Manipulator control using the configuration space method

M.H.Raibert, California Institute of Technology and B.K.P.Horn, Massachusetts Institute of Technology, U.S.A.

SUMMARY

The throughput of a manipulation process depends upon the arm's speed of operation, but many existing controllers provide accurate trajectory control only at low or moderate velocities. We propose a control method that explicitly compensates for configuration-dependent gravity, acceleration, and velocity forces - the latter being especially important during rapid simultaneous motions of a number of joints. A tabular form of the equations of motion is used in real-time in conjunction with a configuration space memory organized by positional variables. The contents of the memory are precomputed only once for each manipulator and are usable for all possible movements. A planned implementation of this method for the Stanford Schienman arm that uses about 250K memory locations and requires about $n^3 + 3n^2$ arithmetic operations per evaluation is discussed, where n is the number of degrees of freedom of the device.

INTRODUCTION

The application of industrial manipulators to parts transfer and assembly operations is still limited by their high cost. A useful figure of merit for such devices is the ratio of the number of operations they can perform per unit time to their cost. The more cycles the manipulator performs per unit time, the more rapidly it will pay back the investment(2). There is, we believe, a threshold for this figure of merit above which the application of manipulation to a wide variety of new tasks becomes economically feasible. If the figure of merit were to rise above this threshold, the increase in feasibility would make mass production of manipulators possible resulting in further drops in unit cost and yet wider application.

It is unlikely that this revolutionary sequence of events will be triggered by a reduction in the cost of manipulators, since the technology for building reliable devices in the numbers now used appears to be fairly stable. It is possible, however, to decrease task cycle time with equally dramatic results. Decreasing cycle time means increasing the rate of manipulation - the speed the arm moves during transfer and during manipulation. There are two areas that present good opportunities for improving this manipulation speed:

Stability consideration limit top speeds during "gross" motion for a number of large manipulators. If an arm is moved too quickly non-linearities and time varying parameters make control quite difficult. Since these problems are accentuated at high velocity (when servo bandwidth is lower relative to the frequency of changes in the state variables, and when dynamic terms have more effect), stability now is maintained by operating at lower speeds.

Workers at the Charles Stark Draper Laboratory found that about 60% of the automatic assembly cycle period is spent making "fine" motions (7). These motions are made slowly because high precision is required, and highly precise trajectory control is difficult to achieve at high velocities. The reasons are similar to those cited above: the effects of non-linear inertial terms and velocity interactions are minimized at low speeds.

In this paper we present a method for evaluating the equations of motion that makes real-time compensation possible for ALL important terms. Our goal is to improve the quality of control for gross and fine manipulator motions so that they can be executed with greater speed - in less time. Stability at high velocity becomes less of a problem once a nominal control trajectory is chosen. Fine motions can be made rapidly without sacrificing precision because velocity dependent terms, non-linear inertial terms, and inertial interaction terms are no longer left uncompensated.

Better planning of the assembly station, optimization of the required assembly steps, placement of fixtures, and choice of tools can also contribute to reductions in assembly cycle time, but these problems are not addressed here.

Before discussing the Configuration Space Method (CSM) for manipulator control, we present a brief review of the manipulator control problem and a few comments on previous approaches.

THE PROBLEM

Presently manipulator controllers typically employ simple fixed-analog servo loops closed separately around each degree-of-freedom. Though suitable for control of a set of independent second-order systems with fixed inertias and damping, such control is not appropriate for devices with non-linear, time-varying behaviour. Performance is adequate at low speeds provided the actuators are strong enough and the properties of the device do not change too dramatically with configuration.

If we look at equations of motion that relate the trajectory followed by a manipulator to the torque applied to each of its joints, we see that control of such a device is more complicated than for simple one degree-of-freedom second-order systems. Neglecting friction, one obtains for a device with n degrees of freedom (8):

$$t_i = \sum_{l=i}^{n} [m_l \, G \, U_{l\,i} \, R_l] + \sum_{l=i}^{n} \sum_{k=1}^{l} \mathrm{Tr}\,[U_{lk} H_l U_{l\,i}^T] \, . \, \ddot{\theta}_k$$
$$+ \sum_{k=1}^{n} \sum_{m=1}^{n} \sum_{l=i}^{n} \mathrm{Tr}[U_{lkm} H_l \, U_{l\,i}^T] \, . \, \dot{\theta}_l \, . \, \dot{\theta}_k \qquad \text{(Eq.1)}$$

where:

t_i	is the motor torque acting on the i'th joint.
m_i	is the mass of the i'th link.
G	is the gravity vector.
R_i	is the center of mass vector for the i'th link.
H_i	describes distribution of masses in the i'th link.
U_{il}	describes the kinematic relationship between the i'th and j'th links.
$\dot{\theta}_i, \ddot{\theta}_i$	are the velocity and acceleration of the i'th joints.

The coefficents of $\ddot{\theta}_i$ in this expression are not constant but vary, indicating changing moments of inertia. A desired response can no longer be achieved by selecting fixed values for the feedback gains, since no single value will apply to all inertial situations. One sees that there are terms containing $\ddot{\theta}_k$ in the expressions for t_i, when $i \neq k$. This cross-coupling means that torques applied to one single joint can cause accelerations at other joints in the arm. Finally, one sees numerous centrifugal and Coriolis force terms that are multiples of products of angular velocities. These terms, though insignificant at low angular rates, can become quite large when more than one manipulator joint is moved simultaneously at moderate and high velocities[5].

Despite these complications of the control problem, the vast majority of manipulators are presently controlled using simple servo controllers that are closed separately around each degree-of-freedom. Though these systems may perform adequately over a limited range of situations, high performance trajectory control is usually not achieved.

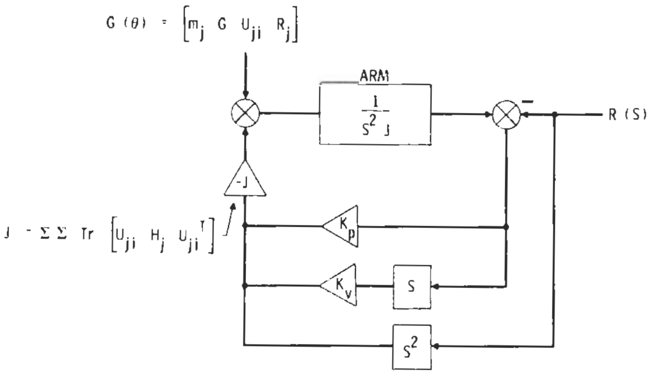
A number of researchers at the Stanford Artificial Intelligence Laboratory took a bold step to solve this problem. They attempted to evaluate certain of the terms in the equations of motion in order to deal with the complications outlined above [4, 8, 9]. Their scheme was to combine a closed-loop controller with nominal control signals computed on the basis of the equations of motion. Fig.1 shows the control scheme they implemented. Here are its important features:

Compensation is provided for time varying gravitational forces.

The feedback gains are adjusted throughout each movement to reflect changes in moment of inertia.

An acceleration feedforward term is included to compensate for changes made to the reference point.

Unfortunately, their implementation suffers from computational distress. Calculation of the coeffients shown in Eq.1 requires evaluation of trigonometric expressions having hundreds of terms. (See Appendix A of Ref.4). Therefore, as a concession to these computational difficulties, the Stanford control algorithm makes no provision for the off-diagonal moment of inertia terms that mediate interactions between joint, and provides no compensation for the numerous centrifugal and Coriolis forces that are generated when the manipulator is not stationary. Furthermore, the terms that are included require so much computation that they have to be pre-computed off-line. This means that the trajectory of every motion has to be known in advance, and a compile-and-execute paradigm is required. Further, significant departures from the planned trajectory cannot be tolerated.



$G(\theta) = [m_j \; G \; U_{ji} \; R_j]$
ARM
$\frac{1}{S^2 J}$
R (S)
-J
$J = \Sigma\Sigma \; Tr \; [U_{ji} \; H_j \; U_{ji}^T]$
K_p
K_v
S
S^2

Fig.1 Block diagram of the Stanford controller. From (Ref.8)

THE CONFIGURATION SPACE METHOD

We rewrite Eq.1 in a form given by Bejczy [1]:

$$t_i = g_i(\underline{\theta}) + \sum_{l=1}^{n} j_{il}(\underline{\theta}) \cdot \ddot{\theta}_l + \sum_{l=1}^{n} \sum_{k=1}^{n} c_{ilk}(\underline{\theta}) \cdot \dot{\theta}_l \cdot \dot{\theta}_k \tag{Eq.2}$$

where:

t_i	is the motor torque acting on the i'th joint.
g_i	is the gravitational force acting on the i'th joint.
J_{il}	is the inertial force acting on the i'th joint as a result of accelerations of the l'th joint.
c_{ilk}	are velocity dependent forces acting on the i'th joint as a result of rotations about the l'th and k'th joints.

These n equations can also be written in matrix form:

$$T = G(\underline{\theta}) + J(\underline{\theta}) \cdot \ddot{\underline{\theta}} + \begin{bmatrix} \dot{\underline{\theta}}^T \cdot C_1(\underline{\theta}) \cdot \dot{\underline{\theta}} \\ \dot{\underline{\theta}}^T \cdot C_2(\underline{\theta}) \cdot \dot{\underline{\theta}} \\ \cdot \\ \cdot \\ \dot{\underline{\theta}}^T \cdot C_n(\underline{\theta}) \cdot \dot{\underline{\theta}} \end{bmatrix} \tag{Eq.2b}$$

where:

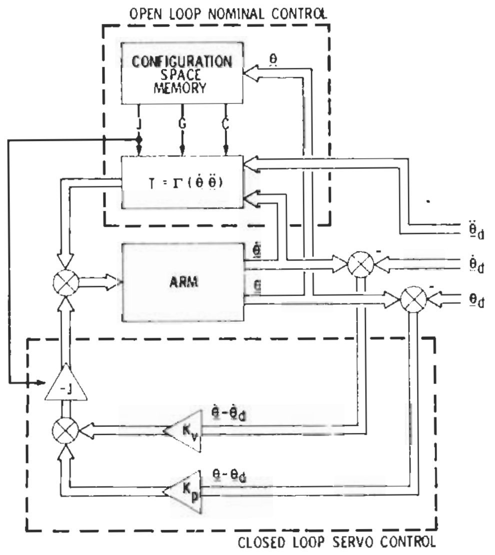
T and G are n-vectors.

J and each C_i are nxn matrices

These equations are quite complicated. In fact most of the computational difficulties encountered in evaluating manipulator equations derive from the kinematic relationships expressed here. The point to notice is that each of the coefficients in these equations, G, J, and the C_i's, is only dependent on the configuration of the arm, the position vector, 0. Rather than compute these coefficients each time they are needed, our approach is to look them up in a pre-defined, multi-dimensional memory organized by these positional variables.

We define a space that has one dimension for each joint of the manipulator. Each point in this space corresponds to a configuration, that is, a set of values for the elements of the arm's position vector, and therefore, to a set of values for the coefficients of Eq.2. If we create a multi-dimensional memory that corresponds to this configuration space, then the values of these coefficients can be computed and stored for future use.

Any time the equations of motion are to be evaluated the position vector is measured and used as an index into the configuration space memory. The coefficients appropriate to the current configuration of the arm are retrieved from the indexed region of the memory and used in Eq.2 to compute nominal control torques. These nominal control signals are used in combination with a low gain servo. (See Fig.2).



OPEN LOOP NOMINAL CONTROL
CONFIGURATION SPACE MEMORY
$\underline{\theta}$
J
G
C
$T = \Gamma(\dot{\theta}\,\ddot{\theta})$
ARM
$\dot{\underline{\theta}}$
$\underline{\theta}$
$\ddot{\underline{\theta}}_d$
$\dot{\underline{\theta}}_d$
$\underline{\theta}_d$
-J
$\dot{\underline{\theta}}-\dot{\underline{\theta}}_d$
K_v
$\underline{\theta}-\underline{\theta}_d$
K_p
CLOSED LOOP SERVO CONTROL

Fig.2 Block diagram of the configuration space controller

For this scheme to be workable the memory must be of finite size. Therefore, the index is computed by quantizing each element of $\underline{\theta}$ into one of M levels. This causes the entire memory to be subdivided into a large number of small n-dimensional hyper-regions. Then, assuming for simplicity that M is the same for each dimension, the M^n sets of coefficients G, J, and C are stored in memory, one set per hyper-region. Selection of a value for M depends on the particular application, but we estimate that values ranging from 5 to 12 will be adequate in most cases.

COMPUTATIONAL AND STORAGE COSTS

Nominally there are n dimensions in configuration space. However, the J and C terms do not depend on the position of the first joint; that connecting the arm to mechanical ground. When the axis of this joint is parallel to the gravity vector, (commonly the case for manipulators now in use), the G term is also independent of θ_1. Since these coefficients do not vary with changes in θ_1, there is no reason to store duplicate values along that dimension of the memory. Consequently the size of the memory can be reduced by a factor of M - - M^{n-1} sets of coefficients are required.

Another important reduction in storage requirements is sometimes possible. When the distribution of important masses in the highest numbered link, the hand, has circular symmetry about its axis of rotation, the equations of motion do not depend on that link's position, θ_n - - M^{n-2} sets are required.

If both of these conditions are true, the stored tables need only be of dimension n-2. Even when both conditions are not true a certain amount of storage savings can be achieved by using two tables; one for J and C of dimension n-1 or n-2, and another for G of dimension n or n-1.

The next question is, how much memory is needed to store a single set of coefficients; those corresponding to one hyper-region of the configuration space? Nominally there are $n + n^2 + n^3$ coefficients in Eq.2. But due to symmetry relationships each J and C_i only requires n (n + 1)/2 different coefficients (5). Another relationship indicates than n (n - 1)/4 coefficients in the C matrices are duplicates, and need not be stored. In general, only about $(1/4)n^3 + (5/4)n^2 + (3/2)n$ coefficients are needed to represent Eq.2. In addition to these simplifications, however, there are often special relationships that apply to the particular arm under consideration, and these relationships can result in further reductions. Such additional terms can be identified when Eq.1 is used to find the coefficients shown in Eq.2. (Eg. 25 additional terms can be dropped when writing the equations for the Stanford Scheinman arm.)(5).

So far the discussion has centered on eliminating coefficients that are identically zero, but there are also terms that make no significant contribution to the control torque, while still being non-zero. The configuration space memory can be tailored to include or exclude any of the G, J, or C coefficients, depending on their importance. Therefore, the system is suited to the dynamic properties of a particular arm and application.

Since the number of dimensions in the memory, the number of quantized levels along each dimension of the memory, and the exact number of coefficients to be stored in each hyper-region of the memory are factors that can only be determined after a particular manipulator has been selected, we cannot make precise statements of how much memory is required to implement CSM. In an initial implementation for the Stanford Scheinman arm we plan to use 4-dimensional memory, (n - 2), with about 60 coefficients stored per hyper-region. Setting M = 8 will require a memory of about 250K words of 12 or 16 bit memory.

The amount of computation required per evaluation of the equation is, taking the symmetry relations into account, $n(n + 1)/2 + n$ multiples and adds per joint, or $n^3 + 3n^2$ operations. Of course, this too is an upper limit, since non-zero coefficients that are excluded to reduce storage do not have to be included here. Notice that no trigonometric evaluations are required.

DISCUSSION

Actually, our proposal for manipulator control is a modification of the Stanford approach(8). Rather than use an analytic form for the equations of motion however, we suggest that a semi-tabular form of these equations be employed in connection with a pre-computed, multi-dimensional table. Because the amount of run-time computation is reduced, more terms from the equations of motion can be included and the compile-and-execute paradigm is avoided. The combined use of a nominal control signal plus low gain closed-loop control is retained.

By using a configuration space memory in conjunction with Eq.2, CSM avoids the computational demands that dominate evaluation of analytical equations. At the same time, the extremely large memories associated with total reliance on table look-up are not required. We feel that the method outlined here provides an attractive compromise between computational and storage costs(10).

The amount of memory required by the Configuration Space Method is large when compared to the size of main memory found in today's mini and micro computers. Disc storage devices with multi-megabyte capacities are common, but are rather expensive by standards of computers used for industrial control applications. However, the cost of solid state memory is dropping dramatically, and bubble memory technology is still improving. In 1978 the cost of a configurations space memory may only be feasible for laboratory use. But we are quite confident that memories of suitable size and speed will be available for substantially less than $1000.00 in the next year or two.

Before CSM can be used, values must be pre-computed for the tabular memory. Since these computations only have to be done once for each manipulator, the premium is placed on ease of programming rather than computational efficiency. We have implemented a preliminary coefficient generator program using a LISP-based symbolic manipulation program, MACSYMA(6). This program accepts a standardized description of the manipulator having sliding and/or revolute joints, and generates expressions for all of the coefficients described above. The entire program is only about one page of code.

The CSM can increase maximum velocity when stability control problems related to uncompensated time-varying inertial terms or non-linear velocity dependent terms are at fault. This is usually the case for large arms having powerful actuators and large link masses. Little improvement in top speed is expected for smaller arms where maximum actuator torque and friction are dominating factors. Limitations in trajectory accuracy are unlikely to be limiting factors for gross motions, since they are usually not made when the hand is near obstacles. Small deviations in the path are then of little consequence.

While many workers think that speeding up the manipulator means making fast motions faster, we are emphasizing the effect of speeding up slow motions. These "fine" motions should be especially fertile ground for improvement, since they comprise about 60% of the assembly cycle time, while "gross" motions only take up about 25%(7).

CONCLUSIONS

Economic factors recommend the development of techniques that will increase manipulation speed for both gross and fine motions. Present controllers are limited by their inability to deal with the time-varying and non-linear terms found in the manipulator's equations of motion. This results in limitations on maximum velocity for certain large manipulators, and limitations on fine motion requiring precise trajectory control.

The Configuration Space Method is a means of evaluating the equations of motion for a manipulator that allows all important terms to be included. Since computational costs and storage costs are balanced, all calculations can be done in real-time and all storage requirements are affordable.

ACKNOWLEDGEMENTS

The authors would like to thank Matt Mason, Tomas Lozano Perez, and Nancy Cornelius for their helpful comments. This report describes research done in part at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643. Supplementary support was provided by the Jet Propulsion Laboratory, California Institute of Technology, under NASA contract NAS 7-100.

REFERENCES

1. Bejczy,A.K. "Robot arm dynamics and control", NASA-JPL Memorandum 33-669, (February, 1974).

2. Heginbotham,W.B. "Can robots beat inflation?," Second North American Symposium on Industrial Robots. Organized by Society of Manufacturing Engineers,Detroit,(1977)

3. Horn,B.K.P., Hirokawa,K. and Vazirani,V. "Dynamics of a three-degree of freedom kinematic chain," MIT Artificial Intelligence Laboratory Working Paper No.155, (October, 1977).

4. Kahn,M.E. "The near-minimum time control of open-loop articulated kinematic chains," Stanford Artificial Intelligence Memo No.106, (December, 1969).

5. Lewis,R.A. "Autonomous manipulation on a robot: summary of manipulator software functions," NASA-JPL Technical Memorandum 33-679, (March, 1974).

6. MACSYMA Manual, MIT Laboratory for Computer Science, Mathlab Group, (1977).

7. Nevins,J., Whitney,D., Dunlavey,M., Drake,S., Kiloran,D., Kondoleon,A., Mogged,C., Seltzer,D., Simunovic,S., Wang.S. and Watson,P. "Exploratory research in industrial modular assembly," Charles Stark Draper Laboratories, Inc. Report R-1111, (1977).

8. Paul,R. "Modelling and trajectory calculations and servoing of a computer controlled arm," Stanford Artificial Intelligence Memo No.11, (November 1972).

9. Pieper,D.L. "The kinematics of manipulators under computer control," Stanford Artificial Intelligence Laboratory Memo No.72, (1968).

10. Raibert,M.H. "Analytical equations vs. table look-up: a unifying concept," Proc.IEEE Conference on Decision and Control, New Orleans, (December, 1977).

INDUSTRIAL ROBOTS - GRIPPER REVIEW

by G.Lundstrom, Fluid Technology Laboratory Aeronautical Research Institute of Sweden

Translated from the Swedish by Birgitta Glemme and the author

English Edition edited by: Dr.B.W.Rooks, Dept. Mechanical Engineering, University of Birmingham

88 pages, 96 figures and 27 references. Published by International Fluidics Services Ltd., November 1977 at £14.00 U.K. Orders. US$38.00 Elsewhere.

The Swedish industrial robot manufacturers ASEA, Ekstroms, Electrolux, Kaufeldt, Retab and Philips Tele, together with the Swedish Board for Technical Development sponsored a gripper design project from 1973 to 1976 at the Fluid Technology Laboratory in Stockholm. Within this project grippers in general have been studied and specific problems have been analysed and solved.

Different new ideas were tried and the results continuously forwarded to the sponsors. An agreement for the co-operation and for inventions within the project was worked out at the initial stage.

This book summarizes sixteen smaller reports and patent applications resulting from this project. The material has been structured to satisfy the needs of designers, buyers and production managers who are working with industrial robots and other material handling equipment.

Grippers can account for up to 10% of the industrial robot cost. This book will, therefore help suppliers and users of industrial robots to reduce their costs by showing what is already available and those which have been tried successfully in industry.

THE INDUSTRIAL ROBOT

an international quarterly journal

volume 5 no.2 june 1978