Communications

Hybrid Impedance Control of Robotic Manipulators

ROBERT J. ANDERSON AND MARK W. SPONG

Abstract—**In order for robots to improve their adaptability, it is necessary for the underlying control strategies to become more sophisticated. This is especially true in contact tasks such as grinding or deburring where it is desired not only for the robot to apply a steady force, but also to reject undesirable high-frequency disturbances. This work presents a foundation for such force control strategies. Realizing that the type of control strategy that is employed depends fundamentally on the characteristics of the environment, this correspondence classifies the types of environments by use of a duality condition, and demonstrates which control is appropriate for each type of environment. A general control approach is introduced, called hybrid impedance control (HIC), which in its simplest forms reduces to Khatib's operational space control [3], or to Hogan's impedance control [14]. The control law is formulated in a general enough fashion, however, to allow for higher order controllers.**

I. Introduction

Force control of robotic manipulators has been an active area of research for many reasons. The inclusion of force information in the control of robots increases their adaptability to uncertain environments, such as are found in deburring, grinding, and assembly tasks, and provides safety against breakage due to excessive contact forces. Typically, forces are measured by a force sensor connected to the wrist of the robot [1]. It is assumed that this force sensor can accurately delineate forces along six degrees of freedom (DOF).

A. Definitions and Terms

Force control implicitly involves contact with the "environment," which we define to be any element connected to or contacting the robot anywhere past the wrist force sensor. With this definition the end-effector's mass and compliance are both considered to be part of the environment.

Two descriptions of the end-effector position are of interest. The first coincides with the Denavit–Hartenberg [2] representation where the generalized variable q_i represents joint angles, and generalized joint torques are given by τ_i. This representation of the manipulator's position is called joint space. The second description is called task space or constraint space. The six DOF vector x consists of the three orthogonal displacements, which represent the distance between the end-effector and a fixed Cartesian coordinate frame, and the three respective angles, which represent the orientation of the end-effector with respect to the fixed Cartesian coordinate frame. Similarly, the six DOF vector F consists of the three orthogonal force components and the three orthogonal torque components operating at the end-effector with respect to the fixed Cartesian coordinate system.

When the task space is not fixed (e.g., when following a contour) a third, "world" space is necessary. This space is a fixed Cartesian coordinate frame, which serves as a reference point for the task space. For simplicity we shall consider world space and task space to be identical.

Manuscript received April 7, 1987; revised January 5, 1988. This research was partially supported by the National Science Foundation under Grant DMC-8516091. Part of the material in this correspondence was presented at the IEEE International Conference on Robotics and Automation, Raleigh, NC, March 1987.

The authors are with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

IEEE Log Number 8820705.

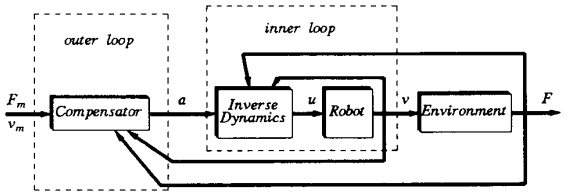

Fig. 1. Inner/outer loop control.

The equations of motion of an n-degree-of-freedom robot can be written using task space coordinates [3] as

$$D(x)\ddot{x} + h(x, \dot{x}) - F = J^{-T}u \tag{1}$$

where $D(x)$ is the $n \times n$ inertia matrix in task space, $h(x, \dot{x})$ contains the Coriolis, centripetal, and gravitational force terms, F represents contact forces, J represents the Jacobian, and u is the input torque.

B. Inner/Outer Loop Control Strategy

The control problem for the system (1) is to choose the input $u(t)$ to cause the end-effector to execute a desired motion in task space while regulating the forces of interaction of the end-effector with the environment. Depending on the task, this can be specified either as a trajectory tracking problem, that is, as the problem of tracking a given motion and/or force trajectory in Cartesian space, or as the problem of obtaining a desired impedance, for example, controlling the manipulator to respond as a second-order system with stiffness K, damping B, and mass M as

$$M\ddot{x} + B\dot{x} + Kx = Kx_m. \tag{2}$$

We will use the concept of inner/outer loop control shown in Fig. 1, where the inner loop is a nonlinear feedback linearization or inverse dynamics control [4] and the outer loop is an additional control to achieve the more classical control theoretic goals such as tracking, disturbance rejection, robustness, etc. Given the system (1), the inner loop control law is assumed to be of the form

$$u = J^T(D(x)a + h(x, \dot{x}) - F) \tag{3}$$

and cancels the nonlinearities present in (1). Since the inertia matrix is invertible and assuming that we are in a region free of kinematic singularities, the control law (3) applied to (1) results in the familiar double integrator system

$$\ddot{x} = a = \dot{v} \tag{4}$$

where a represents the outer loop control whose design constitutes the main thrust of this correspondence.

Two comments are in order at this point. First, the control law (3) differs from the standard inverse dynamics control considered by many authors in the sense that the end-effector forces, represented by F, are also canceled. We are assuming that these forces are directly measurable and hence F in (3) is a measured rather than a computed term. Second, we do not consider the robustness issue in this

correspondence that arises from inexact cancellation of the nonlinearities. We are concerned here only with the outer loop design in the absence of inner loop uncertainty. Numerous robust algorithms [5]-[8] have been applied to inverse dynamics controllers and may also be applied to HIC. The initial presentation [9] of the HIC algorithm demonstrated how a Model Reference Adaptive Controller (MRAC) might be implemented within the inner loop of the HIC controller.

II. Research in Force Control

A number of force control algorithms have been proposed with such names as damping control, stiffness control, and explicit force control. Recently, Whitney surveyed the bulk of these approaches, and gave a historical perspective to the field [10]. Despite the diversity of approaches, however, most of these approaches can be separated into two classes, *hybrid control* and *impedance control.*

A. Hybrid Control

Hybrid position/force control, or hybrid control for short, was first proposed by Raibert and Craig [11], based on an orthogonal decomposition of task space [12]. Improvements have been suggested since then [13], but the central concept remains the same. The position of the end-effector and the contact force with the environment along one DOF cannot be controlled independently. The task space is split into two subspaces, called the position-controlled and force-controlled subspaces. As the name would imply, positions are commanded and controlled along one subspace, and forces are commanded and controlled along the other. A selection matrix S is used to determine which DOF are to be position-controlled, and which are to be force-controlled.

B. Impedance Control

Impedance control is more an approach to force control than an established algorithm. Hogan realized that although both position and force cannot be controlled simultaneously, by controlling the amount of compliance, or "impedance" in the manipulator, the contact forces can be regulated [14].

Impedance control has been implemented in many forms. In its simplest form it can be considered a generalization of damping and stiffness control schemes [10]. In this form, it is essentially a PD position controller, with position and velocity feedback gains adjusted to obtain different apparent impedances. Unfortunately, given constant feedback gains, the robot impedance changes with configuration, due to the high nonlinearity of the robot dynamics, making it difficult to determine exactly what impedance the environment will see.

In order to achieve a constant, known impedance, independent of manipulator configuration, some sort of inverse dynamics is necessary. Hogan [15] suggests one way to implement this.

C. Problems with Earlier Approaches

Hybrid control is a highly intuitive approach to force control, which properly recognizes the distinction between force-controlled and position-controlled subspaces. The problem with hybrid control is its failure to recognize the importance of manipulator impedance.

Impedance control considers the effects of impedance on robot/environment interactions. When performed in task space, a known impedance can be maintained for all configurations. It is considered, however, to be solely a position-control scheme, with small adjustments made to react to contact forces. Positions are commanded, and impedances are adjusted to obtain the proper force response. No attempt is made to follow a commanded force trajectory and any distinction between force-controlled subspaces and position-controlled subspaces is ignored. Furthermore, the impedance schemes published to date consider only second-order impedances for the manipulator, where in some cases higher order impedances are desirable.

III. Hybrid Impedance Control

The control approach introduced in this section, namely, hybrid impedance control (HIC), combines impedance control and hybrid position/force control into one strategy, while allowing for more sophisticated impedances.

For the remainder of this correspondence we shall assume that an inverse dynamics controller has been implemented in task space, and has effectively decoupled the manipulator into single-DOF linear subsystems. When only linear systems are considered, systems concepts, such as Norton equivalence, Thèvenin equivalence, and impedance, may be readily applied to manipulator/environment interaction. Our interest will now be directed to these subsystems.

A. The System Approach

From a systems point of view, the input/output behavior of a linear continuous system of the type considered here is described by the ratio of two variables, effort (F) and flow ($v = \dot{x}$). For a mechanical system, effort is represented by force and torque, and flow is represented by linear and angular velocity. Motors and batteries are considered equivalent, in a system sense, both being effort sources. Similarly, a current generator or a rotating cam shaft are both flow sources.

Passive elements are characterized by resistance (B), capacitance (K), and inertia (M). Resistance represents the proportional relationship between effort and flow, $B = F/v$, capacitance represents the integral relationship between effort and flow, $K = F/\int v\, dt$, and inertia represents the differential relationship between effort and flow, $M = F/\dot{v}$. For linear, time-invariant continuous systems, the impedance Z may be defined as the ratio of the Laplace transform of the effort $F(s)$ to the Laplace transform of the flow $v(s)$. For nonlinear systems, the term impedance can still be used to describe the relationship between effort and flow. In this case, the impedance is operating point dependent. That is, the impedance of the nonlinear system is defined as the equivalent linear impedance for the system linearized about a particular operating point.

As any electrical engineer knows, the determination of how two complex subcircuits interact is greatly simplified when the subcircuits are represented by their Norton or Thèvenin equivalent circuits. The same is true for force control. We are concerned with how two complex systems, the robot and the environment, will react when connected together. In the spirit of systems theory, the robot and the environment can both be represented by Norton and Thèvenin equivalent circuit models.

B. Modeling the Environment

The environment is central to any force control strategy. For instance, no force algorithm, no matter how sophisticated, can command a force with the end-effector in free space, nor command a motion while the end-effector is held fast. The environment is usually modeled as a linear spring K_e which is sometimes in parallel with a dashpot B_e. Both are considered to be known and constant, and a force law is chosen accordingly. This may be valid when a rigid tool meets a compliant workpiece, but is far too simplistic to represent the entire spectrum of environments.

We will describe the robot and environment systems based on the low-frequency behavior of the impedance. For a linear environment, the impedance is defined as the ratio of the Laplace transforms of effort and flow. For any given frequency ω, this is a complex number with real part $R(\omega)$ and imaginary part $X(\omega)$

$$Z(\omega) = R(\omega) + jX(\omega). \tag{5}$$

As ω approaches zero, one of three things can happen to the magnitude of the environment's impedance. It can approach infinity, it can approach a nonzero finite number, or it can approach zero. We introduce the following definitions:

Definition 3.1: A system with impedance given by (5) is *inertial* iff $|Z(0)| = 0$.
Definition 3.2: A system with impedance given by (5) is *resistive* iff $|Z(0)| = c$, where $0 < c < \infty$.
Definition 3.3: A system with impedance given by (5) is *capacitive* iff $|Z(0)| = \infty$.

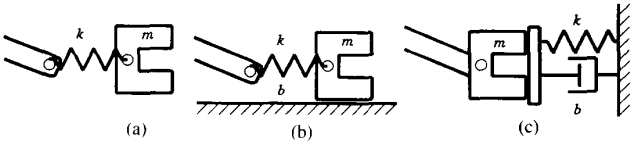

Fig. 2. Environment types. (a) Inertial. (b) Resistive. (c) Capacitive.

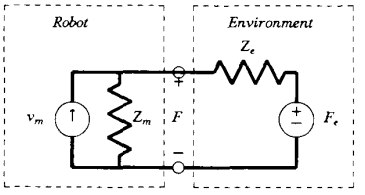

Fig. 3. Position-control model.

Fig. 2 shows examples of three environments (recall that the environment, by definition, includes everything past the wrist force sensor). The first environment is a mass-spring system with impedance $Z_e = mks/(ms + k)$ and, by application of the defintion, is inertial. The second environment consists of a compliant mass sliding across a viscous surface. It has an impedance of $Z_e = b + mks/(ms + k)$ and is resistive. The final environment consists of a damped, compliant surface with impedance $Z_e = ms + b + k/s$ and is, by definition, capacitive.

Capacitive and inertial environments represent dual impedances in the sense that the inverse of a capacitive system is inertial, and the inverse of an inertial system is capacitive. A resistive environment is self-dual. To represent this duality we shall use Norton and Thèvenin equivalents.

Recall that a Norton equivalent consists of an impedance in parallel with a flow source, and a Thèvenin equivalent consists of an impedance in series with an effort source. We shall use a Norton equivalent to represent a capacitive system, and conversely, we shall use a Thèvenin equivalent to represent an inertial system. Either representation will suffice for representing a resistive environment.

C. Duality

Once the environment has been properly modeled, the desired manipulator response may be determined. A fundamental goal for designing a controller is zero steady-state error to a step input. This will be obtained if we adhere to the following duality principle.

Duality Principle:
The manipulator should be controlled to respond as the dual of the environment.

This statement is most easily described in terms of Norton and Thèvenin equivalents. When the environment is capacitive we represent it as an impedance in parallel with a flow source, and the corresponding manipulator dual is an effort source in series with a noncapacitive impedance. When the environment is inertial we represent it as an impedance in series with an effort source, and the corresponding manipulator dual is a flow source in parallel with a noninertial impedance. When the environment is resistive, either equivalent may be used but the dual manipulator impedance must be nonresistive. Simply stated, capacitive environments require a force-controlled manipulator, inertial environments require a position-controlled manipulator, and resistive environments allow either position or force control. Once the type of servo control is known, the manipulator impedances should be chosen accordingly.

To show that this condition insures zero steady-state error to a step (assuming no environmental inputs) is straightforward. First assume that the environment is found to be inertial so that $Z_e(0) = 0$. Fig. 3 shows the environment, and the corresponding manipulator, where Z_e is the environment's impedance, Z_m is the manipulator's impedance, and v_m is the input velocity. The input/output transfer function

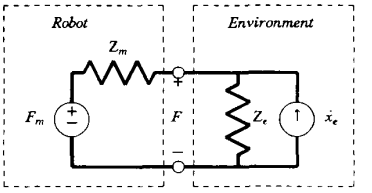

Fig. 4. Force-control model.

for velocity is given as

$$\frac{v}{v_m} = \frac{Z_m(s)}{Z_m(s) + Z_e(s)}. \tag{6}$$

Assuming stability, the steady-state error to a step input $1/s$ is given by the final value theorem as

$$e_{ss} = \lim_{t\to\infty} (v - v_m) = \frac{-Z_e(0)}{Z_m(0) + Z_e(0)} = 0 \tag{7}$$

as long as $Z_m(0) \neq 0$, i.e., as long as the manipulator impedance is noninertial. Next assume that the environment is capacitive so that $Z_e(0) = \infty$. Fig. 4 shows the environment and the corresponding manipulator. The input/output transfer function for force is given by

$$\frac{F}{F_m} = \frac{Z_e(s)}{Z_m(s) + Z_e(s)} \tag{8}$$

and the steady-state error to a step is likewise given by

$$e_{ss} = \lim_{t\to\infty} (F - F_m) = \frac{-Z_m(0)}{Z_m(0) + Z_e(0)} = 0 \tag{9}$$

as long as $Z_m(0)$ is finite, i.e., as long as the manipulator impedance is noncapacitive. It is simple to show that zero steady-state error is also obtained for resistive environments, as long as either $Z_m(0) = 0$ and the manipulator is force-controlled, or $Z_m(0) = \infty$ and the manipulator is position-controlled.

The duality condition encompasses the notion that neither two different flows nor two different efforts can be maintained simultaneously at the junction of a one-port. An environment following a position trajectory and a position-controlled robot attempting to track this environment are inconsistent. The dual combination of one Norton equivalent flow source and a Thèvenin equivalent effort source, however, can exist simultaneously. Although we have only discussed linear environments, duality is also a desirable property for nonlinear manipulators [14].

HIC requires that duality be maintained along every degree of freedom of the manipulator. This duality principle will allow us to determine, based on the character of the environment, when to use position control or force control in a given task. To demonstrate the application of the duality principle in a force-control problem we give two examples.

First, consider the setup shown in Fig. 5. The environmental impedance is given by $M_e s + B_e$. Because the environment is resistive, the system can be either force-controlled or position-controlled. Duality requires that if we choose to position-control the manipulator then the manipulator impedance should be capacitive, such as $Ms + B + K/s$. The resulting velocity through the environment port is given by

$$\frac{v}{v_m} = \frac{Bs + K}{(M_e + M)s^2 + (B + B_e)s + K}. \tag{10}$$

As a second example consider the system shown in Fig. 6. The environmental impedance is given by $B_e + K_e/s$. Furthermore, we shall assume that the environmental velocity v_e has a high noise content over a narrow band. Because the environment is capacitive it should be modeled by its Norton equivalent. Duality demands that the manipulator appears as a Thèvenin equivalent system, a force source

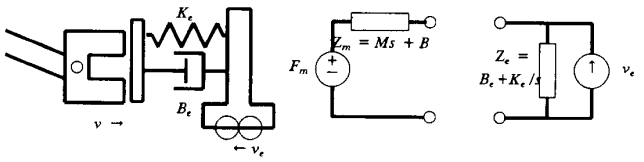

Fig. 5. Inertial environment example. (a) Mechanical representation. (b) Circuit model for manipulator. (c) Circuit model for environment.

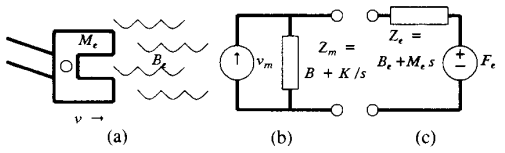

Fig. 6. Capacitive environment example. (a) Mechanical representation. (b) Circuit model for manipulator. (c) Circuit model for environment.

in series with a noncapacitive impedance. In order to filter out the environmental noise (which might occur due to mechanical vibrations in a grinding wheel, for instance) we choose

$$Z_m = Ms + B\frac{(s+\omega)^2}{s^2+\omega^2} \tag{11}$$

where the second term is tuned to the environmental noise frequency. Upon connecting the manipulator port with the environment port the resulting force is given by

$$F = \frac{(s^2+\omega^2)(B_e s+K_e)}{(Ms^2+B_e s+K_e)(s^2+\omega^2)+Bs(s+\omega)^2}F_m - \frac{Ms^2(s^2+\omega^2)(B_e s+K_e)+B(s+\omega)^2(B_e s+K_e)}{(Ms^2+B_e s+K_e)(s^2+\omega^2)+B(s+\omega)^2}v_e. \tag{12}$$

For $F_m = 0$ we see that

$$v = -\frac{Z_e}{Z_m+Z_e}v_e = \frac{(B_e s+K_e)(s^2+\omega^2)}{(Ms^2+B_e s+K_e)(s^2+\omega^2)+B(s+\omega)^2}v_e \tag{13}$$

and the manipulator is responsive to all environmental motions, except for those motions in the narrow band of disturbances.

Position-controlled and force-controlled subsystems are shown in Figs. 3 and 4, respectively. The control engineer has no influence over either the environment's impedance Z_e or the environment sources v_e and F_e. The control engineer can, however, determine Z_m by properly designing a controller and feeding back state variables. How this may be done, without interfering with the inner loop controller, is the subject of the next two sections.

D. Position-Controlled Subsystem

The transfer function for the position-controlled circuit shown in Fig. 3 is

$$v = \frac{Z_m}{Z_m+Z_e}v_m + \frac{1}{Z_m+Z_e}F_e. \tag{14}$$

This response can be realized be feedback of the contact force, combined with information about the desired acceleration for the manipulator. Fig. 7 shows a block diagram of the position-control implementation.

The commanded acceleration a is given implicitly by

$$a = \dot{v} = \frac{d}{dt}\left(v_m - \frac{F}{Z_m}\right). \tag{15}$$

In practice, we would like to obtain the control a explicitly, without differentiators and using only measurements of F, x, and v. This is possible if the impedance can be written as

$$Z_m = Ms + Z_{\text{rem}} \tag{16}$$

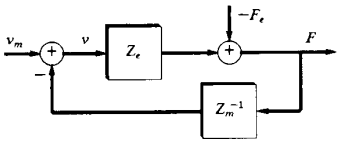

Fig. 7. Position-control block diagram.

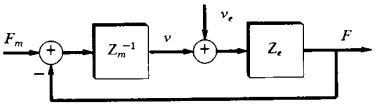

Fig. 8. Force-control block diagram.

where the remaining impedance Z_{rem} is proper. It should be mentioned here that Z_m must still satisfy the duality condition, and will do so only if Z_{rem} contains the appropriate capacitive elements. The position control law (15) can now be rewritten as

$$a = \frac{d}{dt}\left(v_m - \frac{F}{Ms+Z_{\text{rem}}}\right) = \dot{v}_m + \frac{1}{M}Z_{\text{rem}}(v_m - v) - \frac{1}{M}F. \tag{17}$$

As an example consider the system in Fig. 5, where the manipulator impedance was given as $Z_m = Ms + B + K/s$. Here, $Z_{\text{rem}} = B + K/s$, and the commanded acceleration a is,

$$a = \dot{v}_m + \frac{B}{M}(v_m - v) + \frac{K}{M}(x_m - x) - \frac{1}{M}F. \tag{18}$$

The position control law (18) is seen to be equivalent to Hogan's impedance control law [15].

E. Force-Controlled Subsystem

The transfer function for the force-controlled subsystem shown in Fig. 4 is

$$F = \frac{Z_e}{Z_m+Z_e}F_m + \frac{Z_e Z_m}{Z_m+Z_e}v_e. \tag{19}$$

This response can be realized by feedback of the force signal, in combination with a controller with transfer function Z_m^{-1}. Fig. 8 shows a block diagram of the resulting system. The outer loop control a is defined implicitly by

$$a = \frac{d}{dt}\frac{(F_m - F)}{Z_m} \tag{20}$$

where the force F is a measured quantity given by

$$F = Z_e(v - v_e) \tag{21}$$

and F_m is the input command corresponding to a desired Thèvenin equivalent force. Note that if the inputs v_m and F_m are set to zero then the force control law (20) and the position control law (15) are identical.

As before, we would like to obtain a without differentiators, and using only measurements of F, x, and v. Again this is possible if the impedance can be written as

$$Z_m = Ms + Z_{\text{rem}} \tag{22}$$

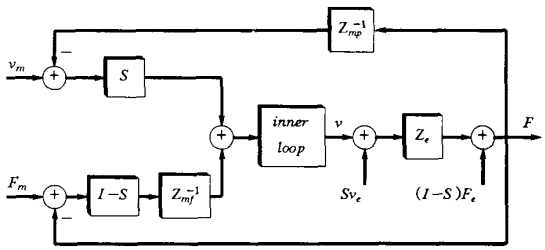

Fig. 9. Hybrid impedance block diagram.

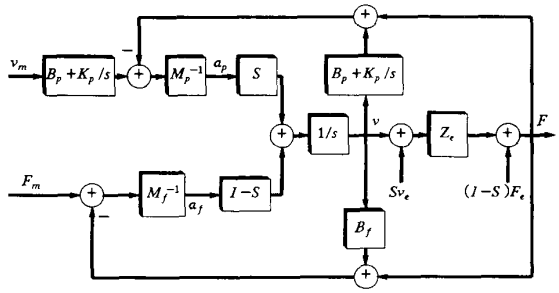

Fig. 10. HIC controller for two-link arm.

where the remaining impedance Z_{rem} is proper. In this case we can rewrite our control law (20) as

$$a=\frac{d}{dt}\frac{(F_m-F)}{Ms+Z_{\text{rem}}}=-\frac{1}{M}Z_{\text{rem}}v+\frac{1}{M}(F_m-F). \tag{23}$$

As an illustration, consider the system shown previously in Fig. 6, in which the robot impedance was chosen to be $Z_m = Ms + B(s + \omega)^2/(s^2 + \omega^2)$. The commanded acceleration a is given by

$$a=M^{-1}((F_m-F)-B(v-v_f)) \tag{24}$$

where v_f is an additional filtering compensation term obtained from the filter $v_f = 2\omega s/(s^2 + \omega^2)v$.

F. Combining the Subsystems

Now that the force-controlled and position-controlled subsystems have been described in detail for a single DOF, the entire system may be put together. The hybrid system is similar to Raibert's and Craig's hybrid controller, but with control loops operating in task space rather than in joint space. This has been achieved by using an inverse dynamics inner loop controller with additional end-effector force cancellation. A selection matrix S, which consists of ones and zeros down the diagonal, is used to separate the force-controlled and position-controlled subspaces.

It should be reiterated that as the manipulator encounters different environments, different controller gains are necessary to maintain a desired response. A robot packaging eggs needs far more internal compliance than a robot packaging cold cuts. An intelligent controller would not just determine the elements of the selection matrix, but should also be actively adjusting system parameters. This might be done by adaptive control, table look-up, or other means. The details of such an implementation, however, are beyond the intended scope of this work.

The hybrid impedance control algorithm for a six-DOF manipulator is given in block diagram form in Fig. 9. Here, the robot impedance terms Z_{mp} and Z_{mf} represent diagonal matrices with terms equal to the impedance along each degree of freedom. The control signal a is now a vector and may be defined implicitly as

$$\begin{aligned}a&=Sa_p+(I-S)a_f\\&=S\frac{d}{dt}(v_m-Z_{mp}^{-1}F)+(I-S)\frac{d}{dt}Z_{mf}^{-1}(F_m-F)\end{aligned} \tag{25}$$

which to be implemented should be of the form

$$\begin{aligned}a=S(\dot{v}_m+M_p^{-1}Z_{\text{rem}\,p}(v_m-v)-M_p^{-1}F)\\-(I-S)(M_f^{-1}Z_{\text{rem}\,f}v+M_f^{-1}(F_m-F)).\end{aligned} \tag{26}$$

IV. Simulations

In order to demonstrate the implementation of HIC, an assembly task was simulated. The robot arm is modeled in continuous time, using a second-order Runge–Kutta integration scheme, while the control is implemented in discrete time. The simulations were run in Fortran.

The two-link revolute-prismatic arm (Fig. 11), was chosen since it was the simplest nonlinear manipulator which could be simultaneously force- and position-controlled. The first subsection develops the HIC equations for this manipulator. The second subsection shows the results of a simulation involving the two-link manipulator and an assembly task.

A. The Two-Link Manipulator

The dynamics and kinematics for the circular manipulator have been derived from the Euler–Lagrange equation. The actuators are assumed to be ideal, and the effects of friction have been ignored. The dynamics equation for this system is

$$\begin{bmatrix}u_1\\u_2\end{bmatrix}=\begin{bmatrix}I_1&0\\0&m_2\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\\ddot{q}_2\end{bmatrix}+\begin{bmatrix}2m_2q_2\dot{q}_2\dot{q}_1+m_2gq_2\cos(q_1)\\-m_2q_2\dot{q}_1^2-m_2g\sin(q_1)\end{bmatrix}. \tag{27}$$

The inverse dynamics equation, (3), can be applied with $D(x)$ and $h(x, v)$ given by

$$D(x)=\begin{bmatrix}m_2+\dfrac{\sin^2(q_1)}{q_2^2}I_1 & \dfrac{-I_1\sin(q_1)\cos(q_1)}{q_2^2}\\[2ex]\dfrac{-I_1\sin(q_1)\cos(q_1)}{q_2^2} & m_2+\dfrac{\cos^2(q_1)}{q_2^2}I_1\end{bmatrix} \tag{28}$$

$$h(x,v)=\begin{bmatrix}2\sin(q_1)m_2\dot{q}_2\dot{q}_1-2\cos(q_1)m_2q_2\dot{q}_1^2\\-2\cos(q_1)m_2\dot{q}_2\dot{q}_1-2\sin(q_1)m_2q_2\dot{q}_1^2-m_2g\end{bmatrix}. \tag{29}$$

In free space, the end-effector impedance is given by $Z_e = mK_eS(mIs^2 + K_e)^{-1}$, and the second-order manipulator impedance was chosen accordingly as $Z_{mp} = M_ps + B_p + K_p/s$, where M_p, B_p, and K_p are all diagonal matrices.

When the end-effector comes in contact with a stiff surface, the end-effector impedance becomes $Z_e = K_e/s$ and the corresponding manipulator impedance is chosen as $Z_{mf} = M_fs + B_f(M_fs^2 + B_fs + K_f)^{-1}$. The inverse dynamics inputs a are obtained as in Section I-B. In two dimensions, the position subspace inputs are

$$\begin{aligned}a_p=&\begin{bmatrix}\dot{v}_{m1}\\\dot{v}_{m2}\end{bmatrix}+M_p^{-1}\begin{bmatrix}B_p&0\\0&B_p\end{bmatrix}\begin{bmatrix}v_{m1}-v_1\\v_{m2}-v_2\end{bmatrix}\\&+M_p^{-1}\begin{bmatrix}K_p&0\\0&K_p\end{bmatrix}\begin{bmatrix}x_{m1}-x_1\\x_{m2}-x_2\end{bmatrix}-\begin{bmatrix}F_1\\F_2\end{bmatrix}\end{aligned} \tag{30}$$

and the force subspace inputs are

$$a_f=M_f^{-1}\begin{bmatrix}F_{m1}-F_1\\F_{m2}-F_2\end{bmatrix}-B_fM_f^{-1}\begin{bmatrix}v_1\\v_2\end{bmatrix}. \tag{31}$$

The resulting control signal a is obtained from a_f and a_p as in (26). A block diagram of the HIC controller with the inner loop suppressed is shown in Fig. 10.

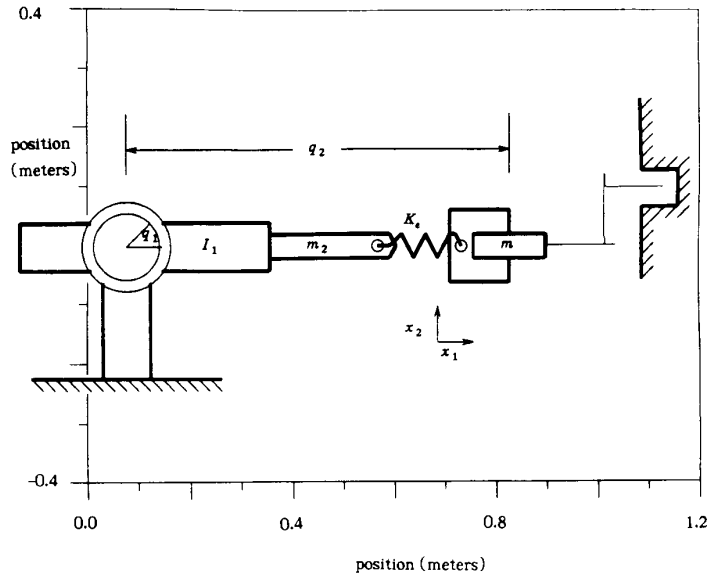

Fig. 11. Manipulator and assembly task.

B. Assembly Problem

The peg-in-the-hole problem is a classic problem, which involves moving a peg to a surface, moving along the surface until a hole is discovered, centering over the hole, and finally driving the peg into the hole. What is easily done by a human is not so easily implemented in a robot. Very subtle mechanisms are used to monitor forces. Slight adjustments are made to prevent jamming, and excessive forces. Determining algorithms is difficult because the task is done on such a low level. We do not think out our reflexes. Nevertheless, a robust systematic approach to the peg-in-the-hole problem may be implemented using hybrid impedance control.

The task may be broken into six stages:

1) Move to the surface along the x_1 direction, beneath the peg-hole with constant impedance along both DOF. When measured forces exceed preset thresholds, the surface has been contacted.
2) With the x_1-axis force controlled and the x_2-axis position controlled, maintain constant position, and let the force along the x_1 direction settle to within a reasonable bound.
3) Proceed in the x_2 direction, with constant force in the x_1 direction. Maintain a reasonable impedance along x_2 so that friction does not cause skipping. Continue until the x_1 position jumps sharply. The hole has been found.
4) Switch the x_2 direction to force control, and the x_1 position to position control. Back away from the far wall of the hole by commanding the force in the x_2 direction to zero. Keep the x_1 position constant.
5) Once the forces along x_2 have settled, proceed into the hole in the positive x_1 position. Maintain sufficient flexibility so that the peg does not jam. Continue until the force in the x_1 direction exceeds preset bounds.
6) Finally, let the forces felt from exerting pressure on the bottom of the hole settle to within a prescribed minimum. The peg is now securely in the hole.

This algorithm was implemented in Fortran using the HIC strategy. Instead of changing the impedance to best match each stage of the task, however, a constant force impedance, and a constant position impedance were used.

The results of the simulations are shown in Figs. 11–13. The six stages begin at $t = 0.0$, $t = 0.95$, $t = 1.50$, $t = 2.15$, $t = 2.9$, and $t = 3.5$ s, respectively. Fig. 11 shows the manipulator operating in the plane. The environment with peg-hole is also shown. The solid black line represents the movement of the wrist. Fig. 12 shows both the wrist position along x_1 and the desired wrist position along x_1 as a function of time. It is shown that when x_1 is position-controlled, x_1 tracks the desired position. When movement along the x-axis is force-controlled, however, actual and desired positions deviate. Fig. 13 shows the force and position in the x_1 direction both plotted against time. When movement is force-controlled, the force approaches the desired force exponentially. Small oscillations in force are due to vibrations of the passive end-effector spring.

V. Conclusion

Hybrid impedance control represents a unification of many previous approaches to force and position control of robotics. By combining recent advances in inverse dynamics with already existing force control strategies, such as impedance control and hybrid control, a cohesive general plan has been developed.

The careful modeling of the environment has been shown to be instrumental in determining the proper control strategy. Proper models with appropriate controls make it possible for output responses to match input commands in the steady state. Duality between Thèvenin and Norton equivalents insures that a robot is being controlled consistently with respect to the environment.

The implementation of HIC was demonstrated for a simple two-link manipulator, applied to the peg-in-the-hole task. Inverse dynamics in task space were calculated, and the general impedance notation used in the development of HIC was made specific in terms of spring and damping constants.

Despite the generality of hybrid impedance control, a number of issues remain. In the simulations, a constant impedance was maintained, and only force and position commands were adjusted. Human impedances change constantly, however, and such an approach might prove advantageous in robotics. Programming is also significantly more difficult since it is hard to understand a task in terms of applied forces. Perhaps an AI system could be developed for determining appropriate force and position commands for a certain task. Furthermore, numerous simplifications were made in the model. Gear backlash, flexibility, static friction, torque constraints, and noise all detract from the ideal system. It was assumed that a devoted inverse dynamics controller existed which not only computed inverse dynamics in task space, but also included measured force signals as part of the process.

Hybrid impedance control is just one step towards developing more

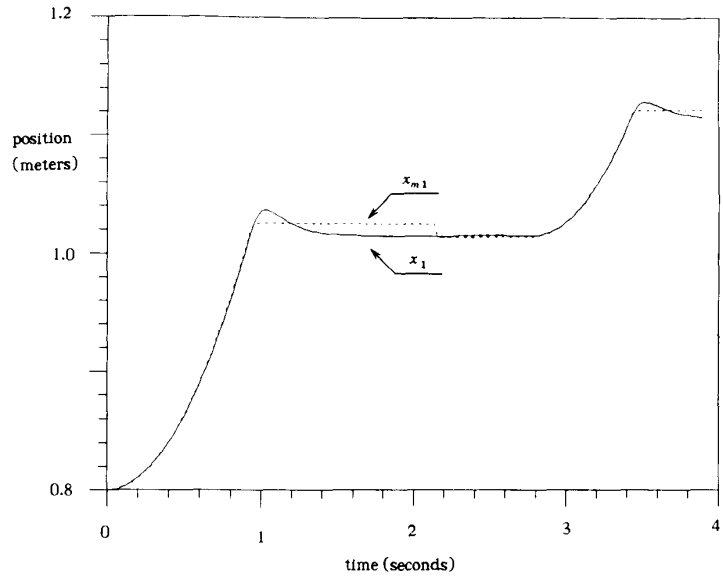

Fig. 12. Wrist position during assembly.

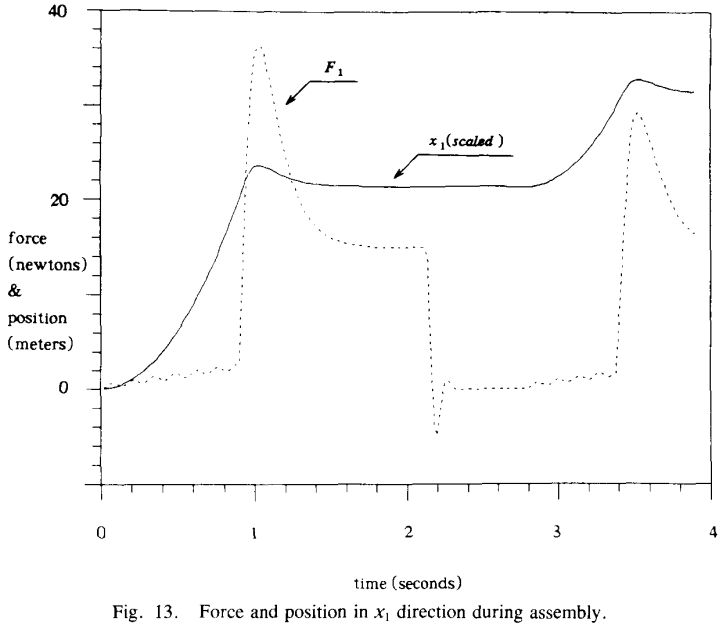

Fig. 13. Force and position in x_1 direction during assembly.

general and more sophisticated robots. Further reduction in the cost of computing power and the increased availability of sensors should insure, however, the future of such control strategies.

References

[1] B. Shimano and B. Roth, "On force sensing information and its use in controlling manipulators," in *Proc. 8th Int. Symp. on Industrial Robots* (Washington, DC), pp. 119–126.

[2] A. K. Bejczy, "Robot arm dynamics and control," NASA-JPL Tech. Memo., 33–69, 1974.

[3] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," in *Proc. 1986 Int. Conf. on Robotics and Automation*, pp. 1381–1386.

[4] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-25, no. 3, pp. 195–200, June 1980.

[5] S. Dubowsky and D. T. DesForges, "The application of model-reference adaptive control to robotic manipulators," *Trans. ASME, J. Dyn. Syst., Meas. Contr.*, vol. 101, pp. 193–200, Sept. 1979.

[6] J. Slotine, "Robustness issues in robot control," in *Proc. 1985 Int. Conf. on Robotics and Automation* (St. Louis, MO), pp. 656-661.

[7] M. Spong, J. Thorp, and J. Kleinwaks, "Robust microprocessor control of robot manipulators," *Automatica*, vol. 23, no. 3, pp. 373–379, 1987.

[8] M. Spong, "Robust stabilization for a class of nonlinear systems," in *Theory and Applications of Nonlinear Control Systems*. Amsterdam, The Netherlands: North-Holland, 1986, pp. 155–165.

[9] R. J. Anderson, "Hybrid admittance/impedance force control of robotic manipulators," Masters thesis submitted to the University of Illinois at Urbana-Champaign, Oct. 1986.

[10] D. E. Whitney, "Historical perspective and state of the art in robot force control," in *Proc. 1985 IEEE Int. Conf. on Robotics and Automation* (St. Louis, MO), pp. 262–268.

[11] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *J. Dyn. Syst. Contr.*, vol. 102, June 1981.
[12] M. T. Mason, "Compliance and force control for computed controlled manipulators," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-11, no. 6, pp. 418-432, June 1981.
[13] H. Zhang and R. P. Paul, "Hybrid control of robot manipulators," in *Proc. 1985 IEEE Int. Conf. on Robotics and Automation* (St. Louis, MO), pp. 602-607.
[14] N. Hogan, "Impedance control: An approach to manipulation," *J. Dyn. Syst., Meas., Contr.*, vol. 107, pp. 1-7, Mar. 1985.
[15] ——, "Beyond regulators: Modeling control systems as physical systems," preprint. Nov. 1986.

A Variable Structure Model Following Control Design for Robotics Applications

K-K. DAVID YOUNG

Abstract—**The increased demand on robotic system performance leads to the use of advanced control strategies. In this communication a variable structure model following control design is proposed for robotics applications.**

Introduction

Adaptive control methodologies have been recently proposed for robotic manipulator control design [1]-[7]. These existing designs are based on one of the following adaptive design methodologies for linear plants with unknown parameters. They are the design methods based on hyperstability theory, the Lyapunov stability methods, and the self-tuning regulator type techniques. While these methods, in general, provide an adaptive controller that stabilizes the linear plant, transient response performance of the closed-loop system cannot be specified directly in the design. Transient response specifications in these designs are limited to specifying the speed of convergence of the output error norm to zero which is typically related to the magnitude of the adaptive controller gains. A variable structure following control (VSMFC) system design was first proposed in [8] as an alternative to adaptive model following control (AMFC) design. The work presented in [9] provides the later developments on this approach. The advantage of the VSMFC design lies in its ability to prescribe transient response requirements as well as providing a stabilizing controller. In these earlier works, linear plants with unknown plant parameters constitute the class of systems for which VSMFC design was derived. In this communication we consider the extensions of VSMFC design to nonlinear plants and, in particular, for robotics applications.

VSMFC Formulation

Robotic system dynamics are generally described by the following nonlinear differential equations:

$$J(\theta)\ddot{\theta} = G(\theta)g + F(\theta, \dot{\theta}) + u \tag{1}$$

Manuscript received November 20, 1986; revised January 25, 1988. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48. A preliminary version of this communication was presented at the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 7-10, 1986.

The author is with the Lawrence Livermore National Laboratory, University of California, Livermore, CA 94550.

IEEE Log Number 8821743.

where θ is a vector of m-joint shaft angular displacements, u is a vector of m control torques, $J(\theta)$ is the effective moment of inertia matrix, $G(\theta)g$ represents the gravitational torques, and $F(\theta, \dot{\theta})$ denotes the velocity-dependent torques. Defining x to be the state vector

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{2a}$$

the corresponding state-space model is

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ J(\theta)^{-1}[G(\theta)g + F(\theta, \dot{\theta})] \end{bmatrix} + \begin{bmatrix} 0 \\ J(\theta)^{-1} \end{bmatrix} u. \tag{2b}$$

In adaptive model following control (AMFC) design, the desired behavior of the plant (the robotic system) is expressed through the use of a reference model driven by a reference input. Typically, linear models are used. For robotic applications, we select a reference model of the form

$$\dot{x}_r = \begin{bmatrix} 0 & I \\ A_{r_1} & A_{r_2} \end{bmatrix} x_r + \begin{bmatrix} 0 \\ B_{r_2} \end{bmatrix} r \tag{3a}$$

where

$$x = \begin{bmatrix} \theta_r \\ \dot{\theta}_r \end{bmatrix} \tag{3b}$$

and dim θ_r = dim θ = m and dim r = p. This reference system is asymptotically stable when $r = 0$ and it is controllable with respect to the reference input r. The goal of the AMFC design is then to force θ and $\dot{\theta}$ to track the desired reference model trajectories θ_r and $\dot{\theta}_r$, respectively.

In VSMFC design, the goal is the same as in AMFC design; the approach used in accomplishing this goal is, however, different: it exploits a variable structure feedback control mechanism which induces sliding mode to occur during the transient process in which θ approaches θ_r. The concept of VSMFC design for linear plants with unknown parameters was introduced in [8]. This concept is readily applicable to handle the types of nonlinear plants, exemplified by the robotic system equation (1). The theory of variable structure systems and sliding mode has first been shown in [10] to be powerful for designing robot arm controllers. Further investigations [11] confirmed the utility of sliding mode in robot arm control design.

A VSMFC design for robotics applications, in general, involves a variable structure feedback control of the type

$$u_i(x, x_r, r) = \begin{cases} u_i^+(x, x_r, r), & s_i(e) > 0 \\ u_i^-(x, x_r, r), & s_i(e) < 0 \end{cases} \tag{4}$$

for $i = 1, \cdots, m$, where

$$e \equiv x_r - x \tag{5}$$

denotes the tracking error. The feedback functionals in (4) are discontinuous across switching hyperplanes s_i, $i = 1, \cdots, m$ which are designed to be hyperplanes in the tracking error space; $s_i(e)$ denotes the ith component of

$$s = Ge. \tag{6}$$

The design objective of a VSMFC system is to determine the feedback functionals u_i^+, u_i^-, and the switching hyperplane matrix G such that sliding mode occurs on the switching hyperplanes, the tracking error has an acceptable transient response, and it goes to zero asymptotically as $t \to \infty$. For this design, it is not necessary to know the functionals $J(\theta)$, $G(\theta)$, and $F(\theta, \dot{\theta})$ except for element-wise upper and lower bounds. Since $J(\theta)$ is positive definite for all θ, the