largely, which will be shown in Figure 10.

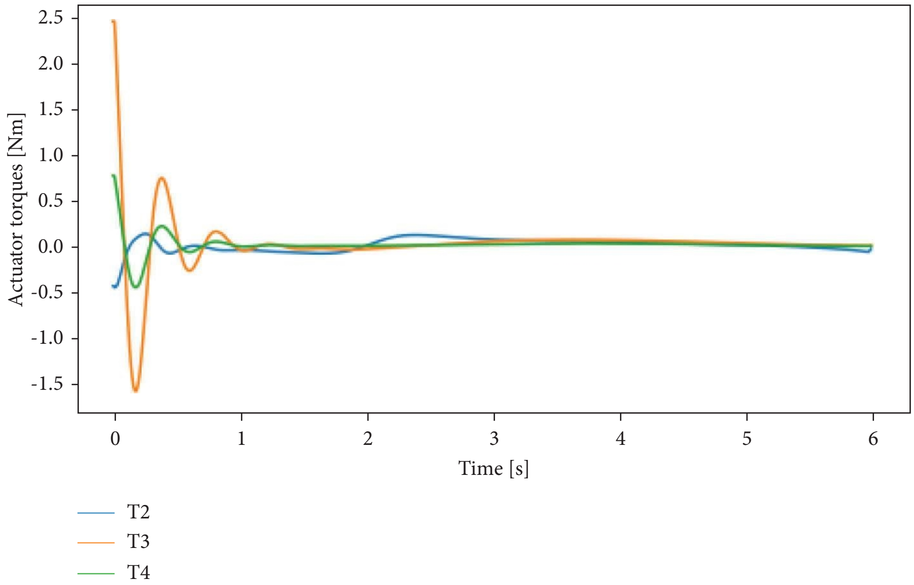


Figure 7: Actuation torques.

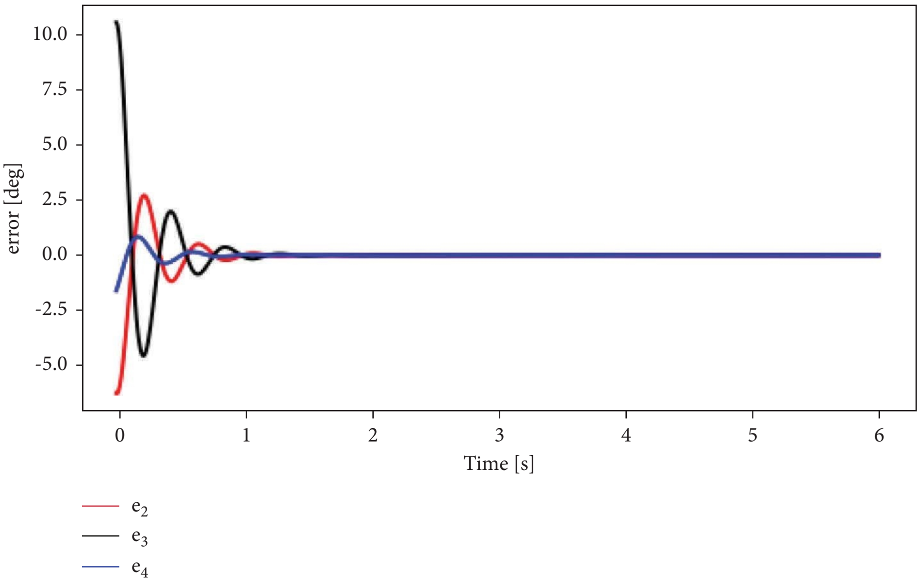


Figure 8: The trajectory tracking error in joint space using perfect inverse dynamic.

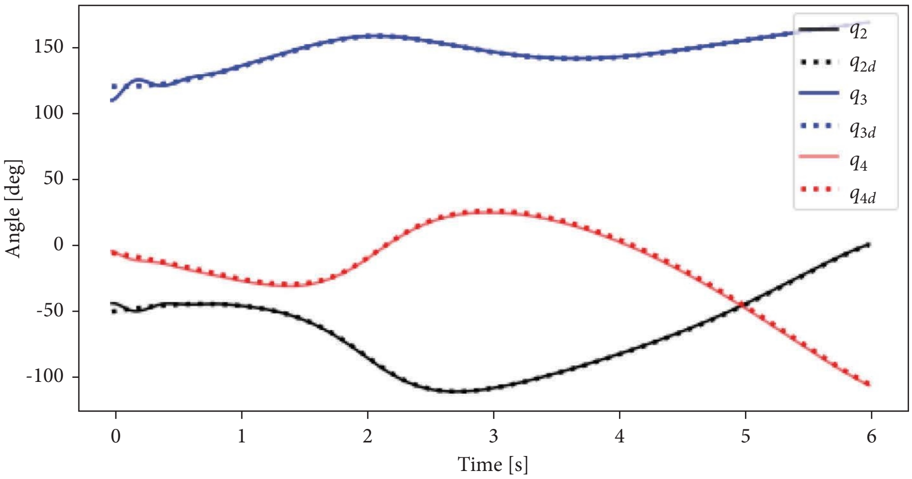


Figure 9: The desired and actual joint trajectories.

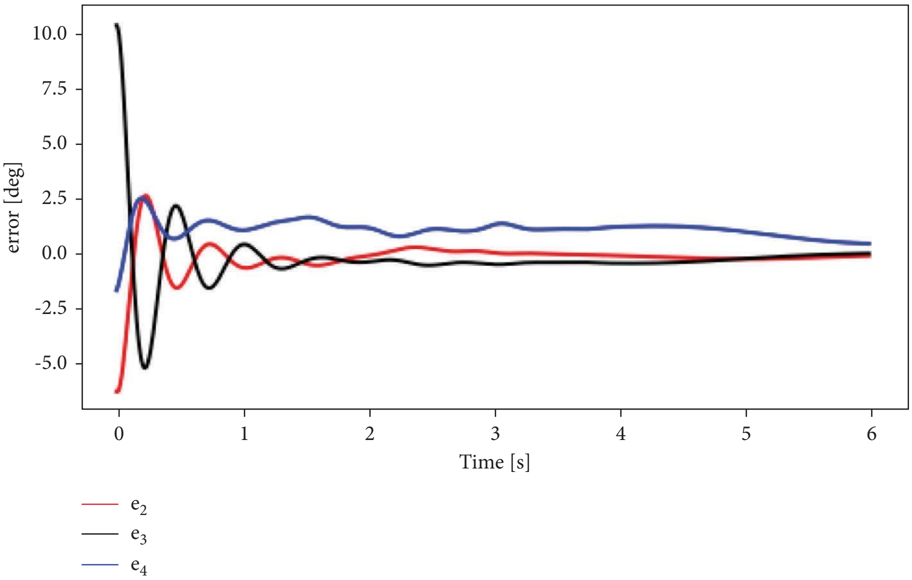


Figure 10: Trajectory tracking error using imperfect inverse dynamic.

6. Conclusion

In this paper, the domain of multibody dynamics is restudied, and a new systematic approach is extended to derive explicit closed-form dynamic equations for a general serial manipulator (i.e., systems with revolute and/or prismatic joints). The method depends on matrix algebra prior to performing differentiation. Owing to the use of a 3×3 matrix instead of a 4×4 matrix and the fact to isolate the effect of each body with respect to its inertia and mass and with respect to the type of joint (translational or rotational), the computational redundancy is reduced from the modeling procedure, making it also an efficient automated method of model formation. Furthermore, it is shown that the inertia and centrifugal/Coriolis matrices can be calculated separately and independently of generalized velocities and accelerations, which makes the determination of the equations of motion straightforward and more compact. Based on the demonstration of the developed model, a 5-DOF SCARA (selective compliant assembly robot arm) manipulator is used to illustrate its simplicity and correctness as compared with other formalisms, which are proved by simulations. A comparative study of dynamics formulation complexity has demonstrated the power of the new formulation with respect to Kane's method, Lagrange–Euler method, and GIM-Christoffel's algorithm. It is shown that the derivation process using the new formulation for the explicit closed-form equations of motion is more efficient, and the computational cost is low enough for online model-based control. In future work, it is important to adapt the new formulation to parallel manipulators.

Data Availability

The datasets used to support the results of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory Methods and Algorithms*, Springer, Berlin, Germany, 2007.

[2] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Prentice Hall, Hoboken, NJ, USA, 2005.

[3] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*, Butterworth-Heinemann, Oxford, UK, 2004.

[4] G. Xin, H. Deng, and G. Zhong, "Closed-form dynamics of a 3-DOF spatial parallel manipulator by combining the lagrangian formulation with the virtual work principle," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 1329–1347, 2016.

[5] W. Khalil, "Dynamic modeling of robots using newton-euler formulation," *Informatics in Control Automation and Robotics, Lecture Notes in Electrical Engineering*, vol. 89, 2011.

[6] B. Dasgupta and T. S. Mruthyunjaya, "A newton-euler formulation for the inverse dynamics of the stewart platform manipulator," *Mechanism and Machine Theory*, vol. 33, no. 8, pp. 1135–1152, 1998.

[7] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69–76, 1980.

[8] K. Miller and R. Clavel, "The lagrange-based model of delta-4 robot dynamics," *Roboter Systeme*, vol. 8, pp. 49–54, 1992.

[9] M. E. Kahn, *The near minimum-time control of open articulated kinematic chains*, Ph.D. thesis, Stanford University, Stanford, CA, USA, 1969.

[10] C. Urrea and J. Kern, "Trajectory tracking control of a real redundant manipulator of the SCARA type," *Journal of Electrical Engineering and Technology*, vol. 11, no. 1, pp. 215–226, 2016.

[11] J. M. Hollerbach, “A recursive lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 11, pp. 730–736, 1980.

[12] J. Angeles, O. Ma, and A. Rojas, “An algorithm for the inverse dynamics of *n*-axis general manipulators using kane’s equations,” *Computers & Mathematics with Applications*, vol. 17, no. 12, pp. 1545–1561, 1989.

[13] Z. Hussain and N. Z. Azlan, *Kane’s Method for Dynamic Modeling*I2CACIS, Malaysia, 2016.

[14] S. M. Mirtaheri and H. Zohoor, “Efficient formulation of the Gibbs-Appell equations for constrained multibody systems,” *Multibody System Dynamics*, vol. 53, no. 3, pp. 303–325, 2021.

[15] S. Bertrand and O. Bruneau, “A clear description of system dynamics through the physical parameters and generalized coordinates,” *Multibody System Dynamics*, vol. 29, no. 2, pp. 213–233, 2013.

[16] M. Vulliez, S. Lavernhe, and O. Bruneau, “Dynamic approach of the feedrate interpolation for trajectory planning process in multi-axis machining,” *International Journal of Advanced Manufacturing Technology*, vol. 88, pp. 2085–2096, 2017.

[17] R. Featherstone, *Rigid Body Dynamics Algorithms*, Springer, New York, NY, USA, 2008.

[18] M. Bucolo, A. Buscarino, C. Famoso, L. Fortuna, and S. Gagliano, “Imperfections in integrated devices allow the emergence of unexpected strange attractors in electronic circuits,” *IEEE Access*, vol. 9, pp. 29573–29583, 2021.