Constrained Articulated Body Dynamics Algorithms

Ajay Suresha Sathya[1,2] and Justin Carpentier[1]

I. Introduction

Efficient rigid body dynamics algorithms [1] have played an essential role in robotics development. They enable dynamics evaluation in chip sets with limited resources and at high frequencies for demanding applications (e.g., computed torque control, model predictive control, large-scale simulation, reinforcement learning, etc.). Most simulators [2]–[6] use low-complexity algorithms such as the articulated body algorithm (ABA) [1], [7]–[9], which has $O(n)$ complexity, where n is the robot's degrees-of-freedom (DoF) only in constraint-free settings. In constrained settings, the simulators resort to Featherstone's sparsity-exploiting LTL algorithm in the joint space [10], [11], which has high computational complexity of $O(nd^2+m^2d+md^2+m^3)$, where m and d are the constraint dimensionality and the kinematic tree depth respectively.

A few low-complexity algorithms have been proposed for constrained dynamical systems, such as the Popov-Vereshchagin algorithm (PV algorithm) [12], [13] for kinematic trees with $O(n + m^2d + m^3)$ complexity. The PV algorithm was independendly discovered and further extended to kinematic loops in [14], [15] with the same computational complexity. See [16] for an expository derivation of the PV algorithm by solving the Gauss' principle of least constraint (GPLC) [17] using an equivalent linear quadratic regulator (LQR) formulation. [16] also proposed PV-soft and PV-early algorithms, each with only $O(n + m)$ complexity. PV-soft relaxes all motion constraints using quadratic penalties, while PV-early relies on the expensive singular value decomposition [18] (SVD). However, these efficient algorithms suffer from being fairly complex to derive and implement and perhaps due to which, their usage in simulators is currently low. Moreover, they cannot adequately deal with singular cases (e.g., redundant constraints, singular constraints, etc.), in which cases, they resort to Tikhonov regularization (which biases solutions towards origin adversely affecting constraint satisfaction) or the expensive SVD algorithms.

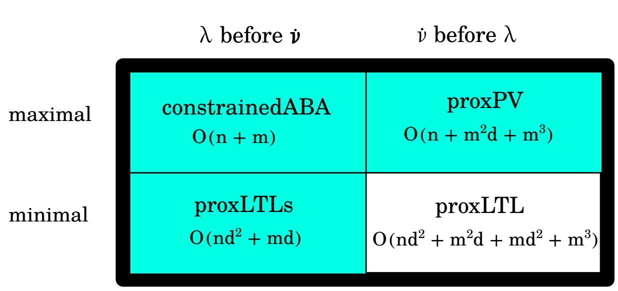

Fig. 1: Overview of the proximal dynamics algorithms.

[1] Inria - Département d'Informatique de l'École normale supérieure, PSL Research University. {ajay.sathya,justin.carpentier}@inria.fr

[2] MECO Research Team, Department of Mechanical Engineering, KU Leuven and Flanders Make@KU Leuven, Belgium {ajay.sathya}@kuleuven.be

Addressing these issues, we present three new constrained dynamics algorithms (CDAs) constrainedABA, proxPV and proxLTLs based on proximal algorithms [19] that are simple and effectively handle singular cases. These algorithms are closely related to the proxLTL [6] algorithm and arise depending on whether joint accelerations or constraint forces are eliminated first and depending on the usage of maximal or minimal coordinates as shown in the overview figure Fig. 1.

II. Proximal Reformulation of Constrained Dynamics

A. Constrained dynamics in generalized coordinates

Constrained dynamics. According to GPLC [17], [20], [21], the acceleration $\dot{\boldsymbol{\nu}}$ of a constrained system at state $(\mathbf{q}, \boldsymbol{\nu})$, when acted upon by $\boldsymbol{\tau}$, is the minimizer of the following equality-constrained strongly convex quadratic program (QP):

$$\underset{\dot{\boldsymbol{\nu}}}{\textbf{minimize}} \quad \frac{1}{2}\|\dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\nu}}_{\text{free}}(\mathbf{q}, \dot{\boldsymbol{\nu}}, \boldsymbol{\tau})\|^2_{M(\mathbf{q})} \tag{1a}$$

$$\textbf{subject to} \quad J_{\mathbf{f}_c}(\mathbf{q})\dot{\boldsymbol{\nu}} + \dot{J}_{\mathbf{f}_c}(\mathbf{q}, \boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{a}_c^* - \boldsymbol{\gamma}_{\mathbf{f}_c}(\mathbf{q}, \boldsymbol{\nu}), \tag{1b}$$

where $M(\mathbf{q}) \in \mathbb{S}^n_{++}$ is the joint-space inertia matrix (JSIM) and unconstrained joint-space acceleration

$$\dot{\boldsymbol{\nu}}_{\text{free}}(\mathbf{q}, \dot{\boldsymbol{\nu}}, \boldsymbol{\tau}) := M^{-1}(\mathbf{q})\left(\boldsymbol{\tau} - \mathbf{c}(\mathbf{q}, \boldsymbol{\nu})\right), \tag{2}$$

where $\mathbf{c}(\mathbf{q}, \boldsymbol{\nu})$ is the generalized force vector due to gravity, Coriolis and centripetal effects. Eq. (1b) is motion constraint expressed at the acceleration level. The variable dependencies will be dropped for brevity whenever obvious from the context.

Constrained dynamics Lagrangian. The solution to the QP above is the primal-dual saddle point of the Lagrangian [22]

$$(\dot{\boldsymbol{\nu}}^*, \boldsymbol{\lambda}^*) = \textbf{arg}\,\underset{\boldsymbol{\lambda}}{\textbf{max}}\,\underset{\dot{\boldsymbol{\nu}}}{\textbf{min}}\,\mathcal{L}(\dot{\boldsymbol{\nu}}, \boldsymbol{\lambda}), \tag{3}$$

where

$$\mathcal{L}(\dot{\boldsymbol{\nu}}, \boldsymbol{\lambda}) := \frac{1}{2}\|\dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\nu}}_{\text{free}}\|^2_M + \boldsymbol{\lambda}^T(J_{\mathbf{f}_c}\dot{\boldsymbol{\nu}} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^*). \tag{4}$$

Eliminating $\dot{\boldsymbol{\nu}}$ using

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_{\text{free}} - M^{-1}J_{\mathbf{f}_c}^T\boldsymbol{\lambda}, \tag{5}$$

and back-substituting in Eq. (3), gives the dual function

$$g(\boldsymbol{\lambda}) = -\frac{1}{2}\boldsymbol{\lambda}^T\Lambda^{-1}\boldsymbol{\lambda} + \left(J_{\mathbf{f}_c}\dot{\boldsymbol{\nu}}_{\text{free}} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^*\right)^T\boldsymbol{\lambda}, \tag{6}$$

where $\Lambda^{-1}(\mathbf{q}) := J_{\mathbf{f}_c}M^{-1}J_{\mathbf{f}_c}^T$ is the so-called Delassus matrix [23], [24], also known as the inverse operational space inertial matrix (inverse OSIM) [25]. When Λ^{-1} is full rank, the optimal Lagrange multipliers is obtained by solving

$$\Lambda^{-1}\boldsymbol{\lambda}^* = J_{\mathbf{f}_c}\dot{\boldsymbol{\nu}}_{\text{free}} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^*. \tag{7}$$

However, in practice, Λ^{-1} often does not have full rank due to redundant constraints or kinematic singularities and we propose using two popular optimization approaches to solving the QP in Eq. (1), that are mathematically equivalent but differ in their computational cost.

1) Dual proximal point method (proxLTL): An exact and efficient alternative to Tikhonov regularization or SVD that we will leverage is the proximal point algorithm (PPA) [19], [26], which is effective for robotics problems [27]–[29], and most often requiring few iterations (each of which is efficient) to converge for robot dynamics problems [27]. Applying PPA to optimize the dual function in Eq. (6) gives

$$\boldsymbol{\lambda}^{k+1} = \text{prox}_{\mu,-g}(\boldsymbol{\lambda}^k) = \mathop{\textbf{arg min}}_{\boldsymbol{\lambda}} \; -g(\boldsymbol{\lambda}) + \frac{1}{2\mu}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^k\|^2, \tag{8a}$$

$$= \Lambda_\mu \left(J_{\mathbf{f}_c} \dot{\boldsymbol{\nu}}_{\text{free}} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^* + \frac{1}{\mu}\boldsymbol{\lambda}^k \right), \tag{8b}$$

where $\Lambda_\mu^{-1} := \Lambda^{-1} + \frac{1}{\mu} I$ is the damped Delassus matrix. $\Lambda_\mu^{-1} \in \mathbb{S}_{++}^m$ and can be factorized efficiently using Cholesky decomposition. This algorithm, which we will call proxLTL, has already been implemented in the PINOCCHIO library [27]. Even if the primal problem in Eq. (1) is infeasible (e.g., due to redundant constraints and Baumgarte terms [30]), it has been shown [31]–[33] that the primal residuals converges to a desirable least squares residual solution during PPA iterations.

2) Augmented Lagrangian method (proxLTLs): An alternative to proxLTL that can solve the QP in Eq. (1) exactly is the augmented Lagrangian method [34], [35] (ALM), where the augmented Lagrangian function is

$$\mathcal{L}^A(\dot{\boldsymbol{\nu}}, \boldsymbol{\lambda}) := \mathcal{L}(\dot{\boldsymbol{\nu}}, \boldsymbol{\lambda}) + \frac{\mu}{2}\|J_{\mathbf{f}_c}\dot{\boldsymbol{\nu}} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^*\|^2. \tag{9}$$

ALM iterations alternately optimize $\mathcal{L}^A$ over its primal and dual variables

$$\dot{\boldsymbol{\nu}}^{k+1} = M_\mu^{-1}\left\{ M\dot{\boldsymbol{\nu}}_{\text{free}} - J_{\mathbf{f}_c}^T \left(\boldsymbol{\lambda}^k + \mu\left(\boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^* \right) \right) \right\}, \tag{10a}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mu\left(J_{\mathbf{f}_c}\dot{\boldsymbol{\nu}}^{k+1} + \boldsymbol{\gamma}_{\mathbf{f}_c} - \boldsymbol{a}_c^* \right), \tag{10b}$$

where $M_\mu := M + \mu J_{\mathbf{f}_c}^T J_{\mathbf{f}_c}$ is the augmented JSIM, with the influence of the constraints from the quadratic term in the augmented Lagrangian function.

B. Constrained dynamics in maximal coordinates

In the so-called 'maximal' coordinates, we will use Featherstone's spatial algebra [1] to refer to rigid body quantities. Both proxLTL and proxLTLs algorithms have lower complexity counterparts proxPV and constrainedABA respectively, that can be derived by applying dynamic programming (DP) on the problem of Gauss' principle in the so-called maximal coordinates [21]

$$\mathop{\textbf{minimize}}_{\dot{\boldsymbol{\nu}}, \boldsymbol{a}} \quad \sum_{i=1}^{n_b} \left\{ \frac{1}{2}\mathbf{a}_i^T H_i \mathbf{a}_i - \mathbf{f}_i^T \mathbf{a}_i \right\} \tag{11a}$$

$$\textbf{subject to} \quad \mathbf{a}_i = \mathbf{a}_{\pi(i)} + S_i \dot{\boldsymbol{\nu}}_i + \mathbf{a}_{b,i}, \quad i = 1, 2, \ldots, n_b, \tag{11b}$$

$$K_i \mathbf{a}_i = \mathbf{k}_i, \quad i = 1, 2, \ldots, n_b, \tag{11c}$$

$$\mathbf{a}_0 = -\mathbf{a}_{\text{grav}}, \tag{11d}$$

All the spatial quantities are expressed in the inertial frame in our subsequent derivations for simplicity of notation. $\pi(i)$ is the parent link of the i^{th} link in the kinematic tree, $\boldsymbol{\nu}_i \in \mathbb{R}^{n_i}$ is the i^{th} joint's generalized velocities, $\dot{\boldsymbol{\nu}}_i \in \mathbb{R}^{n_i}$ is the i^{th} joint's generalized accelerations. S_i is the i^{th} joint's motion subspace matrix of size $6 \times n_i$, with n_i being the i^{th} joint's DoF.

III. COMPUTATIONAL BENCHMARKING, DISCUSSION AND CONCLUSIONS

We now present the benchmarking results of this paper's algorithms on diverse robots, namely Kuka Iiwa (7 DoF chain), Solo (18 DoF tree) [36], Talos (50 DoF tree) [37], and Atlas with two shadow-hands attached to each wrist (84 DoF tree). Constraints on a hand, fingertip, or feet in the benchmarks are represented as H_{m_i}, T_{m_i} or F_{m_i} respectively where m_i is the constraint dimension. We implemented proxPV, proxLTLs and constrainedABA in `C++` within the PINOCCHIO library [6], and compared them along with the proxLTL algorithm [27] already available in PINOCCHIO. All timings were benchmarked on a `13th Gen Intel® Core™ i9-13950HX` laptop CPU running `Ubuntu 22.04LTS` operating system. The code was compiled using `Clang-14` compiler with the usual optimized compilation flags `-O3 -march=native`. Table I lists the benchmarking results.

TABLE I: Computational timings of the algorithms in μs with `Turbo Boost` enabled. Each algorithm is allowed three proximal iterations.

System	cABA	PV	LTLs	LTL
Iiwa - H_3	0.97	**0.95**	0.99	1.51
Iiwa - H_6	1.02	1.33	**1.0**	2.0
Solo - F_3^2	**1.81**	2.11	2.06	3.08
Solo - F_3^4	**2.25**	2.84	2.27	4.06
Talos - F_6^2	**5.02**	6.27	7.14	10.2
Talos - $\mathrm{F}_6^2\mathrm{H}_6^2$	**6.21**	8.84	7.57	14.5
Atlas SR - F_6^2	**7.90**	9.56	13.4	17.7
Atlas SR - $\mathrm{F}_6^2\mathrm{T}_3^5$	**9.95**	14.0	14.7	26.5
Atlas SR - $\mathrm{F}_6^2\mathrm{T}_3^{10}$	**11.7**	20.8	15.0	36.8

ConstrainedABA is the first linear complexity algorithm that deals with singular cases without resorting to expensive SVD computation, that we are aware of, and perhaps the simplest. ConstrainedABA emerged as the fastest out of the four algorithms for larger robots like quadrupeds and humanoids, being over 2x faster than proxLTL, the previously existing state-of-the-art `C++` implementation. The higher complexity proxLTLs surprisingly remained competitive even for larger robots, when heavily constrained, due to linear complexity in constraint dimension and an efficient vectorized `C++` implementation.

The proximal formulation generalizes and establishes connections between existing algorithms like MUJOCO's solver, PV-soft, and PV-early. It is a powerful formulation that enables efficient trading-off between MUJOCO-style compliance and an expensive SVD-style rigid contact during singular cases (and also in general) through fast proximal iterations. ConstrainedABA and proxLTLs, in particular, are fairly straightforward to implement, by introducing only a few new lines of additional code compared to Featherstone's original ABA and LTL algorithms.

We invite interested readers to refer to [38] for a complete description, derivation, analysis and detailed benchmarking of the presented algorithms. [38] also proposes cABA-OSIM, which computes the damped Delassus inverse matrix with the optimal complexity of $O(n + m^2)$.

Acknowledgements

This work was supported by the Belgium government through the FWO project G0D1119N of the Research Foundation - Flanders (FWO - Flanders), by the French government under the management of Agence Nationale de la Recherche through the project INEXACT (ANR-22-CE33-0007-01) and as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), by the European Union through the AGIMUS project (GA no.101070165) and the Louis Vuitton ENS Chair on Artificial Intelligence. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

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

[2] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[3] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, "DART: Dynamic animation and robotics toolkit," *The Journal of Open Source Software*, vol. 3, no. 22, p. 500, Feb 2018. [Online]. Available: https://doi.org/10.21105/joss.00500

[4] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 895–902, 2018. [Online]. Available: www.raisim.com

[5] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Int. Robots. Syst.* IEEE, 2012, pp. 5026–5033.

[6] 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 *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2019, pp. 614–619.

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

[8] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Robot. Res.*, vol. 2, no. 1, pp. 13–30, 1983.

[9] H. Brandl, R. Johanni, and M. Otter, "A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix," *IFAC Proceedings Volumes*, vol. 19, no. 14, pp. 95–100, 1986.

[10] R. Featherstone, "Efficient factorization of the joint-space inertia matrix for branched kinematic trees," *Int. J. Robot. Res.*, vol. 24, no. 6, pp. 487–500, 2005.

[11] ——, "Exploiting sparsity in operational-space dynamics," *Int. J. Robot. Res.*, vol. 29, no. 10, pp. 1353–1368, 2010.

[12] J. P. Popov, A. F. Vereshchagin, and S. L. Zenkevič, *Manipuljacionnyje roboty: Dinamika i algoritmy*. Nauka, 1978.

[13] A. F. Vereshchagin, "Modeling and control of motion of manipulational robots," *Soviet Journal of Computer and Systems Sciences*, vol. 27, no. 5, pp. 29–38, 1989.

[14] M. Otter, H. Brandl, and R. Johanni, "An algorithm for the simulation of multibody systems with kinematic loops," in *Proceedings of the 7th World Congress on Theory of Machines and Mechanisms, IFToMM, Sevilla, Spain*, 1987.

[15] D.-S. Bae and E. J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part ii. closed loop systems," *Journal of Structural Mechanics*, vol. 15, no. 4, pp. 481–506, 1987.

[16] A. S. Sathya, H. Bruyninckx, W. Decré, and G. Pipeleers, "Efficient constrained dynamics algorithms based on an equivalent lqr formulation using gauss' principle of least constraint," *IEEE Transactions on Robotics*, vol. 40, pp. 729–749, 2024.

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

[18] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[19] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[20] F. E. Udwadia and R. E. Kalaba, *Analytical dynamics : a new approach*. Cambridge: Cambridge University press, 1996.

[21] H. Bruyninckx and O. Khatib, "Gauss' principle and the dynamics of redundant and constrained manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3. IEEE, 2000, pp. 2563–2568.

[22] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[23] É. Delassus, "Mémoire sur la théorie des liaisons finies unilatérales," in *Annales scientifiques de l'École normale supérieure*, vol. 34, 1917, pp. 95–179.

[24] B. Brogliato, *Nonsmooth mechanics*. Springer, 1999, vol. 3.

[25] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational-space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[26] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[27] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and sparse resolution of constrained dynamic equations," in *Proc. Robot., Sci. Syst.*, 2021.

[28] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.

[29] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained differential dynamic programming: A primal-dual augmented lagrangian approach," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 371–13 378.

[30] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.

[31] O. Güler, "On the convergence of the proximal point algorithm for convex minimization," *SIAM journal on control and optimization*, vol. 29, no. 2, pp. 403–419, 1991.

[32] A. Chiche and J. C. Gilbert, "How the augmented lagrangian algorithm can deal with an infeasible convex quadratic optimization problem," *Journal of Convex Analysis*, vol. 23, no. 2, 2016.

[33] A. Bambade, F. Schramm, A. Taylor, and J. Carpentier, "QPLayer: efficient differentiation of convex quadratic optimization," in *2024 International Conference on Learning Representations (ICLR)*, 2023.

[34] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[35] M. J. Powell, "A method for nonlinear constraints in minimization problems," *Optimization*, pp. 283–298, 1969.

[36] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols *et al.*, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.

[37] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 689–695.

[38] A. Sathya and J. Carpentier, "Constrained Articulated Body Dynamics Algorithms," Feb. 2024, working paper or preprint. [Online]. Available: https://hal.science/hal-04443056