

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/343098270>

Analytical Inverse of the Joint Space Inertia Matrix

Technical Report · May 2018

CITATIONS

0

READS

28

1 author:



Justin Carpentier

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)

34 PUBLICATIONS 440 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Compliant feedback control of legged robots [View project](#)



Memory of Motion (MEMMO) [View project](#)

Analytical Inverse of the Joint Space Inertia Matrix

Justin Carpentier

Laboratoire d'Analyse et d'Architecture des Systèmes

Université de Toulouse

7 avenue du Colonel Roche, Toulouse, FRANCE

Email: justin.carpentier@laas.fr

Abstract—In this short report, we briefly introduce a new algorithm to compute the analytical inverse of the joint space inertia matrix. Such inverse may be useful for several applications like the computation of the analytical derivatives of the forward dynamics. We also provide a free C++ implementation of this algorithm inside our C++ framework for rigid-body dynamics computations called Pinocchio.

I. CONTENT

The algorithm that we introduce now allows to compute in recursive manner (i.e. by exploiting the sparsity induced by the kinematic tree) the inverse of the joint space inertia matrix, without explicitly computing the joint space inertia matrix itself. It formally exploits the computations done in the articulated body algorithms by removing the affine terms. To the best of our knowledge, this is the first time that this algorithm is introduced and detailed.

This algorithm is composed of three passed, similarly to ABA. We use the same notations that have been used so far in our paper on the analytical derivatives [2] and which follows the ones introduced in [4]. We denote by M_{inv} the inverse of M et we use the notation $M_{\text{inv}}[i, :]$ to designate the rows of M_{inv} which correspond to the joint i . Using this notation, $M_{\text{inv}}[i, \text{subtree}(i)]$ are the columns of $M_{\text{inv}}[i, :]$ supported by the joint i including i itself. In addition, we exploit the Pythonic notation $M_{\text{inv}}[i, i :]$ to express all the columns of $M_{\text{inv}}[i, :]$ starting at index i until the last column of M_{inv} . We also exploit the fact that M^{-1} is symmetric.

In Algo. 1, F_i is the force-set collecting the contributions of the supporting tree rooted at i and P_i is a motion-set which contains the contributions of all the parents of joint i .

II. IMPLEMENTATION

This algorithm is made freely available in our open-source C++ framework called Pinocchio [3]. Pinocchio allows to compute in efficient manner the forward and inverse dynamics of poly-articulated systems such as humanoid robots, robotic manipulators, quadrupedal robots. Pinocchio is now at the hearth of many softwares developed in the Gepetto team at LAAS-CNRS for planing and control of robots [5, 1]. In addition, Pinocchio comes with a complete Python interface for easy code prototyping.

ACKNOWLEDGMENTS

This work is supported by the RoboCom++ FLAG-ERA JTC 2016 proposal.

REFERENCES

- [1] Justin Carpentier and Nicolas Mansard. Multi-contact locomotion of legged robots. *Submitted to IEEE Transaction on Robotics*, 2018.
- [2] Justin Carpentier and Nicolas Mansard. Analytical derivatives of rigid body dynamics algorithms. In *Robotics: Science and Systems*, 2018.
- [3] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al. Pinocchio: fast forward and inverse dynamics for poly-articulated systems, 2015–2018. URL <https://stack-of-tasks.github.io/pinocchio>.
- [4] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008.
- [5] Joseph Mirabel, Steve Tonneau, Pierre Fernbach, Anna-Kaarina Seppälä, Mylene Campana, Nicolas Mansard, and Florent Lamiroux. HPP: A new software for constrained motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

Relationship to Featherstone's ABA (when $\dot{q}=0$, $a_{grv}=0$)

$$F_i \tau = p_i^A \leftarrow \text{Articulated Bias Force}$$

$$P_i \tau = a_i \leftarrow \text{Spatial acceleration}$$

Algorithm 1 - Pseudo code of the algorithm to directly compute the inverse of the joint space inertia matrix and which is inspired from ABA exposed by Featherstone [4, p.132] and follows the same notations.

1 First forward pass:

2 **for** $i = 1$ **to** N_B **do**

3 $[X_J, S_i] = \text{jcalc}(\text{jtype}(i), q_i, \dot{q}_i)$

4 ${}^i X_{\lambda(i)} = X_J X_T(i)$

5 $I_i^A = I_i$

6 **end**

7 Backward pass:

8 **for** $i = N_B$ **to** 1 **do**

9 $U_i = I_i^A S_i$

10 $D_i = S_i^T U_i$

11 $M_{\text{inv}}[i, i] = D_i^{-1}$

12 $M_{\text{inv}}[i, \text{subtree}(i)] = M_{\text{inv}}[i, \text{subtree}(i)] - D_i^{-T} S_i^T F_i[:, \text{subtree}(i)]$

13 **if** $\lambda(i) \neq 0$ **then**

14 ~~$F_{\lambda(i)}[:, \text{subtree}(i)] = F_{\lambda(i)}[:, \text{subtree}(i)] + \lambda(i) X_i^* U_i M_{\text{inv}}[i, \text{subtree}(i)]$~~

15 $I_i^a = I_i^A - U_i D_i^{-1} U_i^T$

16 $I_{\lambda(i)}^A = I_{\lambda(i)}^A + \lambda(i) X_i^* I_i^a X_{\lambda(i)}$

17 **end**

18 **end**

19 Second forward pass:

20 **for** $i = 1$ **to** N_B **do**

21 **if** $\lambda(i) \neq 0$ **then**

22 $M_{\text{inv}}[i, i:] = M_{\text{inv}}[i, i:] - D_i^{-1} U_i^T {}^i X_{\lambda(i)} P_{\lambda(i)}[i, i:]$

23 **end**

24 $P_i[i, i:] = S_i M_{\text{inv}}[i, i:]$

25 **if** $\lambda(i) \neq 0$ **then**

26 $P_i[i, i:] = P_i[i, i:] + {}^i X_{\lambda(i)} P_{\lambda(i)}[i, i:]$

27 **end**

28 **end**

At this point in the algorithm $M_{\text{inv}}[i, :]$ is not its final value. Compared to ABA it satisfies:

$$M_{\text{inv}}[i, :] \tau = D_i^{-1} u_i$$

← Replace w/lines below

I_i^a is the portion of I_i^A transmitted across the joint

Replacement for line 14:

$$F_i^a[:, \text{subtree}(i)] = F_i[:, \text{subtree}(i)] + U_i M_{\text{inv}}[i, \text{subtree}(i)]$$

$$F_{\lambda(i)}[:, \text{subtree}(i)] = F_{\lambda(i)}[:, \text{subtree}(i)] + X_i^* F_i^a[:, \text{subtree}(i)]$$

• F_i^a is the portion of F_i transmitted across the joint

• Relates to ABA on pg. 132 of [4] as $F_i^a \tau = p_i^a$