A Programmatic Dynamics Modeling Method for Robotic Mechanical Systems Postprint

Authors: Zhuang Wei, Kang Wenjie, Huang Yonghua, Gao Zhenyu, Yang Jiwei, Huang Yonghua

Date: 2022-09-26T00:00:00+00:00

Abstract

To address the issues of complex robot dynamics models and low computational efficiency caused by redundant calculations, a Programmatic Modeling Method (PMM) is proposed. Taking the 6-DOF Stanford manipulator as an example, the dynamics model based on Lagrange' s equations is established using this method. Following the core concept of "forward analysis, reverse output," the recursive process of the model is analyzed in detail. On the basis of verifying the model' s correctness, the "size" and runtime metrics of the Stanford manipulator dynamics models established using PMM and the conventional Lagrange' s equation method in the computer are compared. Experimental results demonstrate that compared with the conventional Lagrange method, the complexity of the model established by PMM is reduced by 67.6%, and the computational efficiency is improved by 66.3%. The Stanford manipulator is a holonomic constrained system; PMM is extended to underactuated nonholonomic constrained systems. Numerical simulation and physical prototype experimental analysis are conducted using a partial feedback linearization control algorithm closely related to the model, which verifies the reliability and effectiveness of PMM and provides an efficient and strongly general dynamics modeling method for different types of robots.

Full Text

Preamble

Classification Number: TP242

Abstract

A Programmed Dynamic Modeling Method for Robot Mechanical Systems

ZHUANG Wenjie, HUANG Yonghua, ZHEN Yu, JI Wei
School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, China

This paper proposes a programmed dynamic modeling method (PMM) for robot mechanical systems. Addressing the problems of complex dynamics models and low computational efficiency caused by redundant calculations in robotic systems, this method establishes a dynamics model based on Lagrange equations. Using a six-degree-of-freedom Stanford manipulator as an example, the model complexity and computational efficiency are analyzed, with key metrics including model dimensionality and runtime. Experimental results demonstrate that compared with conventional Lagrange methods, the proposed model reduces complexity by 67.6% and improves computational efficiency by 66.3%. The Stanford manipulator represents a holonomically constrained system. To extend the method to underactuated nonholonomic constrained systems, numerical simulations and physical prototype experiments are conducted using a partial feedback linearization control algorithm closely related to the model. The reliability and effectiveness of the programmed modeling method are verified, providing an efficient and versatile dynamics modeling approach for different types of robots.

Keywords: dynamics; modeling; programmed; output expansion; computational efficiency; robot

1. Introduction

Complex system dynamics models form the foundation for motion control, while the modeling approach largely determines model accuracy and computational efficiency, thereby affecting control performance. This has attracted widespread attention from scholars. Dynamics modeling methods for robots primarily fall into two categories.

The first approach analyzes system force and torque vectors, belonging to the Newtonian classical mechanics framework. Kvrgic et al. [1] consider this method highly effective for deriving manipulator dynamics models, applying it to a six-degree-of-freedom robot and comparing its computational efficiency with other approaches. Wang Jian et al. [2] established a nonlinear model for quadrotor aircraft based on forward kinematics analysis and inverse-order output of system dynamics models, implementing dual-loop control. Zhao Limei [3] and Ghariblu et al. [4] employed this approach to develop dynamics models for snake robots and novel mobile spherical robots, respectively, achieving model-based motion control studies. However, traditional methods calculate dynamics equation elements step-by-step from system generalized coordinates, essentially performing symbolic computation that iterates variable expressions layer by layer. This can lead to model output expansion and redundant calculations, causing low efficiency. While some techniques have been developed to address these issues, they cannot completely guarantee the most concise model or prevent problems like variable redefinition and sequencing errors.

The second approach analyzes system energy and work, belonging to analytical mechanics. Methods based on Lagrange equations or Euler-Lagrange equations are typical examples. Zhang et al. [13] used Lagrange equations to model a six-degree-of-freedom manipulator and completed dynamics simulation in ADAMS. Ruan et al. [14] utilized Euler-Lagrange equations for unicycle robot modeling, achieving straight-line and curved motion control. Li Mengfei [15] established a Stanford manipulator dynamics model based on Lagrange equations and verified its correctness. These applications demonstrate Lagrange equation methods across various manipulators and multi-degree-of-freedom robots [16-19]. However, energy analysis methods can suffer from dimensionality curse, bringing repeated calculations and low operational efficiency. Sometimes the most reasonable output cannot be guaranteed during actual modeling.

To solve these problems, this paper proposes a programmed modeling method (PMM) that systematically eliminates redundant variables and implements variable sorting algorithms.

2. The Programmed Modeling Method (PMM)

2.1 Core Concept

The core idea involves forward analysis and inverse-order output. Forward analysis begins with generalized coordinates, proceeding through velocity analysis, motion constraint analysis, etc., and terminates at the system' s kinetic energy function. The key is establishing variable dependency relationships after introducing intermediate variables. Inverse-order output starts from partial derivatives of kinetic energy with respect to generalized coordinates or velocities, propagating backward through variable dependency relationships according to the chain rule in reverse order, ultimately deriving the most primitive partial derivative terms.

2.2 Variable Definition and Forward Analysis

The forward analysis process defines system variables systematically. For a system with components Bi (i = 1~n), the variables include: - qi: generalized coordinates (joint angles or displacements) - q•i: generalized velocities - ωi: angular velocities - vi: center-of-mass linear velocities - Position vectors described in coordinate frames e(Bi) - Rotation transformation matrices between frames - Gravity acceleration vector

The forward analysis calculates these variables sequentially from base to end-effector using rigid body kinematics principles.

2.3 Inverse-Order Output Process

The inverse-order output process begins with kinetic energy T. By expressing T as a sum of products of coefficients and velocity terms, intermediate variables ul (l = 1~m) are introduced. The kinetic energy becomes a function of these intermediate variables: T = T(ul(q • n, q • k)). The partial derivatives T/ qi and T/ q • i are then computed in reverse order through the variable dependency chain, avoiding model expansion and redundant calculations.

3. Stanford Manipulator Modeling and Simulation

3.1 System Description

To intuitively demonstrate PMM and verify its efficiency, we establish a Stanford manipulator dynamics model [11]. The manipulator consists of six components Bi (i = 1~6), connected through sliding and rotating joints. Coordinate frames e(i) (i = 0~6) are defined, with e(0) as the ground frame. The generalized coordinates are qi (i = 1~6), representing joint angles or displacements.

3.2 Forward Analysis

Using robotic transformation formulas, the rotation matrices between coordinate frames are calculated. For example, the transformation from e(0) to e(4) involves successive rotations about different axes. Angular velocities ωi are computed recursively, and linear velocities of each link' s center of mass are derived based on the velocity of preceding links.

3.3 Kinetic and Potential Energy

The total kinetic energy T comprises rotational and translational components for each rigid body: - Rotational kinetic energy: (1/2)ωi^T Ji ωi - Translational kinetic energy: (1/2)Mi vi^T vi

where Ji is the inertia matrix and Mi is the mass of link Bi. The potential energy V is calculated based on the height of each link' s center of mass relative to a reference plane.

3.4 Dynamics Model

Using Lagrange equations, the dynamics model is expressed as: τi = d/dt(T/ q• i) - T/ qi + V/ qi

where τi represents joint driving torques. The inverse-order output process computes the required partial derivatives efficiently through the intermediate variable structure.

3.5 Implementation and Comparison

The model is implemented in Mathematica with automatic format conversion scripts for controller design in MATLAB/Simulink. Simulation experiments compare PMM with conventional Lagrange methods. Running on an industrial PC (2.0 GHz CPU, 2 GB RAM, Win7 OS), the PMM model shows: - 57.9% reduction in variable count - 67.6% decrease in model complexity - 66.3% improvement in computational efficiency

The simulation curves match those from ADAMS and literature [15], verifying model correctness.

4. Validation on Underactuated Nonholonomic Systems

4.1 Two-Wheeled Robot Application

To demonstrate broader applicability, PMM is extended to underactuated nonholonomic systems. A variable-structure two-wheeled robot, which is statically unstable but dynamically stable, serves as a test case. The system consists of a vehicle body B1 and wheels B2, B3. The dynamics model is established using PMM and integrated with a partial feedback linearization controller.

The control law uses the pitch angle q3 as input: v = kp(q3 - q3d) + kd(q · 3 - q · 3d)

where v is a virtual control variable, and DD, EE are model-dependent parameters. The wheel torque is computed as τw = DD · v + EE.

Physical prototype experiments demonstrate that the pitch angle and wheel torque converge within finite time, achieving balanced stance. The model runs on a DSP platform (TMS320F28335, 150 MHz) with a servo period of 5 ms, occupying only 0.04% of the servo cycle, leaving ample time for data acquisition and control computation.

4.2 Unicycle Robot Application

Further validation is performed on a unicycle robot for straight-line positioning control. The robot' s roll and pitch angles stabilize within $±$0.05 rad, with y-direction displacement variation of only $±$0.005 m, confirming straight-line motion. The positioning error is minimal, achieving the control objective. Running on the same industrial PC platform, the model execution time occupies merely 0.05% of the 5 ms servo cycle, meeting real-time requirements.

5. Conclusion

This paper proposes a programmed dynamic modeling method (PMM) and demonstrates its effectiveness through Stanford manipulator modeling and simulation. Compared with conventional Lagrange methods, PMM reduces model complexity by 67.6% and improves computational efficiency by 66.3% when implemented on a 2.0 GHz CPU platform. By extending the method to underactuated nonholonomic systems (two-wheeled and unicycle robots) through numerical simulation and physical prototype experiments, the method' s reliability and versatility are verified. The dynamics models satisfy real-time computation requirements for various physical prototypes and experiments, demonstrating PMM' s characteristics of high efficiency, broad applicability, and strong versatility. This provides valuable guidance for dynamics modeling in robotics.

References

[1] KVRGIC V, VIDAKOVIC J. Efficient dynamics forward computation[J]. Mechanism and Machine Theory, 2020, 145: 103680.

[2] WANG Jian, et al. Research on dynamics modeling and control method for snake robots[J]. Journal of Guilin University of Electronic Technology, 2018, 38(5): 406-411.

[3] ZHAO Limei. Hovering control of quadrotor aircraft based on adaptive fuzzy algorithm[J]. Beijing University of Chemical Technology, 2016.

[4] GHARIBLU H, et al. Dynamic modeling of mobile robot[J]. Applied Mathematical Modelling, 2015, 39(10-11): 3103-3115.

[5] LUH J Y S, WALKER M W, PAUL R P C. On-line computational scheme for mechanical manipulators[J]. Journal of Dynamic Systems, Measurement and Control, 1980, 102(2): 69-76.

[6] RACKL W, LAMPARIELLO R, ALBU-SCHAFFER A. Dynamic modeling and control of a front-wheel drive bicycle[C]//2010 IEEE International Conference on Artificial Intelligence and Computational Intelligence. Los Alamitos, CA: IEEE Computer Society, 2010: 546-551.

[7] HUSSAIN Z, AZLAN N Z. KANE's method for robotics[J]. The International Journal of Robotics Research, 1983, 2(3): 3-21.

[8] HUANG Qizheng, WEI Yonghua, LIAO Shimin, et al. Research on dynamics of front-wheel drive bicycle using screw theory[C]//2010 IEEE International Conference on Artificial Intelligence and Computational Intelligence. Los Alamitos, CA: IEEE Computer Society, 2010: 546-551.

[9] HUSSAIN Z, AZLAN N Z. KANE's method for robotics[J]. The International Journal of Robotics Research, 1983, 2(3): 3-21.

[10] HUANG Qizheng, et al. Dynamics modeling of three-degree-of-freedom mechanism using KANE' s method[J]. Mechanical Science and Technology, 2010, 29(8): 1035-1041.

[11] KANE R, LEVINSON D. Dynamical equations for robot manipulators[J]. The International Journal of Robotics Research, 1983, 2(3): 3-21.

[12] ZHUANG Weijie, et al. Modeling and analysis of a spherical robot with telescopic arm[J]. Mechanical Science and Technology, 2010, 29(8): 1035-1041.

[13] ZHANG Qin, JIA Qingxuan, SUN Hanxu, et al. Application of genetic algorithm-based controller for spherical robot[C]//2009 IEEE International Conference on Control and Automation. Piscataway, NJ: IEEE Press, 2009: 180-184.

[14] RUAN Jingmin, WANG Xiaogang, HU Qiyuan. Modeling and analysis of unicycle robot using Euler-Lagrange equation[C]//2009 Second International

Conference on Intelligent Computation Technology and Automation. Piscataway, NJ: IEEE Press, 2009.

[15] LI Mengfei. Dynamics modeling and simulation of 6-DOF industrial manipulator[J]. Science and Technology Innovation, 2019(25): 39-40.

[16] BOUGHDIRI R, BEZINE H, NASSER H, et al. Dynamic modeling and grasping control task for multi-fingered robot[C]//International Symposium on Intelligent Robotics and Sensors (IRIS 2012). Piscataway, NJ: IEEE Press, 2012, 41: 923-931.

[17] SAFEEA M, NETO P, BEAREE R. Robot dynamics: efficient calculation algorithm with symbols[J]. Mechanism and Machine Theory, 2019, 142: 103589.

[18] XING Li, WANG Xiaofeng, WANG Jianhui. A simplified Lagrange dynamic modeling method for multi-DOF robot[J]. Journal of Intelligent & Robotic Systems, 2016, 31(4): 2393-2401.

[19] WANG Yonghua. Design and motion control research of variable-structure two-wheeled robot[D]. Guilin University of Electronic Technology, 2017.

[20] WANG Yonghua, et al. Straight-line positioning control of unicycle robot[J]. Mechanical Design and Research, 2020, 36(6).

Note: Figure translations are in progress. See original paper for figures.