Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 3 (2012) 54 – 59

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Modeling, Identification and Control

An United Recursive Robot Dynamics Based on Screws

Bing Shao*, En-tao Yuan

College of Mechanical Engineering, Shanghai Dianji University, Jiangchuan Road 1201 of minhang district, Shanghai 200240, China

Abstract

The key point of robot dynamics is optimal design and control. The efficiency of robot dynamics has been the goal of researchers in recent years. Screws are used to describe dynamic problems in this paper, and an O(N) recursive robot forward dynamic algorithm is given on this. It can be easily extended to tree topology, closed loop and spatial robot systems. And three classic methods of robot dynamics are compared for easy of use. The results show that dynamics described with screws are helpful in high efficient dynamics modelling. The dynamical expressions based on screws are concise and clear. It's efficiency is high of O(N) and is linear to the degree of freedom. With the improvement of computation efficiency, it will make the real-time dynamics control become possible.

Keywords: Robot, dynamics, recursive algorithm, screws

1. Introduction

For the simulation and control of astronautics robots and complex mechanical systems, computation efficiency is very important. In the many methods of analyzing space machine, screw is an efficient tool. Screw can be expressed by a pair of vectors, such as angular velocity and linear velocity, force and moment. In 3-D space, screw can be described as six scalar quantities. The use of screw theory has been very effective.

* Corresponding author. Tel.: +86-15021216409.
E-mail address: shaobox@tom.com.

doi:10.1016/j.aasri.2012.11.010

Also, it is important in developing algorithms which have applicability to robotic mechanisms with general geometries and joint structures. So it is easily applied in kinematics and dynamics and it is excellent in clear geometry conception and simple expression form.

The Lie algebras are the algebraic structure of Lie groups $SE(3)$ expressed by screws. The Lie groups and Lie algebras have been very effective in the methods of analyzing space machines. Also, it is important in developing algorithms which have applicability to mechanisms with general geometries and joint structures.

The classical dynamical modelling method is the Lagrange method [1], it leads to $O(N^4)$ algorithms. For inverse robot dynamics, the first O(N) algorithm was the Newton-Euler(NE) formula [2]. The inverse dynamical algorithm of Walker and Orin [3] was the basic of forward dynamics which computed the inertial parameters of composite sets of rigid bodies at the outer end of the manipulator chain. The algorithm was $O(N^3)$.

The O(N) algorithm of Featherstone was called the Articulated-Body Algorithm (ABA) [4]. In terms of the total number of arithmetic operations required, the ABA was more efficient than the CRBA for $N > 9$.

Rodriguez [5] recognized the similarities between the forward dynamics problem and the concepts of Kalman filtering, and developed an O(N) algorithm called the spatial operator algebra(SOA) method for the multi-body dynamics. Jain [6] used the spatial operator algebra framework to provide a unified formulation for manipulator dynamics. He compared the various $O(N^3)$, $O(N^2)$, and O(N) algorithms that had been previously published. J. García de Jalón[7] put forward a dynamical algorithm based on full Cartesian coordinates, it can also obtain O(N) complexity.

In this paper, an O(N) recursive robot forward dynamic algorithm is described on Screws. Then the computation complexity is interpreted in detail and compared with other methods. At last it gives the different notations of three classic O(N) recursive dynamics for ease of understanding.

2. Inverse and Forward Recursive Dynamics

2.1. The inverse dynamics of Newton-Euler(NE) in screw theory

In recent years, modern differential geometry has been very important in robotics. So we define motion screw as: $\hat{v} = [\omega \quad v]^T \in R^6$, and force screw as: $\hat{f} = [f \quad \tau]^T \in R^6$.

A robot mechanism can be expressed by: N rigid body links, marked as 1…N, a fixed base, a set of N joints which join the base and the links together. Figure 1 is the velocity and acceleration of two adjacent links, the velocity of link i is $\hat{v}_i$, and

$$\hat{v}_i = \hat{v}_{i-1} + h_i \dot{q}_i \tag{1}$$

Where $h_i \in R^6$ spanning the motion freedom subspace of joint i, and $\dot{q}_i$ is the vector of joint velocity.

The time-derivative of above equation is:

$$\hat{a}_i = \hat{a}_{i-1} + h_i \ddot{q}_i + \hat{c}_i \tag{2}$$

Where $\hat{a}_i$ is the acceleration of link i, composed by the angular acceleration and line acceleration. $\ddot{q}_i$ is a vector of joint acceleration. $\hat{c}_i$ is the bias acceleration contains the centrifugal acceleration and the Coriolis acceleration, and $\hat{c}_i = \dot{h}_i \dot{q}_i$.

$$\hat{X}_{i-1}^{i} = \begin{bmatrix} 1 & 0 \\ -\tilde{r} & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \Rightarrow \hat{X}_{i-1}^{i} = \begin{bmatrix} R & 0 \\ -\tilde{r} \cdot R & R \end{bmatrix} \tag{3}$$

Where X_{i-1}^{i} is the coordinate transformation matrix from i-1 to i, then

The algorithm of NE in screw is:

$$\begin{cases} \hat{v}_0 = 0; \hat{a}_0 = 0 \\ for \quad i = 1:n \\ \hat{v}_i = \hat{X}_{i-1}^{i} \hat{v}_{i-1} + h_i \dot{q}_i \\ \dot{h}_i = \hat{v}_i \times h_i \\ \hat{a}_i = \hat{X}_{i-1}^{i} \hat{a}_{i-1} + \dot{h}_i \dot{q}_i + h_i \ddot{q}_i \\ end \end{cases} \tag{4}$$

2.2. The forward dynamics in screw theory

According to Featherstone, there are a matrix and a vector which

$$\hat{f}_i^I = \hat{I}_i^A \hat{a}_i + \hat{Z}_i^A \tag{5}$$

Where $\hat{I}_i^A$ is the articulated body inertia of link i independent of joint velocity. $\hat{Z}_i^A$ is the articulated body zero force of link i independent of joint acceleration.

Force of any body in figure 2 can be written as:

$$\hat{f}_{i-1}^I = \hat{I}_{i-1} \hat{a}_{i-1} + \hat{Z}_{i-1} - \hat{f}_{i-1}^O \tag{6}$$

The superscript of I and O represent the inward joint force and outward force acting on body B_i separately.

$$\hat{f}_{i-1}^O = -\hat{X}_i^{i-1} \hat{f}_i^I \tag{7}$$

Substitute equation 7 in 6 and then in equation 5,

$$\hat{f}_{i-1}^I = \hat{I}_{i-1} \hat{a}_{i-1} + \hat{Z}_{i-1} + \hat{X}_i^{i-1} (\hat{I}_i^A \hat{a}_i + \hat{Z}_i^A) \tag{8}$$

Substitute equation 2 in the above equation

$$\hat{f}_{i-1}^I = (\hat{I}_{i-1} + \hat{X}_i^{i-1} \hat{I}_i^A \hat{X}_{i-1}^{i}) \hat{a}_{i-1} + \hat{Z}_{i-1} + \hat{X}_i^{i-1} (\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i + \hat{I}_i^A h_i \ddot{q}_i) \tag{9}$$

The fore $\hat{f}_i$ projecting on the axe is the drive force of joint. $Q_i = h_i^T \hat{f}_i$, combine this with equation 2 and 7,

$$Q_i = h_i^T \hat{I}_i^A (\hat{X}_{i-1}^i \hat{a}_{i-1} + \ddot{q}_i h_i + \hat{c}_i) + h_i^T \hat{Z}_i^A \tag{10}$$

From which the acceleration can be solved:

$$\ddot{q}_i = \frac{Q_i - h_i^T \hat{I}_i^A \hat{X}_{i-1}^i \hat{a}_{i-1} - h_i^T (\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i)}{h_i^T \hat{I}_i^A h_i} \tag{11}$$

Substitute equation 12 in equation 10

$$\hat{f}_{i-1}^I = \left[\hat{I}_{i-1} + \hat{X}_i^{i-1} \left(\hat{I}_i^A - \frac{\hat{I}_i^A h_i h_i^T \hat{I}_i^A}{h_i^T \hat{I}_i^A h_i} \right) \hat{X}_{i-1}^i \right] \hat{a}_{i-1} + \hat{Z}_{i-1} + \hat{X}_i^{i-1} \left[\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i + \frac{\hat{I}_i^A h_i \left[Q_i - h_i^T \left(\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i \right) \right]}{h_i^T \hat{I}_i^A h_i} \right] \tag{12}$$

Compare this with equation 7, it can be concluded that:

$$\hat{I}_{i-1}^A = \hat{I}_{i-1} + \hat{X}_i^{i-1} \left(\hat{I}_i^A - \frac{\hat{I}_i^A h_i h_i^T \hat{I}_i^A}{h_i^T \hat{I}_i^A h_i} \right) \hat{X}_{i-1}^i \tag{13}$$

$$\hat{Z}_{i-1}^A = \hat{Z}_{i-1} + \hat{X}_i^{i-1} \left[\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i + \frac{\hat{I}_i^A h_i \left[Q_i - h_i^T \left(\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i \right) \right]}{h_i^T \hat{I}_i^A h_i} \right] \tag{14}$$

Where $\hat{I}_n^A = \hat{I}_n$, $\hat{Z}_n^A = \hat{Z}_n$.

All these above equations lead to O(N) forward dynamics in screw theory:

$$\left\{ \begin{array}{l} for \quad i = 1:n \\ \quad \hat{Z}_i = \begin{bmatrix} -m_i g \\ \omega_i \times I_i \omega_i \end{bmatrix} \\ \quad \hat{I}_i = \begin{bmatrix} 0 & M_i \\ I_i & 0 \end{bmatrix} \\ \quad \hat{c}_i = \hat{v}_i \times h_i \\ end \end{array} \right. \quad \left\{ \begin{array}{l} for \quad i = n:2 \\ \quad \hat{I}_{i-1}^A = \hat{I}_{i-1} + \hat{X}_i^{i-1} \left(\hat{I}_i^A - \frac{\hat{I}_i^A h_i h_i^T \hat{I}_i^A}{h_i^T \hat{I}_i^A h_i} \right) \hat{X}_{i-1}^i \\ \quad \hat{Z}_{i-1}^A = \hat{Z}_{i-1} + \hat{X}_i^{i-1} \cdot \\ \quad \left[\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i + \frac{\hat{I}_i^A h_i \left[Q_i - h_i^T \left(\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i \right) \right]}{h_i^T \hat{I}_i^A h_i} \right] \\ end \end{array} \right. \quad \left\{ \begin{array}{l} for \quad i = 1:n \\ \quad \ddot{q}_i = \frac{Q_i - h_i^T \hat{I}_i^A \hat{X}_{i-1}^i \hat{a}_{i-1} - h_i^T (\hat{Z}_i^A + \hat{I}_i^A \hat{c}_i)}{h_i^T \hat{I}_i^A h_i} \\ \quad \hat{a}_i = \hat{X}_{i-1}^i \hat{a}_{i-1} + \hat{c}_i + h_i \ddot{q}_i \\ end \end{array} \right. \tag{15}$$

Equations 15 are forward dynamics of robots, composed by four steps:

1. Recursively computing velocity for each link from base to end.
2. Initializing $\hat{Z}_i$ and $\hat{I}_i$, computing bias acceleration $\hat{c}_i$.
3. Recursively computing articulated body inertia and zero acceleration force for each link from end to base.

4. Recursively computing the acceleration force for each link from base to end.

It can be seen that the computation complexity of above equations grows linearly with DOFs of robot, that is the algorithm is O(N). The above-mentioned algorithms can be easily extended to tree topology, closed loop and spatial robot systems.

3. Computational complexity

Table 1 shows the computation complexity of three kinds of different forward dynamics cost. The algorithms in screw and that of Rodriguez are both O(N) methods. But the SOA method presented by Rodriguez lead to a complex process of forward dynamics. The algorithm put forward by Walker and Orin is $O(N^3)$. When the DOF of a robot is N＝6, the above methods are the same. If the robot is complex of N＝17, the O(N) method is more efficient of half cost.

Tab.1 : Computational complexity of forward dynamics

Forward dynamics algorithms	Computational complexity	N =6	N =17
Walker and Orin[3]	(1/3)N3+65N2+(197/6)N-33	2567	18948
Rodriguez formula [6]	477N-503	2359	7606
Screw formula	594N-567	2997	9531

4. Three classic methods of O(N) recursive dynamics

Table 2 gives the different notations of the above three O(N) recursive dynamics for ease of understanding.

Tab.2 Notations of three O(N) recursive dynamics

O(N) recursive dynamics	SOA	ABA	Lie algorithm
Body count	$n-k-1$	k	k
Joint velocity	$\beta(k)$	$\dot{q}_k$	$\dot{q}_k$
From joint to state space	$H^*(k)$	$\hat{s}_k$	s_k
Move velocity to prior link	$\phi^*(k+1,k)$	${}_k\hat{X}_{k+1}$	$Ad_{T_{k,k+1}^{-1}}$
Move force to next link	$\phi(k+1,k)$	${}_k\hat{X}^*_{k+1}$	$Ad^*_{T_{k,k+1}^{-1}}$
Spatial velocity	$V(k)$	$\hat{v}_k$	V_k
Spatial acceleration	$\alpha(k)$	$\hat{a}_k$	$\dot{V}_k$
Spatial force	$f(k)$	$\hat{f}_k$	F_k
Rigid link inertia	$M(k)$	$\hat{I}_k$	J_k
Kalman gain	$G(k)=P(k)H^*(k)D^{-1}(k)$	$\dfrac{\hat{I}_k^A\hat{s}_k^*}{\hat{s}_k^*\hat{I}_k^A\hat{s}_k}$	$\dfrac{\hat{J}_k s_k^*}{s_k^*\hat{J}_k s_k}$
Articulated inertia	$P(k)$	$\hat{I}_k^A$	$\hat{J}_k$

Force of zero acceleration	$z(k)$	$\hat{P}_k^A$	B_k

There are three kinds of O(N) recursive dynamics, the spatial operator algebra(SOA), the articulated body algorithms(ABA) and the Lie Algebras. For inverse dynamics, these dynamical algorithms are all based on recursive Newton-Euler algorithm. It can be seen the same from this point of view. For forward dynamics, the authors of SOA recognized the analogy between Kalman filtering and robot dynamics and got the inverse of the mass matrix. The ABA used the articulated body to creatively divide the spatial force into two parts, the force related to acceleration and the force of zero acceleration. The Lie Algebras also applied the articulated body and put forward two dual Adjoint operators.

The mechanism in these three recursive methods is: movement parameters and their changes in accordance with the articulated body inertia are linked the outward and inward force. The end of the movement is the burden of low order body. The implication is the role of the relationship between the force and the reaction force. The body is linked with outside articulated body definitely which determine the basis of recursion. This relationship is an objective reality, different scholars have proposed three awareness means.

5. Conclusions

Screw is used to describe the robot dynamics in this paper. Similar to Newton-Euler(NE) formula, the inverse dynamics in screw theory is of more simple expression form and applicable to large robot systems. The O(N) recursive forward dynamics in screw can be clearly expressed by equations 15. This algorithm can be easily extended to closed loop robot systems, flexible robot systems and spaceflight robot systems. The O(N) algorithm is efficient especially for large systems. It can be seen that screw is helpful to high efficient dynamics modelling and the formulae of robot dynamics in screw are concise and clear. It also gives the different notations of three O(N) recursive dynamics for ease of understanding.

Acknowledgements

This research is partially supported by Leading Academic Discipline Project of Shanghai Municipal Education Commission (project number: J51902) and Scientific Research Project Start-up Fund of Shanghai Dianji University (project number: 11c404).

References

[1] M. E. Kahn and B. Roth, The Near Minimum time Control of Open-loop Articulated Kinematic Chains[J], Journal of Dynamic Systems, Measurement, and Control, vol. 93, pp. 164-172, 1971.
[2] Xiong Y. L, The basic of Robot Technology [M], Huazhong University of Science and Technology Press, 2002.
[3] M. W. Walker and D. E. Orin, Efficient Dynamic Computer Simulation of Robotic Mechanisms[J], Trans. ASME, J. Dynamic Systems, Measurement & Control, vol. 104, pp. 205-211, 1982.
[4] R. Featherstone, Efficient Factorization of the Joint Space Inertia Matrix for Branched Kinematic Trees [J], Int. J. Robotics Research, vol. 24, no. 6, pp. 487-500, 2005.
[5] A. Jain, G. Rodriguez, Multibody Mass Matrix Sensitivity Analysis Using Spatial Operators [J], Int. Journal for Multiscale Computational Engineering, vol.1, Issue 2-3, 2003.
[6] A. Jain, Unified Formulation of Dynamics for Serial Rigid Multibody Systems [J], Journal of Guidance, Control, and Dynamics, vol. 14, no. 3, pp. 531-542, 1991.
[7] J. García de Jalón, Twenty-five years of natural coordinates, Multibody System Dynamics [J]. vol.18, pp. 15–33, 2007.