applied sciences

MDPI

Review

A Review of Mathematical Models in Robotics

Pubudu Suranga Dasanayake, Virginijus Baranauskas *, Gintaras Dervinis and Leonas Balasevicius

Department of Automation, Faculty of Electrical and Electronics Engineering, Kaunas University of Technology, Studentu St. 48-320, LT-51367 Kaunas, Lithuania; pubudusdasanayake@gmail.com (P.S.D.); gintaras.dervinis@ktu.lt (G.D.); leonas.balasevicius@ktu.lt (L.B.)
* Correspondence: virginijus.baranauskas@ktu.lt; Tel.: +370-37-300-291

Abstract

In robotics, much emphasis is placed on mathematical modeling, as the creation, control, and optimization of robots for a wide field of work must be achieved precisely and adaptively. The aim of this paper is to present a systematic and structured approach to the literature review of mathematical models in robotics, critically considering mathematical frameworks that influence and shape robotics in light of current and prevailing trends. The paper underlines the complexities of maintaining accurate dynamic representations in robotic systems, revealing the challenges that arise from numerical simplifications. The study outlines the development of efficient remote-control systems that consider dynamic relationships among the components comprising the robot. The findings of the recent simulation prove that the developed mathematical model effectively supports designing an adaptive control system with artificial intelligence features, especially for autonomous mobile robotics with manipulators that are inherently complex and networked systems. If models are to accelerate robotics progress toward increasingly intelligent, adaptive, and efficient systems, they must learn to overcome some of the computational challenges while leveraging disciplinary synergies.

Keywords: mathematical models; robotics; adaptive robotic systems

check for updates

Academic Editors: Anton Civit, Pavol Božek, Tibor Krenicky and Yury Nikitin

Received: 15 May 2025
Revised: 15 July 2025
Accepted: 18 July 2025
Published: 21 July 2025

Citation: Dasanayake, P.S.; Baranauskas, V.; Dervinis, G.; Balasevicius, L. A Review of Mathematical Models in Robotics. *Appl. Sci.* **2025**, *15*, 8093. https://doi.org/10.3390/app15148093

1. Introduction

Mathematical modeling is the backbone of robotics in terms of understanding, design, and control. The growth of demanding and flexible mathematical models has rapidly increased due to the growing importance of their applications to a wide range of diverse disciplines such as manufacturing, health, logistics, and exploration problems. Such models are highly essential for robust decision-making in complicated dynamic scenarios, optimization, and prediction of robot behaviors. Some of the key areas where mathematical models have significantly advanced robotics include soft continuum robots, flexible robots, dual-quaternion algebra, simplified dynamic models, fractional order models, autonomous mobile robots, and artificial intelligence in robotics.

Modeling soft robots is particularly challenging mathematically due to their constantly deforming three-dimensional forms when subjected to the action of external loads. As one of the important approaches, the deformation space formulation of the finite element is important. The method encompasses bending, torsion, shear, and axial deformations and deduces a discrete model from the Cosserat rod theory. The model has been proven to be highly accurate and computationally efficient, with errors of less than 1% of the robot's length [1].

Flexible robots usually have inherent properties and the environments they operate in. That is the reason why complex dynamics is often part of the mathematical model for flexible robotics. Advances in computational methods, such as finite element analysis and machine learning, are helping to overcome some of these hurdles. Following the variational principle of Hamilton, the model describing the transverse displacement and revolved angles of an elastic connection was proposed by a system of three nonlinear integrodifferential equations and two elastic partial differential equations [2]. The finite element approach and kineto-elasto-dynamics were used to describe a Delta robot with flexible linkages. The set of linear time-varying differential equations describes flexible motions relative to the rigid body configuration [3].

Dual quaternion algebra is a powerful mathematical framework. Due to its ability to represent geometric and physical processes in an algebraic framework, dual-quaternion algebra has become increasingly popular in the field of robotics. This approach is helpful for both robot modeling and control, as it makes a variety of mathematical concepts and processes easier to represent [4].

Simplified dynamic models are essential in robotics, especially when dealing with complex systems. They enable faster simulations that are crucial for real-time applications and iterative testing. Although computational efficiency makes simplified dynamic models desirable, they can potentially have serious drawbacks. Models are likely to completely ignore the laws of classical physics, leading to wildly inaccurate representations that demand high correction torques from controllers with potentially saturated actuators [5].

Fractional-order models leverage fractional calculus to describe systems with non-integer order dynamics. These models can be used to model robot arms using changeable fractional-order difference equations. It splits the labor into developing a mechanical model and running simulations to compare the data collected. It also provides robotic systems with the freedom to capture dynamics [6]. Furthermore, fractional modeling is applied to various systems such as missile launchers and gantry cranes, whose equations have been established through Euler–Lagrangian formulations for further analysis [7]. The relationship of dynamic parameters in a changeable configuration is considered when modeling the dynamics of autonomous mobile robots with manipulators. The Newton–Euler technique derives the equations of motion with respect to the center of mass, the angular motion, and the movement of the manipulator, by which adaptive control algorithms can be realized that enhance stability and efficiency [8].

By using techniques such as geometric and topological reasoning and even fibered logical spaces for logical reasoning, mathematical modeling used in robotics shares connections with artificial intelligence. These multidisciplinary techniques, in turn, extend the interaction between AI and symbolic mathematical computation while integrating with robotic systems [9]. The equations containing geometric and nonholonomic connections that describe the motion of coupled body systems, Lagrange multipliers offer a way to account for these relationships that helps with the development of robotic control systems for cyclic motions [10]. For robot manipulators, Bound Model Predictive Control (BoundMPC) is a path planning algorithm for robots that optimizes an objective over a planned-ahead horizon based on input/output constraints such as joint limits and collisions. BoundMPC is suitable for adaptive constraint-obeying motion planning based on state-space modeling and error decomposition of path following and is therefore appropriate for use in industries where efficiency and security are required [11]. Dynamic Movement Primitives (DMPs) accommodate point-to-point and cyclic movements and reconstruct complex biological and robot motions through the assistance of differential equations. Quaternion-based DMPs allow orientation changes without singularities. Movement adaptation is facilitated through force feedback and reinforcement learning-based DMPs, as well as through Riemannian extensions to geometry that facilitate object contact and movement into the complex geometric space [12]. An end-to-end control structure for underwater vehicles was presented to avoid interruptions while station-keeping in wave-dominated seas. The system forecasts wave-induced forces and adapts control actions via the combination of a deterministic sea wave predictor (DSWP) and a nonlinear model predictive controller (NMPC). Based on experimental studies, the method improves autonomous vehicle behavior in adverse sea conditions and is robust to noisy predictions and delays [13].

A new wheel motion generator to track the centroidal motion of a quadruped-on-wheel robot, which can cross various rough terrains using model-based whole-body torque control, was proposed. The wheel contact model and the whole-body inverse kinematics model are derived using spatial vectors. The wheel motion is extracted out mathematically depending on the base and the legged motions, which serve as the kinematics model. The wheel motion generator combines both the kinematics model and the robot centroidal momentum/dynamics model. The models are decomposed into three components related to the base motion, the legged motion, and the wheel motion [14].

A horizontal-stability control framework was designed for a wheel-legged hybrid robot to ensure the stability and horizontal orientation of the robot's trunk when encountering unknown and rough terrain conditions. This framework primarily comprises a compliance controller and a terrain adaptation controller. The compliance controller is geared towards establishing compliant interactions with the terrain and tracking the desired ground reaction forces. This is achieved through the implementation of a novel adaptive impedance control method to uphold torque equilibrium in the robot's trunk [15].

Deep learning improves robot arm motion planning for e-commerce picking by creating coarse motions as a warm start for an optimizing motion planner. Unlike existing motion planning approaches, this solution improves warehouse automation efficiency by lowering calculation time drastically without sacrificing kinematically and dynamically feasible motions [16]. A memristor-based hybrid analog-digital computation system improves the robot's energy consumption and speed as it processes input of high frequencies, while decision-making is taken care of by digital. Experiments establish better performance when compared to typical digital systems and open the doorway to sophisticated, self-governing, and energy-efficient applications in harsh robotic environments [17].

Since intelligent, adaptive, and efficient systems are always in demand, surpassing computational challenges with such techniques can result in the discoveries in robotics in the future. Through methodically outlining and classifying a broad array of mathematical models and their usage, this research adds to the amount of literature already in circulation regarding mathematical modeling for robotics. It also signifies new developments and avenues that can be explored within the discipline. The paper focuses on the conceptual foundations of these models and their real-world implications and limitations through critical examination of diverse categories [18–20].

2. Methodology

The research design presents a systematic and structured approach to the literature review of mathematical models in robotics and ensures that the review is rigorous, includes high-quality sources, and provides a critical synthesis of existing research.

The scope and objectives of the investigation cover different mathematical backgrounds that form the basis for the design, analysis, and control of robotic systems. These include three main elements: first, the categorization of the mathematical models applied in these specific robotic fields for the thematic understanding of their theoretical and practical implications; second, the analysis of their limitations and computational challenges; and third, the identification of literature gaps and emerging trends that may guide future research. This objective provides a basis for understanding how mathematical models form the fields of traditional and modern robotics.

Literature Search Strategy. An extensive search of the literature was conducted to ensure that the review captured the most relevant and high-quality studies. The literature search and selection processes are shown in Figure 1. The key sources of materials for this review are well-established databases, including Web of Science Core Collection (WOS, Clarivate Analytics, Philadelphia, PA, USA), IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library, and Google Scholar. These have been supplemented by textbooks, doctoral theses, and conference proceedings to consider both foundational and applied research. Specific keywords and search terms such as 'robotics', 'mathematical model', 'flexible robot', 'simplified dynamic', 'autonomous mobile', and 'artificial intelligence' were used to ensure a thorough coverage of each domain. Boolean operators and advanced search techniques were used to refine the results and identify the most relevant studies. An extensive search of the literature provided 3532 references in the field.

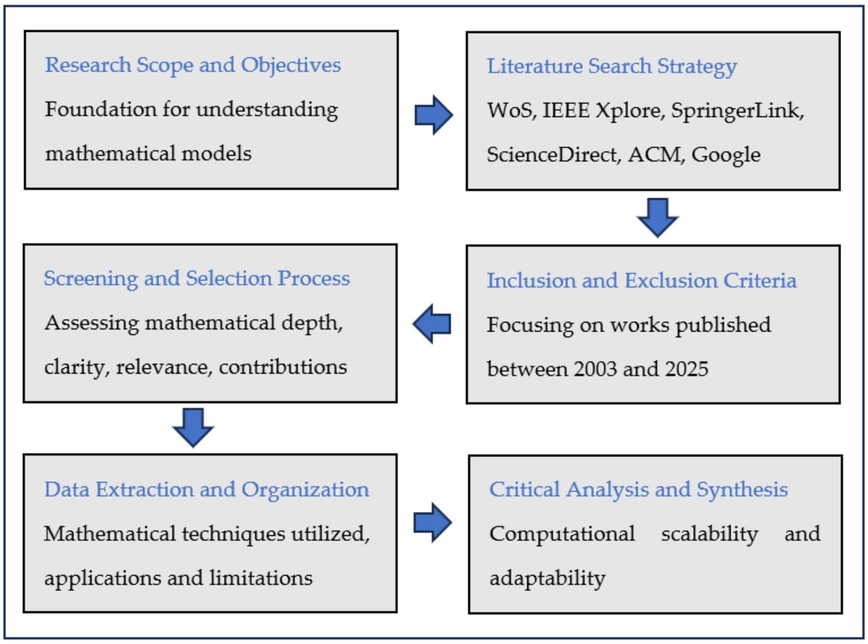

Figure 1. Literature review process.

Inclusion and Exclusion Criteria. The mathematical derivation inclusion criteria limited the review to only 309 peer-reviewed journal articles and conference papers, focusing on works published between 2003 and 2025, although seminal contributions that remain foundational to the field were considered (see Table 1). Studies with a clear emphasis on mathematical modeling were selected in one or more of the specified domains. Some of the exclusion criteria included a lack of theoretical or mathematical depth in the articles and others that were purely experimental or hardware-based without any model-based analytical content.

Screening and Selection Process. The screening of studies first began with the title and abstract screening for relevance with respect to the objective of this study. Studies retrieved from more than one database were considered duplicates and therefore removed. Subsequently, full-text reviews of the shortlisted papers were performed to confirm their alignment with the research objectives. The selected studies were then classified according to thematic areas, including but not limited to flexible robots, continuous soft robots, and autonomous mobile robots. Quality assessment criteria were aimed at assessing the mathematical depth, clarity, relevance, and contributions of each study to theoretical and practical advances in robotics. Ultimately, 60 papers were retained for review in this paper.

Table 1. Number of WOS publications in the field.

Year	Flexible Robots	Simplified Dynamic	Autonomous Mobile Robots	AI and Robotics
2025	11	9	4	10
2024	33	12	8	11
2023	13	10	4	10
2022	15	3	5	10
2021	18	7	4	7
2020	11	5	0	5
2019	6	3	5	2
2018	12	1	5	3
2017	9	2	5	1
2016	6	2	1	0

Data Extraction and Organization. Key information from selected studies was systematically extracted to facilitate comparative analysis and thematic synthesis. This included identifying the specific robotic domain addressed, the mathematical techniques utilized, and the applications and limitations of the proposed models. Data were organized into structured categories that corresponded to each focus area, providing a clear framework for analysis.

Critical Analysis and Synthesis. This work compared mathematical approaches in different domains to evaluate effectiveness, find common points, and highlight unique aspects of modeling techniques. It examined trends such as the embedding of AI into traditional models and vice versa, as well as interdisciplinary applications; it also pointed to several persistent gaps, including computational scalability and adaptability in dynamic environments. A synthesis of the findings provided insight into the evolution and possible future directions of mathematical models in robotics.

Validation of the Results. To ensure the reliability of the findings, insights were checked against multiple studies to show consistency. Several informal consultations were conducted with experts in robotics and applied mathematics to validate interpretations and identify overlooked aspects.

Presentation of results. The findings have been categorized into three main sections: a theoretical overview that summarizes the key mathematical models and their principles, domain-specific discussions that highlight findings in each robotic area, and the exploration of emerging trends and unresolved challenges. This provides a coherent narrative that bridges theoretical advancements with practical implementations. Limitations of the review include the fact that studies not in the English language and/or unpublished works may have been missed and that the interpretation of the relevance of mathematical models to domains is inherently subjective. These limitations set a framework in which the findings must be interpreted and underscore the need for ongoing research. This methodology provides a robust framework for conducting a comprehensive literature review, ensuring that the study offers meaningful insights into the contributions to the understanding and advancement of mathematical models in robotics.

3. Literature Classification and Analysis

The reviewed literature on mathematical models in robotics is categorized and analyzed in the paper for seven key domains. Each category represents a certain cross section of theoretical modeling and practical applications in robotics. Soft continuum robots are considered for the application of continuum mechanics and energy methods to address the deformable and bioinspired design. Flexible robots are analyzed in terms of their reliance on advanced kinematic and dynamic models to handle non-rigid behavior, enabling adaptability and precision in tasks like manipulation and exploration. Dual quaternion algebra is highlighted for its mathematical elegance in representing rotations and translations simultaneously, proving critical in robotic motion planning and control. Simplified dynamic models that mainly focus on reducing computational complexity while maintaining the accuracy necessary for real-time control, which is important in robotics. Fractional-order models arise from fractional calculus and can model viscoelastic and memory-dependent systems with greater accuracy, thereby providing enhanced designs for actuators and sensors. Autonomous mobile robots make use of path planning algorithms and probabilistic models to perform navigation and decision-making in dynamic environments. In general, AI in robotics brings data-driven models that complement more traditional approaches, allowing for improved perception, adaptability, and autonomy. This classification provides a systematic basis for discussing theoretical underpinnings, computational challenges, and application-driven advances in each domain, thus offering a comprehensive view of the status and future of mathematical modeling in robotics.

3.1. Soft Continuum Robots

3.1.1. A Geometrically Exact Model for Soft Continuum Robots

Finite element deformation space formulation soft robots, which have the capability of continuous deformation, raises unique modeling challenges due to their highly flexible structure and complex dynamics [21]. Traditional rigid robotics models cannot accurately capture the behavior of these systems. The finite element deformation space formulation proposes a novel, geometrically exact modeling framework that tries to balance accuracy and computational efficiency, addressing key limitations of existing approaches. It is based on Cosserat rod theory, extended to include geometric mechanics, and implemented with finite element methods. This allows an accurate representation of bending, torsion, shear, and axial deformations in soft robots under external loads [22]. Utilizing the latest mathematical tools, such as Lie groups and Lie algebras, this framework accurately captures nonlinearities in the deformations of soft robots, thus allowing very detailed but efficient simulations. The model adopts a helical shape function for spatial discretization and a geometric time-integration scheme. This maintains accuracy in the simulation of large deformations while maintaining computational efficiency. Unlike traditional constant-curvature models, which approximate robot shapes as a series of arcs, this method captures more complex and realistic deformations under diverse loading conditions [23]. The derivation of a finite element-based geometric Jacobian that relates the robot's structural velocities to the deformation dynamics follows. Such innovation will then easily enable static and dynamic analyses, and the model would represent motion and forces using the principle of virtual work. The implementation of the model in the software framework of Sim SOFT v1.0 provides an avenue for dynamic simulation and control of soft robots. Sim SOFT integrates the deformation space formulation, allowing real-time simulation of robots in diverse scenarios with significant improvements in computational efficiency over previous methods. The model is validated against experimental benchmarks, reaching positional accuracy with errors below 1% of the robot length. Dynamic simulations show the versatility of the framework, including examples of helical motion under combined bending and torsion loads. Applications span minimally invasive surgery, robotic rehabilitation, and industrial inspection. Traditional soft robot models are often based on simplifying assumptions, such as constant curvature or small-deformation approximations, which make them poorly applicable under extreme conditions. In contrast, the finite element deformation space formulation naturally accommodates large deformations and non-uniform strain distributions, making it more robust for practical applications. The Lie group-based approach avoids explicit global parameterization and therefore avoids typical drawbacks such as singularities, and improving numerical stability [24]. The framework allows modular designs; thus, complex multi-segment soft robots and parallel continuum manipulators can be simulated. Its computational efficiency is reflected in dynamic simulations that run up to twice as fast as real-time on standard hardware, which is a significant advance over existing models. The integration of analytical solutions for special cases, such as planar bending and torsion, provides intuitive insight with model manageability. Possible improvements include the optimization of computational algorithms and their extension for consideration of material heterogeneity and anisotropic properties. In this manner, the integration of real-time sensory feedback and machine learning models with the Sim SOFT platform will enhance adaptability within unstructured environments. This establishes the groundwork for more intelligent, adaptive, and versatile robotic systems that can perform complex tasks in real-world settings by extending the theoretical and computational bases of soft robotics [25].

Mathematical Models

The developed continuous model of Cosserat rods on a Lie group infers that their three-dimensional form, under an external stress field, deforms continuously. This model is established through a finite element method, which includes embedding a geometric structure for temporal integration of the form configuration and its spatial discretization. The model involves the distributed laws of internal deformations and is defined by the following equations [1]:

$$\frac{d}{ds}\left(EI\frac{^2u}{s^2}\right) + N = 0. \tag{1}$$

This equation represents the equilibrium of forces and moments in the rod, where E is the Young modulus, I is the area moment of inertia, u is the displacement vector, and N is the internal force vector.

The discrete model using deformation space formulation is developed on the basis of a geometric technique for the temporal integration of the robot shape configuration, while a helicoidal shape function has been used for spatial discretization. It considers significant deformations caused by bending, torsion, shearing, and extension. This model can be defined with the following sets of equations [1]:

$$M\ddot{q} + C\dot{q} + Kq = F, \tag{2}$$

where M is the mass matrix, C is the damping matrix, K is the stiffness matrix, q is the vector of generalized coordinates, and F is the vector of external forces.

The implicit generalized Lie group scheme, which is used for soft robot dynamics, is used to carry out the geometric time integration. The following equations define the scheme [1]:

$$q_{n+1} = exp(\Delta t v_{n+1}) \cdot q_n. \tag{3}$$

This equation represents the update of the soft robot at the next step, where q_{n+1} is the configuration at time n + 1, Δt is the time step, v_{n+1} is the velocity at time n + 1, and $q_{\boldsymbol{n}}$ is the configuration at time n.

Considering the deformations and external loading conditions of soft continuum robots, these mathematical models provide a thorough understanding of their dynamics and behavior.

Two test cases were utilized to examine the discrete model, pure bending and rotation of a soft arm under various environmental conditions, and comparisons of results with the Princeton experiment were utilized to quantify the discrete model against benchmarks. With every simulation having errors less than 1%, the outcomes for the average error in tip distance between the simulated and real robot were only 1.32 mm. With a 0.01 s step, it averaged only 2 s to compute 1 s of actual activity. This is a significant improvement over previous models and indicates the possibility for real-time application in dynamic environments. The models promise to be good for advanced model-based control methods in soft robotic manipulators because they are computationally efficient and are good at modeling geometric nonlinearities. Sim SOFT, a special-purpose simulation library for the modeling of soft robots, was created due to their implementation and is still being developed today. All things considered, the mathematical models proposed successfully combine geometric spatial and temporal dynamics, laying the groundwork for future studies of material and geometric nonlinearities in soft robotics [1].

3.1.2. Soft Robot Modeling: A Structured Overview

The different mathematical models that are being used in soft robotics are basically rooted in continuum mechanics. The key idea is that the material coordinates are linked with the spatial configuration through a definition of the motion of soft entities by means of displacement fields, mathematically represented as $r(x,t)$. The balance of forces is introduced through Cauchy's stress tensor σ, which connects internal and external forces acting on the material. Kinematic maps are an integral part of mapping generalized coordinates q to the configuration of the robot in three dimensions. The system dynamics are dictated by principles such as Newton's laws, expressed in generalized form that includes both internal and external generalized forces. FEM is also used in the research as a computational technique to discretize the equations of motion and deformation that control the robot. In this way, difficult partial differential equations can be transformed into algebraic equations that can be solved computationally. This can be addressed by a constitutive law coupling stress with strain, which can be expressed as $\sigma = D\epsilon$, able to simulate the material behavior and essentially convey how the soft body reacts under different loads. These mathematical models together provide a general framework to study the mechanics of soft robots, thereby helping in their improvement in view of application, control, and design in diverse settings. In this approach, further axiomatically fostered are the dynamic interactions and responses of the soft robotic systems [26].

Mathematical Models

In the continuum mechanics models, the governing equation describes how the momentum of the robot changes due to both internal stresses and external forces, which is fundamental in dynamic modeling. The motion of a soft robot modeled as a continuum is governed by the following balance equations:

$$\rho \frac{^2 r(x,t)}{t^2} = \nabla \cdot \sigma + f, \tag{4}$$

where ρ is the density of the material, $r(x,t)$ is the position vector of the material points in the continuum at the position x and time t, σ is the Cauchy stress tensor that describes the internal forces, and f represents body forces (e.g., gravitational forces).

In the kinematic models the robot's configuration is often described using a kinematic mapping that relates generalized coordinates to its shape [26]:

$$r(x,t) = g(q(t)) \in SE(3), \tag{5}$$

where $g(q(t))$ is a mapping function that takes the generalized coordinates $q(t)$ (which could include positions, angles, etc.) and translates them to a configuration in 3D space represented by $SE(3)$, the special Euclidean group. This captures both the position and orientation of the robot, essential for understanding its motion and deformations.

Geometric models based on the functional approaches can describe the deformed configuration using specific mathematical functions.

$$r(x,t) = h(x,t). \tag{6}$$

The function h can represent various geometries (e.g., curves) that the soft robot might assume during deformations. For instance, a serpenoid or sine curve can model a snake-like robot's motion under actuation. This approach simplifies the modeling of complex deformations by selecting appropriate functions that capture the essential shape characteristics of the robot.

The main effect is that currently, it is unrealistic to simulate soft robots with great realism and real-time calculation ability. Hence, depending on the user requirements, choosing a modeling approach is a trade-off between accuracy and computation efficiency. To seek innovative actuator concepts, designers may initially emphasize realism. Subsequently, they can move towards efficient models for control and optimization. Surrogate (data-driven) models need vast amounts of data, which provide speed and flexibility. Physics-based models, especially the discrete rod and continuum mechanics ones, are more accurate and variable but computationally expensive. Commercial FEM tools, due to their computationally expensive nature, are usually not ideal for real-time control applications, although they are excellent in complex design and phenomena [2].

It was demonstrated that the condensed FEM model formalism can be used to jointly learn both the design and control of a soft robot. Contact modeling has been integrated into the learned condensed FEM model framework. Contacts are treated as constraints, similar to actuators and effectors. It was shown that the model for both open-loop and closed-loop control can be deployed on an embedded controller. The more complex the robot, the greater the potential speedup. However, significant changes in actuation or morphology would require relearning the model, because the model has been tested with a limited number of design parameters. Expanding this approach to more complex parameterization, with a greater number of parameters or increased expressivity, would require rethinking network scaling and sampling strategies. Another limitation of the method is its dependence on fixed material and mechanical parameters, which may change over time or across different robots. However, this can be mitigated using domain randomization, as a single network can be used with different geometric or mechanical properties. Finally, to address the limitation of the offline sampling process, one potential extension is to use online learning techniques [27].

3.2. Flexible Robots

3.2.1. Mathematical Modeling and Analysis of the Delta Robot with Flexible Links

The first thing is the necessity of precise mathematical modeling for robotic systems, especially when operating at high speeds and under high loads in processes such as laser cutting or machining. This highlights the importance of considering the flexibility of joints, links, and fixed/moving bases when performing the mathematical modeling of manipulators [28]. Under the assumption that flexible robot links are rigid entities, recent studies have concentrated on the kinematic and dynamic analysis of the Delta robot. However, it is argued that, especially for lightweight mechanisms intended for high-speed motion with heavy loads, mathematical modeling should consider the flexibility of links, joints, or fixed/moving bases. It was underlined that each link of the robot was modeled with multiple beam elements, comprising an axial displacement, an axial torsion, and two transverse displacements [29]. The investigation now describes the flexible motion of the manipulator by defining a set of global variables using the DH technique and presents a different approach to combining the elements. Using this approach, any beam element can be assembled and yields a set of linear time-varying ordinary differential equations that are easy to solve by most of the numerical integrations. A few issues related to the mathematical modeling and analysis of the flexible link Delta robot with flexible links are described [30]. Among others, finite element equations, work of a tensile load, kinetic energy, and gravitational potential energy have been presented. Numerical simulations based on circular motion and an inverted U route are also presented to replicate an industrial pick-to-place motion. These numerical simulations are supported by natural frequency analysis and convergence analyses. The results of these simulations highlight how changes in the position and attitude of the end effector result as a function of linkage flexibility [31].

Mathematical Models

The dynamic equations of the robot can be derived by the Euler–Lagrange equations as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q}\right) - \frac{\partial L}{\partial q} = F, \tag{7}$$

where q refers to the generalized coordinates (X_P, Y_P, Z_P) of point P and angles ϕ_i (i = 1, 2, 3); F refers to the applied forces (F_{PX}, F_{PY}, and F_{PZ}) at point P and the applied torques τ_i (i = 1, 2, 3) to the active joints; L is the Lagrangian, which is obtained by the following equation as

$$L = T - V \tag{8}$$

where T and V refer to the kinetic energy and the potential energy, respectively. Please refer to [3] for more details about T and V expressions.

To formulate the finite element equations of the robot, one models a spatial translating and rotating beam element, which is subjected to an axial deformation, a torsional deformation, and two bending deformations. These deformations result in three displacements and one torsional angle. The modeling of these elastic deformations is based on the Euler–Bernoulli beam theory [3].

Let us assume that the beam element has a uniform density and a constant cross-sectional area. The kinetic energy of the beam element then is given as

$$T_e = \frac{1}{2}\rho A\int_0^l V_e^2 dx + \frac{1}{2}\rho I_1\int_0^l (\omega_{0X} + \dot{\varphi})^2 dx \tag{9}$$

where ρ is the mass density, A is the cross-sectional area, l is the length of the beam element, I_1 is the moment of area along the x-axis, ω_{0X} is the angular velocity of the beam element, and φ is the torsional angle on the x-axis.

The beam element is considered with axial, lateral, and torsional strains. Thus, the strain energy is given as

$$\begin{gathered} V_e = \tfrac{1}{2}EA\int_0^l \left(\tfrac{\partial u}{\partial x}\right)^2 dx + \tfrac{1}{2}EI_z\int_0^l \left(\tfrac{\partial^2 v}{\partial x^2}\right)^2 dx + \tfrac{1}{2}EI_y\int_0^l \left(\tfrac{\partial^2 \omega}{\partial x^2}\right)^2 dx + \\ \tfrac{1}{2}GI_x\int_0^l \left(\tfrac{\partial \varphi}{\partial x}\right)^2 dx\,, \end{gathered} \tag{10}$$

where E and G are the elastic and shear moduli; A is the cross-sectional area, I_x, I_y and I_z are the moments of area with respect to x-, y- and z-axes, respectively; u, v, and ω are the elastic displacements of the arbitrary point on the x-, y-, and z-axes, respectively; and l is the length of the beam element.

The gravitational potential energy of the beam element is considered as

$$V_g = \rho A \int_0^l g(r_0 + r_e) dx, \tag{11}$$

where ρ is the mass density, A is the cross-sectional area, g is the gravitational acceleration vector, and r_0 is the vector from the global reference frame to the left endpoint of the beam element, r_e is the position vector of the arbitrary point on the beam element, and l is the length of the beam element.

The work performed by a tensile load undergoing transverse deflections is given by

$$W_t = \frac{1}{2} \int_0^l P(x,\ t) \left[\left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial \omega}{\partial x} \right)^2 \right] dx, \tag{12}$$

where $P(x, t)$ is an axial inertial force, applied at position x and time t, v and w are transverse deflections, $\partial v / \partial x$ and $\partial w / \partial x$ are derivatives of the transverse deflections with respect to position x, and l is a length of the beam element.

Based on the aforementioned energy and work, the Lagrangian of the beam element can be written as

$$L_e = T_e - V_e - V_g + W_t, \tag{13}$$

Applying the Euler–Lagrange equation shown in (7), the finite element equations of the beam element are expressed as [3]

$$M_e \ddot{\delta}_e + 2 B_e \dot{\delta}_e + \left(K_e + K_d \right) \delta_e = P_e \tag{14}$$

where δ_e is a column vector in terms of the nodal variables of the beam element. Please refer to [3] for more details about M_e, B_e, K_e, K_d, and P_e expressions.

To assemble all elements, a set of global variables should be defined, which are used to describe the elastic motion of the robot. The relationship between the global variables and the nodal variables can be expressed as

$$\delta_e = S_i \delta \tag{15}$$

where δ is a column vector consisting of the global variables, and S_i is the transformation matrix from the global variables to the nodal variables for the ith beam element.

To assemble all the beam elements, applying (15) leads to a set of global equations for the robot in the form of (2). Note that the terms associated with the kinetic energy and the gravitational potential energy of the end-effector should be added into (2) [3].

According to a circular motion simulation, the natural frequencies of the robot oscillate every 120 degrees and change within a given range because of its geometry. Higher-order polynomials accelerate convergence and minimize degrees of freedom, making the computation faster, as per convergence analysis. Even though attitude changes were noted in all directions in the circular motion simulation, position oscillations of the end-effector were negligible in the X-Y plane and about 1.5 mm in the Z-axis direction, reflecting some flexural motion that deteriorated accuracy. Similarly, caused by the flexible connections, the end-effector had large attitude deviations and the largest position oscillations of about 10 mm in the inverted-U path simulation. These results emphasize the need for link flexibility consideration in planning a path for high-precision Delta robot operation, especially when precision matters in high-speed applications. The model highlights the need for adaptive path planning to achieve accurate placement and possible deviations caused by flexible links [3].

The characteristics and behavior of the Delta robot with flexible linkages can be understood by knowing these mathematical models. They provide a framework for studying and modeling the motion and flexibility of the robot, helping to create accurate mathematical representations of robotic systems.

Recently proposed a novel rigid-flexible coupling robot for industrial narrow space manipulations that incorporates the strengths of both rigid and flexible arms to enhance stiffness while ensuring reachability. The forward kinematics of the robot is uniformly derived using screw theory. Decoupling the robot's position and orientation results in closed-form inverse kinematics, guaranteeing both accuracy and efficiency. Furthermore, a two-stage search strategy is proposed to obtain the inverse kinematics of the robot that fails to conform to Pieper's solution based on the closed-form solution. A series of experiments demonstrated the feasibility and effectiveness of the inverse kinematics and the proposed rigid-dominat control method within the robot's working space. If Pieper's solution is satisfied, the closed-form inverse kinematics has achieved a 100% success rate. For the case where the Pieper solution is not satisfied, the inverse kinematics method demonstrated a higher success rate (99.1%) compared to the Jacobian-based inverse kinematics (98.7%). Furthermore, the inexact solutions converge near the target configuration [32].

3.3. Dual Quaternion Algebra

3.3.1. DQ Robotics

DQ Robotics is a computational library that uses dual quaternion algebra for robot modeling and control. Dual quaternion algebra has gained significant traction in robotics because of its capability to represent geometric transformations in a very compact and efficient way, which makes it into a very useful tool for both modeling and control applications. The dual quaternions can represent mathematical objects such as points, lines, planes, and rigid transformations easily. They emphasize that while dual quaternions enjoy some advantages, such as reduced computational complexity compared to homogeneous transformation matrices, their use has been rather limited due to a lack of efficient and user-friendly computational tools. Few existing libraries focus on quaternion algebra, let alone dual quaternion algebra. In this sense, the gap between theory and implementation must be bridged [33]. To fill this gap, the DQ Robotics library is proposed. It is suitable for self-study and education but at the same time computationally efficient for practical applications. DQ Robotics is implemented in Python (PyPI 23.4.0a61), MATLAB R2020b, and C++ 20, all sharing a common programming style that makes it easy to switch between languages. The library has been designed to resemble mathematical notation as much as possible to let the user to translate their thoughts directly into code [34]. The tutorial starts with an introduction to dual quaternion notation and basic operations, followed by robot modeling and control, and ends with a complete example of two robots collaborating on a task while avoiding obstacles. This tutorial will help provide practical information on how to use the library; code snippets are mainly in MATLAB but can also be applied in Python and C++ [35]. The object-oriented design of the library architecture is explained with simplified UML diagrams. The library provides a broad set of mathematical operations and methods for manipulating dual quaternions that makes it versatile for various robotic applications. It also supports the modeling of different types of robots, such as serial manipulators and mobile bases, and allows for the integration of these components into more complex robotic systems, such as mobile manipulators and bimanual systems. The library provides tools for both kinematic modeling and control, enabling users to compute forward kinematics and Jacobians and implement various motion controllers based on dual quaternion representations. The compatibility of the library with ROS (Robot Operating System) and V-REP—a versatile robot simulation framework—demonstrates the utility of the library in practical robotic scenarios, is presented. The performance of DQ Robotics through comparative benchmarks, showing its efficiency in dual quaternion computations compared to MATLAB and native Python implementations, is highlighted [36]. This will increase the acceptance and use of DQ-algebra, thereby significantly advancing robotic technologies and methodologies.

Mathematical Models

The representatives using dual quaternions are points, lines, planes, spheres, infinite cylinders, coordinate systems, twists, and wrenches. Given the imaginary units $\hat{\imath}$, $\hat{\jmath}$, $\hat{k}$ that satisfy $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$ and the dual unit ε, which satisfies $\varepsilon \neq 0$ and $\varepsilon^2 = 0$, the quaternion set is defined as [4]:

$$H \triangleq \left\{h_1 + \hat{\imath}h_2 + \hat{\jmath}h_3 + \hat{k}h_4 : h_1, h_2, h_3, h_4 \,\epsilon R\right\}, \tag{16}$$

and the dual quaternion set is defined as

$$\mathcal{H} \triangleq \{h_1 + \varepsilon h_2 : h_1, h_2 \,\epsilon H\}. \tag{17}$$

where h_1, h_2, h_3, h_4 are scalar components of the dual quaternion, and they belong to the set of real numbers. $\hat{\imath}$, $\hat{\jmath}$, $\hat{k}$ are imaginary unit vectors in quaternion algebra, which extend the concept of complex numbers to four dimensions. $\hat{\imath}$ represents the unit vector along the x-axis, $\hat{\jmath}$ represents the unit vector along the y-axis, and $\hat{k}$ represents the unit vector along the z-axis. This set is used to represent various geometric and kinematic properties of robots. The set of dual quaternions is particularly useful for representing rotations, translations, rigid motions, twists, wrenches, and other geometric primitives. For instance, the set $S^3 \triangleq \{h \in H : \|h\| = 1\}$ is used to represent rotations, with $\|h\| \triangleq \sqrt{hh^*} = \sqrt{h_1^2 + h_2^2 + h_3^2 + h_4^2}$ being the quaternion norm and $h^* = h_1 - \left(h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}\right)$ being the conjugate of $h = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$. Translations and rotations can be grouped into dual quaternions.

In dual-quaternion algebra, geometric primitives like points, planes, and lines have representatives, which is particularly useful for incorporating geometric constraints into robot motion controllers. For example, a plane expressed in a frame $\mathcal{F}_a$ is given by [4]:

$$\underline{\pi}^a = n^a + \epsilon d^a \tag{18}$$

where $n^a = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$ is the normal to the plane, $d^a = \langle p^a, n^a \rangle$ is the signed distance between the plane and the origin of $\mathcal{F}_a$, p^a is an arbitrary point on the plane and ε is a dual unit used in dual-quaternion algebra. A line in a frame $\mathcal{F}_a$, with direction given by $l^a = l_x\hat{\imath} + l_y\hat{\jmath} + l_z\hat{k}$ and passing through point $s^a = p_x\hat{\imath} + p_y\hat{\jmath} + p_z\hat{k}$, is represented in dual quaternion algebra as:

$$\underline{l}^a = l^a + \epsilon s^a \times l^a. \tag{19}$$

One test, for example, showed a mobile manipulator tracing a circle while avoiding a wall as well as cylindrical obstacles, effectively evading obstacles despite slight impairments in the tracking of a trajectory. Furthermore, dual quaternion algebra robotics is being adopted because of the ease of use presented by the design of DQ Robotics, so it can be applied to education as well as self-learning. The versatility and efficiency in practical use are also demonstrated through its capacity for independently creating robot models and managing multiple robotic systems [4].

Dynamic movement primitives aim to learn complex behaviors in such a way, representing tasks as stable, well-understood dynamical systems. By modeling movements over the SE(3) group, modeled primitives can be generalized for any robotic manipulator capable of full end-effector 3D movement. Liendo, F. et al. presented a robot-agnostic formulation of discrete DMP based on the dual quaternion algebra, oriented towards modeling throwing movements. DMP considers adapted initial and final poses and velocities, all computed from a projectile kinematic model and from the goal at which the projectile is aimed. Simulated and real environment experimental demonstrations support the effectiveness of the improved method formulation [37].

3.4. Simplified Dynamic Models

3.4.1. Mathematical Deficiencies of Numerically Simplified Dynamic Robot Models

Typically, model-based algorithms are used in robot control due to the complex, coupled, and non-linear dynamics of robots. This generally involves evaluation of the complete dynamic model in real time, which can be computationally expensive. Hence, many practitioners will implement simplified robot models within their controllers [38]. Such models cannot accurately represent complex robot dynamics and thus may give adverse results, including oscillations or overshooting of the end effector. They emphasize that the consequences of these simplifications are not easily quantifiable, as experimental results can often contradict simulation results. The numerically simplified models can contravene fundamental principles of classical mechanics, resulting in models that do not accurately reflect any feasible manipulator. This can force the controller to exert excessive corrective torques, potentially saturating the actuators, which can be detrimental to performance [39]. Simplification starts with a review of robot dynamics, nomenclature, and requirements for simple dynamic models with an introduction to a systematic approach toward simplification. The practical example, involving the JPL/Stanford manipulator, shows how simplifications can lead to loss of positive definiteness in the inertial matrix, a crucial property to ensure stability in control applications [40]. A systematic simplification procedure, based on a numerical significance analysis of dynamic coefficients, may lead to great errors or even negative values in the determinant of the inertial matrix—the condition of positive definiteness violated and the stability is endangered. The results show that small values of the determinant result in very large driving forces, which jeopardize the functionality of the controller. The reduced models must be compensated for in trajectory planning; thus, there is a loss of available workspace and a need for the definition of buffer zones to avoid performance loss. Careful use of simplified models must be considered when designing controllers for robots, and further research is needed to systematically simplify dynamic models without losing the crucial properties of dynamic models [41]. Ultimately, the research serves as a cautionary note for engineers and researchers in robotics, advocating for a more robust understanding of dynamic modeling to enhance control system performance and reliability.

Mathematical Models

Let us rewrite Equation (2) to describe the dynamic model of an open-chain robot with N rigid links in slightly different form [5]:

$$D(q)\ddot{q} + h(q, \dot{q}) = F, \tag{20}$$

where $D(q) = [d_{ij}] \in R^{NxN}$ is the positive-definite inertial matrix, q is the vector of generalized coordinates, $h(q, \dot{q}) \in R^N$ is the coupling vector containing centrifugal and Coriolis, gravitational and frictional effects, and F is the joint forces/torques actuating vector.

Let $d(q_1, q_2, \ldots, q_N)$ denote a generic inertial coefficient, whose functional dependency on the joint coordinates may be expressed as

$$d(q_1, q_2, \ldots, q_N) = \sum_{i=1}^{M} k_i \alpha_i(q_1, q_2, \ldots, q_N), \tag{21}$$

where each α_i is written as a product of some sines and cosines (and their powers) of rotational joint coordinates and of some (normalized) translational joint coordinates. The constants k_i depend on the geometrical and mass properties of the links as well as the range of the translational joint coordinates. Without loss of generality, let us assume that the terms in (21) have been ordered so that $|k_1| \le |k_2| \le \ldots \le |k_M| = k_{max}$. It suffices to note that

$$|\alpha_i(q_1, q_2, \ldots, q_N)| \le 1, \ \forall i. \tag{22}$$

To simplify then let us rewrite (21) under consideration as

$$d(q_1, q_2, \ldots, q_N) = k_{max} \sum_{i=1}^{M} p_i \alpha_i(q_1, q_2, \ldots, q_N), \tag{23}$$

where $p_i \equiv k_i / k_{max} so |p_i| \le 1, \forall i$. Find the greatest integer value of $L \le M$ for which

$$\sum_{i=1}^{L} |p_i| \le \varepsilon, \tag{24}$$

where ε $(0 < \varepsilon < 1)$ is tolerance. Then the simplified coefficient $\hat{d}$ is evaluated as

$$\hat{d}(q_1, q_2, \ldots, q_N) = k_{max} \sum_{i=1}^{M} \hat{p}_i \alpha_i(q_1, q_2, \ldots, q_N), \tag{25}$$

where

$$\hat{p}_i = \begin{cases} p_i & if\ i > L \\ 0 & if\ i \le L. \end{cases} \tag{26}$$

The final simplified coefficient d is verified by the numerical significance analysis of the dynamic coefficients. Under this procedure, numerically significant terms are retained while the computational complexity is reduced and the way towards real-time implementations is opened [5].

3.5. Fractional-Order Models

There are various definitions of fractional order calculus, and in fractional order neural network modeling, the three most used definitions are the Riemann–Liouville fractional order calculus, the Caputo fractional order calculus, and the Grunwald–Letnikov fractional order calculus. Some new stability criteria and boundedness results for fractional fuzzy cellular neural networks with impulse and reaction–diffusion is derived by way of the Lyapunov function method, fixed point theorem, and integral inequality techniques. The fractional order terms are slated by the fractional order differentiation and its derivative definition. The stability criteria are presented for global Mittag–Leffler stability. In addition, a new sufficient condition for the boundedness of fractional-order reaction–diffusion delayed fuzzy cellular neural networks with impulse is presented. Fuzzy cellular neural networks have unique applications in the field of image recognition. This approach aims to enhance their properties and improve their applications in robotics in the future [42].

3.5.1. New Approach to the Variable Fractional-Order Model of a Robot Arm

This is dedicated to research on mathematical models of the robot arm, including a detailed analysis of two specific models—one based on the Newton–Euler formalism, and the second using variable fractional-order difference equations [43]. The idea was also to compare these models with real measurements to determine their effectiveness in simulation. Newton–Euler formalism is a traditional method that has provided a long-standing basis for modeling robotic systems, essentially based on classical mechanics principles. The variable fractional-order difference equations model involves the use of fractional calculus, which enables a superior representation of systems with memory and hereditary properties. The final model structure integrates nonlinear behavior and variations in input signal characteristics [44]. The detailed experiments were conducted on a robot arm as part of a larger mobile platform project. The robot utilizes a DC electric drive motor with harmonic gears controlled via a single-phase PWM inverter. The study analyzed the torque profiles (M0, M1) necessary for various positions of the manipulator over time, focusing on the starting phase (increasing torque), steady phase (constant torque), reduction phase (decreasing torque to zero), and stop phase (zero torque). The results of the robot arm were obtained through simulations and real measurements [45]. Two performance indices were used, IAE and ISE. The proposed fractional-order variable model gave a better response than the classical model by presenting higher accuracy in the realization of the real arm. Graphical results were obtained representing the response of the robotic arm. Notable figures included plots of measured responses versus simulated results for both models, demonstrating the superiority of the fractional-order approach. This determined the correct order of functions to relate real-time signals of the system. The research corroborates that the variable fractional order difference equation model provides a much better conceptual and estimated framework to understand and predict the dynamics of a robot arm compared to other classical methods [46]. This gives another degree of freedom for possible enhancement of robotic control systems in a real environment.

Mathematical Models

The trade-off between model accuracy and its complexity should also be taken into consideration. The classical mathematical model (2) of the robot arm is inspired by the well-known Newton–Euler formalism. Such a dynamic system can be described by two classical differential second-order equations. This approach is based on a non-deformable suspension and a fixed mobile platform. As a discrete time, classical mathematical model, one takes an approximation of the derivative by a difference [6].

The variable fractional-order difference equation model approach is based on the general form of a linear, time-invariant variable fractional-order difference equation:

$$\sum_{i=0}^{n} A_i {}_0^{GL}\Delta_k^{(v_k)} y_k = \sum_{j=0}^{m} B_j {}_0^{GL}\Delta_k^{(\mu_k)} u_k, \tag{27}$$

where A_i, B_j are constant coefficients with $A_n = 1$; $v_{n,k} > v_{n-1,k} > \ldots > v_{1,k} > v_{0,k} = 0$ are order functions; $\mu_{m,k} > \mu_{m-1,k} > \ldots > \mu_{1,k} > \mu_{0,k} \geq 0$ are order functions; u_k, y_k are the robot arm input and output functions.

The robot arm possesses the integration element properties. Hence, if a classical backward difference is considered, then from (27) one obtains the equation:

$${}_0^{GL}\Delta_k^{(v_k)} y_k = B_0 u_k, \tag{28}$$

Describing a variable fractional-order integrator, two parameters to be selected: a coefficient B_0 and an order function v_k. Note that this is a generalization of the fractional-order discrete integrator, which in turn is a generalization of the classical (first-order) integrator. It was shown that (28) can be transformed to an equivalent form.

$$y_k = B_0 {}_0^{GL}\Delta_k^{(\mu_k)} u_k, \tag{29}$$

where $\mu_k = -v_k$. The non-linear behavior may be expected, so it can be taken as $B_0(u_k)$.

The final model is represented by Equation [6]:

$$y_k = B_0(u_k)_0^{GL}\Delta_k^{[\mu_k(u_k)]}u_k, \tag{30}$$

where $\boldsymbol{y_k}$ represents the output or response of the system at discrete time k, $B_0(u_k)$ represents the coefficient or function associated with the input u_k at time k, and $\Delta_{\boldsymbol{k}}^{[\boldsymbol{\mu_k}(\boldsymbol{u_k})]}$ represents the variable fractional-order backward difference of the response y_k at time k with respect to the order μ_k.

Comparing these models with the observed values of the robot arm, the model based on the variable fractional-order difference equation yields better results. Further research is needed in two areas, according to the article: one is the effect of platform movement on the tested system, and the other is the selection of the appropriate order function with respect to the recorded real-time data [6].

The variable fractional-order difference equations model performed better than the classical Newton–Euler formalism model, as presented by the final simulation results. Integral Squared Error and Integral Absolute Error were the two performance criteria that were utilized for judgment. It was found that the variable fractional-order model, with its higher number of parameters than the classical model, more accurately depicts the behavior of the robot arm in practice. The open problem of the choice of a suitable order function that can be effectively applied to real-time system signals is one of the prominent problems of the research. It is suggested that future work investigate the impact of the mobility of the movable platform on the robot arm's performance. In addition, the robustness of the suggested model was verified through comparisons of simulated outcomes with experimental results. Overall, the results indicate that this new direction has a very high chance of increasing the accuracy in modeling transitory behaviors and creating sophisticated control schemes for use in robotics [7].

3.6. Autonomous Mobile Robots

3.6.1. Construction of a Mathematical Model of the Dynamics of an Autonomous Mobile Robot of Variable Configuration

AMR manipulators, with an emphasis on hazardous applications, are meant to perform tasks that are unsafe or impossible for humans, especially in military, space, and nuclear contexts. The development of efficient remote-control systems that take into consideration dynamic relationships among the components comprising the robot is crucial. Key components in AMR design are kinematic schemes. The design should incorporate a kinematic scheme that details the movement and operation of the AMR with its manipulator. An information system should help remotely monitor and control AMR. The mathematical model should include the dynamic relationships among the control channels to ensure the accurate movement and operations of the AMR. The formulation of algorithms that enable real-time robot control is crucial [47]. Some of the challenges to modeling dynamics for AMRs include nonholonomic constraints. The constraint that it will not slip or skid during operation makes the control equations cumbersome. It is important to consider how dynamic parameters interact, as their negligence can result in incorrect control and operation. Extreme and unpredictable conditions will need to be considered in design; both the mechanical and control systems are influenced. The Gibbs–Appel recursive formula is used to construct the kinematic and dynamic equations of motion. The use of 3×3 and 3×1 matrices improves computational efficiency during calculations. Research advocates adaptive control methods that adjust based on real-time analysis of dynamic parameters [48]. AMRs with manipulators can significantly improve operational efficiency in high-risk environments without human intervention. Therefore, this indicates the need for more sophisticated models and control systems to be able to operate successfully within dynamic conditions. Further research on the integration of feedback mechanisms is required to improve operator control and interaction with the AMR [49].

Mathematical Models

The mathematical model is created using the formalism of robotics based on the geometric description of the modified Denavit–Hartenberg method. The method considers that the autonomous vehicle is a multi-articulated system consisting of n bodies wherein the chassis is the movable base and the wheels are the terminals. Each body is connected to its antecedent by a joint, which represents a translational or a rotational degree of freedom. A body can be virtual or real; the virtual bodies are introduced to describe joints with multiple degrees of freedom or to introduce intermediate fixed frames. The mixed Euler–Lagrange model is obtained from two recurrences of the algorithm of Newton–Euler in the following way. The forward recursive equations (from the mobile base to the effectors) compute the total forces ${}^{j}F_j$ and moments ${}^{j}Mo_j$ on each link j by calculating the angular and the linear speeds and accelerations of each body. The backward recurrence (from the effectors to the mobile base) computes the forces ${}^{j}f_j$ and the moments ${}^{j}mo_j$ exerted on each body by its antecedent, considering the external forces applied to the robot. The torque τ_j applied on the body C_j is calculated by projecting, according to the type of the joint j, the vector of force ${}^{j}f_j$ or moment ${}^{j}mo_j$ on the axle of movement [8].

$$\tau_j = \left(\sigma_j{}^{j}f_j + \overline{\sigma}_j{}^{j}mo_j\right)^{tj}a_j, \tag{31}$$

where ${}^{j}a_j = [0\;0\;1]^t$, $\sigma_j = 1$ if the joint j is translational and $\sigma_j = 0$ if the joint j is rotational. If there is no degree of freedom between two frames that are fixed with respect to each other, $\sigma_j = 2$.

The inverse dynamic model of a robot with a mobile base can be written as [8]

$$\tau = f\left(q, \dot{q}, \ddot{q}, f_e\right) = A(q)\ddot{q} + H(q, \dot{q}) + J(q)f_e \tag{32}$$

where $\boldsymbol{\tau}$ is the vector of the actuators torques or forces, q, $\dot{q}$ and $\ddot{q}$ are the vectors of positions, velocities, and accelerations of all the joints, including the variables of the chassis, and f_e is the vector of external forces. H is the vector of centrifugal, Coriolis, and gravity terms; J is the jacobien matrix; Jf_e is the vector of generalized efforts representing the projection of external forces and torques on the joint axis; and A is the inertial matrix of the system.

The direct dynamic model is then given by [8]:

$$\ddot{q} = [A(q)]^{-1}(\tau - H(q, \dot{q}) - J(q)f_e). \tag{33}$$

Once the expression of τ is determined by the Newton–Euler algorithm, the direct dynamic model for calculating the matrices A, H, and J is as follows [8]. The column c_a of the matrix A is computed by

$$A(:, c_a) = \frac{\partial \tau}{\partial \ddot{q}(c_a)},\ c_a \in [1,\ l] \tag{34}$$

where l represents the number of degrees of freedom in the system, which is the dimension of the vector q. The matrix J is computed similarly by

$$J(:, c_j) = \frac{\partial \tau}{\partial f_e(c_j)},\ c_j \in \left[1,\ r_f\right], \tag{35}$$

where r_f represents the dimension of the vector f_e. The matrix H is obtained using $H(q,\dot{q}) = \tau$ when $\ddot{q} = f_e = 0$. Then

$$H = f(q,\dot{q},0,0) \tag{36}$$

The adopted mixed Euler–Lagrange formalism allowed for directly elaborating the dynamic model of the autonomous mobile robot with minimum computation time. This formalism was used to develop a four-wheeled vehicle model. The model was validated using the Scanner-Studio simulator, and the results showed its validity for a large range of driving conditions [8].

3.7. AI and Robotics

3.7.1. On Mathematical Modeling in Robotics

The papers presents the geometric reasoning, topology, and logical aspects of multitasking robots. "Logical fiberings" can be put into practice to serve practical purposes in a robotic context; a simple robot scenario example is shown to illustrate this concept. It also discusses the use of polynomial inequalities in the modeling of geometric objects and, related to this, the importance of semi-algebraic geometry [50,51]. The paper also covers the inverse kinematics problem, its formulation, and the issues related to singular configurations. It also mentions symbolic mathematical computation and artificial intelligence in robotics and the potential for interdisciplinary collaboration. It also explains the role of logical reasoning and fibered systems in the development of many-valued logical systems. A descriptive account of the concept and how it might be applied to robotics is given [52]. Another discussion revolves around path planning in robotics. Much emphasis has been placed on problems revolving around collision detection and collision-free path construction due to their complex nature. The paper concludes by recognizing the limitations of the presented topics and recommends further reading to gain a deeper understanding of the subject matter.

Mathematical Models

The models of kinematic nature are a description of the motion without regard to the forces that cause it. They can usually be represented using transformation matrices. The general formula for forward kinematics using Denavit–Hartenberg (DH) parameters is

$$T = T_1 \cdot \ T_2 \cdot \ldots \cdot \ T_n, \tag{37}$$

where T_i is the transformation matrix for the i-th joint, defined as

$$T_i = \begin{bmatrix} \cos\Theta_i & -\sin\Theta_i \cos\alpha_i & \sin\Theta_i \sin\alpha_i & a_i \cos\Theta_i \\ \sin\Theta_i & \cos\Theta_i \cos\alpha_i & -\cos\Theta_i \sin\alpha_i & a_i \sin\Theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{38}$$

where Θ_i is the joint angle, d_i is the link offset, a_i is the link length, and α_i is the link twist. This matrix multiplication results in the transformation matrix T, which describes the position and orientation of the end effector relative to the base frame. With some given joint settings, this framework defines the procedure for computing the end-effector's pose.

The inverse kinematics method involves determining the settings of the joints that will provide the desired location for the robot's end effector. This can be represented as

$$f(\theta) = (x, y, z), \tag{39}$$

where f is a function that maps joint angles to end effector locations. This solves for the joint angles θ given a desired position (x, y, z).

Polynomial inequalities are usually applied to provide limits and restrictions in robotic motions. These inequalities can represent various physical scenarios such as collision avoidance, joint limits, and workspace boundaries.

Geometric reasoning is used to solve the problems of robot motion and configuration based on the utilization of geometrical characteristics and relationships. It typically employs equations of lines, planes, and curves in space, such as for a 3D plane.

$$ax + by + cz + d = 0, \tag{40}$$

Topological models help in understanding a robot's configuration space and the relations between different configurations. Topological spaces can express notions such as homotropy and connectedness, which are not usually expressed in straightforward tequations.

The findings of the recent simulation prove that the developed mathematical model effectively supports designing an adaptive control system with artificial intelligence features, especially for autonomous mobile robots with manipulators that are inherently complicated and networked systems. By cross-connecting control channels, the model makes it easy to design adaptive control algorithms and improves the system's efficiency.

3.7.2. Reinforcement Learning in Robot Control

Traditional control techniques like trajectory optimization, model-predictive control (MPC), and PID controllers are being replaced by reinforcement learning (RL) increasingly, especially for sophisticated robotic systems with uncertain dynamics or highly variable environments. Through trial-and-error both in simulation and actual environments, reinforcement learning teaches agents to maximize reward signals without having to learn robust and flexible policies. For example, by learning obstacle avoidance and real-time computing capabilities with low pre-modeling, the MAPPO-IK model introduced a deep RL method for six-DOF manipulators apart from standard inverse kinematic approaches. Similarly, joint-level controllers learned by model-free reinforcement learning can ease control pipelines, efficiently address redundancy and joint constraints, and achieve sub-centimeter precision through sim-to-real transfer.

Mathematical Models

Reinforcement Learning Policy (MAPPO) Equation. In the RL framework, an agent selects joint actions based on the current environmental state to learn how to control the manipulator. The policy NN utilizes the MAPPO algorithm to predict the optimal steps needed to navigate the manipulator toward a specified position while avoiding both obstacles and joint limits. The agent maximizes a cumulative reward function, which encourages precise hitting, significant movement minimization, and collision-free navigation. This approach enables the manipulator to learn sophisticated motion sequences and inverse kinematics through interaction with the environment rather than through purely analytical methods.

The policy that maps the current state to an action

$$a_t = \pi_\vartheta(s_t) \tag{41}$$

where s_t is the state at time t, π_ϑ is the neural network policy parameterized by θ, and a_t is the action (joint angle adjustments).

And the optimization objective (expected cumulative reward)

$$J(\theta) = E_{s_t,a_t}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\right], \tag{42}$$

represents the expected cumulative discounted reward in a reinforcement learning framework, parameterized by the policy π_ϑ. Here is a breakdown: $J(\theta)$ is the objective function is to be maximized, representing the expected total reward under the policy with parameters θ. $E_{s_t,a_t} \sim \pi_\vartheta$ is the expectation over state-action trajectories generated by following the policy π_ϑ. $\sum_{t=0}^{T} \gamma^t r(s_t, a_t)$ is the sum of discounted rewards collected from time step t = 0 to T, where $r(s_t, a_t)$ is the reward received at time t, when the agent acts in state s_t. $\Gamma \in [0, 1]$: The discount factor that prioritizes immediate rewards over distant future rewards.

These two equations capture the core mathematical models for the manipulator's kinematics and the reinforcement learning control strategy.

The RL equation, particularly the policy function $a_t = \pi_\vartheta(s_t)$, outlines how the agent (controller) moves from the current state s_t (containing information about the manipulator's pose and the surrounding environment) to action a_t (joint angle adjustments). The aims are, as described by the cumulative reward function, to predict an optimal outcome and retrieve maximum value by improving policy parameters θ. By means of interactions governed by these equations, an inverse kinematics problem is solved, which is learning to identify the joint angles for a specified end-effector position [53].

4. Discussion

Mathematical modeling in robotics is a rather extensive and dynamic field that develops robot design, control, and application. Robotic systems often include high-dimensional state spaces, resulting in complex and time-consuming computation of models and solutions. The limited computational resources on the robot can limit the complexity of models used in real time. Models must be robust to deal with unexpected situations and ensure the safety of robots and their environment. Many robotic systems have nonlinear behavior, which complicates modeling and requires sophisticated numerical methods. Real-time applications such as robotic surgery require models that can process data and make decisions with minimal delay. Artificial intelligence models require a large amount of training data that may not always be available in real scenarios and cannot be easily collected. Since robots are limited in power supply, the model must optimize energy efficiency.

This reviewed research spans a wide variety of robotic systems, which reflects the complexity and breadth of this area of study. From soft continuum robots to autonomous mobile robots and artificial intelligence in robotics, each domain presents unique challenges that demand tailored mathematical frameworks and computational techniques. For soft continuum robots, the modeling difficulty arises from their deformable, three-dimensional nature, which defies conventional rigid-body assumptions. The application of Cosserat rod theory combined with finite element methods has emerged as a robust solution that offers an accurate and computationally efficient model to predict deformations under external loads. This methodology not only achieves validation with minimal error but also opens avenues for enhancing the precision of the control and interaction of soft robots with unstructured environments.

Advanced mathematical models constitute the backbone of soft robotics development, which is continuously growing, especially for biomedical applications and exploratory tasks. Other challenging models involve flexible robots with elastic links and complex dynamic behaviors. Approaches based on Hamilton's variational principle and kineto-elasto-dynamics are found capable of capturing elastic deformations, transverse displacements, and flexible motions quite accurately. These partial differential equations and finite element-based models have reached far beyond theoretical development into the practical realm of applications involving Delta robots. The insights provided by these models are very important in the development of the precision and adaptability of flexible robots for their application in high-speed industrial operations and fine-manipulation tasks.

The dual quaternion algebra illustrates well the use of both geometric and algebraic frameworks in robotics modeling and control. This method is particularly effective for dealing with transformations, rotations, and translations in robotic systems, as it may capture the essence of the geometric phenomena involved within a much simpler algebraic structure. Its ease of use does not stop at kinematic modeling, but plunges into dynamic systems and control where computational efficiency and clarity are needed. The dual quaternion algebra emphasizes the increasing interest in applying mathematical abstraction to practical challenges in robotics. While simplified dynamic models are highly appealing for their computational efficiency, they come with significant trade-offs. The review exposes the dangers of oversimplification, such as the neglect of classical mechanics principles, which could result in unrealistic representations and excessive demands on control systems. This is a cautionary note for researchers and practitioners to balance simplification with accuracy, ensuring that computational models remain grounded in physical reality.

The emergence of fractional-order models represents a new frontier in robotics modeling. Using fractional-order variable difference equations, researchers have developed a flexible framework that accommodates the intricate dynamics of robotic systems. These models, validated through simulations and empirical data, have proven to be particularly effective at capturing the nuanced behavior of systems with complex dynamic characteristics, such as gantry cranes and missile launching vehicles. The fractional-order approach also demonstrates how mathematical techniques can be adapted to accommodate a wide range of robotic applications, from heavy machinery to fine robotic arms.

The modeling of autonomous mobile robots incorporates dynamics with adaptability to variable configurations as part of their multifaceted functionality. The Newton–Euler method, used to derive the equations of motion, gives a basis for adaptive control algorithms that enhance stability and efficiency. This is especially important in cases where AMRs interact with manipulators, as the interaction of the mobility of the robot with its manipulation requires a delicate balance of dynamic forces. It is through such modeling that continued development of autonomous systems for effective operation in dynamic and unstructured environments has been possible.

The combination of mathematical modeling and artificial intelligence (AI) brings a paradigm shift to robotics. Methods such as geometric and topological reasoning, along with the notion of fibered logical spaces, allow for the embedding of logical reasoning and decision-making capabilities in robots. This cross-disciplinary approach reinforces the development of symbolic computation combined with artificial intelligence to reach not only mechanical but also intelligent autonomous decision-making. The interplay of AI with mathematical modeling is going to push the next generation of robotic systems to bridge the gap between physical dynamics and cognitive capabilities. Such models are instrumental in developing control systems that optimize the coordination and stability of these systems, paving the way for advances in modular robotics and cooperative robotic tasks. From the works reviewed, it is evident that advanced mathematical modeling has played an indispensable role in taming the multifaceted challenges of robotics. Each method—classical mechanics, algebraic abstractions, and fractional calculus—finds its unique perspective and solution set, reflecting the diversity of robotic systems and their applications. It is in this marriage of mathematical precision with computational techniques that these models do not stop being theoretical constructs but serve to provide actionable insight into real-world robotics. The reviewed studies also emphasize the importance of validation and interdisciplinary collaboration in robotics modeling. Techniques such as empirical validation against experimental data, simulation studies, and cross-disciplinary integration with AI demonstrate how mathematical models evolve from theoretical frameworks to practical tools. This evolution is essential for bridging the gap between academic research and industrial applications, ensuring that robotics continues to innovate and address complex societal needs.

To summarize the diverse range of mathematical models applied in robotics, a comparative analysis has been conducted based on key performance parameters such as accuracy, computational complexity, adaptability, and application domains. Table 2 below provides a structured overview of each model's strengths and limitations, highlighting their suitability for different robotic tasks. This comparative insight is essential for guiding model selection in research and real-world implementations, especially as robotics continues to advance toward greater intelligence, flexibility, and autonomy.

Table 2. A model selection guide.

Model	Accuracy	Complexity	Flexibility	Applications	Notes
Soft Continuum Robots	Very high (error ~1.3 mm)	Medium	Very high	Surgical robotics, inspection, rehabilitation	Cosserat rod theory + FEM
Flexible Robots	High (error < 1%)	Medium-high	High	Laser cutting, high-speed pick-and-place	FEM and kineto-elasto-dynamics
Dual Quaternion Algebra	High	Low-medium	High	Mobile manipulators, robot kinematics/control	Compact, singularity-free
Simplified Dynamic Models	Low-medium	Low	Low	Fast control loops	Risk of violating physical laws
Fractional Order Models	Very high (ISE/IAE optimized)	Medium	High	Flexible actuators, adaptive robot arms	Superior transient modeling
AMR	High	Medium	Medium-high	Hazardous environments, exploration	Newton–Euler dynamics with adaptive control
AI-Based Models	High (data dependent)	High	Very high	Path planning, obstacle avoidance	Uses logic, symbolic computation, and geometry

5. Future Directions

Looking ahead, a few emerging trends and challenges can be identified: the increasing complexity of robotic systems, including soft materials, flexible components, and intelligent algorithms, requires an increasingly sophisticated mathematical model. For many applications, especially when real-time performance is required—for example, autonomous navigation or dynamic manipulation—computational efficiency remains a pressing concern. Therefore, the trade-off between model accuracy and computational tractability will remain a significant focus in the field for the foreseeable future.

Another important area is the representation of uncertainty and variability within robotic models. Real-world settings are essentially indeterminate; any hopes for the performance of robots in real-world conditions are contingent upon sound mathematical bases that can handle such an indeterminacy. Probabilistic modeling, stochastic differential equations, and reinforcement learning algorithms are some of the techniques that could be applied to this end and could provide a route to more robust and adaptive robotics.

The future directions of robotics mathematical models focus on improving adaptation, efficiency, and real-time capability. The main areas of development include the integration of foundation models, the development of model learning, and the improvement of kinematic modeling. Foundation models, based on large-scale datasets, could enhance the autonomy of robots by improving perception, decision-making, and control capabilities. They offer superior generalization and can solve problems that are not present in training data, although problems such as data scarcity and safety remain [54]. Mathematical models for soft robots are improving using FEM to accurately simulate deformations under different loads. These models aim to improve efficiency and accuracy in dynamic simulations [55].

The mathematical modeling of robotics stands as the very cornerstone of this discipline, providing theoretical and computational tools to understand, design, and control these increasingly complex systems. The methodologies that were discussed, from Cosserat rod theory to dual quaternion algebra and even to fractional-order calculus, represent examples of ingenuity and the ability of researchers to adaptively face the challenges presented by robotic systems. As robotics continues to intersect with the boundaries of AI, material sciences, and computational biology, the role of mathematical modeling will only grow, leading to innovation and expanding the horizons of the things robots can achieve.

The development of artificial intelligence and computer technology is rapidly developing mathematical modeling in robotics. AI improves simulations by providing a realistic scenario and high-precision predictions. The integration of artificial intelligence and mathematical modeling is an engine for the development of more robust and reliable robotic systems. These systems can cope with uncertainty and work consistently under diverse conditions. Modern models are dynamic and adaptable, allowing robots to adapt their behavior based on real-time data and changing conditions. Advanced mathematical models are used to simulate and virtually test robotic systems. These simulations enable researchers to test and improve the model before physical implementation, saving time and resources [56,57]. This synthesis of mathematics and robotics not only furthers technological capabilities but also extends insight into complex systems, paving the way to a future where robots will become commonplace in everyday life and industry.

Optimal models for inverse kinematics are being developed, especially for complex robots such as six-degree-of-freedom anthropomorphic robots, to balance accuracy and computational efficiency. Explainable artificial intelligence technology is also used to improve model interpretability [58]. Large-scale language models are integrated into robotic systems to improve real-time performance and reduce latency. These models help in path planning and task management without extensive training data [59]. Generative AI has become one of the most influential trends of recent years and is also reshaping industries, and robotics is no exception. This technology can simplify and enhance the way robots operate, making advanced robotics more accessible and versatile. One notable example in this field is Google's Robotics Transformer 2 (RT-2). RT-2 exemplifies the potential of by enabling robots to perceive, plan, and act with a level of adaptability previously unattainable [60].

Author Contributions: Conceptualization, L.B., G.D., V.B. and P.S.D.; methodology, L.B., G.D., V.B. and P.S.D.; validation, L.B., G.D., V.B. and P.S.D.; formal analysis, L.B., G.D., V.B. and P.S.D.; investigation, L.B., G.D., V.B. and P.S.D.; resources, L.B., V.B. and P.S.D.; data curation, L.B., G.D., V.B. and P.S.D.; writing—original draft preparation, L.B., G.D., V.B. and P.S.D.; writing—review and editing, L.B., G.D., V.B. and P.S.D.; visualization, G.D.; supervision, L.B.; project administration, G.D. and V.B.; funding acquisition, V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: Authors acknowledge PI of research group "Edge Artificial Intelligence and Robotics" Vidas Raudonis, for his inspiration and support.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
AMR	Autonomous mobile robot
BoundMPC	Bound model predictive control
DH	Denavit–Hartenberg
DMP	Dynamic movement primitives
DOF	Degree of freedom
DQ	Dual Quaternion
DSWP	Deterministic sea wave predictor
FEM	Finite element method
IAE	Integral of absolute error
ISE	Integral of squared error
MPC	Model predictive control
NMPC	Nonlinear model predictive control
PWM	Pulse width modulation
RL	Reinforcement learning
ROS	Robot operating system
UML	Unified modeling language
WOS	Web of Science Core Collection

References

1. Grazioso, S.; Di Gironimo, G.; Siciliano, B. A Geometrically Exact Model for Soft Continuum Robots: The Finite Element Deformation Space Formulation. *Soft Robot.* **2019**, *6*, 790–811. [CrossRef] [PubMed]
2. Yu, W. Mathematical modelling for a class of flexible robot. *Appl. Math. Model.* **1995**, *19*, 537–542. [CrossRef]
3. Kuo, Y. Mathematical modeling and analysis of the Delta robot with flexible links. *Comput. Math. Appl.* **2016**, *71*, 1973–1989. [CrossRef]
4. Adorno, B.V.; Marinho, M.M. DQ Robotics: A Library for Robot Modeling and Control. *IEEE Robot. Autom. Mag.* **2021**, *28*, 102–116. [CrossRef]
5. Tang, K.-M.W.; Tourassis, V.D. Mathematical deficiencies of numerically simplified dynamic robot models. *IEEE Trans. Autom. Control* **1989**, *34*, 1109–1111. [CrossRef]
6. Bąkała, M.; Duch, P.; Ostalczyk, P. New Approach of the Variable Fractional Order Model of a Robot Arm. *Appl. Sci.* **2023**, *13*, 3304. [CrossRef]
7. Singh, A.P.; Deb, D.; Agrawal, H.; Balas, V.E. Fractional Modeling of Robotic Systems. In *Fractional Modeling and Controller Design of Robotic Manipulators*; Intelligent Systems Reference Library; Springer: Cham, Switzerland, 2020; Volume 194, pp. 19–43. [CrossRef]
8. Chebly, A.; Talj, R.; Charara, A. Coupled Longitudinal and Lateral Control for Autonomous Vehicle Dynamics Modeled Using a Robotics Formalism. *IFAC-PapersOnLine* **2017**, *50*, 12526–12532. [CrossRef]
9. Pfalzgraf, J. On Mathematical Modeling in Robotics. In *Artificial Intelligence and Symbolic Mathematical Computing*; Calmet, J., Campbell, J.A., Eds.; AISMC 1992; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1992; Volume 737, pp. 116–132. [CrossRef]
10. Polyanina, A. Construction of mathematical model for the controlled dynamics of connected bodies system in robotics. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1030*, 012055. [CrossRef]
11. Oelerich, T.; Beck, F.; Hartl-Nesic, C.; Kugi, A. BoundMPC: Cartesian path following with error bounds based on model predictive control in the joint space. *Int. J. Robot. Res.* **2025**, *44*, 02783649241309354. [CrossRef]
12. Saveriano, M.; Abu-Dakka, F.J.; Kramberger, A.; Peternel, L. Dynamic movement primitives in robotics: A tutorial survey. *Int. J. Robot. Res.* **2023**, *42*, 1133–1184. [CrossRef]

13. Walker, K.L.; Jordan, L.-B.; Giorgio-Serchi, F. Nonlinear model predictive dynamic positioning of a remotely operated vehicle with wave disturbance preview. *Int. J. Robot. Res.* **2024**, *44*, 932–951. [CrossRef]
14. Du, W.; Fnadi, M.; Benamar, F. Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics. *Mech. Mach. Theory* **2020**, *153*, 103984. [CrossRef]
15. Xu, K.; Wang, S.; Shi, L.; Li, J.; Yue, B. Horizon-stability control for wheel-legged robot driving over unknow, rough terrain. *Mech. Mach. Theory* **2025**, *205*, 105887. [CrossRef]
16. Ichnowski, J.; Avigal, Y.; Satish, V.; Goldberg, K. Deep learning can accelerate grasp-optimized motion planning. *Sci. Robot.* **2020**, *5*, eabd7710. [CrossRef] [PubMed]
17. Chen, B.; Yang, H.; Song, B.; Meng, D.; Yan, X.; Li, Y.; Wang, Y.; Hu, P.; Ou, T.H.; Barnell, M.; et al. A memristor-based hybrid analog-digital computing platform for mobile robotics. *Sci. Robot.* **2020**, *5*, eabb6938. [CrossRef] [PubMed]
18. My, C.A.; Bien, D.X.; Hiep, T.X.; Dinh, N.C.; Duc, V.M.; Nguyen, N.T.; Le, C.H.; Alandoli, E.A. A Review of Mathematical Methods for Flexible Robot Dynamics Modeling and Simulation. In *Machine Learning and Mechanics Based Soft Computing Applications*; Nguyen, T.D.L., Lu, J., Eds.; Studies in Computational Intelligence; Springer: Singapore, 2023; Volume 1068. [CrossRef]
19. Qin, L.; Peng, H.; Huang, X.; Liu, M.; Huang, W. Modeling and Simulation of Dynamics in Soft Robotics: A Review of Numerical Approaches. *Curr. Robot. Rep.* **2024**, *5*, 1–13. [CrossRef]
20. Ortigosa-Martínez, R.; Martínez-Frutos, J.; Mora-Corral, C.; Pedregal, P.; Periago, F. Mathematical modeling, analysis and control in soft robotics: A survey. *SeMA J.* **2024**, *81*, 147–164. [CrossRef]
21. Trivedi, D.; Rahn, C.D.; Kier, W.M.; Walker, I.D. Soft robotics: Biological inspiration, state of the art, and future research. *Appl. Bionics Biomech.* **2008**, *5*, 99–117. [CrossRef]
22. Majidi, C. Soft Robotics: A Perspective-Current Trends and Prospects for the Future. *Soft Robot.* **2014**, *1*, 5–11. [CrossRef]
23. Burgner-Kahrs, J.; Rucker, D.C.; Choset, H. Continuum Robots for Medical Applications: A Survey. *IEEE Trans. Robot.* **2015**, *31*, 1261–1280. [CrossRef]
24. Park, Y.L.; Chen, B.; Perez-Arancibia, N.; Young, D.; Stirling, L.; Wood, R.J.; Goldfield, E.C.; Nagpal, R. Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation. *Bioinspir. Biomim.* **2014**, *9*, 016007. [CrossRef] [PubMed]
25. Polygerinos, P.; Wang, Z.; Galloway, K.C.; Wood, R.J.; Walsh, C.J. Soft robotic glove for combined assistance and at-home rehabilitation. *Robot. Auton. Syst.* **2015**, *73*, 135–143. [CrossRef]
26. Armanini, C.; Boyer, F.; Mathew, A.T.; Duriez, C.; Renda, F. Soft Robots Modeling: A Structured Overview. *IEEE Trans. Robot.* **2023**, *39*, 1728–1748. [CrossRef]
27. Ménager, E.; Navez, T.; Chaillou, P.; Goury, O.; Kruszewski, A.; Duriez, C. Modeling, Embedded Control and Design of Soft Robots using a Learned Condensed FEM Model. *arXiv* **2025**, arXiv:2503.15009. [CrossRef]
28. Boër, C.R.; Molinari-Tosatti, L.; Smith, K.S. *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements*; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
29. Laribi, M.A.; Romdhane, L.; Zeghloul, S. Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace. *Mech. Mach. Theory* **2007**, *42*, 859–870. [CrossRef]
30. Zhang, L.; Song, Y. Optimal design of the Delta robot based on dynamics. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 336–341. [CrossRef]
31. Piras, G.; Cleghorn, W.L.; Mills, J.K. Dynamic finite-element analysis of a planar high speed, high-precision parallel manipulator with flexible links. *Mech. Mach. Theory* **2005**, *40*, 849–862. [CrossRef]
32. Chen, K.; Zhao, H.; Zhang, J.; Feng, M.; Wang, Y.; Wang, J.; Ding, H. Modeling and control of a rigid-flexible coupling robot for narrow space manipulations. *Sci. China Technol. Sci.* **2025**, *68*, 1220302. [CrossRef]
33. Adorno, B.V. Two-Arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration [Contribution à la Manipulation à Deux Bras: Des Manipulateurs à la Collaboration Homme-Robot]. Ph.D. Dissertation, University of Montpellier, Montpellier, France, 2011. Available online: https://theses.hal.science/tel-00641678/ (accessed on 27 November 2024).
34. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Modeling and Control*, 2nd ed.; Wiley: New York, NY, USA, 2020. Available online: https://www.perlego.com/book/2774514/robot-modeling-and-control-pdf (accessed on 15 October 2024).
35. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*, 1st ed.; Springer: London, UK, 2008.
36. Leclercq, G.; Lefèvre, P.; Blohm, G. 3D kinematics using dual quaternions: Theory and applications in neuroscience. *Front. Behav. Neurosci.* **2013**, *7*, 7. [CrossRef] [PubMed]
37. Liendo, F.; Hernández, C.; Galez, C. An Improved Dual Quaternion Dynamic Movement Primitives-Based Algorithm for Robot-Agnostic Learning and Execution of Throwing Tasks. *J. Mech. Robot.* **2025**, *17*, 091012. [CrossRef]
38. Govindarajan, V.; Bhattacharya, S.; Kumar, V. Human-robot collaborative topological exploration for search and rescue applications. In *Distributed Autonomous Robotic Systems*; Springer: Tokyo, Japan, 2016; pp. 17–32. [CrossRef]
39. Brito, T.; Queiroz, J.; Piardi, L.; Fernandes, L.A.; Lima, J.; Leitão, P. A machine learning approach for collaborative robot smart manufacturing inspection for quality control systems. *Procedia Manuf.* **2020**, *51*, 11–18. [CrossRef]

40. Kilbas, A.A.; Srivastava, H.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier Science Inc.: New York, NY, USA, 2006; Volume 204. Available online: https://www.sciencedirect.com/bookseries/north-holland-mathematics-studies/vol/204 (accessed on 6 May 2025).
41. Sabatier, J.; Melchior, P.; Tenreiro Machado, J.A.; Vinagre, B.M. Fractional Order Systems. Applications in modelling, identification, and control. *J. Eur. Des. Syst. Autom.* **2008**, *42*, 625.
42. Xiong, X.; Xiong, M.; Chen, L.; Luo, M.; Yu, Y. Existence uniqueness and stability analysis of solutions for the fractional fuzzy cellular neural networks with impulse and reaction diffusion. *Math. Comput. Model. Dyn. Syst.* **2025**, *31*, 2515875. [CrossRef]
43. Alagoz, B.B.; Tepljakov, A.; Ates, A.; Petlenkov, E.; Yeroglu, C. Time-domain identification of One Noninteger Order Plus Time Delay models from step response measurements. *Int. J. Model. Simul. Sci. Comput.* **2019**, *10*, 1941011:1–1941011:22. [CrossRef]
44. Lopota, A.V.; Spassky, B.A. Mobile ground-based robot systems for professional use. *Robot. Tech. Cybern.* **2020**, *8*, 5–17. [CrossRef]
45. Tsarichenko, S.; Antokhin, E.; Chernova, P.; Dementey, V. The state and problems of standardization and unification of military ground robot systems. *Robot. Tech. Cybern.* **2020**, *8*, 18–23. [CrossRef]
46. Liu, X.-F.; Li, H.-Q.; Chen, Y.-J.; Cai, G.-P. Dynamics and control of space robot considering joint friction. *Acta Astronaut.* **2015**, *111*, 1–18. [CrossRef]
47. Liu, G.; Geng, X.; Liu, L.; Wang, Y. Haptic based teleoperation with master slave motion mapping and haptic rendering for space exploration. *Chin. J. Aeronaut.* **2019**, *32*, 723–736. [CrossRef]
48. Angeles, J. *Rational Kinematics*; Springer: New York, NY, USA; Berlin, Germany, 1988.
49. Baker, D.R. Some topological problems in robotics. *Math. Intell.* **1990**, *12*, 66–76. [CrossRef]
50. Bochnak, J.; Coste, M.; Roy, M.-F. *Real Algebraic Geometry*; Springer: Berlin/Heidelberg, Germany, 1987.
51. Boissonat, J.-D.; Laumond, J.-E. *Geometry and Robotics*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1989; Volume 391.
52. Lapshin, V.V. *Mechanics and Motion Control Walking Machines*; MSTU named after N.E. Bauman: Moscow, Russia, 2012; p. 200.
53. Zhao, C.; Wei, Y.; Xiao, J.; Sun, Y.; Zhang, D.; Guo, Q.; Yang, J. Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning. *Sci. Rep.* **2024**, *14*, 12467. [CrossRef] [PubMed]
54. Firoozi, R.; Tucker, J.; Tian, S.; Majumdar, A.; Sun, J.; Liu, W.; Zhu, Y.; Song, S.; Kapoor, A.; Hausman, K.; et al. Foundation models in robotics: Applications, challenges, and the future. *Int. J. Robot. Res.* **2024**, *44*, 701–739. [CrossRef]
55. Shufeng, T.; Jingfang, J.; Wei, Y.; Shijie, G.; Hong, C.; Xuewei, Z. Shape modeling and experimental validation of continuum robots. *Appl. Math. Model.* **2024**, *135*, 306–318. [CrossRef]
56. Diveev, A.; Sofronova, E.; Konyrbaev, N.; Abdullayev, O. Advanced Model with a Trajectory Tracking Stabilisation System and Feasible Solution of the Optimal Control Problem. *Mathematics* **2024**, *12*, 3193. [CrossRef]
57. Ohue, M.; Yasuo, N.; Takata, M. Innovations in mathematical modeling, AI, and optimization techniques. *J. Supercomput.* **2025**, *81*, 340. [CrossRef]
58. Rakesh, C.J.; Jaynendra, K.R.; Radim, B.; Malay, K.D. Optimized inverse kinematics modeling and joint angle prediction for six-degree-of-freedom anthropomorphic robots with Explainable AI. *ISA Trans.* **2025**, *157*, 340–356. [CrossRef]
59. Gao, Y.; Luo, W.; Wang, X.; Zhang, S.; Goh, P. LAMARS: Large Language Model-Based Anticipation Mechanism Acceleration in Real-Time Robotic Systems. *IEEE Access* **2025**, *13*, 3864–3880. [CrossRef]
60. Autonomous Systems. Design, Develop, and Test Autonomous Systems with MATLAB. Available online: https://blogs.mathworks.com/autonomous-systems/2025/02/27/let-robots-move-easy-the-power-of-generative-ai-for-robotics/ (accessed on 2 April 2025).