'86韓國自動制御學術會議 論文集 VOL.1 1986. 10

컴퓨터를 이용한 로보트 설계 소프트웨어 팩키지 개발

오 세영**, °김 호연*, 강 영국*
플로리다 대학**, 대우중공업(주) 기술연구소*

CARDS : A COMPUTER-AIDED DESIGN SOFTWARE PACKAGE FOR ROBOT MANIPULATORS AND THEIR CONTROLLERS

SE-YOUNG OH**, HO-YOUN KIM*, YOUNG-GOOK KANG*
UNIVERSITY OF FLORIDA**, DAEWOO HEAVY INDUSTRIES*

Abstract

A software package CARDS for general purpose robot design, control, and simulation has been developed and is presented here. CARDS (Computer Aided Robot Design and Simulation) consists of a collection of standardized subroutine modules that carry out typical kinematic, dynamic, and control computations so that the user only needs to write a main program that further defines a particular robot configuration and the task to be performed.

It provides users a complete simulation environment, so that it will be a valuable engineering tool for mechanical designers as well as electric control designers.

I. Introduction

The next decade will bring the age of intelligent and high performance robot. High performance is predicated primarily on fast and accurate motion, and consequently high productivity.

The majority of existing industrial robots must be taught their path either by manual guidance or by a teach pendant. Recent developments, however, have enabled some advanced robots to employ the inverse kinematic analysis [1] so that each joystick can directly control movements in Cartesian workspace. Using the trajectory obtained through inverse kinematics, an independent servo is closed around each joint to reduce the position and/or velocity errors. Although such a control scheme will work for relatively slow motions, it is not capable of controlling dynamics at higher speeds, thus alone would still limit productivity. Dynamic analysis which computes the forces required to drive the robot through a desired motion can therefore be employed. These forces can be added as bias terms to the joint servos to compensate for changing dynamics. From this, the relationship between robotic servo design and its kinematic and dynamic analyses is evident. As far as the authors know, other simulation system such as IMP[2] and ADAMS [3] exist. However, these systems were developed primarily for mechanical designers of complex closed-chain system but not specifically for robotic system. CARDS differs from these two in that it has been designed specifically robotic systems and therefore it further includes algorithms for trajectory planning, servo control and performance evaluation routines. It will define a robot's motion trajectory, compute the equivalent joint space trajectory and the necessary joint torques for that trajectory, apply these torques through a control law to the simulated robot, and evaluate the control performance.

II. Construction for the simulation package

When developing computer programs for the simulation and control of robotic systems, it is a very inefficient, time-consuming, and error-prone process to work out a separate program for each mechanism or task considered . It is convenient to have available standardized programming modules that perform typical computations required for most of the kinds of tasks and mechanisms to be considered.

In general, the following requirements should be met for the modules described above. First, the program code should not change from mechanism to mechanism. Rather, each mechanism to be considered should be parametrized and this data furnished to the program. Second, the modules should be flexible enough so that the user can select from the package only those modules needed to satisfy his particular requirements. Third, the input and output data structures and data transformation functions of each module must clearly be defined so that the user can easily prepare the data and parameters in a form compatible with the modules without the need to know any details of the program. Finally, the package must simultaneously include all three design aspects: mechanical design, motion trajectory design, and servo control design. For it is the combination of all these that influence the overall performance.

1. Typical Configuration of Modules

The software will consist of a set of general purpose standardized subroutine modules which may be assembled along with the main program. This main program will allow a mechanical specification of the robot to be simulated and its different application tasks. Its user would not be required to derive any kinematic or dynamic equations or even to understand the basic mechanical principles governing robot motion.

The algorithmic structure is shown in Figure I which describes the various computational modules required for a typical dynamic computer simulation environment for control law or algorithm development. As trained or commanded by a human operator or as automatically generated through programmed algorithms, the complete trajectory is furnished by the motion planning section. This motion is then transformed into joint coordinates through inverse kinematics. Further, from this joint motion, the rigid body equations of motion are formulated. The inverse dynamic analysis is then carried out to determine the necessary joint torques(or forces) to drive a robot through its desired trajectory for open chains.

Through a control law, thus calculated joint torques are applied through the actuators to the mechanism. The direct dynamic equations simulate the actual robot's dynamic behavior by formulating the joint accelerations given a particular joint torque set and a system state. This allows testing of different control laws without building the actual robot hardware. Finally through integration, the actual trajectory is determined and the control performance will be evaluated.

2. Detailed description of modules

For kinematic models for an arbitrary geometric configurations, the Denavit-Hartenberg convention is used. Link coordinates are attached to each link and their relations are described with four parameters, namely link length, joint offset, twist angle, and the joint angle. Based on this model, a complete set of kinematic equations are derived[4]. For dynamic analysis, however, the mass and inertia properties need to be further specified. In addition, the location of the center of mass needs to be defined. The center of mass position and the inertia are constant when referred to their own link coordinates[4].

1) Motion Planning Subroutine

This translates the input task into a time sequence of end-effector positions, velocities, and accelerations in Cartesian workspace. Different joint interpolation schemes can be used accordingly, their influence upon control performance may be evaluated.

2) Inverse Kinematic Subroutine

This performs the coordinate transformation from the Cartesian end-effector workspace into the joint articulation space. All three transforms, position, rate, and acceleration are necessary to define a complete motion. A numerical solution to this problem [1] must be used to simulate different robots as opposed to special solutions. The motion output from this subroutine is used as input to the following Inverse Dynamics Subroutine.

3) Inverse Dynamic Subroutine

Given a joint space trajectory as calculated from Inverse Kinematics, this computes the joint torques required to drive the joint actuators through a desired trajectory for an open-chain mechanism.

4) Digital Control Subroutine

This subroutine utilizes various custom-designed linear and nonlinear control techniques. Its output generates the corrective control torques in the face of noise and disturbances.

5) Direct Kinematic Subroutine

Given the joint space trajectory, this computes the robot trajectory in Cartesian workspace. As shown in Figure 1, this trajectory is used to graphically display motion of the robot to test and evaluate its dynamic performance or for workspace analysis.

6) Direct Dynamic Subroutine

This virtually replaces the actual robot with its dynamic computer model. It achieves this by computing the motion trajectory as a function of the given input joint torques as commanded by the Digital Control Subroutine

7) Control Performance Evaluation Subroutine

This subroutine calls the Digital Control, Direct Dynamic, Direct Kinematic, and Graphic Display Subroutines to observe and test the performance of a particular combination of the trajectory and servo control design.

8) Graphic Display Subroutine

This dynamically displays the robot motion in Cartesian space through computer graphics.

Ⅲ. Simulation and Model Accuracy

In this section, the use of CARDS in a real world situation is discussed. Since simulation is performed on a computer model of the system, its results will be only as useful as its model accuracy. When the program is used for design, the problem disappears by the nature of design: an appropriate model is sought to guide its implementation. However, when the package modules are used to control a real manipulator, it is very important to have a good model of every component of the system that is relevant for designing the controller. This is an impossible task. For example, dynamic parameters, especially due to the acuators and associated components. To add to these difficulties, the following uncertainties also exist.

1) Actuator modeling
2) Discretization of analog circuitry
3) Gear backlash, friction, link flexibility

Despite all these real world problems, the use of simulation for control design still gives some valuable insights into various factors affecting the overall performance. To put in easy terms, even if the model is only 90% accurate, still the same amount of efforts has been saved. 90% accurate answer is far better than no answer at all. Furthermore, for design, simulation is an essential step and can save a great deal of efforts later on.

Ⅳ. Summary and Conclusion

CARDS is a user-friendly software package for general purpose robotic design and simulation. Users will be able to reduce their robotic control design and simulation efforts substantially through use of this package. The standardized subroutine modules will perform all typical robotic computations so that the user does not have to write them individually. Using CARDS will greatly reduce the turn around time between design and testing. More importantly, the user will be able to devote important time to designing and iterating on different controllers, testing them for different trajectories, and repeating all these even for different robots. A graphic display will be an invaluable component of the system providing the user with an immediate visual feedback of the control performance.

REFERENCES

[1] S. Y. Oh, D.E. Orin, and M. Bach, "An Inverse Kinematic Solution to Kinematically Redundant Robot Manipulators," Journal of Robotic Systems, Vol.1, No.3, PP. 235-249, September 1984.

[2] P. N. Sheth and J.J. Uicker, Jr., "IMP(Integrated Mechanisms Program),a computer-aided design analysis system for mechanisms and linkage." Trans. ASME, Journal of Engineering for Industry, pp.454-464, May 1972.

[3] N. Orlandea, D.A. Calahan, and M.A. Chace, "A sparsity-oriented approach to the dynamic analysis and design of mechanical systems--Part 2," Trans. ASME, Journal of Engineering for Industry, pp.780-784, August 1977.

[4] M.W. Walker and D.E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms, " ASME Journal of Dynamic Systems, Measurement and Control, Vol. 104,pp.205-211, September 1982.

Figure 1. Robotic Simulation and Control
Block Diagram