MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 458 December 1977

CONFIGURATION SPACE CONTROL

Berthold K. P. Horn and Marc H. Raibert*

ABSTRACT. Complicated systems with non-linear time-varying behavior are difficult to control using classical linear feedback methods applied separately to individual degrees of freedom. At the present, mechanical manipulators, for example, are limited in their rate of movement by the inability of traditional feedback systems to deal with time-varying inertia, torque coupling effects between links and Coriolis forces. Analysis of the dynamics of such systems, however, provides the basic information needed to achieve adequate control.

Implementation of a control system based on such analysis is not straightforward, however, since impractical amounts of computation or memory may be called for. We propose a new method that balances the trade-off between computational and storage costs. The actuator torques required to move a manipulator along a trajectory are calculated using coefficients found in a look-up table indexed by the configuration of the manipulator. Feedback plays only an indirect role in correcting for small differences between the state of the actual device and that of a dynamic model.

This report describes research done 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 Office of Naval Research under Office of Naval Research contract N00014-77-C-0389.

* now at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91103.

MOTIVATION.

The application of industrial manipulators to parts transfer is still limited by their high cost. A useful figure of merit for such a device is the ratio of the number of operations it can perform per unit time to its cost. The more cycles the manipulator performs per unit time, the more rapidly it will pay back the investment. There is, we believe, a threshold for this figure of merit above which the application of machine manipulation to a wide variety of 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 fairly stable. It is possible, however, to decrease task cycle times with equally dramatic results. Decreasing cycle time means increasing the rate of manipulation -- the speed the arm moves during transfer and during manipulation.

Presently, many mechanical manipulators are limited by their controllers. Such systems 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 behavior. Performance is adequate at low speeds provided the actuators are strong enough and the properties of the devices do not change too dramatically with configuration. At higher speeds, however, problems are caused by:

(1) Varying effective moments of inertia,

(2) Torque coupling between degrees of freedom, and

(3) Coriolis forces proportional to velocity product terms.

Naturally, other factors, such as the mechanical strength of the device and the power available from the actuators, also limit ultimate performance. For many manipulators, however, these are not the limiting factors.

BACKGROUND.

Mechanical manipulators used for parts transfer typically consist of rigid components, called links, attached to each other at joints, each joint being powered by an actuator (see for example figure 1). Measurements of joint position and velocity are available to the control system that supplies commands to the actuators. Most commonly, manipulators are attached to a fixed base at one end and carry a terminal device or tool at the other. The time-varying actuator commands are intended to cause this terminal device to follow a given trajectory through space.

Many arrangements of links and joints are possible; in this paper, we concentrate on a kinematic chain arranged in a popular serial or cascaded structure using rotary joints. This choice allows us to be concrete and to avoid repeated use of phrases such as "joint-angle or joint-extension" and "actuator torque or actuator force". Similar methods apply to devices with linear motions and to those with parallel degrees of freedom.

The particular device shown in figure 1 was designed and built by Victor Scheinman for the Artificial Intelligence Laboratory [1]. It has six degrees of freedom, the minimum necessary to reach points in the work space with arbitrary orientation of the terminal device. This manipulator is driven by six direct current torque motors, and has potentiometers for joint-angle measurement and tachometers for joint-angle-rates or angular velocities. The supporting electronics permit direct control of motor currents and, consequently, actuator torques.

We designate θ_i the angle of the i^{th} joint, and $\dot{\theta}_i$ the angular velocity of this joint. Similarly, $\ddot{\theta}_i$ is the angular acceleration and T_i the torque applied by the i^{th} actuator. Frequently it is helpful to group values for all degrees of freedom using vector notation. Thus for a system with n degrees of freedom, we call $\underset{\sim}{\theta}$ the configuration, where:

$$\underset{\sim}{\theta} = (\theta_1, \theta_2, \theta_3, \quad \ldots \quad \theta_n)$$

Similarly, we refer to the combination of $\underset{\sim}{\theta}$ and $\dot{\underset{\sim}{\theta}}$ as the state, where:

$$\dot{\underset{\sim}{\theta}} = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \quad \ldots \quad \dot{\theta}_n)$$

The torque vector is similarly defined as

$$\underset{\sim}{T} = (T_1, T_2, T_3, \quad \ldots \quad T_n)$$

With this notation we see that the function of the control system is to produce appropriate actuator torques $\underset{\sim}{T}(t)$, so that the actual joint angles, $\underset{\sim}{\theta}^a(t)$, follow a given trajectory of desired joint angles, $\underset{\sim}{\theta}^d(t)$ (see figure 2). We will see later that this task may at times be simplified if the control system also has access to both actual angular velocities, $\dot{\underset{\sim}{\theta}}^a(t)$, as well as desired angular velocities, $\dot{\underset{\sim}{\theta}}^d(t)$.

A SYSTEM WITH ONE DEGREE OF FREEDOM.

To introduce some of the notions used later on, let us consider a very simple system illustrated in figure 3 -- a one degree-of-freedom "manipulator". Here a motor produces a torque, T, which drives a shaft. The shaft carries a rod of mass m and length ℓ. The angular departure from vertical, θ, is measured by a potentiometer, while a tachometer measures the angular velocity, $\dot{\theta}$. Clearly the system is governed by an equation of the form

$$T = I\ddot{\theta} - k\sin(\theta)$$

where $I = m\ell^2/3$, g is the acceleration due to gravity and $k = (m\ell/2)g$. A typical control system for such a second-order system is shown in figure 4. Here,

$$T = \beta(\dot{\theta}^d - \dot{\theta}^a) + \alpha(\theta^d - \theta^a)$$

where the superscripts denote desired and actual values, while α and β are parameters yet to be determined. Combining the two equations we find

$$I\ddot{\theta}^a + \beta\dot{\theta}^a + (\alpha\theta^a - k\sin(\theta^a)) = \beta\dot{\theta}^d + \alpha\theta^d$$

If the actual angle, θ^a, is to follow the desired angle θ^d closely, $\alpha \gg k$. In this case, the poles of the system are approximately at the roots of the polynomial

$$I s^2 + \beta s + \alpha = 0$$

That is,

$$s = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha I}}{2I}$$

The speed of response of the overall system depends on $\sqrt{\alpha / I}$; so, for rapid response, α should be large.

To obtain good damping of transient oscillations, we choose β so that $\beta \approx 2\sqrt{\alpha I}$. The details of this are not very important other than to show that such feedback systems can achieve adequate control of simple second-order mechanical systems and that the parameters of the feedback system must be chosen by considering the parameters of the system itself. When system parameters change, a different set of feedback parameters is used to provide best performance.

If the parameters of the system vary greatly and the control system is not altered, unsatisfactory performance can be anticipated. This may take the form of sluggish response, excessive overshoot, or undamped oscillations.

INVERSE SYSTEMS.

Analysis of the dynamics of a system often leads to equations that can be used to implement an "inverse" system. The system originally analyzed can be viewed as an analog computer for calculating position (and its derivatives) given actuator outputs, while the inverse system computes actuator outputs from position (and its derivatives). A simple illustration will make this clear.

A one degree-of-freedom system was shown in figure 3, governed by the equation

$$I \ddot{\theta} - k \sin(\theta) = T$$

This system can be viewed as an analog computer solving this differential equation for $\theta(t)$, given the input $T(t)$. If the constants in the equation are known, one can turn this analysis around and calculate the values of torque, $T(t)$, needed to achieve the desired joint-angle variations with time, $\theta(t)$. This inverse procedure is important in solving the control problem. An open-loop control system based on this notion is shown in figure 5a. We will take up later the question of errors in trajectory which result from small differences between the actual system and the model used in deriving the inverse system. For now, note that in view of this possibility the actual state of the system should be used in the inverse calculation rather than the desired state (see figure 5b). With this modification, the inverse system takes as its prime input the angular acceleration, and produces actuator torque as its output.

The straightforward kind of control based on an inverse system and illustrated here applied to a linear, time-invariant, one degree-of-freedom system, will now be extended to control of more complex systems such as manipulators. Before we can do this, we have to understand the dynamics of these devices. Considerable work has been done in this area as can be seen from references [2, 3, 4, 5, 6, 7, 8] for example.

DYNAMICS OF MANIPULATORS.

The most direct route leading to a detailed understanding of the dynamics of a complex mechanical system such as a manipulator is an analysis based on the Euler-Lagrange equations [8],

$$Q_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}$$

Here the q_i represents generalized coordinates, Q_i generalized forces and L is the Lagrangian, or "kinetic potential",

$$L = K - P$$

where K is the kinetic energy and P the potential energy of the whole system. In our case the most convenient generalized coordinates are the joint-angles, θ_i, and then the generalized forces become the actuator torques T_i. Furthermore, since the potential energy is a function of joint-angles only, it is convenient to separate the calculation of torques required to compensate for gravitational forces,

$$T_i' = \frac{\partial P}{\partial \theta_i}$$

from the calculation of torques required to support the motion if gravity were not present

$$T_i = \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_i}\right) - \frac{\partial K}{\partial \theta_i}$$

Also, since the total kinetic energy is the sum of the kinetic energies of each of the links, it is helpful to separate the calculation into components of the form,

$$T_{ij} = \frac{d}{dt}\left(\frac{\partial K_j}{\partial \dot{\theta}_i}\right) - \frac{\partial K_j}{\partial \theta_i}$$

where T_{ij} is the torque required of the i^{th} actuator to support the motion of the j^{th} link. The total torque required at each actuator is then obtained by summation of these terms.

The derivation of the Euler-Lagrange equations requires difficult mathematical arguments; however the use of these equations is straightforward. Application of these equations to manipulator control was pioneered by Uicker Pieper, Kahn and Paul [5, 6, 7, 8]. A practical difficulty is the potentially explosive growth in algebraic manipulation that accompanies analysis of systems with several degrees of freedom. A computer system such as MACSYMA [9], able to carry out manipulations of symbolic mathematical expressions is very helpful in these cases.

Much of the earlier work on this problem made use of a general representation, with a coordinate system erected in each link and matrices describing the transformations between coordinate systems of connected links [5, 6, 7, 8]. While perfectly general, this kind of analysis leads to very complicated results and the need to perform thousands of arithmetic operations in order to calculate required joint-torques. All hope of performing these calculations in real-time was abandoned as a result.

ILLUSTRATION USING THE SINGLE DEGREE-OF-FREEDOM SYSTEM.

For the system shown in figure 3, it is clear that $K = \frac{1}{2} I\dot{\theta}^2$ and $P = k \cos\theta$, so that

$$L = \frac{1}{2} I\dot{\theta}^2 - k \cos\theta$$

Consequently,

$$T = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta}$$

gives us

$$T = I\ddot{\theta} - k \sin\theta$$

as before. Here, of course, little is gained by using this method. It is, however, invaluable for complex devices.

DYNAMICS OF A THREE-LINK DEVICE.

Recent work has shown that some devices can be analyzed easily if calculations use joint-angles directly and if links are modelled as thin rods [10, 11, 12, 13]. As an example, we present the results for a three-link device with offsets shown in figure 6. This corresponds to the first three joints of the arm discussed earlier, shown in figure 1.

$$T_1 = [I_1 + m_2(\delta_2^2 + \frac{\ell_2^2}{3} s_2^2) + m_3(\ell_2^2 s_2^2 + \ell_2\ell_3 s_2 s_{23} + \frac{\ell_3^2}{3} s_{23}^2 + \delta_3^2)]\ \ddot{\theta}_1 -$$

$$[\frac{m_2}{2}\delta_2\ell_2 c_2 + \frac{m_3}{2}\delta_3(2\ell_2 c_2 + \ell_3 c_{23})]\ \ddot{\theta}_2 -$$

$$[\frac{m_3}{2}\delta_3\ell_3 c_{23}]\ \ddot{\theta}_3 +$$

$$[m_2\frac{2\ell_2^2}{3} s_2 c_2 + m_3(2\ell_2^2 s_2 c_2 + \ell_2\ell_3(c_2 s_{23} + s_2 c_{23}) + \frac{2\ell_3^2}{3} s_{23} c_{23}]\ \dot{\theta}_1\dot{\theta}_2 +$$

$$[\frac{m_2}{2}\delta_2\ell_2 s_2 + \frac{m_3}{2}\delta_3(2\ \ell_2 s_2 + \ell_3 s_{23})]\ \dot{\theta}_2^2 + [m_3\delta_3\ell_3 s_{23}]\ \dot{\theta}_2\dot{\theta}_3 +$$

$$[\frac{m_3}{2}\delta_3\ell_3 s_{23}]\ \dot{\theta}_3^2 + [m_3\ell_3 c_{23}(\ell_2 s_2 + \frac{2\ell_3}{3} s_{23})]\ \dot{\theta}_3\dot{\theta}_1$$

$$T_2 = -[\frac{m_2}{2}\,\delta_2\ell_2 c_2 + \frac{m_3}{2}\,\delta_3(2\ell_2 c_2 + \ell_3 c_{23})]\,\ddot{\theta}_1 +$$

$$[m_2\,\frac{\ell_2^2}{3} + m_3(\ell_2^2 + \ell_2\ell_3 c_3 + \frac{\ell_3^2}{3})]\ddot{\theta}_2 +$$

$$[\frac{m_3}{2}\,(\ell_2\ell_3 c_3 + \frac{2\ell_3^2}{3})]\ddot{\theta}_3 -$$

$$[m_2\,\frac{\ell_2^2}{3}\,s_2c_2 + \frac{m_3}{2}\,(2\ell_2^2 s_2c_2 + \ell_2\ell_3(c_2s_{23} + s_2c_{23}) + \frac{2\ell_3^2}{3}\,s_{23}c_{23}]\,\dot{\theta}_1^2 -$$

$$[m_3\ell_2\ell_3 s_3]\dot{\theta}_2\dot{\theta}_3 - [\frac{m_3}{2}\,\ell_2\ell_3 s_3]\,\dot{\theta}_3^2 -$$

$$[m_2\,\frac{\ell_2}{2}\,s_2 + m_3(\ell_2 s_2 + \frac{\ell_3}{2}\,s_{23})]g$$

$$T_3 = -[\frac{m_3}{2}\,\delta_3\ell_3 c_{23}]\,\ddot{\theta}_1 + [\frac{m_3}{2}\,(\ell_2\ell_3 c_3 + \frac{2\ell_3^2}{3})]\,\ddot{\theta}_2 + [m_3\,\frac{\ell_3^2}{3}]\ddot{\theta}_3 -$$

$$[\frac{m_3}{2}\,\ell_3 c_{23}(\ell_2 s_2 + \frac{2\ell_3}{3}\,s_{23})]\dot{\theta}_1^2 + [\frac{m_3}{2}\,\ell_2\ell_3 s_3\,]\,\dot{\theta}_2^2 -$$

$$[m_3\,\frac{\ell_3}{2}\,s_{23}]\,g$$

Here ℓ_1, ℓ_2, ℓ_3 are the lengths of the three links. The upright column is modelled as a cylinder with inertia I_1 about its axis, while the other two links are modelled as thin rods of mass m_2 and m_3 respectively. An offset of δ_2 occurs between the long axis of the upright column and the plane in which the second link rotates. A similar offset δ_3 separates the plane in which link 3 rotates from this vertical axis (see figure 6).

A shorthand notation is used for trigonometric terms. That is,

$$c_i = \cos(\theta_i) \qquad s_i = \sin(\theta_i)$$

and

$$c_{ij} = \cos(\theta_i + \theta_j) \qquad s_{ij} = \sin(\theta_i + \theta_j)$$

The terms containing $\ddot{\theta}_1$, $\ddot{\theta}_2$ or $\ddot{\theta}_3$ are inertial torques (required to accelerate the links), while terms containing angular velocity products of the form $\dot{\theta}_i\dot{\theta}_j$ are Coriolis force components. The third class of terms contain g, the gravitational constant, and are thus the torques required to compensate for the gravitational load.

Roughly a hundred arithmetic operations are required to calculate the required joint torques given joint angles, $\underset{\sim}{\theta}$, and angular rates, $\underset{\sim}{\dot{\theta}}$, as well as desired accelerations, $\underset{\sim}{\ddot{\theta}}$. Such a direct calculation might be used as the basis of a control system. In fact, if δ_2 and δ_3 are zero, several terms fall out and the calculation becomes simpler. If, on the other hand, we consider a device with either more degrees of freedom or links that have to be modelled by full inertia matrices, instead of the diagonal form appropriate to thin rods, then these calculations become quite intractable (For an example, see appendix A in reference [7]).

WHY THE STRAIGHTFORWARD CONTROL METHOD FAILS.

If we look at the equations for the actuator torques we see that they have some characteristics that make control more complex than it is for a simple one degree-of-freedom second-order system. First of all, the coefficients of $\ddot{\theta}_i$ in the expression for T_i are not constant, indicating variable effective inertia. Ordinarily, as we have seen, the feedback coefficients are constants tuned for proper operation at some fixed inertia, so control will not be good for inertias very different from this design value.

Next, one sees that there are terms containing $\ddot{\theta}_j$ in the expressions for T_i, when $i \neq j$. This cross-coupling, too, may produce problems since accelerations of one joint require coordinated torques at all joints. Loops closed separately around each joint cannot easily deal with this problem.

Finally one sees numerous Coriolis force terms, multiples of products of joint-angle-rates. At high speeds, these dominate the inertial and gravitational torques, and actuator torques produced by traditional control systems may not be appropriate for stable control. Such problems become most significant for long movements, when velocities can build up to a point where velocity product terms exceed acceleration terms.

FORM OF THE EQUATIONS OF MOTION.

We now examine the form of the equations for the actuator torques in an attempt to find a reasonable computing scheme for control of such a device. Clearly each torque is a function of joint angles, $\underset{\sim}{\theta}$, angular rates $\underset{\sim}{\dot{\theta}}$, and angular accelerations, $\underset{\sim}{\ddot{\theta}}$. One approach then is to calculate the required actuator torques directly from the equations; techniques used by Paul are similar to this [8]. In most cases, however, this approach involves an inordinate amount of computing time and drastic simplifications have to be introduced to make this at all feasible [8].

The other extreme is based on a look-up table indexed on $\underset{\sim}{\theta}$, $\underset{\sim}{\dot{\theta}}$ and $\underset{\sim}{\ddot{\theta}}$. Each of the dimensions is quantized into m intervals. No calculation is required, but the look-up table has 3 n dimensions for a device with n degrees of freedom and is thus quite unmanageable even when each dimension is quantized coarsely, that is, when m is small. If we use the convention that subscripts correspond to variables with discrete sets of values, then we may represent this scheme by the equation

$$T_i = F_{i,(\underset{\sim}{\theta},\underset{\sim}{\dot{\theta}},\underset{\sim}{\ddot{\theta}})}$$

where the values of F_i are pre-calculated for a discrete set of values of $\underset{\sim}{\theta}$, $\underset{\sim}{\dot{\theta}}$ and $\underset{\sim}{\ddot{\theta}}$, and stored in a table [14]. Albus' manipulation scheme is similar to this [15].

Fortunately, however, the torques are linear functions of the accelerations, as we have seen, and the equation can be rewritten in the form

$$T_i = K_i(\underset{\sim}{\theta},\dot{\underset{\sim}{\theta}}) + \sum_{j=1}^{n} I_{ij}(\underset{\sim}{\theta},\dot{\underset{\sim}{\theta}})\,\ddot{\theta}_j$$

This leads to a second implementation, using one look-up table for K_i and one for I_{ij}, both indexed on $\underset{\sim}{\theta}$ and $\dot{\underset{\sim}{\theta}}$. These tables are now only of dimension 2n. The computation required after table look-up is simple, namely n multiplications and additions per joint. Raibert's manipulator control schemes are based on a similar formulation of the problem [16,17]. He derives the table entries by "learning" rather than calculation from the model -- that is, the manipulator performs test motions to estimate experimentally the multipliers K_i and I_{ij}. In general the look-up tables are still too large to be useful. (In the equation above, the inertial terms I_{ij} are written as functions of both angles and angular rates to indicate the indexing of look-up tables in this scheme. In fact, the inertial terms do not depend on the angular rates, something we will exploit next.)

CONFIGURATION SPACE CONTROL.

It turns out that the equations have a rather special form [8, 10] and can be written as

$$T_i = G_i(\underset{\sim}{\theta}) + \sum_{j=1}^{n} I_{ij}(\underset{\sim}{\theta})\,\ddot{\theta}_j + \sum_{j=1}^{n}\ \sum_{k=j}^{n} C_{ijk}(\underset{\sim}{\theta})\dot{\theta}_j\dot{\theta}_k$$

This is a consequence of the form of the expressions for the Lagrangian. Here, G_i is the gravity compensation, I_{ij} are inertial terms and C_{ijk} are Coriolis force coefficients. Each of these is a polynomial in the sines and cosines of the joint-angles, the link lengths and masses. Clearly these could be pre-calculated and stored in lookup tables indexed on $\underset{\sim}{\theta}$. Such tables would be of dimension n and thus manageable in terms of storage space. A little more calculation is required; namely, $n(n + 1)/2 + n$ multiplications and $n(n+1)/2+n$ additions per joint. We call this method configuration space control, since the look-up table is indexed on $\underset{\sim}{\theta}$, the configuration of the manipulator.

We should immediately add, that while nominally there are n dimensions in configuration space, some economy of storage is possible by noting that the I_{ij} and C_{ijk} terms are not functions of the position of the first joint; that connecting the manipulator to its base. If in addition the axis of this joint is parallel to the gravity vector (as is often the case), the G_i term is also independent of the position of the first joint. Furthermore, if the important masses in the last, or highest numbered link, the terminal device, are symmetrically distributed, then the equations of motion do not depend on the position of the last joint either. If both these conditions are true, the stored tables need only be of dimension (n - 2), a considerable saving. Further simplifications may apply to specific classes of manipulators.

Similarly, the storage required per table entry can be economized when it is realized that the inertia matrix is symmetric, that only the upper half of the Coriolis force coefficient matrix is needed, and that there exist relationships between the coefficients C_{ijk} and C_{kji} [4]. It takes a little more work to exploit the fact that for a given manipulator geometry many of these coefficients are actually zero, or so small as to be negligible, as can be seen from the equations we presented earlier for a three link device (where only 10 of a possible 18 terms were non-zero).

We have now explored a spectrum of methods for computing the required joint torques (see figure 7). It is our contention that both ends of the spectrum represent techniques which are impractical and that the configuration space method provides a near optimum balance between storage and computational costs. Note for example that state space control requires more storage capacity than configuration space control for any system with $n > 1$ and $m > 1$.

The notions of the inverse system and configuration space look-up can now be brought together in an overall system like that shown in figure 8. If one does not take advantage of the economies mentioned above, then about $n^2(n + 3)/2$ multiplications and additions are required per calculation cycle. If each dimension is quantized into m sections, then the look-up table has a total of $[n(n+1)(n+2) / 2]\, m^n$ numeric entries. These numbers are large, but manageable, especially in view of recent trends in the cost of computer storage.

MISMATCHES BETWEEN THE DYNAMIC SYSTEM AND THE MODEL.

Glancing at figure 8, one notices that the input to the system is the angular acceleration, presumably supplied by a trajectory planner. Intuition suggests that such a system is likely to suffer from the ill effects of approximate numerical differentiation and in general behave in a fashion that has the actual joint-angles drifting away from the desired joint angles. The rate of accumulation of errors will depend on how accurate an inverse one can build to the dynamic system. For low speed movement it appears that the limiting factor in this regard will be friction, which tends to be both difficult to predict and not a repeatable function of joint angles and angular velocities. This suggests that we have to augment the elegant open-loop system with sub-systems capable of correcting for small departures of the actual trajectory from the desired one.

Some form of negative feedback is needed. Note, however, that feedback plays quite a different role here than it did in the simple control system shown earlier for a second-order dynamical system. In that situation, feedback produces the actuator forces; error signal are in some sense the prime movers. Here feedback is added only to correct for minor departures of the dynamical system from the model used in deriving the inverse system, with the main component of actuator torque coming from the open-loop calculation. For this reason the design of this feedback system is much less critical, with small feedback gains acceptable and consequently there need by little concern over stability.

INTRODUCTION OF CORRECTIVE FEEDBACK

There are a number of alternate ways of introducing feedback to correct for the departures of actual position discussed in the previous section. Perhaps the most obvious has corrections proportional to the errors applied to the inputs of the dynamic system (see figure 9). That is, the actuator torque now is the sum of the calculated open-loop torque required to follow the trajectory and terms proportional to errors in position and velocity. Such a system would differ from the traditional control system in that the input is first passed through the inverse system and that the feedback gains would be much smaller. To some extent, this kind of system would however suffer from some of the short-comings of the traditional system, unless these feedback gains were at least adjusted according to the current configuration.

If suitable costs can be associated with departures from the correct trajectory and if costs can be assigned to control inputs, then optimal time-varying feedback gains can be determined using the techniques of modern control theory [18]. In a system with more than one degree of freedom, one has to use a feed-back matrix. This too could be conventionally obtained from a look-up table indexed on the configuration.

A different system can be obtained by applying the error signals to the inputs to the inverse system instead (see figure 10). This has several advantages. First, the input to the overall system from the trajectory planner is now composed of the joint angles and the angular velocities instead of the angular accelerations. Secondly, this system can be analyzed more readily. For example, if the inverse system really is an exact inverse for the dynamic system, then their cascade connection is simply an identity system. In this case, the overall system degenerates into an ordinary linear, time-invariant second-order system. The designer can now freely choose the response by picking the gains α and β — that is, the poles can be arbitrarily assigned.

Noteworthy is the linearization and decoupling of degrees of freedom obtained in this fashion [19]. In a system with more than one degree of freedom, feedback can now be applied separately to individual degrees of freedom, that is, a feedback matrix is not required. Furthermore, the feedback gains do not depend on the configuration and can be fixed. A remaining analytic difficulty is the determination of the effects of small differences between the actual device and the dynamic model used in the derivation of the inverse system.

SUMMARY AND CONCLUSIONS

Straightforward feedback control is unable to deal correctly with varying effective inertias, joint torque coupling and Coriolis forces encountered in high-speed movements of mechanical manipulators. The precision of manipulation for slower movement is similarly limited. Analysis of the dynamics of the kinematic chain leads to equations representing an inverse system, able to compute required joint torques from desired joint accelerations given the state of the device. Unfortunately this computation is quite unwieldy and essentially useless for real time control of devices with more than two or three degrees of freedom. On the other hand, performing the computation completely by look-up in a table indexed on the state of the device leads to a requirement for excessive amounts of memory.

A compromise on this space-time trade-off is a method based on configuration-space look-up tables. These precomputed tables are of manageable size and the computations performed using the entries found there are relatively straightforward. Each computational cycle requires about $n^2(n+3)/2$ arithmetic operations for a device with n degrees of freedom. The total size of the look-up table for this computation is less than $[n\ (n+1)\ (n+2)/2]\,m^n$ if each dimension is quantized into m segments.

ACKNOWLEDGMENTS

Drawings by Karen Prendergast and Suzin Jabari. Helpful comments were provided by Matt Mason, Patrick Winston and Brian Schunck.

REFERENCES.

1. Horn, B. K. P. and Inoue, H., "Kinematics of the MIT-AI-VICARM Manipulator," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Working Paper No. 69, May 1974.

2. Markiewicz, B. R., "Analysis of the computed torque drive method and comparison with conventional position servo for a computer controlled manipulator," NASA-JPL Technical Memorandum 33-601, August 1973

3. Bejczy, A. K., "Robot arm dynamics and control," NASA-JPL Technical Memorandum 33-669, February 1974.

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

5. Uicker, J. J., "Dynamic force analysis of spatial linkages," Trans. of ASME, 1967.

6. Pieper, D. L., "The kinematics of manipulators under computer control," Stanford University, Artificial Intelligence Group Memo No. 72, 1968.

7. Kahn, M. E., "The near minimum-time control of open-loop articulated kinematic chains," Stanford University, Artificial Intelligence Group, Memo No. 106, 1969.

8. Paul, R., "Modelling, trajectory calculations and servoing of a computer controlled arm," Stanford University, Artificial Intelligence Group, Memo No. 177, November 1972.

9. MACSYMA Manual, Massachusetts Institute of Technology, Laboratory for Computer Science, MATHLAB Group, 1977.

10. Horn, B. K. P., "Kinematics, Statics, and Dynamics of two-d manipulators," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Working Paper No. 99, June 1975.

11. Freund, E., "A non-linear control concept for computer-controlled manipulators," Proc. IFAC-Symposium on Multivariable Technology Systems, Frederickton, Canada, July 1977.

12. Horn, B. K. P., Hirokawa, K., and Vazirani, V., "Dynamics of a three-degree of freedom kinematic chain," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Working Paper No. 155, October 1977.

13. Renaud, M., "Automatic Equation Generation of Articulated Mechanisms," Yugos. Symp. on Indust. Robots, Belgrade, November 1977.

14. Raibert, M. H., "Analytical equations vs. table lookup: A unifying concept," Proc. IEEE Conference on Decision and Control, New Orleans, December 1977.

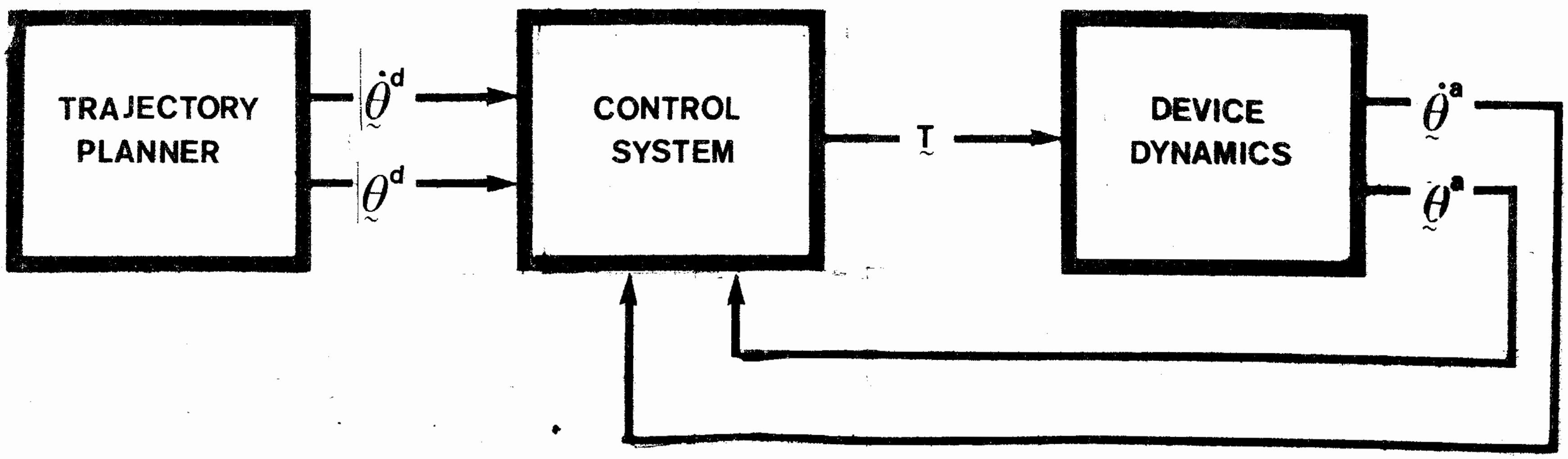
15. Albus, J. S., "A new approach to manipulator control: the cerebellar model articulation controlled (CMAC)," J. Dynamic Systems, Measurement and Control, Trans. of ASME, Series G., Vol. 97, pp 220-227, September 1975.

16. Raibert, M. H., "A state space model for sensorimotor control and learning," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo No. 351, January 1976.

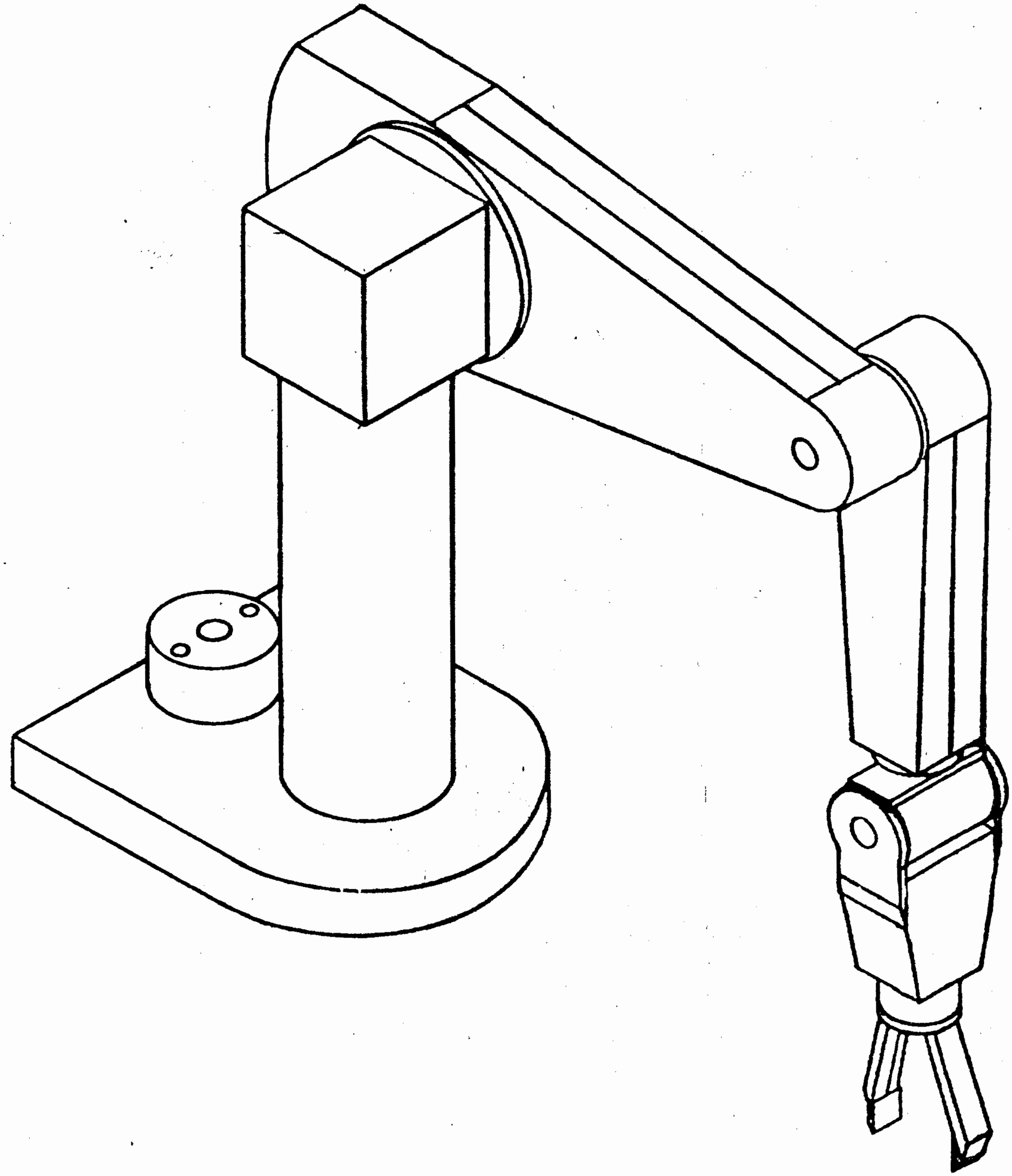
17. Raibert, M. H., "Control and learning by the state-space model: Experimental findings," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Memo No. 412, March 1977.

18. Kwakernaak & Sivan, "Linear Optimal Control".

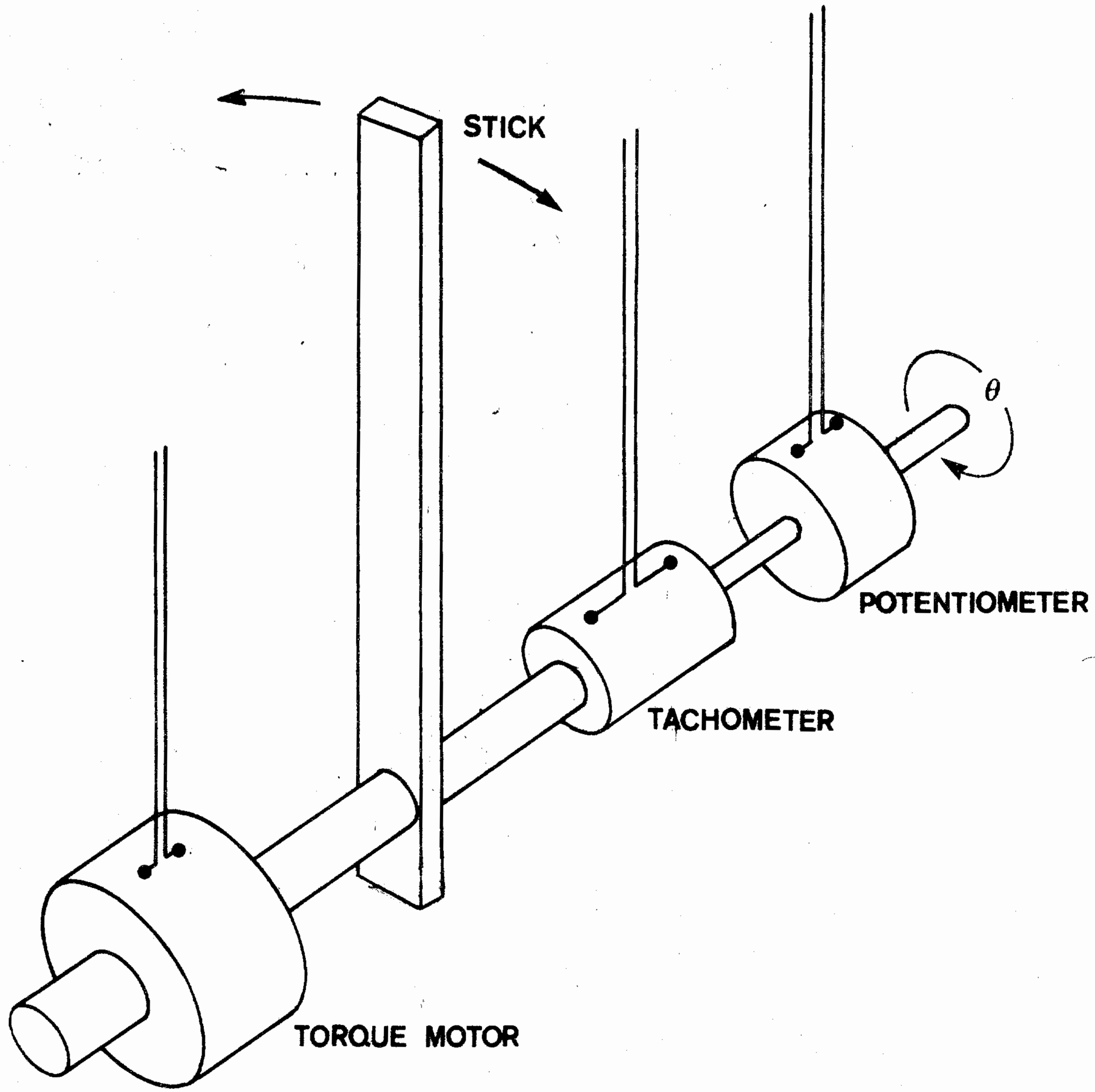
19. Porter, W. A., "Diagonalization and inverse for non-linear systems," Int. J. Control, Vol 11, No. 1, pp 67-76, 1970.



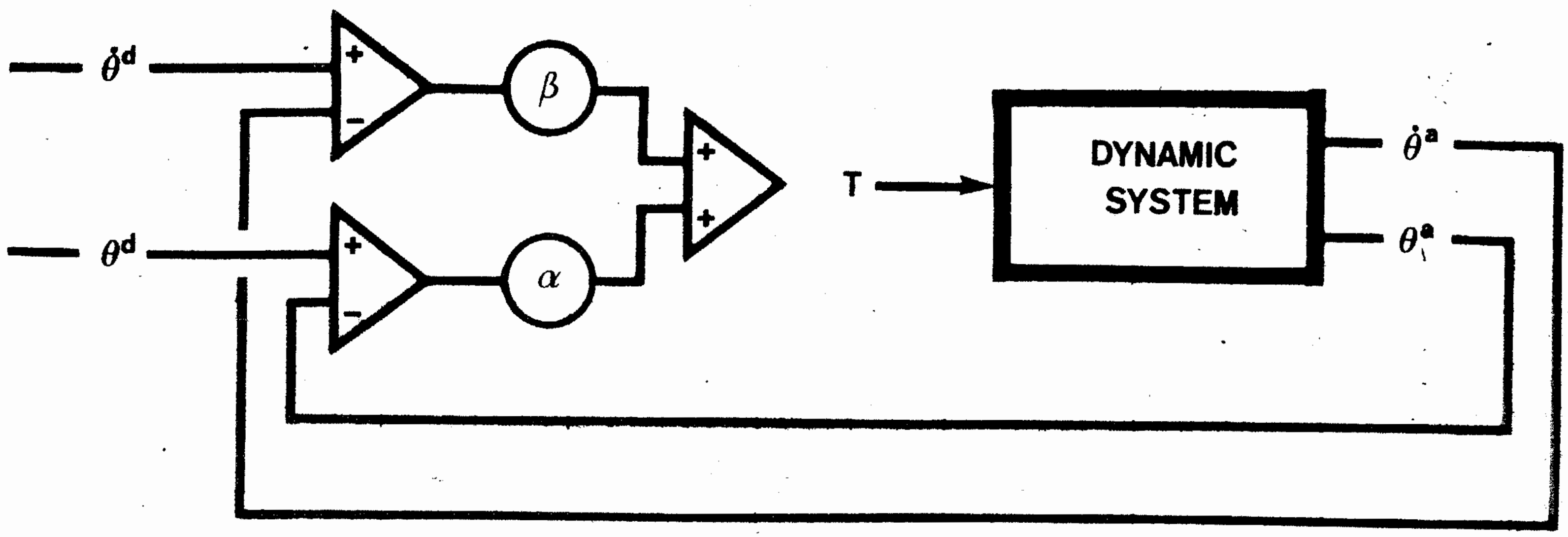
2. Block diagram of the manipulator system. The function of the control system is to generate appropriate actuator torques so that the actual joint angles $\tilde{\theta}^a$, reproduce the desired joint angles $\tilde{\theta}^d$.



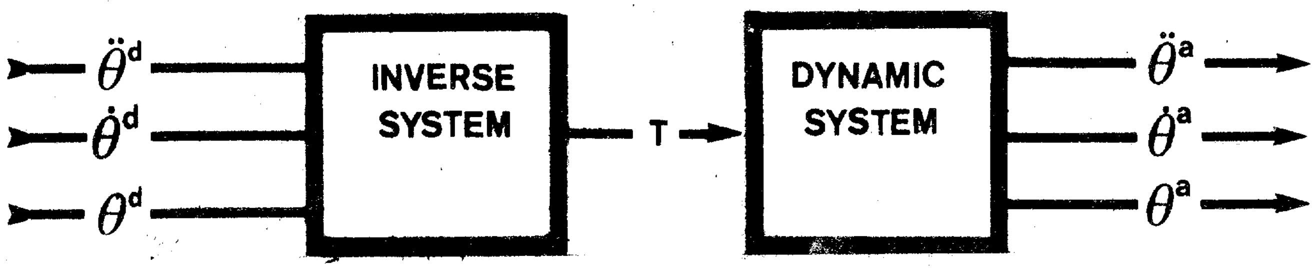
1. Outline drawing of the MIT-AI-VICARM electric manipulator. This is a typical computer controlled device with six degrees of freedom, DC torque motor actuators, potentiometers for position readout and tachometers for the determination of angular velocities.



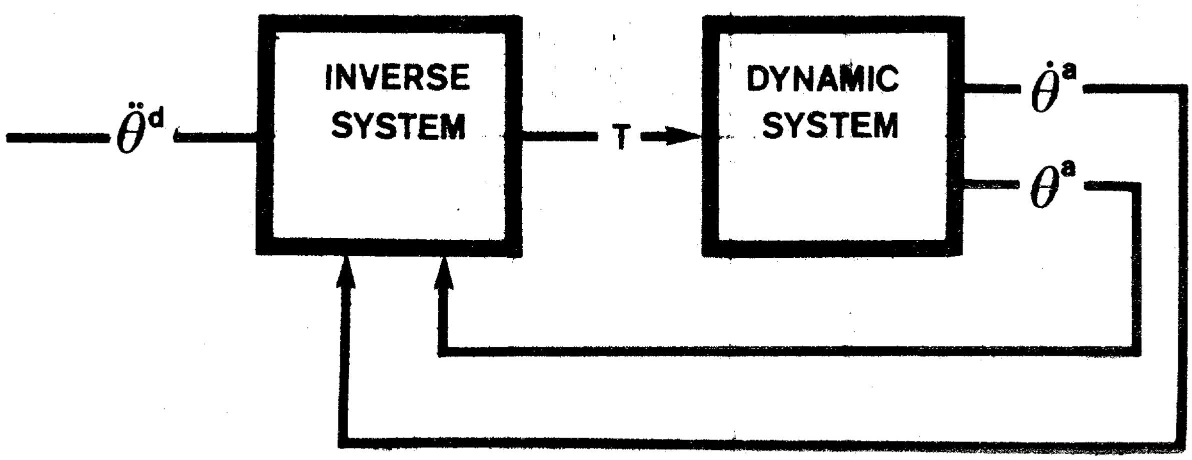
3. Simple one degree-of-freedom mechanical system used to illustrate control schemes. The control system has access to the angle of rotation of the shaft as well as the angular velocity and in turn controls the motor torque.



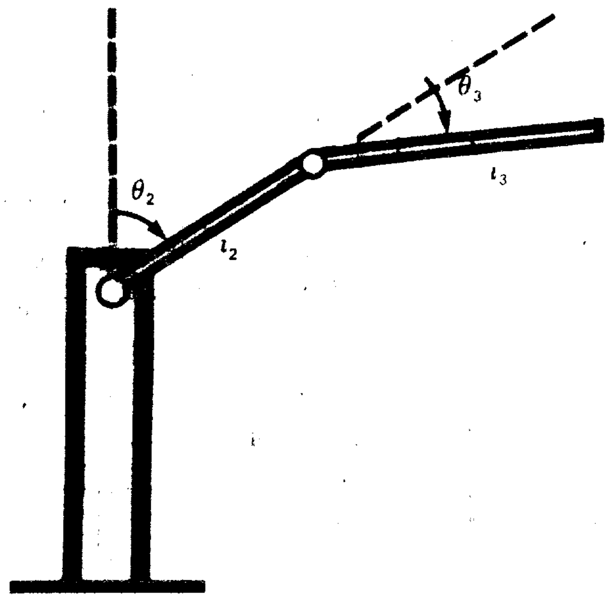
4. Simple feedback control system for single degree-of-freedom, second-order mechanical system with fixed properties. The feedback gains α and β are selected to provide stable control. Their values depend on the parameters of the mechanical system.



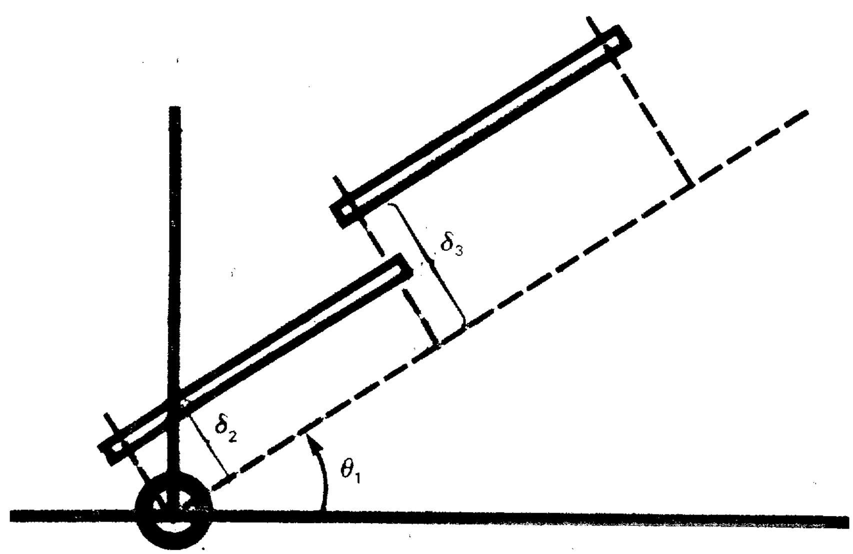
5a. Block-diagram of open-loop control using an inverse system.



5b. Block-diagram of modified open-loop control using actual state rather than predicted state in the calculation of the inverse.



SIDE-VIEW



TOP-VIEW

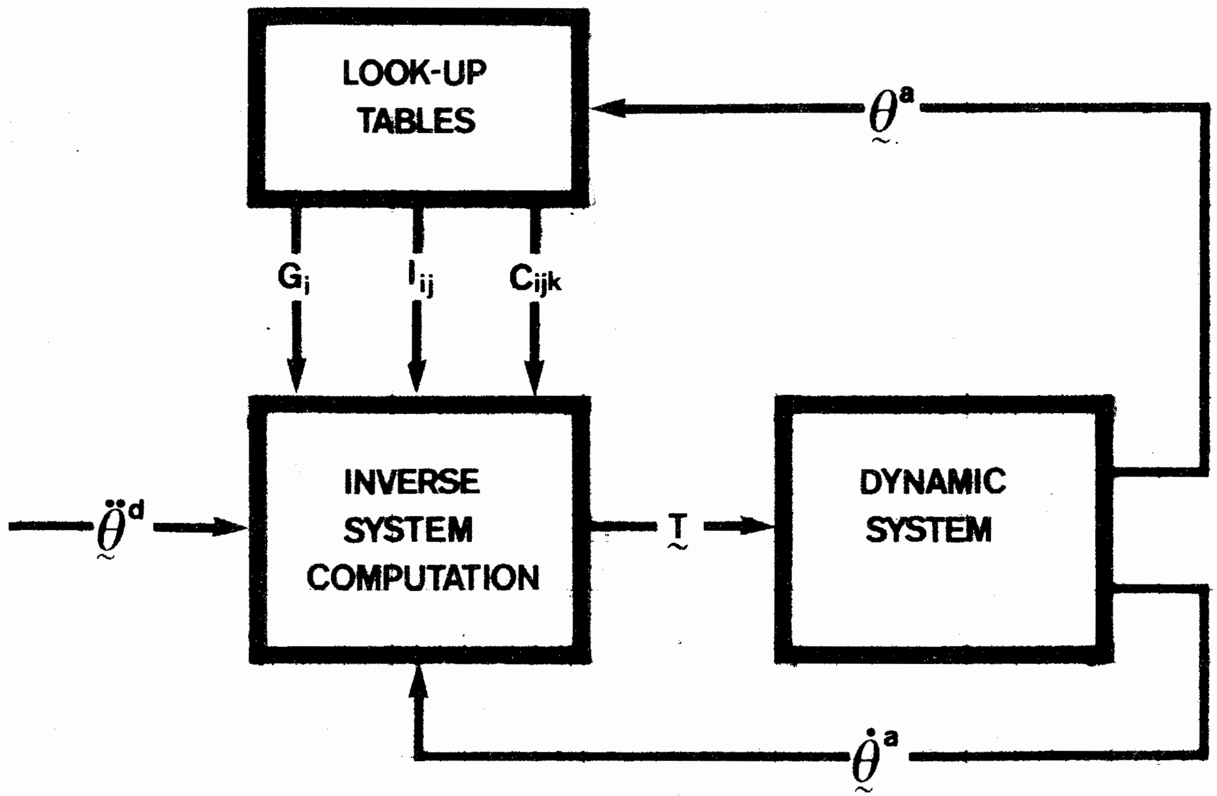
6. Geometry of a three degree-of-freedom manipulator. This could be a diagram of the first three degrees of freedom of the device shown in figure 1, or the drawing for a "leg" on a locomotory device.

7. Table of features of the four methods for calculating actuator torque. The two extremes of the spectrum do not represent viable computational techniques because of excessive computation or storage requirement. Configuration space control appears to provide the optimal balance.

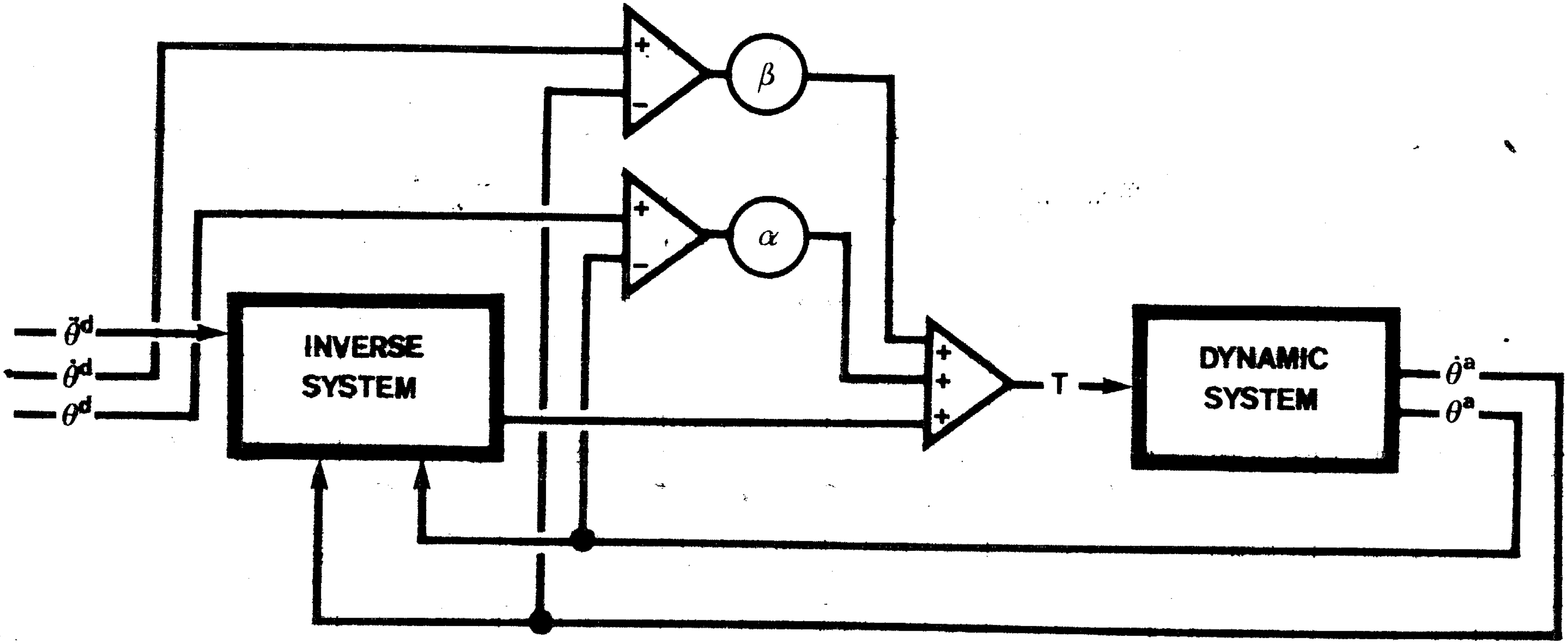
INCREASING STORAGE COSTS ⟵

INCREASING COMPUTATIONAL COST ⟶

	FULL TABLE LOOK-UP	STATE SPACE CONTROL	CONFIGURATION SPACE CONTROL	FULL CALCULATION
EQUATIONS: $T_i =$	$F_{i,(\underset{\sim}{\theta},\underset{\sim}{\dot\theta},\underset{\sim}{\ddot\theta})}$	$K_{i,(\underset{\sim}{\theta},\underset{\sim}{\dot\theta})}$ + $\sum_j I_{i,j,(\underset{\sim}{\theta},\underset{\sim}{\dot\theta})}\ \ddot\theta_j$	$G_{i,(\underset{\sim}{\theta})}$ + $\sum_j I_{i,j,(\underset{\sim}{\theta})}\ \ddot\theta_j$ + $\sum_j\sum_k C_{i,j,k,(\underset{\sim}{\theta})}\ \dot\theta_j\ \dot\theta_k$	$F_i(\underset{\sim}{\theta},\ \underset{\sim}{\dot\theta},\ \underset{\sim}{\ddot\theta})$
DIMENSIONALITY OF LOOK-UP TABLES	3 n	2 n	n	0
COEFFICIENT PER LOOK-UP TABLE ENTRY	1	n + 1	n(n + 1)/2 + n + 1	---
ARITHMETIC OPERATIONS REQUIRED PER JOINT	0	n	n(n + 1)/2 + n	VERY MANY (ROUGHLY ORDER n^3)

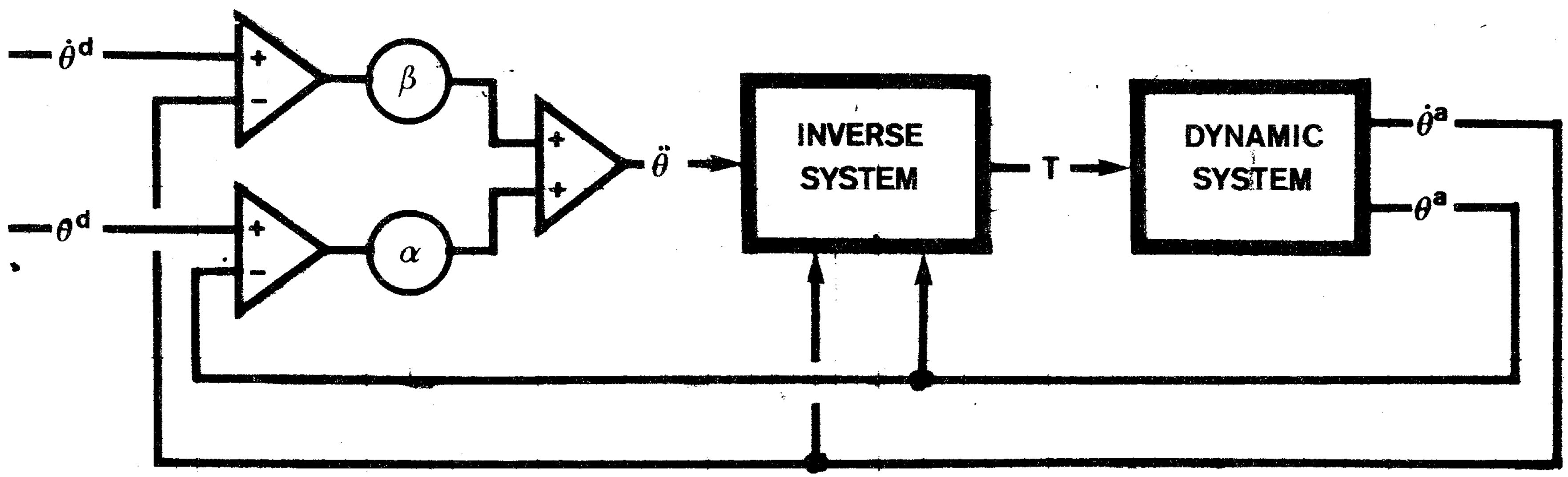


8. Block-diagram of an open-loop control system using the configuration space look-up table method for computing the inverse.



35

9. Block-diagram of a control system with feedback added to take care of small departures of the actual joint angles from the desired joint angles. Here the error signals are applied directly to the actuators after amplification.



10. Block-diagram of a control system with feedback added to take care of small departures of the actual joint angles from the desired joint angles. The error signal here is applied to the input of the inverse system. If the inverse is exact, the overall system becomes a linear time-invariant second-order system. The poles in this system can be freely assigned by manipulating the gains α and β .