Improving Contact-Rich Robotic Simulation with Generalized Rigid-Body Dynamics Algorithms

Matthew Chignoli[1], Nicholas Adrian[2], Sangbae Kim[1], and Patrick M. Wensing[2]

***Abstract*—We propose a novel approach for generalizing rigid-body dynamics algorithms to handle complex sub-mechanisms that are popular in robotic systems. The approach is based on the unification of local constraint embedding and spatial vector algebra, which allows for the development of compact and computationally efficient algorithms for simulating contact-rich robots. We derive the approach intuitively using the Gauss Principle of Least Constraint. Overall, our approach provides a valuable foundation for simulating and controlling complex robotic systems with traditionally difficult-to-simulate designs such as geared motors, differential drives, and four-bar mechanisms.**

I. Introduction

In recent years, the field of humanoid robot design has been dominated by a trend of complex limb designs that aim to expand the robot's range of motion, minimize the inertia of the limbs, and smoothly handle impacts with the environment. Designs with sub-mechanisms such as coupled belt drives [1], differential drives [2], [3], and four bar linkages [4] have emerged as popular options to meet these design criteria. Unfortunately, the dynamics algorithms used to simulate these robotic systems have not kept pace with these recent hardware advancements. Fast, accurate algorithms are crucially important for both model-based controllers as well as "model-free" reinforcement learning, which in practice greatly depends on the quality of the underlying simulator. While the Articulated-Body Algorithm (ABA) [5] has traditionally served the role as that fast, accurate algorithm for simulating robotic systems, it is not suitable for systems with sub-mechanisms like those mentioned above.

In the late 2000s, the concept of Local Constraint Embedding (LCE) was introduced [6], which enabled the development of a forward dynamics algorithm that retained many of the favorable properties of the ABA, while being general to systems with complex sub-mechanisms. However, this work has been largely neglected by the robotics community. Two contributing factors are likely (i) it did not use the ubiqituous Spatial Vector Algebra (SVA) conventions popularized by Featherstone [7] and (ii) its matrix factorizaton-based derivation of the ABA-like algorithms is unintuitive to most roboticists. To that end, we propose an alternative development of constraint embedding-based dynamics algorithms that features:

- A unification of the concepts of LCE and SVA,
- Compact, computationally efficient algorithms needed for simulating contact-rich robotic systems,
- An intuitive derivation based on the "Gauss Principle of Least Constraint" [8].

II. Related Work

The ABA has formed the bedrock of modern dynamic simulators for robotic systems for many years [9]–[11]. The concept dates back to the 1970's [12], but it was not popularized until a decade later when Featherstone developed the algorithm using SVA [5]. In subsequent years, efforts were made to generalize the algorithm to systems with closed-loop constraints. Recognizing the importance of including the complete effects of gear ratios and the gyroscopic effects of the spinning motors, researchers working with high gear ratio manipulators developed minimally modified versions of the algorithm to handle this particular class of sub-mechanism [13], [14]. This work has regained relevance in recent years as proprioceptive actuation has emerged as a popular paradigm for robots designed to make frequent contact with their environments [15]. Featherstone later generalized the ABA for a larger class of sub-mechanisms by introducing the Divide-and-Conquer concept [16], [17]. In the meantime, efficient non-recursive algorithms were developed that could handle arbitrary closure constraints. These algorithms use techniques such as sparse matrix factorization [18] or Lagrange multiplier elimination [19] to efficiently solve the large systems of linear equations arising from the equations of motion and closure constraints.

It wasn't until the introduction of LCE that an algorithm was developed that both maintained the efficient recursive structure while being able to handle arbitrary closure constraints [6], [20]. However, until recently, the concept has been largely ignored by roboticsts. The recent efforts have focused on applications related to parallel kinematic actuators [21], with an emphasis on how hybrid numerical and analytical approaches to dealing with the constraints can lead to improved computational efficiency and accuracy [22], [23]. We instead turn our attention to improving dynamics simulators for more general legged robots and manipulators. Specifically, we rederive an intuitive, generalized version of the ABA, and we discuss the straightforward extension of our work to another state-of-the-art algorithm for computing the Operational-Space Inertia Matrix (OSIM) [24]. Thus, we lay the foundation for a new dynamics engine that combines a generalized ABA for dynamics simulation of unconstrained mechanisms, with a generalized OSIM algorithm for contact simulation. Specifically, the OSIM algorithm will form the

[1] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA.

[2] Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, USA.

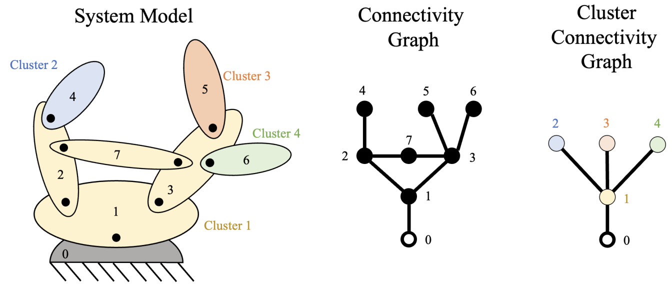

Fig. 1. Comparison of the RB-CG and the C-CG used to represent a system model with closed-loop constraints.

backbone of an efficient time-stepping solver for the contact complementarity problem [25].

III. Modeling

A popular approach to modeling rigid-body systems is to describe the system in terms of its component parts via a *system model* [18]. A system model consists solely of bodies and joints, and can be expressed in the form of a Rigid-Body Connectivity Graph (RB-CG). RB-CGs have the following properties [18]: nodes represent bodies, arcs represent joints, exactly one node represents a fixed base, and the graph is connected and undirected. Conventional recursive algorithms for rigid-body dynamics are only compatible with kinematic trees, i.e., systems whose RB-CGs have the property that there exists exactly one path between any two nodes in the graph. In other words, the recursive algorithms are only compatible for systems with no closure constraints.

LCE can be used to convert arbitrary RB-CGs to kinematic trees where each node of the tree is a *collection of rigid bodies* and each arc is a *collection of joints* [6], as is shown in Fig. 1. Nodes containing more than one rigid body have a closure constraint associated with them that describes how the motions of each contained body are coupled. We assume that all constraints are expressed in explicit form [26]

$$\mathsf{q} = \gamma\left(\mathsf{y}\right), \quad \dot{\mathsf{q}} = \mathsf{G}\dot{\mathsf{y}}, \quad \ddot{\mathsf{q}} = \mathsf{G}\ddot{\mathsf{y}} + \mathsf{g} \tag{1}$$

where q are the spanning tree coordinates and y are the independent coordinates.

In this paper, we refer to these collections of bodies as “body clusters” or simply “clusters”, and we refer to the collections of joints that connect pairs of body clusters as “cluster joints.” Furthermore, we refer to a connectivity graph comprised of clusters and cluster joints as a Cluster Connectivity Graph (C-CG). All C-CGs are kinematic trees. For the most part, C-CGs can be treated the same as RB-CGs. They both follow the widely popular “regular numbering” convention, wherein each node must have a higher number than its parents and arc i connects between node i and its parent [18]. However, in the case of C-CGs, both the individual rigid bodies as well as the clusters are separately numbered. For clarity, we will use indices i and j to exclusively refer to rigid bodies, and we will use indices k and l to exclusively refer to clusters. We use $\mathcal{C}_k$ to refer to the kth cluster in the C-CG.

IV. Spatial Vector Algebra for Clusters

The difference between body clusters and conventional rigid bodies is the dimension of the spatial vectors that describe their motion and force interactions. We define M^n and F^n as vector subspaces for spatial motion vectors and spatial force vectors, respectively. Spatial motion and force vectors for any individual rigid body are elements of the subspaces M^6 and F^6, while the spatial motion and force vectors for a cluster are elements of the subspaces M^{6n_b} and F^{6n_b}, where n_b is the number of bodies contained by the cluster. Thus, we can consider the SVA quantities used to describe the motion and force interactions of rigid-body clusters as the concatenation of the SVA quantities of the cluster’s constituent bodies.

To that end, we define the following cluster concatenation operators: $\mathrm{stack}\left(\{\mathbf{z}_i\}_{i\in\mathcal{C}_k}\right)$ creates a single vector containing the vector quantities $\mathbf{z}$ corresponding to every rigid body in cluster k, while $\mathrm{block}\left(\{\mathbf{Z}_i\}_{i\in\mathcal{C}_k}\right)$ creates a block diagonal matrix of the matrix quantities $\mathbf{Z}$ corresponding to every rigid body in cluster k. Table I distinguishes the notation we use to represent SVA quantities for individual bodies versus clusters.

TABLE I
SVA Notation for Rigid Bodies vs. Clusters

SVA Quantity	Rigid Body	Cluster
Joint Coords.	$\mathbf{q}$, $\mathbf{y}$	q, y
Spatial Vectors	$\mathbf{v}$, $\mathbf{a}$, $\mathbf{f}$	v, a, f
Spatial Matrices	$\mathbf{X}$, $\mathbf{I}$	X, I

An important nuance of SVA for clusters is the motion subspace matrix S. This matrix corresponds to the constraint imposed by the cluster joint and indicates the allowable relative motion between two clusters. Thus, the spatial velocities between any cluster l and its parent cluster k satisfy

$$\mathsf{v}_l - \mathsf{v}_k = \mathsf{S}_l\dot{\mathsf{y}}_l. \tag{2}$$

The nuance comes from the fact that the motion of a body in cluster l might depend on the motion of another body in the same cluster, as is the case for bodies 2, 3, and 7 in Fig. 1. As such, the motion subspace matrix for a cluster l is

$$\mathsf{S}_l = \mathsf{X}_{I(l)}\mathrm{block}\left(\{\mathbf{S}_i\}_{i\in\mathcal{C}_l}\right)\mathsf{G}_l, \tag{3}$$

where $\mathsf{X}_{I(l)}$ is the intra-cluster transformation matrix that captures the relationship between bodies in the same cluster.

V. Forward Dynamics

The original ABA for computing forward dynamics consists of three passes: a forward pass for first order kinematics, a backward pass for updating the articulated-body inertias of each body in the tree, and a forward pass for propagating the accelerations outward from the base. In this section, we use the extension of SVA from Sec. IV and the Gauss Principle of Least Constraint (GPLC) to derive the Cluster-Based Articulated-Body Algorithm (C-ABA) for systems with closed-loop constraints. We focus on ABA since subsequent algorithms, such as the OSIM algorithm, rely on the ABA.

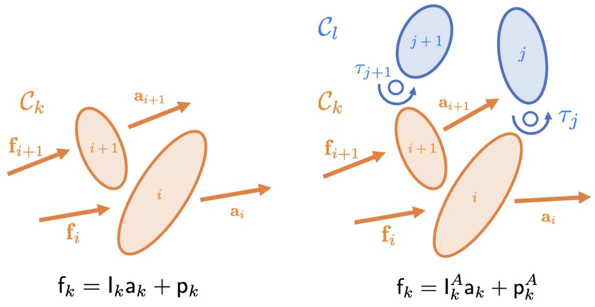

Fig. 2. Comparison of a cluster (left) to an articulated cluster (right).

A. Forward Kinematics

Analagous to the original ABA, we show that the spatial velocities for every body in a cluster can be expressed in terms of the spatial velocities of bodies *in the parent cluster* and the position and velocity of the cluster joint. This relationship can be derived by applying the concatenation operator to the velocities of all of the bodies in an arbitrary cluster k

$$\begin{aligned}\mathsf{v}_k &= \text{stack}\left(\left\{{}^i\mathbf{X}_{\Lambda(i)}\mathbf{v}_{\Lambda(i)}\right\}_{i\in\mathcal{C}_k}\right) + \\ &\qquad \text{stack}\left(\left\{\sum_{j\in\xi(i)} {}^i\mathbf{X}_j\mathbf{S}_j\dot{\mathbf{q}}_j\right\}_{i\in\mathcal{C}_k}\right) \\ &= {}^k\mathsf{X}_{\Lambda(k)}\mathsf{v}_{\Lambda(k)} + \mathbf{X}_{I(k)}\text{block}\left(\{\mathbf{S}\}_{i\in\mathcal{C}_l}\right)_k\dot{\mathsf{q}}_k \\ &= {}^k\mathsf{X}_{\Lambda(k)}\mathsf{v}_{\Lambda(k)} + \mathsf{S}_k\dot{\mathsf{y}}_k,\end{aligned} \tag{4}$$

where $\Lambda(i)$ gives the nearest supporting body of i in the parent cluster and $\xi(i)$ gives all of the supporting bodies of body i that are in its same cluster. Successive iterations of (4) with k ranging from 1 to N_c, where N_c is the number of clusters in the tree, constitute the first forward pass of the C-ABA.

B. Articulated Cluster Inertia

A body's articulated-body inertia describes the inertia that rigid body *appears to have* when it is part of a rigid-body system [5]. The concept extends straightforwardly to clusters of bodies, wherein attaching a cluster l to cluster k via a cluster joint yields the articulated-cluster equation of motion

$$\mathsf{f}_k = \mathsf{I}^A_k\mathsf{a}_k + \mathsf{p}^A_k, \tag{5}$$

where I^A_k and p^A_k are the articulated-*cluster* inertia and bias force of the multi-cluster system, respectively. Articulated-cluster inertias have the same properties as articulated-body inertias: they are symmetric and positive-definite, they map from $\mathbf{M}^{6n_b}$ to $\mathbf{F}^{6n_b}$, they can be transformed between coordinate frames via ${}^l\mathsf{X}^*_k\mathsf{I}_k{}^k\mathsf{X}_l$.

C. Calculating Articulated-Cluster Inertia via the Gauss Principle of Least Constraint

The backward pass of the ABA depends on a recursive relationship between the articulated-body quantities of child and parent bodies, and the second forward pass depends on a recursive relationship between the joint and spatial accelerations of the child and parent bodies. There are multiple ways to derive these recursive relationships, such as the assembly method [5] and the matrix-factorization method [27]. However, arguably the most physically intuitive derivation, especially for roboticists, is based on the GPLC.

Consider an arbitrary cluster of bodies $\mathcal{C}_k$ that is being attached to cluster $\mathcal{C}_l$ via a cluster joint. Applying the GPLC to the interaction between the bodies in these clusters involves evaluating their respective equations of motion in the following context: (i) neglecting the constraint forces at all of the joints, (ii) applying joint torques $\boldsymbol{\tau}_j$ at the actuated joints in the child cluster, and (iii) applying a test spatial force $\mathbf{f}_i$ to each of the bodies in the parent cluster.

The free-body diagram in Fig. 2 illustrates that the parent cluster follows the spatial equation of motion

$$\begin{aligned}\mathsf{I}^A_k\mathsf{a}^{\text{uc}}_k &= \text{stack}\left(\{\mathbf{f}_i\}_{i\in\mathcal{C}_k}\right) - \\ &\qquad \text{stack}\left(\left\{\sum_{j\in\mathcal{C}_l\cap\mu(i)} \boldsymbol{\Psi}^a_j\boldsymbol{\tau}_j\right\}_{i\in\mathcal{C}_k}\right) - \mathsf{p}^A_k \\ &= \mathsf{f}_k - {}^k\mathsf{X}^*_l\Psi^a_l\tau_l - \mathsf{p}^A_k\end{aligned} \tag{6}$$

where $\mu(i)$ gives the children of body i and $\boldsymbol{\Psi}^a_j$ is the active force subspace matrix of the jth joint. Similarly, the child cluster follows

$$\begin{aligned}\mathsf{I}^A_l\mathsf{a}^{\text{uc}}_l &= \text{stack}\left(\{\boldsymbol{\Psi}^a_i\boldsymbol{\tau}_i\}_{i\in\mathcal{C}_l}\right) - \mathsf{p}^A_l \\ &= \Psi^a_l\tau_l - \mathsf{p}_l.\end{aligned} \tag{7}$$

The final step in preparing to apply the GPLC is to relate the accelerations of the child cluster to the acceleration of the parent cluster and the generalized coordinates of the child cluster's joints. This relationship can be derived via a similar process as (4) and is given by

$$\mathsf{a}_l = {}^l\mathsf{X}_k\mathsf{a}_k + \mathsf{S}_l\ddot{\mathsf{q}}_l + \dot{\mathsf{S}}_l\dot{\mathsf{q}}_l. \tag{8}$$

The GPLC states that the motions of the bodies in these two clusters satisfies

$$\begin{aligned}\min_{\mathsf{a}_k,\mathsf{a}_l,\ddot{\mathsf{q}}_l} \; &\frac{1}{2}(\mathsf{a}^{\text{uc}}_k - \mathsf{a}_k)^\top\mathsf{I}^A_k(\mathsf{a}^{\text{uc}}_k - \mathsf{a}_k) \\ &+ \frac{1}{2}(\mathsf{a}^{\text{uc}}_l - \mathsf{a}_l)^\top\mathsf{I}^A_l(\mathsf{a}^{\text{uc}}_l - \mathsf{a}_l) \\ \text{s.t} \quad & \mathsf{a}_l = {}^l\mathsf{X}_k\mathsf{a}_k + \mathsf{S}_l\ddot{\mathsf{q}}_l + \mathsf{c}_l.\end{aligned} \tag{9}$$

Using the substitutions from (6)-(8) and the first order optimality conditions for $\ddot{\mathsf{q}}_l$, we can derive the following recursive relationship with a_k,

$$\ddot{\mathsf{q}}_l = \mathsf{D}^{-1}_l\left(\mathsf{u}_l - \mathsf{U}^\top_l\left({}^l\mathsf{X}_k\mathsf{a}_k + \mathsf{c}_l\right)\right), \tag{10}$$

where

$$\mathsf{U}_k = \mathsf{I}^A_k\mathsf{S}_k, \quad \mathsf{D}_k = \mathsf{S}^\top_k\mathsf{U}_k, \quad \mathsf{u}_k = \tau_k - \mathsf{S}^\top_k\mathsf{p}^A_k. \tag{11}$$

This relationship is used in the second forward pass of the C-ABA to propagate the accelerations of the clusters from the base to the tips of the C-CG.

Lastly, we derive the recursive relationship for the backward pass by considering the spatial equation of motion for the two-cluster system created by joining clusters k and l,

$$\mathsf{f}^A_k = \mathsf{I}^A_k\mathsf{a}_k + \mathsf{p}^A_k + {}^k\mathsf{X}^*_l\left(\mathsf{I}^A_l\mathsf{a}_l + \mathsf{p}^A_l\right). \tag{12}$$

Substituting a_l and $\ddot{\mathsf{q}}_l$ into (12) using (8) and (10) yields

$$\begin{aligned}\mathsf{f}_k^A = &\left(\mathsf{I}_k^A + {}^k\mathsf{X}_l^* \left(\mathsf{I}_l^A - \mathsf{U}_l\mathsf{D}_l^{-1}\mathsf{U}_l\right){}^l\mathsf{X}_k\right)\mathsf{a}_k \\ &+ \mathsf{p}_k^A + {}^k\mathsf{X}_l^*\left(\mathsf{p}_l^A + \mathsf{I}_l^A\mathsf{c}_l + \mathsf{U}_l\mathsf{D}_l^{-1}\mathsf{u}_l\right).\end{aligned} \tag{13}$$

Noticing that (13) has the same form as (5), we have arrived at the recursive relationship for the articulated-cluster inertia and bias force, which are analogous to the original ABA relationships

$$\mathsf{I}_k^A = \mathsf{I}_k + \sum_{l\in\mu(\mathcal{C}_k)} \mathsf{I}_l^a, \quad \mathsf{p}_k^A = \mathsf{p}_k + \sum_{l\in\mu(\mathcal{C}_k)} \mathsf{p}_l^a \tag{14}$$

where

$$\begin{aligned}\mathsf{I}_l^a &= \mathsf{I}_l^A - \mathsf{U}_l\mathsf{D}_l^{-1}\mathsf{U}_l^\top, \\ \mathsf{p}_l^a &= \mathsf{p}_l^A + \mathsf{I}_l^a\mathsf{c}_l + \mathsf{U}_l\mathsf{D}_l^{-1}\left(\tau_l - \mathsf{S}_l^\top\mathsf{p}_l^A\right).\end{aligned} \tag{15}$$

These relationships constitute underpin the backward pass of the C-ABA.

VI. Extensions

For the sake of brevity, we cannot include detailed a derivation of how the proposed approach enables the Cluster-Based Extended-Force-Propagator Algorithm (C-EFPA), which is an extension of the reduced-order recursive algorithm for computing OSIM [24]. Instead, we outline the derivation qualitatively. The derivation is closely related to the derivation of the C-ABA, with the distinction that the C-EFPA isolates the effects of end-effector forces that are needed in operational-space dynamics. The main feature of the EFPA is its use of extended force propagators to provide transformations of spatial forces from end-effector k to rigid body i as if the joints of the robot are free to mode. This is contrary to a standard spatial force transforms, which assume all joints are locked. Deriving the C-EFPA relies on showing that analogous force propagators exist for *clusters of bodies*. Thus, the key step in the deriving the C-EFPA is demonstrating that there exists a recursive relationship that can be used to efficiently compute the extended force propagator matrices between clusters of bodies, rather than between individual bodies.

The development of such an algorithm has significant ramifications for contact-rich dynamic simulation. Numerous contemporary dynamics engines for robotic systems utilize rigid contact models. These dynamics engines employ diverse solution methods to address the inherent complementarity problem for these models [28], [29], but they all share the common requirement of needing to compute the OSIM when solving for contact impulses. Therefore, achieving fast and accurate computations of the OSIM for robots with complex sub-mechanisms holds the potential to greatly enhance the overall efficiency of these contact dynamics solvers for a whole new class of robots.

VII. Results

We present preliminary results demonstrating the computational efficiency of the C-ABA compared to two state-of-the-art algorithms for computing forward dynamics. The algorithms are applied to two robots. Robot A is a serial chain of links that are each actuated with a geared motor. Each link and its corresponding rotor form a cluster that is constrained by

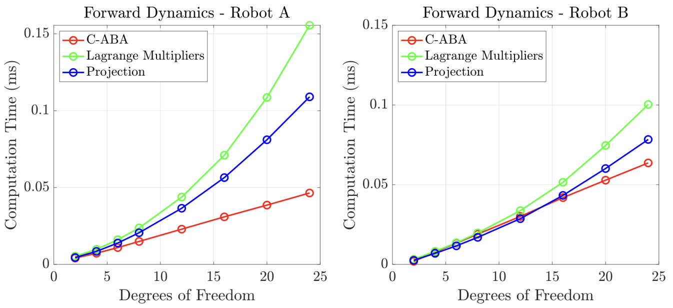

Fig. 3. Comparison of the average computation time for various forward dynamics algorithms for systems with closed-loop constraints.

$$\begin{bmatrix}\mathbf{q}_1\\\mathbf{q}_2\end{bmatrix} = \gamma_A(\mathbf{y}_1) = \begin{bmatrix}\mathbf{y}_1\\\alpha\mathbf{y}_1\end{bmatrix} \tag{16}$$

with $\alpha\in\mathbb{R}$ as a random gear ratio. Robot B is likewise a serial chain of links, but for this robot every *pair* of links in the chain is actuated by a *pair* of geared motors. The distal link in the pair is coupled to the motions of both rotors, such that each cluster is constrained by

$$\begin{bmatrix}\mathbf{q}_1\\\mathbf{q}_2\\\mathbf{q}_3\\\mathbf{q}_4\end{bmatrix} = \gamma_B\left(\begin{bmatrix}\mathbf{y}_1\\\mathbf{y}_2\end{bmatrix}\right) = \begin{bmatrix}\mathbf{y}_1\\\beta_{1,1}\mathbf{y}_1\\\beta_{2,1}\mathbf{y}_1+\beta_{2,2}\mathbf{y}_2\\\mathbf{y}_2\end{bmatrix} \tag{17}$$

with $\beta\in\mathbb{R}$ as random gear ratios. This coupled belt drive, while more complicated to simulate, enables lower inertia limbs since it allows the rotors to be placed closer to the base [1]. The results in Fig. 3 prove that the C-ABA scales linearly with the number of degrees of freedom with the robot. The improvement relative to state-of-the-art is larger for Robot A because the constraints are "more local", so the C-ABA more closely resembles the standard ABA.

VIII. Conclusions

In this paper, we presented novel approach for generalizing rigid-body dynamics algorithms to handle closure constraints in complex robotic systems. We first introduced the C-CG and detailed how SVA can be extended to describe the motion and force inertactions of rigid-body clusters. Next, the C-ABA was developed using the GPLC to derive recursive relationships for the articulated-*cluster* inertias of all of the clusters in the C-CG. We qualitatively described how the approach can extend to other algorithms relevant to contact-rich dynamic simulation. Finally, we presented preliminary results showing the efficiency of our proposed algorithm. Next steps include formal derivations of extended algorithms such as the C-EFPA and developing an open-source dynamics engine that implements these algorithms.

References

[1] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 1–8, IEEE, 2021.

[2] Y. Sim and J. Ramos, "Tello leg: The study of design principles and metrics for dynamic humanoid robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9318–9325, 2022.

[3] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8547–8553, IEEE, 2022.

[4] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[5] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *The international journal of robotics research*, vol. 2, no. 1, pp. 13–30, 1983.

[6] A. Jain, "Recursive algorithms using local constraint embedding for multibody system dynamics," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 49019, pp. 139–147, 2009.

[7] R. Featherstone, "A beginner's guide to 6-d vectors (part 1)," *IEEE robotics & automation magazine*, vol. 17, no. 3, pp. 83–94, 2010.

[8] C. F. Gauß, "Über ein neues allgemeines grundgesetz der mechanik.," 1829.

[9] M. L. Felis, "Rbdl: an efficient rigid-body dynamics library using recursive algorithms," *Autonomous Robots*, pp. 1–17, 2016.

[10] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019.

[11] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.

[12] A. Vereshchagin, "Computer simulation of the dynamics of complicated mechanisms of robot-manipulators," *Eng. Cybernet.*, vol. 12, pp. 65–70, 1974.

[13] S. H. Murphy, J. T. Wen, and G. N. Saridis, "Recursive calculation of geared robot manipulator dynamics," in *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 839–844, IEEE, 1990.

[14] A. Jain and G. Rodriguez, "Recursive dynamics for geared robot manipulators," in *29th IEEE Conference on Decision and Control*, pp. 1983–1988, IEEE, 1990.

[15] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *Ieee transactions on robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[16] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o (log (n)) calculation of rigid-body dynamics. part 1: Basic algorithm," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 867–875, 1999.

[17] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o (log (n)) calculation of rigid-body dynamics. part 2: Trees, loops, and accuracy," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 876–892, 1999.

[18] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[19] H. Brandl, "An algorithm for the simulation of multibody systems with kinematic loops," in *Proc. the IFToMM Seventh World Congress on the Theory of Machines and Mechanisms*, Sevilla, 1987.

[20] A. Jain, "Multibody graph transformations and analysis, part ii: Closed-chain constraint embedding," *Nonlinear dynamics*, vol. 67, no. 4, p. 2153–2170, 2012.

[21] A. Müller, "A constraint embedding approach for dynamics modeling of parallel kinematic manipulators with hybrid limbs," *Robotics and Autonomous Systems*, p. 104187, 2022.

[22] S. Kumar, K. A. v. Szadkowski, A. Mueller, and F. Kirchner, "An analytical and modular software workbench for solving kinematics and dynamics of series-parallel hybrid robots," *Journal of Mechanisms and Robotics*, vol. 12, no. 2, 2020.

[23] R. Kumar, S. Kumar, A. Müller, and F. Kirchner, "Modular and hybrid numerical-analytical approach-a case study on improving computational efficiency for series-parallel hybrid robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3476–3483, IEEE, 2022.

[24] P. Wensing, R. Featherstone, and D. E. Orin, "A reduced-order recursive algorithm for the computation of the operational-space inertia matrix," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4911–4917, IEEE, 2012.

[25] "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[26] F. Marques, A. P. Souto, and P. Flores, "On the constraints violation in forward dynamics of multibody systems," *Multibody System Dynamics*, vol. 39, pp. 385–419, 2017.

[27] A. Jain, *Robot and multibody dynamics: analysis and algorithms*. Springer Science & Business Media, 2010.

[28] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[29] T. Liu and M. Y. Wang, "Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and gauss-seidel methods," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 1, pp. 19–31, 2005.