where A and B are fuzzy subsets of X and Y, respectively. This transforms into the possibility distribution

$$\pi_{u|v}(x,y) = (H(x,y) \wedge \alpha) + (1-\alpha)$$

where

$$H(x,y) = \min(1, 1 - A(x) + B(y)).$$

Conclusion

We have discussed the applicability of the theory of approximate reasoning to rule-based expert systems. The novel aspects of this work concerns our introduction of an approach in this framework for the inclusion of complex rules and the ability to introduce certainty qualification into our system.

References

[1] B. G. Buchanan and R. O. Duda, "Principles of rule-based expert systems, Lab. for Artificial Intelligence Research, Fairchild Camera, Palo Alto, CA, Fairchild Tech. Rep. no. 626, 1982.

[2] B. G. Buchanan, "Partial bibliography of work on expert systems," *Sigart Newsletter*, no. 84, pp. 45–50, 1983.

[3] E. H. Shortliffe, *Computer Based Medical Consultations: MYCIN*. New York: American Elsevier, 1976.

[4] R. Davis, B. G. Buchanan, and E. H. Shortliffe, "Production rules as a representation of a knowledge-based consultation program," *Artificial Intelligence* 8, pp. 15–45, 1977.

[5] W. Van Melle, "A domain independent system that aids in constructing knowledge based consultation programs," Ph.D. dissertation, Stanford Univ. Computer Science Dept., Stanford, CA, Stan CS-80-820, 1980.

[6] L. A. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese 30*, pp. 407–428, 1975.

[7] ——, "The concept of a linguistic variable and its application to approximate reasoning," *Information Science 8 and 9*, pp. 199–249, pp. 301–357, pp. 43–80, 1975.

[8] ——, "A theory of approximate reasoning," in Machine Intelligence 9, J. E. Hayes, D. Michie, and L. I. Kulich, Eds. New York: Wiley 1979, pp. 149–194.

[9] ——, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[10] ——, "PRUF—A meaning representation language for natural languages," *Int. J. of Man–Machine Studies*, vol. 10, pp. 395–460, 1978.

[11] R. R. Yager, "Querying knowledge base systems with linguistic information via knowledge trees," *Int. J. Man–Machine Studies*, vol. 19, 1983.

[12] ——, "Knowledge trees in complex knowledge bases," Machine Intelligence Institute, Iona College, New Rochelle, NY, Tech. Report MII-209, 1982.

[13] L. A. Zadeh, "Fuzzy sets," *Informat. and Contr.*, vol. 8, pp. 338–353, 1965.

[14] ——, "A computational approach to fuzzy quantifiers in natural languages," *Comp. and Machs. with Appl.*, vol. 9, pp. 149–184, 1983.

[15] R. R. Yager, "Quantified propositions in a linguistic logic," in *Proc. 2nd Int. Seminar on Fuzzy Set Theory*, E. P. Klement, Ed. Linz, Austria, 1980.

[16] ——, "General multiple objective decision functions and linguistically quantified statements," Iona College, New Rochelle NY Tech. Rep. MII-302, 1983.

[17] ——, "Generalized probabilities of fuzzy events from fuzzy belief structures," *Inform. Sciences*, vol. 28, pp. 45–62, 1982.

[18] ——, "Measuring tranquility and anxiety in decision making. An application of fuzzy sets," *Int. J. of General Systems*, vol. 8, pp. 139–146, 1982.

Dynamic Modeling of a Three Degree-of-Freedom Robotic Manipulator

HENRY W. STONE AND CHARLES P. NEUMAN

***Abstract*—In this correspondence, transfer function models that can provide physical insight into the dynamic behavior of manipulators and establish a foundation for control system analysis and design are identified. The literature abounds with algorithms to compute, in *real-time*, the joint torques to move a manipulator along a prescribed cartesian trajectory. Emphasis is directed toward increasing the computational efficiency of the manipulator dynamic equations of motion. What is lacking for control engineering applications are physical interpretations of manipulator dynamics. A conceptual three degree-of-freedom manipulator, which represents state-of-the-art configurations, is designed and simulated on a digital computer. The Lagrangian formulation is applied to develop the complete dynamic equations of motion of the manipulator. The simulation model includes the dynamic interactions of the centrifugal, Coriolis, inertial, and gravitational generalized forces along with the kinematic coupling, and it can accommodate load variations. The simulation model is interfaced to the Steiglitz–McBride identification algorithm to extract, from the complete nonlinear model, simplified linear models that characterize the arm in various regions of the work space. The simplified linear models are second-order discrete-time (input–output) transfer functions. The structure and variations of the transfer function parameters in the work space provide physical insight into manipulator dynamics. These models, in turn, establish an engineering foundation in which to evaluate the applicability of alternative controller designs for robotic manipulators.**

Manuscript received October 24, 1983; revised February 1984. This work has been supported, in part, by the Department of Electrical and Computer Engineering and Robotics Institute, Carnegie-Mellon University.

The authors are with the Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213.

I. Introduction

A robot dynamics and control engineering project at Carnegie-Mellon University centers around the dynamic modeling (through computer simulation and hardware experimentation) and feedback control system design and evaluation for robotic manipulators. Control system design for high-performance manipulators requires thorough understanding of robot dynamics [7]. The dynamic equations of motion of robots are highly coupled and nonlinear. The literature abounds with theoretical approaches and both recursive and nonrecursive computational algorithms for formulating these dynamic models [5], [12], [15], [32]. The control engineer needs simplified models that highlight the dynamic interactions. Such models must provide physical insight for control system analysis and design. The objective of this paper is to model and analyze the dynamics of a three degree-of-freedom robotic manipulator, and extract *linear* discrete transfer function models of the complex dynamics [29]. Such models are found to characterize the behavior of the manipulator and provide essential information for control system analysis and design. Interpretation of the features of these models provides a framework in which to evaluate the applicability of control system designs.

II. Strategy

The goal of this work is to develop engineering insight into the dynamic models of robotic manipulators for the design of feedback controllers. A methodology and software tools (including a dynamic simulator and identification algorithm) are developed and applied to identify and interpret *linear*, discrete-time, input–output (transfer function and difference equation) models of robotic manipulators for control engineering applications. Physical understanding of manipulator dynamics is a central robot control engineering problem. Engineers have traditionally designed feedback controllers for linear(ized) systems. Analysis and interpretation of the qualitative characteristics of linear input–output models have become standard engineering practice and these characteristics provide physical insight for control system design. This paper addresses the *complete* dynamic model of the physical manipulator. The characteristics of the linear models are applied to interpret the dynamic behavior of the manipulator in the work space.

Dynamic robot models are formulated in [5], [11], [12], [18], [19], [32]. In the literature, two approaches are followed to obtain *linear* dynamic models of manipulators. The first approach makes simplifying assumptions about the contribution of the generalized force components to the complete dynamic robot model. For

0018-9472/84/0700-0643$01.00

instance, Takegaki and Arimoto [30] assume that the contribution of the centrifugal and Coriolis torques to the overall dynamics is negligible. This assumption contradicts a fundamental *time-scaling* property of manipulator dynamics [6]. Bejczy [2] and Paul [23] simplify the mechanical representation of the manipulator to eliminate the nonlinearities or coupling. Since the resulting models lack the dynamic characteristics of robots, few implications can be drawn for the dynamic modeling and control of manipulators. The second approach linearizes mathematically the complete dynamic robot model around an operating point or trajectory. The range of applicability of these small-signal models is limited and many linearized models are required to span the work space.

The methodology developed in this work differs significantly from the aforementioned approaches. Even though the physical and mathematical structures of the complete dynamic robot model are analytically coupled and nonlinear, observed transient responses of robot dynamics appear to resemble transient responses of linear systems. The search for linear models is motivated by the application of the Weierstrass approximation theorem [3] to the continuous temporal nature of the joint positions, velocities and accelerations in response to the driving joint actuators. According to the Weierstrass approximation theorem, *any continuous function can be approximated uniformly by a polynomial*, and hence by the solution of a linear time-invariant (constant parameter) differential equation. The linear (discrete-time) input-output models are sought to characterize each joint as a single-input–single-output system. The input is the joint actuator voltage and the output is the joint position (and velocity). The input is excited, in simulation, by white Gaussian noise [8] and the output is processed by the Steiglitz–McBride regression algorithm [27] to extract the parameters of the model. The input-output models differ conceptually from the physical state–space models developed in the aforementioned approaches. The identified transfer functions are found (in Section VI) to characterize the manipulator over large regions of the state space. The white noise identification alogirithm insures that the transfer function models are applicable for *all* physical inputs.

The methodology and software package are illustrated through a coordinated plan of activities. To extract the salient dynamics of a realistic robot, a conceptual model of a three degree-of-freedom manipulator arm is designed. While a commercially available manipulator could, in principle, be considered, specification of *all* of the physical parameters (for the simulator) detracts from the essence of the model building activity. Instead, a conceptual arm is conceived to model a state-of-the-art articulated robot (such as the Unimate/Westinghouse Puma 250) and provide a framework in which to interpret manipulator dynamics [29]. The Lagrangian formulation [11], [23] is applied to develop the complete dynamic model of the arm. The model incorporates the kinematic coupling, and the dynamic interactions of the centrifugal, Coriolis, gravitational, and inertial coupling and exogeneous actuator torques that act on the individual links. The model includes realistic features (such as the three dimensional inertia distribution and offsets between adjacent links) that are inherent in state-of-the-art configurations. The actuators are DC torque motors that are modeled by second-order transfer functions. The complete (coupled and nonlinear) dynamic model (which includes ideal gears) is ninth-order; the state variables are the three joint angular positions, the three link velocities, and the three motor currents. This state–space model is the foundation of the arm simulator.

A computer program is developed (in Fortran and the C programming language) to simulate the mathematical model. The simulator produces the temporal profile of the joint angles and link velocities that are processed to identify the parameters of the transfer function models. To facilitate the analysis and interpretation of the manipulator dynamics, each joint (in the simulator) can be constrained independently with a constant angular position, velocity or acceleration, or free to respond to specified joint actuator inputs. A component of the software package implements the Steiglitz–McBride identification algorithm [27]. The identification program uses *synchronously* sampled input and output data from the simulator to identify the transfer functions. Generating, analyzing, and interpreting the models is the focal point of this correspondence.

The Steiglitz–McBride algorithm is applied to identify linear transfer function models of the manipulator in various regions in the work space. Transfer function modeling of manipulator dynamics (which is motivated by the Weierstrass approximation theorem) raises five fundamental issues that are addressed in Section VI. First, what model orders are required to ensure that the linear models will adequately characterize the manipulator? The practical utility of the linear models will decrease as their orders increase. Second, will the model orders vary over the work space? Third, over how large a region in the work space will a particular model be applicable? Fourth, and perhaps most importantly, can the characteristics of the models be related to particular dynamic interactions and hence provide physical insight into the complex manipulator dynamics? Finally, how do the parameters of these models vary in the work space?

The linear models are generated in simulation experiments. An *experiment* is defined as the simulation of the manipulator and identification of a transfer function model for the *target* joint. The transfer function models the input–output (joint input voltage–joint position) characteristics of the target joint. Each experiment concentrates on one target joint and the experimental conditions determine which joints are constrained and the initial configuration for the simulation. The target joint constraint is unconstrained and the voltage input to the target joint is white Gaussian noise. The variance of the noise is chosen by the engineer to move the target joint over a specified angular region. The white noise excitation insures that *all* of the modes of the target joint dynamics are excited and thereby reflected in the identified transfer function model. The models are verified by comparing the responses of the simulator and model to deterministic inputs (such as steps, ramps, and sinusoids). Three hundred simulation experiments have been performed, each experiment investigates a different operating condition. The operating condition define a region in the work space that is associated with an identified linear model. Regions (or *regimes*) in the work space correspond to the dynamic interactions of the manipulator that are excited. The ability to specify the joint constraints (and the regime) to the simulator enables the engineer to analyze (through the linear models) the relative contributions of the dynamic (centrifugal, Coriolis, gravitational, and inertial) interactions.

The linear models are then analyzed to focus upon the following salient features:

- model structure (order and pole and zero locations);
- stability and transient response characteristics;
- minimum and nonminimum phase characteristics;
- continuity and sensitivity of the model parameters to the dominant dynamic interactions in the work space;
- transformation of the linear discrete-time models to continuous-time (analog) counterparts.

Analysis and interpretation of the experimental results provide physical insight into manipulator dynamics for the design of robot control systems.

This work is organized as follows. The manipulator is modeled in Section III. This task consists of conceptual design, mechanical description and dynamic specification, actuator selection, and the complete mathematical modeling through application of the Lagrangian formulation. Section IV describes the simulator, and highlights the features which can be applied to all robotic devices. The Steiglitz–McBride identification algorithm is reviewed in Section V and the linear transfer function models are analyzed and interpreted in detail in Section VI. Finally, in Section VII,

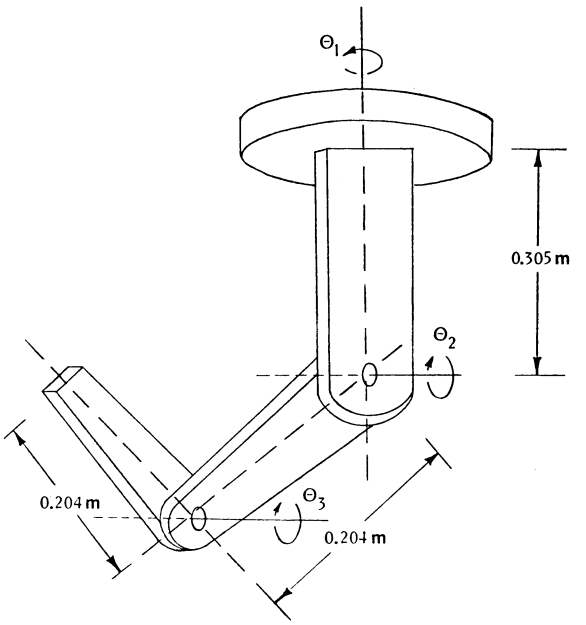

Fig. 1. Physical manipulator model.

implications of the linear models for manipulator control are noted.

III. Modeling the Manipulator Dynamics

A. Introduction

The quality and scope of the physical and mathematical models of a manipulator are fundamental to the analytical development and physical interpretation of manipulator dynamics. Exclusion of dominant manipulator features can result in an oversimplified model. Control system analysis and design for such a model would be of little practical utility. To include all of the interactions in the model would obscure the dominant features. This section outlines the design of a three *degree-of-freedom* (DOF) manipulator and formulates a realistic mathematical model for dynamic analysis and control engineering analysis and design. The features of the transfer function models (identified in Section V) can then be applied to interpret the dynamics of N degree-of-freedom manipulators. Quantitative physical understanding of such a manipulator requires development of a complete mathematical model and simulator.

B. Physical Manipulator

The three link manipulator is displayed in Fig. 1 [29]. The physical manipulator and mathematical model are conceived to exhibit the dynamic behavior which characterizes manipulator structures. Centrifugal, Coriolis, gravitational, and inertial torques are incorporated in the model. The manipulator resembles the configuration of the first three links (or positioning system) of the Unimate/Westinghouse Puma Series 250, 500, and 600 robots [31]. The three-link manipulator, which hangs from a support plate rather than standing on a base, appears as an inverted Puma 250 robot. The three joints are revolute and the axes-of-rotation are shown in Fig. 1. The configuration of the linkage interconnections typifies commercially available arms. The second and third links are offset from the first and second links, respectively. Since the physical links are modeled by cylinders or rectangular bars, the offsets introduce the additional centrifugal and Coriolis coupling torques that are present in arms with irregularly shaped links. Upon including the offsets, the regular link models do not limit the applicability of the physical or mathematical manipulator models or the identified linear models.

TABLE I
Manipulator Characteristics

Link	Weight	Length	Width	Thickness	Radius	Notes
1	51.7 N	0.3050 m	--	--	0.050 m 0.055 m	Inner radius Outer radius
2	8.8 N	0.2040 m	0.040 m	0.010 m	--	0.030 m link 2 offset from link 1
3	8.8 N	0.2040 m	0.040 m	0.010 m	--	0.030 m link 3 offset from link 2

Maximum Payload ..... 31.4 N

TABLE II
Manipulator Model Dynamic Specifications

Joint	Maximum Speed	Maximum Instantaneous Acceleration
1	360 deg/s	1720 deg/s^2
2	360 deg/s	1720 deg/s^2
3	360 deg/s	1720 deg/s^2

The manipulator links are characterized as follows. Link 1 (the base link) is modeled as a hollow metal cylinder that is mounted on a stationary support plate. The drive actuator for link 1 is mounted on the reverse side of the support plate. The actuator shaft passes through a hole in the support plate and is attached to the cylinder. Half of the drive actuator for joint 2 is an integral part of link 1. The joint 2 actuator stator, stator housing, bearings and encoder are fixed to the cylinder. Link 2 is modeled as a solid rectangular bar. The joint 2 actuator rotor, rotor housing, gears and bearings are mounted at one end, and the joint 3 actuator stator, stator housing, bearings and encoder are mounted at the other end. Link 3 is also modeled as a solid rectangular bar with the joint 3 rotor, rotor housing, gears and bearings mounted at one and the payload at the other end. The objective of this correspondence is to address the features of the manipulator that affect the design and performance of a control system. This viewpoint circumvents the need to design the manipulator for construction and specify the mechanical connections of the components. The physical characteristics of the links are listed in Table I.

To insure that the manipulator models a realistic robot, the dynamic specifications (maximum joint speeds and accelerations, and maximum payload ratings) are chosen to approximate those of commercial manipulators (Table II). The physical characteristics (in Table I) and specifications (in Table II) are appropriate for a small parts assembly or interprocess parts transfer robot, and are utilized to specify the actuators. Inland DC torque motors [9] are selected for the joint actuators (i.e., Inland T2809 B and Inland T-5403 G). For each joint, a gear ratio of thirty two is assumed to insure that the motors exemplify actuators of comparable commerical manipulators.

The following assumptions are made to model the manipulator:

- ideal gears between the motor shaft and the joint shaft (i.e., no backlash);
- negligible friction at the joint shafts;
- negligible noise at the joint position transducers and actuator inputs;
- negligible quantization errors in the controller and transducers; and
- absence of actuator saturation limits.

C. Dynamic Manipulator Model

Since the links constitute an interconnected system of rigid bodies, the dynamic model of the manipulator is [11], [19], [23]

$$\boldsymbol{D}(\boldsymbol{\theta})\ddot{\boldsymbol{\omega}} + \boldsymbol{C}(\boldsymbol{\theta},\boldsymbol{\omega}) + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F}(\boldsymbol{\omega}) = \boldsymbol{\Gamma}(t). \quad (1)$$

In (1), $\theta = (\theta_1\theta_2\theta_3)^T$ is the vector of joint angles, $\omega = \dot{\theta}$ is the vector of link velocities and $\dot{\omega}$ is the vector of link accelerations. The dimensions of θ and ω, and the vectors $C(\theta, \omega)$, $F(\omega)$, $G(\theta)$ and $\Gamma(t)$ in (1) equal the number ($N = 3$) of degrees-of-freedom. The first term $D(\theta)\dot{\omega}$, in which $D(\theta)$ is the (3×3) generalized inertial matrix [14], models the inertial coupling. The vectors $C(\theta, \omega)$, $F(\omega)$ and $G(\theta)$, respectively, contain the centrifugal and Coriolis torques, gravitational torques, and frictional torques (that are assumed to be negligible for this arm) experienced at the joints. The input is the vector $\Gamma(t)$ of torques whose components $\Gamma_i(t)$ are generated at the joints by the manipulator motion. Methods for developing the components of $D(\theta)$, $C(\theta, \omega)$ and $G(\theta)$ are detailed in [2], [5], [14], [15], [23], [25], and [32]. For a three degree-of-freedom manipulator, (1) specifies a sixth-order (coupled and nonlinear) dynamic model: the state variables are the three joint angles θ and three joint velocities ω.

The DC torque motors are characterized by second-order linear models with negligible mechanical dissipation [10]:

$$\frac{d}{dt}\begin{bmatrix} i_i \\ \omega_{mi} \end{bmatrix} = \begin{bmatrix} -R_i/L_i & -K_{bi}/L_i \\ \dfrac{K_{ti}}{[J_{mi} + J_{ei}/N^2]} & 0 \end{bmatrix} \cdot \begin{bmatrix} i_i \\ \omega_{mi} \end{bmatrix} + \begin{bmatrix} 1/L_i \\ 0 \end{bmatrix} V_i(t) \qquad (2)$$

and the motor torque

$$\Gamma_{mi}(t) = K_{ti} I_i(t) \qquad \text{for } i = 1,2,3.$$

For each of the i motors in (2), J_{mi} is the rotor inertia, J_{ei} is the external load inertia, K_{bi} is the back emf constant, K_{ti} is the torque constant, $N_i = 32$ is the gear ratio, R_i is the armature resistance, L_i is the armature inductance, V_i is the armature voltage, i_i is the armature current, and ω_{mi} is the *shaft* angular velocity. The gearing mechanically *couples* the actuator models in (2) and the rigid body model in (1). The reflected torque is $\Gamma_{mi}(t)N_i$ and the reflected shaft angular velocity is $\omega_{mi} = \omega_i N_i$. According to Newton's second law, the reflected torque *equals* the ith component $\Gamma_i(t)$ of $\Gamma(t)$ in (1) *plus* the torque required to accelerate the ith motor rotor assembly; that is

$$K_{ti} N_i I_i(t) = \Gamma_i(t) + J_{mi} N_i \dot{\omega}_i. \qquad (3)$$

To develop the simulator [13], (1) and (3) are combined into the state-space model

$$\begin{aligned} \dot{\theta} &= \omega(t) \\ \dot{\omega} &= -[D(\theta) + J_m N]^{-1}\{C(\theta, \omega) + G(\theta)\} \\ &\quad -[D(\theta) + J_m N]^{-1} K_T N i(t) \\ i &= -K_B N L^{-1}\omega(t) - R L^{-1} i(t) + L^{-1} V(t) \end{aligned} \qquad (4)$$

The *diagonal* matrices appearing in (4) are as follows:

Motor rotor inertia matrix	$J_m = Diagonal[J_{m1}\, J_{m2}\, J_{m3}]$
Motor back emf matrix	$K_b = Diagonal[K_{b1}\, K_{b2}\, K_{b3}]$
Motor torque constant matrix	$K_t = Diagonal[K_{t1}\, K_{t2}\, K_{t3}]$
Motor armature inductance matrix	$L = Diagonal[L_1\, L_2\, L_3]$
Motor gear ratio matrix	$N = Diagonal[N_1\, N_2\, N_3]$
Motor armature resistance matrix	$R = Diagonal[R_1\, R_2\, R_3]$.

The state-variables are the three joint angles, the three link velocities, and the three armature currents, and the inputs are the actuator voltages $V(t) = [V_1(t)V_2(t)V_3(t)]^T$. Equation (4) is the complete (ninth-order) mathematical model of the physical manipulator and joint actuator system. The coupled system of nine first-order differential equations (that is, the state-space model) in (4) is directly ammenable to numerical integration and is the foundation of the arm simulator.

IV. Arm Simulator

The design and computer implementation of the manipulator simulator are highlighted in this section. The simulator integrates numerically the dynamic manipulator model (4) and contains features that enable the engineer to model independent joint constraints (Section II). These features enhance the ability of the control engineer to perform detailed analyses of manipulator dynamics and controller designs. The authors anticipate that the versatile properties of the simulator can be applied to a multitude of robotic applications

The simulator is designed, from the control engineering point-of-view, to model the arm dynamics. The inputs to the simulator are the 1) joint constraints, 2) initial joint positions, velocities and accelerations, and 3) motor voltages prescribed as a function of time. The simulator integrates numerically the dynamic manipulator model (4) to produce (*synchronously sampled*) joint positions and velocities. The input voltage and joint position output profiles are processed by the Steiglitz–McBride regression program (Section V) to identify linear transfer function models. Simulator accuracy, rather than speed, is the *design* goal which motivates application of the nonrecursive Lagrangian formulation of the arm dynamics (1) and adoption of the classical *fourth-order* Runge–Kutta algorithm [24] to integrate (4). For the scalar differential equation $\dot{y} = f(y)$, the one-step (classical fourth-order Runge–Kutta) integration across the ith interval proceeds according to

$$y(t_{i+1}) = y(t_i) + \frac{h_i}{6}[K_1 + 2K_2 + 2K_3 + K_4] \qquad (5)$$

where h_i is the step-size for the ith interval and the Kutta coefficients are

$$K_1 = f[y(t_i)] \qquad K_3 = f[y(t_i) + K/2]$$
$$K_2 = f[y(t_i) + K_1/2] \qquad K_4 = f[y(t_i) + K_3/2].$$

Palm [22] points out that "... the fourth-order Runge–Kutta method is probably the most widely used method for engineering applications." Runge–Kutta algorithms 1) are self-starting and the step-size can be changed easily between iterations: 2) do not require derivative evaluation, and 3) display good stability characteristics. Faster and more accurate methods are available, but are more difficult to program [22]. Implementation of the joint constraints (that are developed in the sequel) for a recursive dynamic formulation would be computationally cumbersome.

Since the coupling and nonlinearities in (4) can result in sharp turning points in the state-variables, an internally controlled and adaptive integration step-size augments the Runge–Kutta algorithm. The Collatz criterion [4] on the ratio

$$|K_3 - K_2|/|K_2 - K_1| \qquad (6)$$

is evaluated (after each integration step) to control the step-size and bound the integration error. To the user, the simulator appears as a *synchronous* sampled-data system with the constant sampling period specified by the user. The specified motor inputs are treated as piecewise constant voltages over the sampling periods. An internal function generator creates the input motor voltage waveforms whose parameters are also simulator inputs. One of the following four *independent* constraints can be imposed on the trajectory of *each* of the three joints:

- joint i maintains a constant angular position;
- joint i maintains a constant angular velocity;
- joint i maintains a constant angular acceleration;
- joint i is free to move under the influence of the manipulator dynamics (4) and the *user-specified* joint i input voltage profile.

In a simulation experiment, *for example*, joint 1 can be fixed at a constant position, joint 2 can be forced to move at a constant angular velocity, and the motion of the third joint is then governed by the dynamic interactions and the joint 3 actuator

input voltage. Specific configurations or dynamic interactions can thereby be excited independently and analyzed. The simulator can be applied, for instance, to focus on the Coriolis torques generated at joint 1 due to the relative velocities of joints 1 and 2. By changing the combinations (and relative magnitudes) of the joint constraints, the dynamics can be partitioned into regimes in the work space. For a three degree-of-freedom manipulator, a work space regime is a hypervolume in the nine-dimensional space defined by the state-variable coordinate axes: three joint angular position axes, three joint angular velocity axes, and three joint angular acceleration axes. Velocity and acceleration limitations of the physical manipulator define the extent of reachable regimes in the work space.

With three joints and four constraints, there are 4^3 or 64 distinct constraint combinations that can be specified to the simulator. The initial values of the constrained variables are also specified at run time. Implementation of these constraints is invaluable for generating data for the linear model identification routine and increases significantly the complexity of the simulator. The ability to constrain the state-variable trajectories alters the computation of the derivatives of the state-variables in (4). The Runge–Kutta algorithm integrates the state-variable differential equations $\dot{x} = f(x, V)$. In each iteration of the alogrithm, the nonlinear functions $f(x, V)$ are evaluated explicitly. In the absence of constraints, (4) can be integrated directly. When the position, velocity or acceleration of a joint is constrainted, the formulation of the state–space model (4) must be altered. Joint constraints restrict the values of the elements of the derivative of the state-vector $\dot{x}$. In addition, the input voltages associated with the constrained joints are transformed from specified functions of time to components of the solution vector. When joints are *constrained*, (4) is decomposed into a *reduced-order* state-space model and a system of *algebraic* equations. In the case of second-order actuator models, the order of the reduced state–space model is $(3N - M)$, where N is the number of degrees-of-freedom and $2M$ algebraic equations, where M is the number of constrained joints. The modified input vector in (4) then contains the unconstrained joint input voltages and constrained joint variables. The $2M$ algebraic equations determine the input voltages to the actuators of the constrained joints.

In the aforementioned example, the state-space model (4) is reformulated as follows. Since there are $M = 2$ joint constraints (θ_1 is a constant angular position and ω_2 is a constant angular velocity), the order of the reduced state-space model is $[3N - M]_{N=3, M=2} = 7$, and $2M = 4$ algebraic equations specify the actuator currents and voltages for the constrained joints 1 and 2. To formulate the reduced-order state-space model and algebraic equations, define the *selection* vector

$$z^T = [0\,0\,1] \qquad \text{for the unconstrained actuator currents}$$

and the (3×3) *selection* matrices

$$\boldsymbol{P} = \text{Diagonal}\,[1\,1\,0] \qquad \text{for the constrained joint torques}$$

and

$$\boldsymbol{Q} = \text{Diagonal}\,[0\,0\,1] \text{for the unconstrained joint torques.}$$

The state-variables of the reduced-order model are θ, ω, and $z^T i = i_3$, since i_1 and i_2 are constrained. Upon applying the selection vector z and selection matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ to the complete dynamic model of the manipulator and actuator in (4), the *reduced*-order state-space model becomes

$$\begin{aligned}
\dot{\theta} &= \omega(t) \\
\dot{\omega} &= -\boldsymbol{Q}[\boldsymbol{D}(\theta) + \boldsymbol{J}_m\boldsymbol{N}]^{-1}\{\boldsymbol{C}(\theta, \omega) + \boldsymbol{G}(\theta)\} \\
&\quad -\boldsymbol{Q}[\boldsymbol{D}(\theta) + \boldsymbol{J}_m\boldsymbol{N}]^{-1}\boldsymbol{K}_T\boldsymbol{N}\boldsymbol{Q}i(t) \\
z^T i &= -z^T\boldsymbol{K}_B\boldsymbol{N}\boldsymbol{L}^{-1}\omega(t) - \boldsymbol{Z}^T\boldsymbol{R}\boldsymbol{L}^{-1}i(t) + z^T\boldsymbol{L}^{-1}V(t). \quad (7)
\end{aligned}$$

The $2M = 4$ algebraic equations are the $M = 2$ *torque balance equations*

$$\boldsymbol{P}i = \boldsymbol{P}(\boldsymbol{K}_T\boldsymbol{N})^{-1}\{(\boldsymbol{D}(\theta) + \boldsymbol{J}_m\boldsymbol{N})\dot{\omega} + \boldsymbol{C}(\theta, \omega) + \boldsymbol{G}(\theta)\} \quad (8)$$

in (3) to specify the constrained actuator currents $i_1(t)$ and $i_2(t)$; and the $M = 2$ *Kirchhoff voltage equations*

$$\boldsymbol{P}V = \boldsymbol{P}\{\boldsymbol{L}i + \boldsymbol{R}i + \boldsymbol{K}_B\boldsymbol{N}\omega\} \quad (9)$$

in (2) to specify the constrained actuator voltages $V_1(t)$ and $V_2(t)$. Equation (9) requires the derivatives of the motor currents, i, for the constrained joints. After evaluating (8), the present and past iterates of the constrained joint currents are fit to polynomials. The derivatives of these polynomials (at the current time) are computed to specify the unknown current derivatives for (9).

The simulator automatically reformulates (4) for each of the 64 constraint combinations. The automatic reformulaton of the algorithm (for computing the derivatives of the state-variables and the Runge–Kutta integration algorithm) enables the simulator to calculate the *feedforward* control (motor voltage) inputs to maintain the desired joint constraints. When joint i is constrained to a fixed position, for example, the simulator models the action of mechanical brakes.

To accommodate each of the 64 constraint combinations, the algorithm utilizes the selection vector

$$z^T = [z_1 z_2 z_3]$$

where

$$z_k = \begin{cases} 1, & \text{if current } i_k \text{ is unconstrained} \\ 0, & \text{if current } i_k \text{ is constrained} \end{cases}$$

and the selection matrix

$$\boldsymbol{P} = \text{Diagonal}\,[p_1\, p_2\, p_3]$$

where

$$p_l = \begin{cases} 1, & \text{if joint } l \text{ is unconstrained} \\ 0, & \text{if joint } l \text{ is constrained.} \end{cases}$$

The selection matrix $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{P}$, where $\boldsymbol{I}$ is the (3×3) identity matrix.

The salient feature of the simulator is the ability for the user to specify constraints on the trajectories of the joint variables. The authors anticipate that this feature will be essential for the dynamic analysis of a multitude of robotic manipulators.

V. Identification Algorithm

The iterative Steiglitz–McBride regression algorithm identifies the $2n$ parameters α and β in the nth order digital transfer function:

$$W(z^{-1}) = \frac{\beta_1 z^{-1} + \beta_2 z^{-2} + \cdots + \beta_n z^{-n}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} - \cdots - \alpha_n z^{-n}}. \quad (10)$$

Convergence of the Steiglitz–McBride algorithm is *not* guaranteed; nuances of its convergence properties are detailed in [28]. To assure reasonable identification, the responses of the linear models and nonlinear manipulator simulator to deterministic inputs (such as steps, ramps and sinusoids) are compared. The angular responses of the linear models and the simulator (to the same deterministic inputs) agree to within a few tenths of a degree.

VI. Analysis and Interpretation of the Linear Models

A. Introduction

This section applies the tools of classical control engineering to interpret the linear models. Section VI-B relates the identified parameters of the model to the poles and zeros (of the transfer

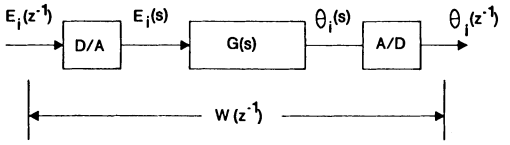

Fig. 2. Analog process discretization.

function) which illuminate the dynamic behavior of the manipulator. Actuator excitation for model identification is highlighted in Section VI-C. The analog characteristics, which are inherent in the discrete linear models, are reviewed in Section VI-D. Section VI-E describes the practical limitations which preclude the identification of linear models for particular state–space regions. The salient features of the identified linear models are documented in Section VI-F. The essence of this correspondence lies in the control engineering analysis of the linear models to interpret the dynamic behavior of manipulators. Section VI-F is, therefore, divided into five subsections to detail the salient features of the identified linear models. The stability and phase characteristics of the models are noted in Section VI-F-1. Sections VI-F-2, 3, and 4 relate the parameters of the linear models to the contributions of the gravitational centrifugal, and Coriolis, and inertial coupling interactions, respectively. Finally Section VI-F-5 notes that the parameters of the linear models vary *smoothly* over the entire state space.

B. Model Structure and Parameters

The Steiglitz–McBride algorithm identifies the linear discrete-time (input–output) transfer function models in (10). The input $E_i(z^{-1})$ is the z-transform of the joint i actuator voltage and the output $\theta_i(z^{-1})$ is the z-transform of the angular position of joint i which is being modeled. The $2n$ parameters α_i and β_i in (10) are explicit functions of the sampling period T_s that is used by the simulator to generate the input–output sequences. As depicted in Fig. 2, the digital transfer function $W(z^{-1})$ in (10) models the cascade of a digital-to-analog (D/A) converter, the analog process model $G(s)$ and an analog-to-digital (A/D) converter. In digital signal processing, the cascade in Fig. 2 is the step-invariant transformation of $G(s)$ [16]. The location of the poles and zeros of the model can provide more insight for control system analysis and design than can the numerical values of the transfer function parameters (α_i and β_i). The simulation experiments have identified accurate *second*-order transfer function models. These models are presented in the equivalent form

$$W(z^{-1}) = K\frac{z^{-1}(a - z_0 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} \tag{11}$$

in which the poles p_1 and p_2 are the roots of the characteristic equation $z^2 - \alpha_1 z - \alpha_2 = 0$; and

$$K = \beta_1 \qquad \alpha_1 = p_1 + p_2$$
$$z_0 = -\beta_2/\beta_1 \qquad \alpha_2 = -p_1 p_2.$$

The gain K, zero z_0, and poles p_1 and p_2 (which are also explicit functions of the sampling period T_s) are interpreted in Section VI-F.

C. Actuator Excitation for Model Identification

When the linear model (10) is sought for joint i, the ith joint actuator (in the simulator) is excited by a white Gaussian noise signal. The resulting input and output sequences (processed by the identifier to estimate the linear model parameters) are the white Gaussian noise input sequence and the ith joint's angular response sequence. Practical reality limits the white noise excitation of the joint actuators to move the joints. The white noise excitation makes it difficult to move the ith joint through a large angular region (during a simulation experiment) unless the standard deviation of the signal exceeds 30 volts. Such a standard deviation causes an appreciable number of the input voltage samples to exceed the tolerable or saturation limits of the motors. These input signals are acceptable (in simulation experiments) to identify linear models. The white noise signal excites *all* of the dynamic (centrifugal, Coriolis, gravitational, and inertial) torque interactions for linear modeling, and the manipulator contributes the dominant dynamics (in comparison with the time-constants of the electromechanical actuators).

D. Analog Characteristics of the Linear Models

The location of the poles and zeros provides engineering insight to evaluate viable control system structures and designs. The parameters of a digital transfer function model of a physical system (such as a robotic manipulator) depend explicitly on the sampling period. Control engineers traditionally characterize analog systems by s-plane pole and zero locations. These analog parameters relate directly to such features as rise-time and overshoot. (The analog poles of *all* of the identified linear models are *real* and the corresponding *time-constants* are the negative reciprocals of the stable s-plane poles.) To analyze the identified linear models in the analog domain requires the transformation of the digital transfer function parameters into *equivalent* analog transfer function parameters. The problem is that the digital transfer function models a sampled *nonlinear* system and the step-invariant transformation (from the analog to the digital transfer function) is developed for linear systems.

In the simulation experiments, the identified discrete poles z_i can be mapped consistently into the analog poles s_i according to the logarithmic transformation $s_i = [\ln z_i]/T_s$.

It would be advantageous to map, if possible, the complete digital transfer function model into an equivalent analog transfer function. Mapping the identified discrete zeros into equivalent analog zeros is significantly more complicated than pole mapping. The applicable transformations (between the s-plane and the z-plane) for mapping a *linear analog* transfer function into a *discrete-time* transfer function utilizes *a priori* information. For instance, transforming a discrete-time transfer function into the analog domain assumes that the physical process can be modeled by a linear analog transfer function and that the sampling period satisfies the Nyquist criterion [21]. For robotic applications, the physical analog process is *nonlinear*. If there is a complete mapping of the second-order digital transfer function, then the analog transfer function must have two poles and, at most, two finite zeros. In [16], a table of digital transfer functions $W(z^{-1})$ is compiled for low-order analog processes. These digital transfer functions are obtained through the step-invariant transformation of the analog process model $G(s)$; that is,

$$W(z^{-1}) = [1 - z^{-1}]^{-1} Z\left\{\frac{G(s)}{s}\right\}. \tag{12}$$

The inability to find the numerators of the analog transfer function implies that the identified linear digital transfer functions model the nonlinear dynamics (at the sampling instants) and that these nonlinearities cannot be analytically transformed into the s-plane. Unfortunately this finding makes it more difficult to extract additional physical insight into the dynamics beyond that obtained (in Section IV-F) from the analog pole mapping. Soderstrom [26] has encountered an analogous phenomenon in attempting to formulate continuous-time counterparts of ARMA models.

E. Limitations in Model Identification

Quantitative comparisons between the parameters of different digital transfer functions require the *same* sampling period. There are limitations that preclude the identification of a family of linear models with the same sampling period. As noted in Section VI-D, the models cannot be *completely* transformed to the s-plane and then back to the z-plane. A second limitation is introduced by the relationship $L = N_s T_s$ between the sampling period $T_s(s)$,

the simulation length $L(s)$ and the number N_s of sample points. This limitation is illustrated by the following example. Three simulation experiments are to be conducted to identify a linear model for joint 2 under the following conditions. Joint 1 is constrained to the constant position $\theta_1(t) = 0°$. Joint 3 is constrained to the constant angular velocity $\omega_3(t)$ and its initial position is $\theta_3(0) = 0°$. The white noise voltage source (driving the joint 2 motor) causes joint 2 to rotate about its initial position, $\theta_2(0) = 90°$.

There are three dynamic situations to model: 1) joint 3 velocity is low ($\omega_3 = 20$ deg/s); 2) joint 3 velocity is medium ($\omega_3 = 90$ deg/s); and 3) joint 3 velocity is high ($\omega_3 = 180$ deg/s). In each situation, the relative contribution of the centrifugal and Coriolis torque components to the manipulator dynamics is different. To compare the three linear models, joint 3 must traverse the same angle to insure that the variations in the effective inertia about joint 2 (as link 3 moves closer to joint 2) remains proportionate in the three situations. For instance, if joint 3 moves 180° during the simulation, the dramatic variations in the effective inertia about joint 2 and the dynamic characteristics prevent the identification of an accurate linear model. To compensate for this limitation the simulation length L is varied, depending upon the magnitude of the velocity of the constrained joints. In the simulation experiments, a minimum number of sample points is maintained to ensure the integrity of the white noise input, and the sampling period is varied to preserve the length of the simulation.

A third limitation is introduced by the numerical sensitivities of the identified parameters. The logarithmic transformation (in Section VI-D) is highly sensitive to variations in the digital poles and this sensitivity increases dramatically as the poles approach the origin in the z-plane. Numerical inaccuracies in calculating the digital poles, and then the analog poles, may lead to misinterpretations of the manipulator dynamics. Consequently, simulation experiments that produce linear models with discrete poles lying close to the origin are repeated with a smaller sampling period. These practical limitations lead to discrete-time manipulator models that are based upon different sampling periods.

F. Salient Features of the Identified Models

The experimentation has produced three hundred *second*-order discrete-time transfer function models [29]. The salient features of the identified models and their relationships to the dynamic torque interactions are described in this section.

1) *Nonminimum Phase and Instability:* Most of the identified transfer function models are *nonminimum* phase [20] because the discrete zero lies outside the unit circle (and typically between $z = -1$ and $z = -6$). Experimentation shows that the transfer function zeros move *toward* the unit circle as the sampling period is *increased*. Astrom *et al.* [1] have demonstrated this property of step-invariant filters. The nonminimum phase property of the linear models indicates that conventional control designs such as pole-zero cancellation or inverse-plant compensation may be inappropriate for manipulator applications.

The identified models for joints 2 and 3 can have an unstable pole lying outside the unit circle in the z-plane. The masses (links, motors, and payload) between the joint being modeled and the end of the manipulator have a center of mass that depends on their relative positions. The modeling data indicate that the position of the center of mass and its interaction with gravity (in a simulation experiment) determine the stability of the identified poles. The two poles are stable if the response of the simulator to the white noise excitation of a joint causes the center of mass to remain below the horizontal plane coincident with the joint axis. One of the identified poles is unstable if the center of mass stays above that horizontal plane. In some experiments, the center of mass moves above and below the plane. In these cases the relative times above and below the plane determine the stability and instability of the pole. This phenomenon is analogous to the stability and instability regions of the simple pendulum. Furthermore the stability region does *not* appear to depend strongly on the motion and velocity of the other two joints. What is important for robot control engineering is the *finite time* stability and dynamic characteristics of manipulators.[1]

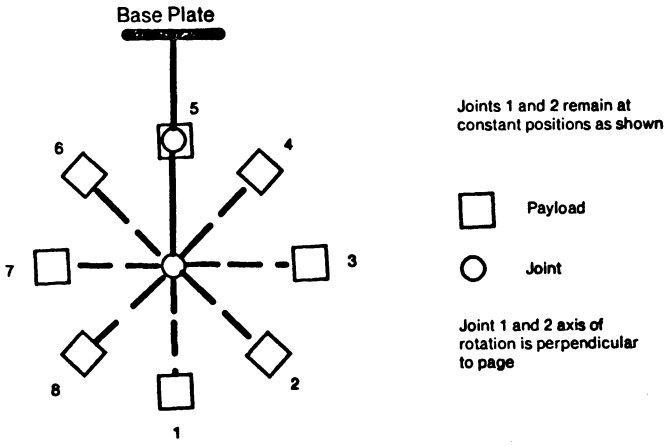

Fig. 3. Experimental configurations for modeling joint 3.

The transfer functions modeling joint 1 (when joint 2 or 3 is moving) also contain an unstable pole. A physical explanation for this observation is lacking because the torque produced at joint 1 is independent of gravity.

The following subsections detail the location and movement of the poles. The extent of the work space leads to the identification of three hundred linear models and examples are selected to illustrate the properties of families of models. The *numerical* values of the transfer function parameters (identified for the manipulator designed in [29]) are included to exemplify dynamic robot characteristics.

2) *Gravitational Torques:* Gravitational torques perturb (the numerical values of) the parameters of the linear models. The models highlighted in this section illustrate configurations in which the two joints not being modeled are fixed in a specified position. Consequently the effective inertia of the part of the manipulator, which rotates about the joint being modeled, remains constant. Experiments are performed for different angular positions about which the joint is modeled. For example, Fig. 3 depicts a set of experiments performed to model joint 3. Dynamic models are identified for motion about the eight positions shown in Fig. 3 and the identified parameters are listed in Table III. All of the models are type *zero* with two *real* poles. The poles move smoothly through the work space. The digital pole p_1 moves between $z = 0.7$ and $z = 0.9$ and the digital pole p_2 hovers around $z = 1$. The analog pole s_1 (corresponding to the digital pole p_1) is essentially constant while the analog pole s_2 (corresponding to the digital pole p_2) varies significantly because of the logarithmic transformation $s_2 = [ln\ p_2]/T_s$ and moves across the imaginary axis (stability boundary) of the s-plane. Fig. 3 and the models in Table III illustrate the identification of stable and unstable models. As the center of mass of link 3 and the payload reach the horizontal plane through the joint 3 axis, the models become unstable. The instability grows until $\theta_3 = 180°$.

The dimensionless damping ratios

$$\zeta = -(s_1 + s_2)/2[s_1 s_2]^{1/2} \tag{13}$$

of the *stable* models in experiments 1 and 2 (and 8) are 1.86 and 2.38, respectively. Since $\zeta > 1.5$, these *overdamped* second-order models behave essentially as *first*-order models with the *equivalent* time-constant [17] $\tau_e = -1/s_2$ ranging from 480 ms (in experiment 1) to 763 ms (in experiments 2 and 8).

[1]Private discussion with A. J. Koivo at the Third Yale Workshop on Applications of Adaptive Systems Theory, June 17, 1983.

TABLE III
IDENTIFIED MODELS FOR JOINT 3

Test	$\theta_3(0)$ (deg)	T_s (s)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
1	0	0.010	$.1693\times10^{-3}$	-1.3005	0.7845	0.9794	-24.30	-2.08
2	45	0.010	$.1682\times10^{-3}$	-1.2977	0.7628	0.9870	-27.08	-1.31
3	90	0.005	$.3322\times10^{-4}$	-2.1340	0.8739	1.0028	-26.96	0.56
4	135	0.010	$.1674\times10^{-3}$	-1.3101	0.7605	1.0149	-27.38	1.48
5	180	0.010	$.1622\times10^{-3}$	-1.2942	0.7396	1.0185	-30.16	1.83
6	225	0.010	$.1674\times10^{-3}$	-1.3081	0.7426	1.0124	-29.76	1.23
7	270	0.005	$.3322\times10^{-4}$	-2.1340	0.8739	1.0028	-26.96	0.56
8	315	0.010	$.1682\times10^{-3}$	-1.2977	0.7628	0.9870	-27.08	-1.31

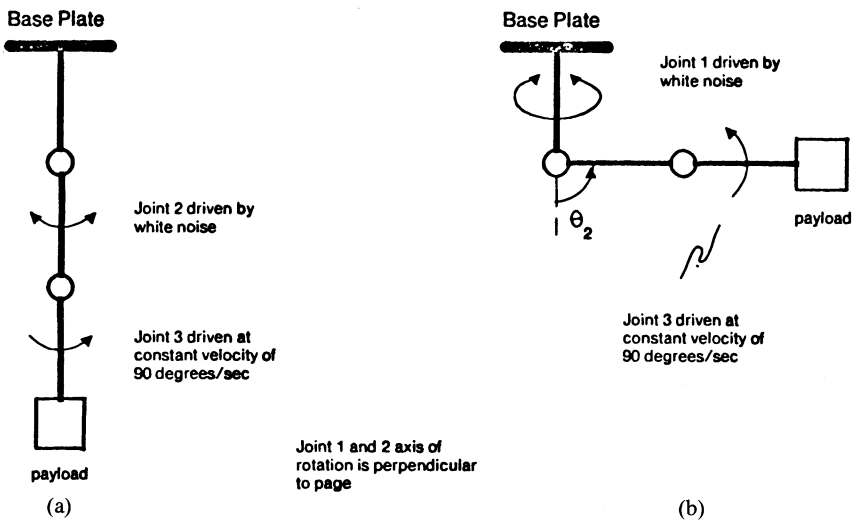

Fig. 4. Experimental configurations for modeling joint 2 and joint 1.

Similar qualitative behavior is observed in the models of joint 2. More experimentation is involved for joint 2 since the models depend on the fixed position of joint 3. The pole variations (in the joint 2 models) follow the relative trends of the joint 3 models with respect to the joint 2 starting position. The magnitudes of the variations, however, are smaller.

Since joint 1 is independent of gravity, the joint 1 models depend only on the effective inertia determined by the *fixed* positions of joints 1 and 2. The gains K and discrete zeros z_0 identified in these experiments remain essentially constant (for a fixed inertia about the modeled joint), varying by less than one percent.

In the joint 2 models, the analog pole s_1 depends inversely on the effective inertia of link 2, link 3, and the payload about the joint 2 axis that in turn depends upon the fixed position θ_3. When $\theta_3 = 0°$ (fully extended), the effective inertia about joint 2 is a maximum and the identified pole $s_2 = -41s^{-1}$ is a minimum. When $\theta_3 = 180°$, the identified pole $s_2 = -233s^{-1}$. The analogous situation arises for joint 1 when θ_2 and θ_3 are fixed. The maximum inertia about the joint 1 axis occurs when $\theta_2 = 90°$ and $\theta_3 = 0°$. The identified analog pole $s_2 = -40s^{-1}$. The minimum inertia occurs when $\theta_2 = \theta_3 = 0°$ for which the identified analog pole $s_2 = -1000s^{-1}$. These simulation results agree with physical intuition.

3) *Centrifugal and Coriolis Torques:* When the centrifugal and Coriolis torques interact, the location and movement of the identified poles and zeros become considerably more complex. Different centrifugal and Coriolis torques are excited, depending upon the motion and velocities of the joints.

The experimental results divide naturally into two groups. *Group 1* models joint i when joint k $(i > k)$ is moving at a fixed angular velocity. *Group 2* models joint i when joint k $(i < k)$ is moving at a fixed velocity. The two groups exhibit different linear second-order model characteristics. The dynamic models suggest that these differences may be attributed to the following phenomenon. In Group 1 $(i > k)$, the inertial reference frame of the joint being modeled is stationary with respect to the world coordinate system. In Group 2 $(i < k)$, the inertial reference frame of the joint being modeled is accelerating with respect to the world coordinate system due to the movement of joint k. The variation in the identified parameters of the Group 1 models is less dramatic than that of the Group 2 models.

Group 1 contains both stable and unstable models, depending upon the configuration (as explained in Section VI-F-2). In Group 1, the observed variation in pole p_2 (which hovers around $z = 1$), with respect to variations in the velocity of the kth joint, is negligible. The second pole p_1, which lies closer to the origin, moves significantly. Figs. 4(a) and (b) depict two experimental configurations belonging to Group 1. Fig. 4(a) models joint 2 as joint 3 moves at various fixed velocities and Fig. 4(b) models joint 1 as joint 3 moves at various fixed velocities. Table IV lists the models identified for joint 2 (corresponding to the experiments in Fig. 4(a)) and Table IV lists the models identified for joint 1 (corresponding to the experiments in Fig. 4(b)). In these experiments, the analog pole s_1 is relatively insensitive to the velocity of joint k. As the angular velocity ω_k increases, the analog pole s_2 moves towards s_1, and the analog pole s_1 varies with the configuration. In the two configurations shown in Fig. 4, the analog pole $s_1 = -40s^{-1}$. When joints 2 and 3 are repositioned (as shown in Fig. 4(b)) and the experiments are repeated, the analog pole s_1 (listed in Table IV) moves to reflect the change in the effective inertia about the joint 2 axis. After each reposi-

TABLE IV
MODELS OF JOINT 2 AS A FUNCTION OF ω_3

ω_3 (deg/s)	T_s (s)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
20	0.0025	$.9537\times10^{-5}$	-1.8725	0.9027	0.9981	-40.9	-0.8
90	0.0025	$.1042\times10^{-4}$	-1.6125	0.9065	0.9956	-39.3	-1.8
180	0.0050	$.4833\times10^{-4}$	-1.1033	0.8200	0.9938	-39.7	-1.2

TABLE V
EXPERIMENTAL CONFIGURATIONS FOR MODELING JOINT 3 WHILE JOINT 2 MOVES WITH FIXED VELOCITY

ω_3 (deg/s)	T_s (s)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
20	0.0050	$.4366\times10^{-4}$	-1.3152	0.8160	1.0	-40.7	0
90	0.0050	$.4396\times10^{-4}$	-1.3073	0.8157	1.0007	-40.7	0.14
180	0.0025	$.9272\times10^{-5}$	-1.8874	0.9034	1.0017	-40.6	0.70

TABLE VI
MODELS OF JOINT 3 FOR REGIONS ABOUT $\theta_3(0)$

Test	$\theta_3(0)$ (deg)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
1	0	$.8859\times10^{-5}$	-1.8387	0.9721	0.9721	-11.30	-11.3
2	45	$.4229\times10^{-5}$	-5.3363	0.9354	1.0037	-26.70	1.48
3	90	$.3930\times10^{-5}$	-5.9930	0.9257	1.0105	-30.90	4.18
4	135	$.4920\times10^{-5}$	-4.6067	0.9236	1.0114	-31.80	4.53
5	180	$.5139\times10^{-5}$	-4.3114	0.9286	1.0074	-29.60	2.95
6	225	$.7188\times10^{-4}$	-2.6465	0.9440	0.9958	-23.10	-1.68
7	270	$.1053\times10^{-4}$	-1.3535	0.9727	0.9727	-11.10	-11.10
8	315	$.1192\times10^{-4}$	-1.0297	0.9739	0.9739	-10.60	-10.60

Sampling period for all discrete models is $T_s = 0.0025$ s.

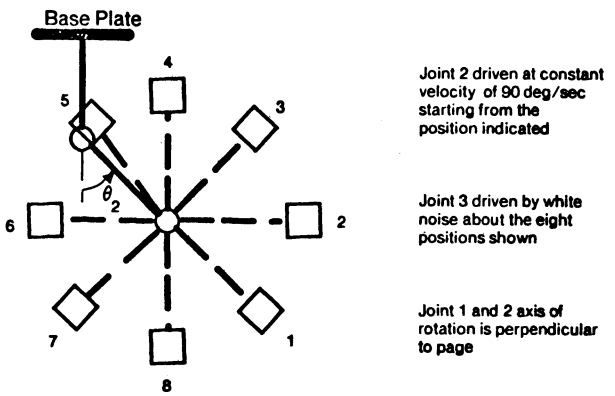

Fig. 5. Experimental configurations for modeling joint 3 while joint 2 moves the fixed velocity.

tioning, the relative invariance of the s_1 pole is preserved. In simulation experiments that can be compared (on the basis of an equivalent sampling period), the discrete gain K and zero z_0 are relatively insensitive to the velocity of joint k.

In Group 2, the variations in the identified model parameters appear to depend upon the relative position of links i and k. Fig. 5 depicts experiments performed to identify models for joint 3. Joint 2 is initially at 45° and rotates with the constant velocity $\omega_2 = 90$ deg/s. Joint 1 is fixed at $\theta_1 = 0°$. Because of the symmetry of the manipulator, the models are invariant to the fixed position of θ_1. The pertinent parameters of the identified models (for the eight experiments depicted in Fig. 5) are listed in Table VI. The analog poles of the unstable, type zero model identified in experiment 4 are $s_1 = -3.18s^{-1}$ (stable) and $s_2 = 4.53s^{-1}$ (unstable). The spread of the analog poles is a maximum and this unstable model occurs when link 3 is vertically upward. As the starting position of joint 3 (around which the models are identified) approaches $\theta_3 = 315°$, the unstable analog pole s_2 moves toward the imaginary axis. In experiments 1, 7, and 8, the analog pole s_2 moves into the left-half plane and becomes stable, and the analog pole s_1 moves toward the origin. In experiments 1, 7, and 8, the two poles coalesce and become a double pole. The pole motion, with respect to $\theta_3(0)$, is smooth. The smooth transition between the distinct and repeated pole systems occurs rapidly, as the opperating configuration is varied. The discrete gains and zeros (listed in Table VI) change smoothly aand appreciably with the configuration. The gain K in (11) increases as the analog pole s_1 moves toward the origin. The nonminimum phase zero moves to the left (along the negative real z-axis) as the analog pole s_1 moves to the left and the analog pole s_2 moves to the right (along the real-axis in the s-plane).

Similar qualitative features emerge for different fixed velocities of joint 2. As the fixed velocity ω_2 of joint 2 approaches 0 deg/s, double pole models are no longer identified. Analog poles s_1 and s_2 approach each other but do not coalese, and the variations in the gain and discrete zero decrease. When the velocity ω_2 approaches 0 deg/s, these joint 2 models converge smoothly to the models described in Section VI-F-2 in which joints 2 and 3 are fixed and joint 3 is modeled when the velocity ω_2 approaches 0 deg/s.

Further experimentation suggests that existence of double pole transfer functions depends upon the relative direction of the joint 2 velocity with respect to the position of link 3 and the payload. For instance, if the velocity of joint 2 in Figure 5 is -90 deg/s, double pole transfer functions are identified when $\theta_3(0)$ is 0° (position 1), 45° (position 2), and 90° (position 3), irrespective of the starting position of joint 2. The parameters of the models identified for joint 1 (when joint 2 or 3 is moving) are far less sensitive to the centrifugal and Coriolis torques than for joints 2 or 3. This is expected because the joint 1 axis-of-rotation is perpendicular to the axes of joints 2 and 3, whereas the axes of joints 2 and 3 are parallel.

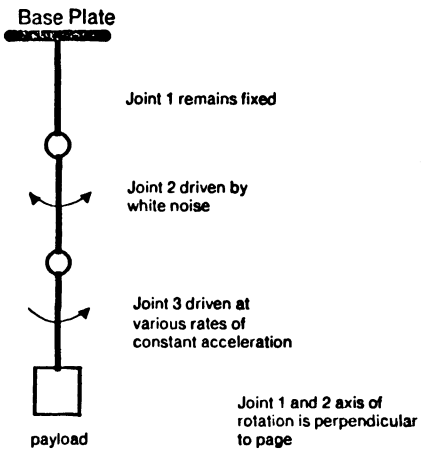

Fig. 6. Joint 2 experimental configuration.

4) Inertial Coupling Torques: This section analyzes the influence of the inertial coupling torques on the identified model parameters. The experiments of Section VI-F-3 model joint i while joint k is moving at a constant angular *velocity*. Consequently the acceleration of joint k is *zero*. The experiments in this section model joint i while joint k is moving at a constant angular *acceleration*, and unavoidably excite the centrifugal and Coriolis torques between joints i and k. Comparison of the experiments in these two sections leads to a *qualitative* separation of the relative contributions. In parallel with the modeling of the centrifugal and Coriolis torques (in Section VI-F-3), the modeling of the inertial coupling torques is divided into two groups. Group 1 models joint i when joint k ($k > i$) is moving at a constant angular acceleration and Group 2 models joint i when joint k ($k < i$) is moving at a constant angular acceleration. The two groups exhibit distinct linear model characteristics and these differences are attributed to the phenomena noted in Section VI-F-3.

All of the identified models are second-order. The parameter variations in the Group 1 models are less dramatic than the parameter variations in the Group 2 models. In Group 1, the observed variation in analog pole s_1 (with respect to variations in the acceleration of joint k) is small and only slightly more significant than when the centrifugal and Coriolis torques are excited. The analog pole s_2 oscillates about the origin. Fig. 6 depicts an experimental Group 1 configuration in which joint 2 is modeled while joint 3 moves at a constant acceleration and Table VII lists the identified models.

The location of analog pole s_1 is insensitive to the acceleration of joint k. As the angular acceleration $\dot{\omega}_k$ increases, the analog pole s_2 moves towards s_1. In Group 1, experiments are performed that differ in the position of joint i, about which models are identified. For instance, experiments are performed in which joint 2 moves about $\theta_2(0) = 135°$ instead of $\theta_2(0) = 0°$ (as shown in Fig. 6). In this case, the analog pole s_2 is unstable, but the same relative variations are observed. The stability of these models is determined by the relative position of the center of mass of link 2, link 3, motor 3 and the payload with respect to the horizontal plane through the joint 2 axis (Section VI-F-2). The location of the analog pole s_1 is determined by the configuration. As the initial position of joint 3 approaches 180°, the effective inertia about the joint 2 axis decreases, and the pole s_1 moves to the left in the s-plane. The discrete zero z_0 approaches the unit circle and the gain K increases slightly as the acceleration of joint k increases. These trends replicate the parameter variations of the Group 1 models in Section VI-F-3. The inertial coupling in Group 1 tends to accentuate mildly the range of the parameter variations exhibited by Group 1 in Section VI-F-3.

The variations in the identified parameters of the Group 2 models in these two sections are also similar. The parameter variations depend upon the relative positions of links i and k. Fig. 7 depicts experiments performed to identify models for joint 3. Joint 2 (which is initially at rest at 0°) accelerates with the constant acceleration $\dot{\omega}_2 = 100$ deg/s^2. Joint 1 is fixed at $\theta_1 = 0°$. These experiments include eight positions for joint 3, about which models are identified. The variations in the identified parameters listed in Table VIII resemble qualitatively the parameter variations in Table VI in which only the centrifugal and Coriolis torques are excited. For example, the analog poles s_1 and s_2 approach each other as the initial position $\theta_3(0)$ of joint 3 reaches $-45°$. If $\dot{\omega}_2(0)$ were -100 deg/s^2, the two analog poles would move together as $\theta_3(0)$ approaches 45°. Excitation of the inertial coupling dampens the pole movement and parameter variations. In Section VI-F-3, the poles coalesce and produce a double pole (for high joint k velocities). With inertial coupling, the poles move together but do not coalesce (even for high joint k accelerations).

Group 2 displays similar qualitative features for different constant accelerations of joint k. As the constant acceleration of joint k approaches zero deg/s^2, the range of the parameter variations decreases. The linear models converge smoothly to the models identified in Section VI-F-2 in which only the joint being modeled is moving. The parameters of the models identified for joint 1 are less sensitive to the inertial coupling. This finding is apparently caused by the fact that the joint 1 axis is perpendicular to the axes of joints 2 and 3.

5) Summary: The linear models exhibit the following fundamental property. *As the manipulator moves from one regime (in the work space) to the next, the analog poles (of the identified linear models) move smoothly.* The examples presented in this paper (and *all* of the experimental data) illustrate this property. Since the parameters of the linear models experience large variations over the entire work space, a small number of fixed coefficient linear models would not be sufficient to characterize the dynamics of the manipulator.

VII. Control Engineering Implications

The experimental results (highlighted in Section VI) indicate that the coupled nonlinear dynamics of a three degree-of-freedom manipulator can be modeled by *second order* (input–output) digital transfer functions (with two real poles and one zero). The collection of models displays two significant properties. 1) The analog poles of the models move gradually as the manipulator moves between adjacent work space regions; and 2) The identified analog poles are commensurate (numerically) with the poles obtained from the classical linearization (in a region of the work space) of the complete dynamic model of the manipulator.

Two implications for the dynamic modeling of manipulators emerge. First, the three degree-of-freedom manipulator embodies the dynamic features which characterize all manipulator devices. The dynamics of N degree-of-freedom manipulators can thus be modeled by transfer functions, as suggested by classical linearization. Second, the local dynamics from the input to joint i to the output of joint k can also be modeled by transfer functions. Identification of these transfer function matrices will lead to linear *multi-input–multi-output* models of the manipulator.

To meet the *practical* need for more versatile and robust manipulators, the control engineer must develop advanced controller designs. Dynamic modeling and engineering insight has thus led to the need to develop *adaptive controllers* for robots. Because the linear transfer function models are *nonminimum* phase, robot control engineering needs multivariable adaptive controllers for nonminimum phase manipulator models. Adaptive control of robotic manipulators is highlighted in [18].

VIII. Conclusion

The fundamental property of the linear models (and its implications for manipulator control) is the contribution of this correspondence. The experimental results suggest the following.

TABLE VII
MODELS OF JOINT 2 AS A FUNCTION OF $\dot{\omega}_3$

$\dot{\omega}_3$ (deg/s^2)	T_s (s)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
-100	0.0050	$.4330\times10^{-4}$	-1.4605	0.8061	0.9982	-43.11	-0.36
0	0.0050	$.4483\times10^{-4}$	-1.3318	0.8121	0.9968	-41.63	-0.64
20	0.0050	$.4518\times10^{-4}$	-1.3056	0.8134	0.9964	-41.31	-0.72
100	0.0050	$.4788\times10^{-4}$	-1.1232	0.8215	0.9943	-39.32	-1.14
500	0.0025	$.4581\times10^{-4}$	-1.1960	0.8222	0.9934	-39.15	-1.32

Sampling period for all discrete models is $T_s = 0.005$ s.

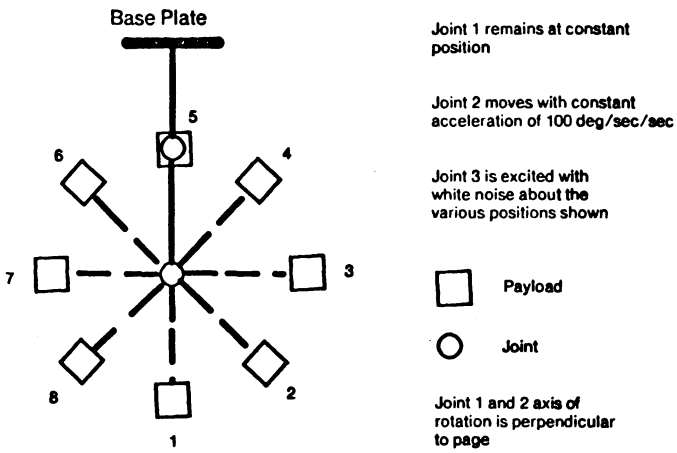

Fig. 7. Experimental configurations for modeling joint 3 while joint 2 moves with fixed acceleration.

- The complex dynamics (including the centrifugal, Coriolis, gravitational, and inertial torque interations) of an N degree-of-freedom robotic manipulator driven by DC torque motors can be modeled by *low-order*, *linear*, discrete-time (input–output) transfer functions.
- The stable and unstable nonminimum phase linear linear transfer functions (which characterize the manipulator dynamics over the entire attainable work space) are *second*-order.

To the authors' knowledge, this work inaugurates the systematic delineation of these characteristics from engineering analyses of the complete dynamic equations-of-motion of a manipulator. Past analyses have constrained the allowable manipulator movements or its environment. The analyses presented in this paper do not make such restrictions, and the findings appear to be applicable to a multitude of robotic manipulators.

TABLE VIII
MODELS OF JOINT 3 FOR REGIONS ABOUT $\theta_3(0)$

Test	$\theta_3(0)$ (deg)	Discrete Gain K	Discrete Zero z_0	Discrete Pole 1 p_1	Discrete Pole 2 p_2	Analog Pole 1 s_1	Analog Pole 2 s_2
1	0	$.5268\times10^{-4}$	-1.2352	0.9210	.09721	-16.46	-5.66
2	45	$.5141\times10^{-4}$	-0.8259	0.9151	0.9776	-17.74	-4.53
3	90	$.3436\times10^{-4}$	-2.0134	0.8757	1.0024	-26.55	0.48
4	135	$.2496\times10^{-4}$	-3.3606	0.8589	1.0126	-30.42	2.50
5	180	$.2473\times10^{-4}$	-3.4471	0.8552	1.0143	-31.28	2.84
6	225	$.2787\times10^{-4}$	-2.9070	0.8602	1.0096	-30.12	1.91
7	270	$.3684\times10^{-4}$	-1.8087	0.8793	0.9977	-25.23	-0.46
8	315	$.5387\times10^{-4}$	-0.7233	0.9254	0.9686	-15.50	-6.38

REFERENCES

[1] K. J. Astrom, P. Hagander, and J. Sternby, "Zeros of sampled systems," in *Proc. 19th IEEE Conf. Decision and Control*, vol. 2, Albuquerque, NM, Dec. 10–12, 1980, pp. 1077–1081.

[2] A. K. Bejczy, *Robot Arm Dynamics and Control*. Jet Propulsion Lab. Pasadena, CA, Tech. Rep. 33-669, Feb. 1974.

[3] G. Birkhoff and G. C. Rota, *Ordinary Differential Equations*. Boston, MA: Ginn and Co., 1962, p. 294.

[4] L. Collatz, *The Numerical Treatment of Differential Equations*. Berlin: Springer-Verlag, 1960, p. 71.

[5] J. M. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, no. 11, pp. 730–736, Nov., 1980.

[6] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," Mass. Institute of Technology, Cambridge, MA, A.I. Memo 700, Jan., 1983.

[7] B. K. P. Horn, K. Hirokawa, and V. V. Vasirani, "Dynamics of a Three Degree-of-Freedom Kinematic Chain," Mass. Institute of Technology, Cambridge, MA, A.I. Memo 478, Oct., 1977.

[8] International Mathematical and Statistical Libraries, Inc., Houston, TX, IMSL Library Reference Manual, 1980.

[9] Inland Motors, Ltd., 501 First Street, Radford, VA 24141, Inland Motor Specification Catalog, 1981.

[10] B. C. Kuo, *Automatic Control Systems, 4th ed*, Englewood Cliffs, NJ: Prentice-Hall, 1982, pp. 181–183.

[11] C. S. G. Lee, "Robot arm kinematics, dynamics and control," *Comput.*, vol. 15, no. 12, pp. 62–80, Dec., 1982.

[12] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *J. Dynamic Systems, Measurement and Control*, vol. 102, no. 2, pp. 69–76, June, 1980.

[13] J. Y. S. Luh, "Conventional controller design for industrial robots—A tutorial," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 3, pp. 298–316, May/June, 1983.

[14] S. S. Mahil, "On the application of Lagrange's method to the description of dynamic systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-12, no. 6, pp. 877–889, Nov./Dec., 1982.

[15] S. Meghed, "New Lagrangian formulation of manipulator dynamics," in *11th Int. Symp. Industrial Robots*, Tokyo, Japan, Oct., 1981, pp. 765–771.

[16] C. P. Neuman and C. S. Baradello, "Digital transfer-functions for microcomputer control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 12, pp. 856–860, Dec., 1979.

[17] C. P. Neuman, "Equivalent time-constants for nonoscillatory second-order processes," *IEEE Trans. Industrial Electronics and Control Instrumentation*, vol. IECI-27, no. 4, pp. 268–270 Nov., 1980.

[18] C. P. Neuman and H. W. Stone, "MRAC control of robotic manipulators," in *3rd Yale Workshop on Applications of Adaptive Systems Theory*, K. S. Narendra, Ed., Yale Univ., New Haven, CT, June 15–17, 1983, pp. 203–210.

[19] C. P. Neuman and V. D. Tourassis, "Robot control: issues and insight," in *Proc. 3rd Yale Workshop on Applications of Adaptive Systems Theory*, K. S. Narendra, ed., Yale Univ., New Haven, CT, June 15–17, 1983, pp. 179–189.

[20] C. P. Neuman and R. L. Morris, "Non-invertible digital transfer functions arising from minimum phase analog process models," *J. Dynamic Systems, Measurement and Control*, vol. 105, no. 1, pp. 46–48, Mar., 1983.

[21] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, p. 29.

[22] W. J. Palm, *Modeling, Analysis and Control of Dynamic Systems*. New York: Wiley, 1983, pp. 202–203.

[23] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, MA: MIT Press, 1981.

[24] J. R. Rice, *Numerical Methods, Software and Analysis*. New York: McGraw-Hill, 1983, p. 277.

[25] W. M. Silver, "On the equivalence of Lagrangian and Newton–Euler dynamics for manipulators," *Int. J. Robotics Res.*, vol. 1, no. 2, pp. 60–70, Summer, 1982.

[26] T. Soderstrom, "On continuous-time counterparts to ARMA models," Institute of Technology, Uppsala Univ. Uppsala, Sweden, Tech. Rep., May 1982.

[27] K. Steiglitz and L. E. McBride, "A technique for the identification of linear systems," *IEEE Trans. Automat. Contr.*, vol AC-10, no. 5, pp. 461-464, Oct., 1965.
[28] P. Stoica and T. Soderstrom, "The Steiglitz-McBride identification algorithm revisited—Convergence analysis and accuracy aspects," *IEEE Trans. Automat. Contr.*, vol. AC-24, no. 3, pp. 712-717, June 1981.
[29] H. W. Stone, "Dynamic modeling and adaptive control of a three degree-of-freedom robotic manipulator," Dept. Electrical and Computer Engineering, Carnegie-Mellon Univ., Pittsburgh, PA, Master's thesis, Feb., 1983.
[30] M. Takegaki and S. Arimoto, "An adaptive trajectory control of manipulators," *Int. J. Control*, vol. 34, no. 2, pp. 219-230, Feb., 1981.
[31] Unimate Inc., Danbury, CT, "Puma 250 Hardware Manual," 1981.
[32] M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *J. Dynamic Systems, Measurement and Control*, vol. 104, no. 3, pp. 205-211, Sept., 1982.

Specifications for Interactive Environmental Management

KAZUHIKO KAWAMURA, MEMBER, IEEE

***Abstract*—The application of large-scale computer modeling to the analysis of public policy issues has received much attention due to efforts of Forrester and others during the 1960's [1]. This trend has continued during the 1970's as we wrestled to resolve many complex issues such as the protection of the environment. In response to the need for comprehensive assessment of environmental quality, the development of a large-scale computer model called SEAS was initiated by EPA in 1972. Although SEAS has provided EPA with a framework for the collection, storage, and treatment of environmental data, it has increasingly become apparent that SEAS has several major limitations that preclude its usefulness as a strategic environmental management tool. In this correspondence, major limitations of using large-scale models like SEAS are reviewed, key design specifications, are offered, and a design process to ameliorate some of the limitations of using such models for envimental management are proposed.**

I. Introduction

During the 1960's, America enjoyed steady economic growth. Unemployment was low and per capita income was increasing. However, this steady economic growth also brought the nation a new set of problems such as air and water pollution. Thus the protection of the environment became a major water public issue by the end of the decade. In order to respond to the growing public demand for protecting the environment, President Nixon signed the National Environmental Policy Act of 1969 (NEPA) on January 1, 1970. Suddenly new agencies were crated throughout the public sector to manage the environment better.

Throughout the seventies, responsibility for monitoring, maintaining and enhancing environmental quality was vested in the Federal government, particularly in the Environmental Protection Agency (EPA). The original legislative mandate necessitated EPA to look at the environment from a holistic, long-range point of view. In the early 1970's, Congress passed a series of air and water pollution control acts known as The Clean Air and Clean Water Acts. However, there existed very few analytical tools available for rapid, systematic, and comprehensive assessment of the impact of such programs. In response to this need, EPA began the development of a large-scale computer model called the strategic environmental assessment system (SEAS)[1] in the fall of 1972 [2], [3]. SEAS was to have a forecasting capability that would detect changes in pollutant levels and the effects that would result from a set of macroeconomic and demographic changes. Also required was the ability to project the impacts of environmental regulations and control technologies on the economy, on the natural environment, and on society.

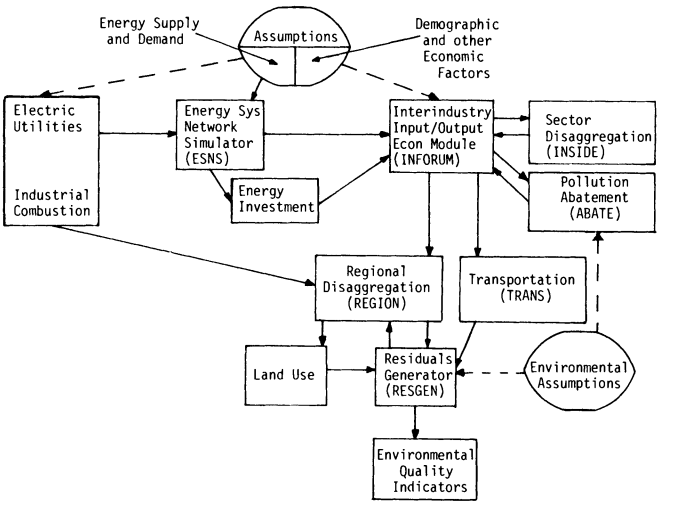

Fig. 1. SEAS block diagram. Source: Environmental Protection Agency, "Strategic environmental assessment systems," draft copy. December 16, 1975.

II. Limitations of SEAS

In his article "Pitfalls in formulation and modeling," Quade lists a number of pitfalls for those making use of the results from large-scale computer models [4]. One area of concern involves the interface between the model builder and the user. Typically large-scale models have been expert-oriented, and often it has been difficult to assess their quality even by experts. This is particularly serious in the environmental management, where the participation by the interested parties in policy making is desirable and indeed essential.

In the following we use SEAS to illustrate the limitations of using large-scale models as environmental management tools.

SEAS Block Diagram

SEAS is a system of special-purpose modules linked to an input–output model of the United States economy developed by the University of Maryland. The basic SEAS block diagram is shown in Fig. 1. The SEAS methodology begins with a set of assumptions on energy supply and demand, economics, and demography (high growth and low growth) and the environment. These assumptions then are fed into an energy model (ESNS) which provides activity levels for energy-related sectors. The outputs from ESNS are fed into an input–output model (Inforum) that computes the values of the intersector flows of goods and services from the various producing and consuming sectors of the economy. These projections of the national flow of goods and services are then fed into a pollution abatement module (ABATE) to estimate air and water pollution abatement costs by sector.

The outputs of Inforum are also fed into a residuals model (RESGEN) through a regional disaggregation module (REGION)

Manuscript received April 11, 1983; revised March 1984.

K. Kawamura is with the Department of Electrical and Biomedical Engineering and the Program in Management of Technology, Vanderbilt University, Nashville, TN 37235. This work is based on one that was presented at the 1982 IEEE International Conference on Cybernetics and Society, Seattle, WA.

[1] In 1981, EPA renamed SEAS as Residuals Accounting Model (RAM). We will, however, keep its original name in this correspondence to minimize confusion.

0018-9472/84/0700-0654$01.00 ©1984 IEEE