Efficient, Generic and Robust Resolution of Constrained Dynamics

Justin Carpentier*, Rohan Budhiraja* and Nicolas Mansard†
*Inria, Département d'informatique de l'ENS, Paris, France
†LAAS-CNRS, Toulouse, France

***Abstract*—Control of robots with kinematic constraints like loop-closure constraints or interactions with the environment requires solving the underlying constrained dynamics equations of motion. Several approaches have been proposed so far in the literature to solve these constrained optimization problems, for instance by either taking advantage in part of the sparsity of the kinematic tree or by considering an explicit formulation of the constraints in the problem resolution. Yet, not all the constraints allow an explicit formulation and in general, approaches of the state of the art suffer from singularity issues, especially in the context of redundant or singular constraints. In this work, we propose a unified approach to solve forward dynamics equations involving constraints in an efficient, generic and robust manner. We notably show how this new approach outperforms current alternatives of the state of the art, providing new algorithmic foundations for robot simulation and control.**

I. Main Contributions

As soon as a robot makes contacts with the world or is endowed with loop closures in its design, its dynamics is governed by the constrained equations of motion. From a phenomenological point of view, these equations of motion follow the so-called least-action principle, also known under the name of the Maupertuis principle which dates back to the 17^{th} century. This principle states that the motion of the system follows the closest possible acceleration to the free-falling acceleration (in the sense of the kinetic metric) which respects the constraints. In other words, solving the constrained equations of motion boils down to solving a constrained optimization problem where forces acts as the Lagrange multipliers of the motion constraints.

A. A generic and numerically robust approach

Constrained dynamics. In particular, if we consider the Lagrangian dynamics of an unconstrained poly-articulated system following the usual notations:

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + c(\boldsymbol{q}, \boldsymbol{v}) + g(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{1}$$

when the system is subject to constraints, the least-action principle reads:

$$\min_{\dot{\boldsymbol{v}}} \quad \frac{1}{2}\|\dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_{\text{free}}(\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\tau})\|^2_{M(q)} \tag{2a}$$

$$\text{subject to} \quad J_c(\boldsymbol{q})\dot{\boldsymbol{v}} + \dot{J}_c(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} = 0, \tag{2b}$$

This work is supported by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

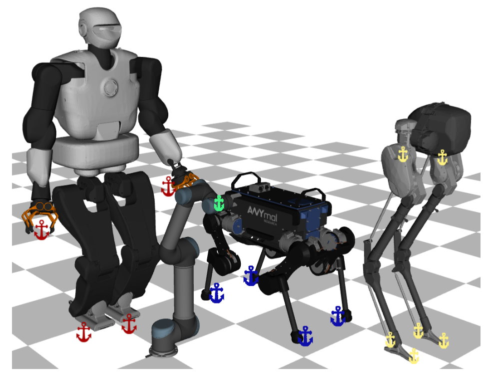


Fig. 1. Robotic systems may be subject to different types of constraints: point contact constraints (quadrupeds), flat foot constraints (humanoids), closed kinematic chains (parallel robots, here the 4-bar linkages of Cassie) or even contact with the end effectors (any robot).

where J_c is the constraint Jacobian, $\|.\|_{M(q)}$ is the kinetic metric and $\dot{\boldsymbol{v}}_{\text{free}}$ is the free acceleration of the system, solution of Eq. (1).

Lagrangian of the constrained dynamics. Problem (2) corresponds to an equality-constrained quadratic program. The associated Lagrangian equation is:

$$L(\dot{\boldsymbol{v}}, \boldsymbol{\lambda}) = \frac{1}{2}\|\dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_{\text{free}}\|^2_M + \boldsymbol{\lambda}^t \left(J_c\dot{\boldsymbol{v}} + \dot{J}_c(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v}\right), \tag{3}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ corresponds to the Lagrange multipliers associated to the constraints. Yet, as soon as the Jacobian is rank-deficient (over-constrained, kinematic singularity, etc.) or the constraint is infeasible, standard approaches are likely to fail [5, 1, 6] or require further regularization terms [2].

Proximal reformulation. To overcome these limitations, we propose instead to exploit a proximal reformulation [7] of Eq. (3). This reads:

$$L_\mu(\dot{\boldsymbol{v}}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_e) = L(\dot{\boldsymbol{v}}, \boldsymbol{\lambda}) - \frac{\mu}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_e\|_2^2, \tag{4}$$

where μ is positive smoothing parameter and $\boldsymbol{\lambda}_e \in \mathbb{R}^m$ is the current estimate of the Lagrange multipliers. This smoothed Lagrangian is now strictly convex in $\dot{\boldsymbol{v}}$ (because M is strictly definite positive) and strictly concave in $\boldsymbol{\lambda}$. Solving the saddle-point problem associated with (4) leads to the primal/dual system of equations:

$$\underbrace{\begin{bmatrix} -\mu I_m & J_{f_c} \\ J_{f_c}^t & M \end{bmatrix}}_{K_\mu(\boldsymbol{q})} \begin{bmatrix} \boldsymbol{\lambda} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} -\dot{J}_c\boldsymbol{v} - \mu\boldsymbol{\lambda}_e \\ M\dot{\boldsymbol{v}}_{\text{free}} \end{bmatrix}, \tag{5}$$

where $K_\mu(\boldsymbol{q})$ is the so-called KKT matrix associated to the saddle-point. From its structure, it appears that K_μ is non-singular, even when J_c is rank deficient, enforcing the well-possness of the formulation. By iteratively solving (5) (updating the value $\boldsymbol{\lambda}_e$ with the optimal value of $\boldsymbol{\lambda}$), it converges to the original saddle-point point associated to (3) [7], following the spirit of proximal algorithms. Notably, the rate of convergence of this approach is linear in μ.

B. *Exploiting at best the underlying problem sparsity*

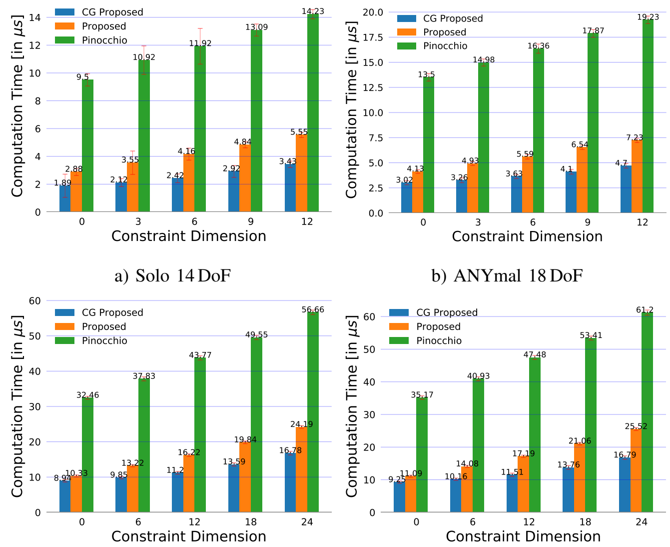
Sparse Cholesky factorization of M. In [4], Featherstone introduces a sparse Cholesky decomposition of M which exploits at best the sparsity pattern induced by the kinematic tree. In particular, he advocates for an upper decomposition of the form $M = UDU^t$, and not for a (more classical) lower Cholesky decomposition $M = LDL^t$, with L and U respectively unitary lower and upper triangular matrices, and D positive diagonal matrix. Indeed and following the remarks made in [4], shaped like that, L will not exhibit any specific pattern resulting in a dense matrix.

Sparse Cholesky factorization of $\boldsymbol{K_\mu}$. Base on this prior work, we introduce a generic algorithm to compute an upper and sparse Cholesky of $\boldsymbol{K_\mu} = U_{K_\mu} D_{K_\mu} U^t_{K_\mu}$. We notably show that this factorization is always well-posed (from a numerical point of view) and admits a sparsity pattern similar to the one exhibited by the constraint Jacobian J_c. Due to space limitation, we skip the presentation of the algorithm in favor of general results illustrating the overall contributions.

II. Results

We implemented in `C++` within the Pinocchio library [3] both the sparse Cholesky factorization and the corresponding constrained dynamics formulation. Comparison of our implementation with the state-of-the-art highlights the benefits of the proposed proximal formulation of the constrained dynamics and the exploitation of the branch-induced sparsity. We benchmark our implementation for multiple robots with different dimensions, kinematic structures, and type of constraints. Additionally, we show the results of our constrained dynamics implementation to simulate the motion of different robots.

We use the time taken for solution of the constrained dynamics to demonstrate the performance of our algorithm against the state-of-the-art approach largely used in the community [5]: . In Fig. 2, we see significant reduction with respect to the state-of-the-art in the timings for the solution of constrained dynamics, which involves the computation of the inverse of the KKT matrix K_μ. We benchmark our algorithm against the `forwardDynamics` as proposed in [5] and implemented within Pinocchio. Moreover, we use the source-code generation tools (CppADCodegen) available in Pinocchio to compile binary code for sparse decomposition of all our (robot, constraints) combinations. We see significant reduction in the timings for constrained dynamics because of the sparse solutions, with our formulation reducing timings by $> 50\%$ in all robots. In addition, we see a small improvement



a) Solo 14 DoF

b) ANYmal 18 DoF

c) iCub 35 DoF

d) Talos 38 DoF

Fig. 2. Proposed Contact Dynamics: The benchmarks show performance of the proposed constrained dynamics algorithm against the state-of-the-art Pinocchio [3] C++ library for different robotic platforms.

because of the binary source-code, which further optimizes over redundant computations. However, this additional performance benefit is capped because of the lack of vectorization in binary compiled functions. In total, our sparse method and source-code generation vastly outperform the state-of-the-art implementation.

III. Conclusion

In this work, we have introduced a numerically robust and sound approach to handle singularities issues occurring in constrained dynamics. Additionally, we have proposed a sparse Cholesky decomposition of the underlying KKT matrix, which enables to exploit at best the sparsity induced by the kinematic tree. This work naturally extends to further contexts (robot control, robot simulation, etc.), that we left as future work.

References

[1] David Baraff. Linear-time dynamics using lagrange multipliers. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 137–146, 1996.

[2] Michele Benzi et al. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[3] Justin et al. Carpentier. The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE/SICE International Symposium on System Integration (SII)*, 2019.

[4] Roy Featherstone. Efficient factorization of the joint-space inertia matrix for branched kinematic trees. *The International Journal of Robotics Research*, 24(6):487–500, 2005.

[5] Roy Featherstone. Exploiting sparsity in operational-space dynamics. *The International Journal of Robotics Research*, 29(10): 1353–1368, 2010.

[6] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[7] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.