論 文
35~5~2

로보트 운동에 대한 공간 좌표 제어

Cartesian Coordinate Control of Robot Motion

盧 瑩 植*·禹 廣 芳**
(Young - Shick Ro · Kwang B. Woo)

요 약

본 논문에서는 공간좌표에서 미리 설정된 로보트 움직임을 제어하기 위한 효과적인 공간 좌표 모델을 제안하고 자기 동조 제어기를 이용하여 다이나믹스에 대한 사전 지식없이 제어할 수 있는 적응 궤환 제어기를 설계하였다. 로보트 손에 작용하는 속도와 힘에 대한 각 혼성 변수 집합이 서로 독립적이라 가정하면 각종 모델화되지 않은 효과를 보상해 주는 항을 갖는 1차 SISO 모델로 모델화할 수 있으며 추정해야할 모델의 파라미터 수는 최소가 된다. 제안된 모델에 대해 주어진 평가지표를 최소로 하는 자기 동조 제어기를 설계하고 계산된 제어력은 Jacobian 행렬을 이용하여 관절에 가해야할 토오크로 변환한다. 제시된 모델과 제어기의 유용성을 입증하기 위해 다른 공간 좌표 제어기와 비교 검토되었다.

Abstract

An effective cartesian coordinate model is presented to control a robot motion along a prescribed time-based hand trajectory in cartesian coordinates and to provide an adaptive feedback design approach utilizing self-tuning control methods without requiring a detailed mathematical description of the system dynamics. Assuming that each of the hybrid variable set of velocities and forces at the cartesian coordinate level is mutually independent, the dynamic model for the cartesian coordinate control is reduced to first-order SISO models for each degree of freedom of robot hand, including a term to represent all unmodeled effects, by which the number of parameters to be identified is minimized. The self-tuners are designed to minimize a chosen performance criterion, and the computed control forces are resolved into applied joint torques by the Jacobian matrix. The robustness of the model and controller is demonstrated by comparing with the other cartesian coordinate controllers.

1. Introduction

The dynamic models play an impotant role in two different phases in robot motion control. The one stems from its use in simulation of robot motion in order to test control strategies and manipulator programs without the expense and mechanical problems of working with actual manipulators[1] and its use in analysis of manipulator dynamics which could aid in the mechanical design of prototype arms [2]. The two main

*正 會 員 : 延世大 大學院 電氣工學科 博士課程
**正 會 員 : 延世大 工大 電氣工學科 教授 · 工博
接受日字 : 1985年 12月 27日

approaches towards deriving the dynamic equations of motion for manipulators have been the Lagrange-Euler equations and the Newton-Euler equations in the form of highly interactive cross coupled non-linear system.

The other is that they are utilized to determine control laws of strategies to achieve the desired system response and performance. A number of models have been proposed: the nonlinear model derived on the basis of Newtonian mechanics[3], the linearized model by introducing a nonlinear feedback loop into the system such that the nonlinear terms in the manipulator model are cancelled[4], the linear perturbation models which are obtained by the linearization about the desired trajectory[5], and the linear autoregressive models which fit the input-output data from the manipulator[6]. However, the parameters in these models are not known exactly, and vary as the payload which is often unknown. For these reasons, there is an increasing interest in adaptive control of mechanical manipulators.

Dubowsky[7] proposed a model referenced adaptive control which uses a linear second-oder differential equation as the reference model for each joint of the robot arm. Koivo[6] proposed an adaptive self-tuning controller using an autoregressive model in joint coordinates. Both algorithms assume that the interaction forces among the joints are negligible; thus, the dynamic model of the manipulator is simplified. However, if the desired motion of a manipulator is specified in terms of a time-based hand trajectory, the equation of motion of the manipulator hand should be derived in cartesian coordinates. Lee[8] developed a resolved motion adaptive control based on the linearized purturbation system in catesian coordinate. Leininger[9] proposed the pole placement self-tuner assuming that the dynamics of hand position error can be represented as second-oder autoregressive form for each degree of freedom in cartesian coordinates.

In this paper, a cartesian coordinate model is proposed which is given in first-order autoregressive type for velocity and force of the manipulator hand. Based on this model, a self-tuning controller is designed to adjust control parameters on-line using the system parameter identification methods so that a chosen performance criterion is minimized. Simplication of the model is discussed with special emphasis on the robustness of scalar self-tuner using SISO model.

The proposed control scheme is applied to the control of JPL—Stanford Arm, and is compared with the other cartesian coordinate control methods.

2. Cartesian Coordinate Self-tuner

The dynamics of a six-joint manipulator can be derived either by Lagrange-Euler or Newton-Euler formulations. In general, the Lagrange-Euler equations of motion of a six-joint manipulator can be expressed in vector matrix notation as,

$$D(q)\ddot{q}(t)+H(q,\ \dot{q})+G(q)=\tau \tag{1}$$

where τ is a 6 by 1 applied torque vector for joint actuators, and q is a 6 by 1 joint position vector.

Since the inertia matrix, D(q), is always nonsingular, $\ddot{q}(t)$ can be obtained from (1):

$$\ddot{q}=D^{-1}(q)[\tau-H(q,\ \dot{q})-G(q)] \tag{2}$$

To apply the STC (Self-Tuning Control) algorithm, a discrete time model of the manipulator is proposed. Using the first-order approximation of the Taylor series expansion about the time, a multivariable discrete time-varing model for the control system can be obtained:

$$\begin{aligned}\dot{q}(k+1)&=\dot{q}(k)+TD^{-1}(q)\ [\tau-H(q,\ \dot{q})-G(q)]\\&=\dot{q}(k)+A(k)\,\tau(k)+a(k)\end{aligned} \tag{3}$$

where T is a sampling time,

$$A(k)=TD^{-1}(q),$$
$$\text{and } a(k)=-TD^{-1}(q)\ [H(q,\ \dot{q})+G(q)]$$

To derive the equation of motion of the manipulator hand in cartesian coordinates, the velocity v and force f of the manipulator hand are defined as follows,

$$v=(v_x,\ v_y,\ v_z,\ \omega_x,\ \omega_y,\ \omega_z)^T \tag{4-a}$$
$$f=(f_x,\ f_y,\ f_z,\ m_x,\ m_y,\ m_z)^T \tag{4-b}$$

where v_x is linear velocity, ω_x is angular velocity, f_x is force, and m_x is moment, along x-axis of reference coordinate frame.

Based on the concept of moving coordinate frame, the joint velocity and torque can be obtained from the hand velocity and force, respectively,

$$\dot{q}=(J(q))^{-1}v \tag{5-a}$$
$$\tau=J^T(q)f \tag{5-b}$$

where J(q) is the 6 by 6 Jacobian matrix whose i th column vector Ji can be found from,

$$J_i=\begin{cases}\begin{bmatrix} z_{i-1} \times (p-p_{i-1}) \\ z_{i-1}\end{bmatrix} & \text{;if joint i is rotational}\\ \begin{bmatrix} z_{i-1} \\ 0\end{bmatrix} & \text{;if joint i is translational}\end{cases}$$

where X indicates cross product, p_{i-1} is the position of the origin of the (i-1)th coordinate frame with respect to the reference frame, z_{i-1} is the unit vector along the axis of motion of joint i, and p is the position of the hand with respect to the reference coordinate frame.

Using (4), (5-a), and (5-b), (3) can be expressed as,

$$\begin{aligned} v(k+1) &= v(k)+J(q)\ A(k)\ J^T(q)\ f(k)+J(q)\ a(k)\\ &= v(k)+B(k)\ f(k)+b(k) \end{aligned} \tag{6}$$

where $B(k)=J(q)\ A(k)\ J^T(q)$, and $b(k)=J(q)\ a(k)$

Computational time delay is accounted by,

$f(k)=u(k-1)$

where u(k) is the control input. Introducing e(k) as the modeling errors, the equation of the manipulator hand in cartesian coordinates can be obtained for STC control system,

$$v(k+1)=v(k)+B(k)\ u(k-1)+b(k)+e(k) \tag{7}$$

Equation (7) is first-order MIMO autoregressive type and contains only two parameters, B(k) and b(k).

The controller aims at minimizing the loss due to the deviation of the output from the reference vector w as well as the loss in energy needed to exert the control effort u. This combined performance criterion is expressed as [10],

$$I(u)=E[\|v(k+2)-w(k+2)\|^2+\|R\ u(k)\|^2] \tag{8}$$

where $w(k+2)=v^d(k+2)+C[y^d(k)-y(k)]/T$, and R, C are positive definite weighting matrices, and v^d, y^d describe the desired velocity and position vector of the hand. The control, minimizing (8), is,

$$\begin{aligned} u(k)=&[B^T(k)B(k)+R^TR]^{-1}\ B^T(k)\\ &[w(k+2)-v(k)-B(k)u(k-1)-2b(k)] \end{aligned} \tag{9}$$

This control law can handle time-varing reference signals as well as certain nonminimum-phase system.

The direct use of (9) is difficult because the calculation of the model parameters is a tedious task and it depends on the payload which is often unknown. Using the principle of STC [11], the model parameters are estimated by recursive identification algorithm, and the estimated parameters are used in optimal control law of (9).

Equation(7) can be rewritten as follows :

$$v(k+1)=v(k)+X^T(k)\theta(k)+e(k+1) \tag{10}$$

where $X(k)=[u^T(k-1),\ 1]^T$, and $\theta(k)=[B(k),\ b(k)]^T$.

The recursive least-squares algorithm is given as,

$$\hat{\theta}(k+1)=\hat{\theta}(k)+L(k)[v(k+1)-v(k)-X^T(k)\hat{\theta}(k)] \tag{11}$$

$$L(k)=P(k)X(k)/[s+X^T(k)P(k)X(k)]$$

$$P(k+1)=[P(k)-L(k)X^T(k)P(k)]/s$$

where the caret refers to estimated values, s is a forgetting factor for exponential decaying of past data in tracking a slow drift in the parameters within the range between $0.9\leq s\leq 1.$, and P is a (n+1) by (n+1) matrix for updating the parameters. Computationally it is more advantageous to use the square-root algorithm, making it possible for the positive definite P-matrix to lose this property.

The estimated parameters are then employed to determine the optimal feedback gain and control force of (9). These control forces can be resolved into applied joint torques, as shown by (5-b):

$$\tau(k+1)=J^T(q)u(k)$$

The overall block diagram of the control system is shown in Fig. 1, where the hand position is represented by the homogeneous transformation matrix H, and Jacobian play a role in interfacing between joint coordinate system and hand coordinate system.

In order to restrict the complexity of the multivariable model of (7), the model variables may be decomposed into two groups with the assumption that the coupling effects of the force and moment vectors

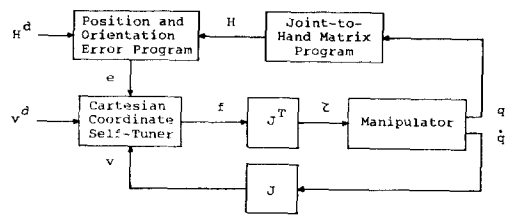


Fig. 1. Block diagram of cartesian coordinate control.

of (4-b) on the linear and angular velocities of (4-a) are negligible. Thus, the 6 by 6 matrix B(k) in (7) is decomposed into two 3 by 3 matrices by which two 3-dim self-tuners are determined for each groups. Also, if interactions between the hybrid variable sets are small in cartesian coordinates, coupling terms in the model of (7) are removed, thus, six scalar self-tuners are designed for each degree of freedom of robot hand.

3. Scalar Self-Tuner

The multivariable autoregressive model of (7) can be decoupled by the assumption that the coupling effects of forces on the velocities of (4) are negligible. Consequently, the decoupled system can be considered to consist of a set of independent single-variable system with first-order SISO model, as follows:

$$v_i(k+1)=v_i(k)+B_iu_i(k-1)+b_i+e_i(k+1) \qquad (12)$$

where $i=1, 2, \ldots, 6$, and b_i contains all unmodeled effects which are related to linear approximation and coupled interactions, as well as gravity, coriolis and centrifugal forces.

The performance criterion is chosen in the following form:

$$I_i(u_i)=E[(v_i(k+2)-w_i(k+2))^2+(R_iu_i(k))^2] \qquad (13)$$

where $w_i(k+2)=v_i^d(k+2)+C_i[y_i^d(k)-y_i(k)]/T$

The optimal control law becomes,

$$u_i(k)=B_i/(B_i^2+R_i^2)^* [w_i(k+2)-v_i(k)-B_iu_i(k-1)-2b_i] \qquad (14)$$

Equations (11) can be programed for on-line computations of the parameter estimation, and (14) for the on-line computation of the controls.

Note that the number of scalar self-tuners equals the number of degree of freedom in cartesian coordinates, and each self-tuner requires to identify only two parameters.

Computer simulation is carried out to demonstrate its applicability.

4. Computer Simulation

Each of the proposed control schemes, 6-dim self-tuner, 3-dim self-tuner, and scalar self-tuner, were applied to the dynamic simulation model of the JPL–Stanford Arm with six degree of freedom[12].

Dynamic equation is solved by using the recursive Newton-Euler algorithm[13] and the 4-th order Runge-Kutta integration method.

A hypothetical problem[3] in which the Stanford manipulator picks up an object on a conveyor with a known velocity of 15 Cm/s, and then places it on table, as illustrated in Fig. 2, was simulated on PDP 11/44 computer. In the figure, s^i, $i=1, 2, \ldots, 8$, is the i-th corner point of the traveling path of the manipulator hand, and it consists of first three components for the location and second three components for the orientation in terms of Euler angle. The kinematic relation between Euler angle and hand orientation refers to Appendix. The input data for s^i are specified in Table1.

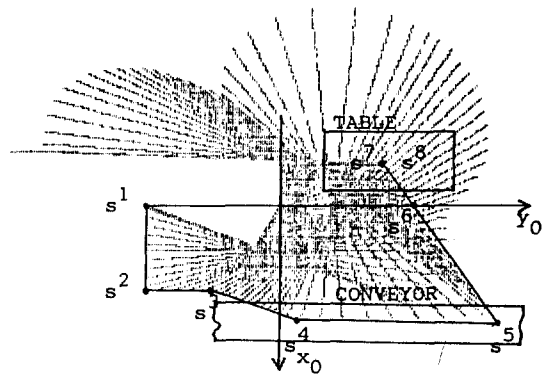


Fig. 2. An Example of simulation (0.5 sec/motion).

First, for analytic simplicity, a simple motion is tested to move the hand from point s^4 to s^5 and to s^4 along straight lines in 10 sec. The preplanned position and velocity trajectory is illustated in Fig. 3. The simulation result for the system with the adaptive control scheme, 6-dim self-tuner, 3-dim self-tuner, and scalar self-tuner, respectively, is shown in Fig. 4, and computed values and estimated values of model parameter at s^4 are represented in Table 2. Also the

Table 1. Input data for a simulation problem.

	s^1	s^2	s^3	s^4	s^5	s^6	s^7	s^8
px(Cm)	0	20	30	40	40	0	−15	−15
py(Cm)	−40	−40	−20	5	65	40	30	30
pz(Cm)	30	30	20	4	4	4	15	5
α (deg)	0	45	90	180	180	225	225	225
β (deg)	135	135	135	150	150	135	180	180
γ (deg)	0	0	0	0	0	0	0	0

Table 2. The Computed value and estimated value of model parameters at s4.

	B						b
Computed Value	1.29E−3	−3.07E−4	1.12E−4	2.26E−4	1.13E−3	8.31E−4	−9.82E−3
	−3.07E−4	3.94E−4	−1.87E−5	−3.04E−4	−3.11E−4	5.37E−4	2.19E−3
	1.12E−4	−1.87E−5	2.71E−4	−2.53E−4	−4.70E−4	1.21E−4	1.76E−3
	2.26E−4	−3.04E−4	−2.53E−4	7.46E−2	−6.07E−3	1.53E−2	−4.29E−2
	1.13E−3	−3.11E−4	−4.70E−4	−6.07E−3	1.92E−1	−1.79E−1	−1.13E−2
	8.31E−4	5.37E−4	1.21E−4	1.53E−2	−1.79E−1	3.90E−1	−1.49E−2
6-dim Self-Tuner	1.30E−3	−2.65E−4	1.25E−4	9.89E−4	−4.98E−3	9.63E−3	−1.02E−2
	−3.10E−4	3.71E−4	−3.00E−5	−1.06E−3	3.57E−3	−4.73E−3	2.53E−3
	1.16E−4	−3.33E−5	2.68E−4	−2.67E−4	1.98E−3	−3.82E−3	1.72E−3
	2.22E−4	−2.53E−4	−2.23E−4	7.61E−2	−1.65E−2	3.01E−2	−4.31E−2
	1.15E−3	−2.54E−4	−4.43E−4	−4.25E−3	1.83E−1	−1.69E−1	−1.21E−2
	7.93E−4	5.61E−4	1.26E−4	1.44E−2	−1.85E−1	4.03E−1	−1.43E−2
3-dim Self-Tuner	9.77E−4	−2.05E−4	3.37E−5	0.	0.	0.	−7.84E−3
	−2.58E−4	3.60E−4	−4.18E−6	0.	0.	0.	1.79E−3
	1.78E−4	−4.26E−5	2.80E−4	0.	0.	0.	1.18E−3
	0.	0.	0.	6.57E−2	4.99E−3	−1.62E−3	−3.41E−3
	0.	0.	0.	−4.94E−3	1.87E−1	−1.72E−1	1.85E−3
	0.	0.	0.	1.33E−2	−1.72E−1	3.73E−1	−7.26E−3
Scalar Self-Tuner	1.19E−3	0.	0.	0.	0.	0.	−1.02E−2
	0.	3.65E−4	0.	0.	0.	0.	−3.89E−4
	0.	0.	2.55E−4	0.	0.	0.	2.43E−3
	0.	0.	0.	7.32E−2	0.	0.	−3.79E−2
	0.	0.	0.	0.	1.39E−1	0.	−1.15E−3
	0.	0.	0.	0.	0.	2.55E−1	−1.23E−3

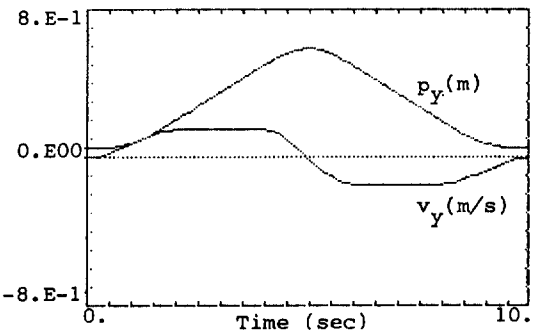


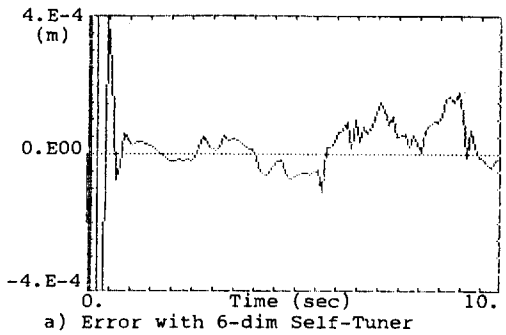
Fig. 3. Preplanned trajectory of py and vy.

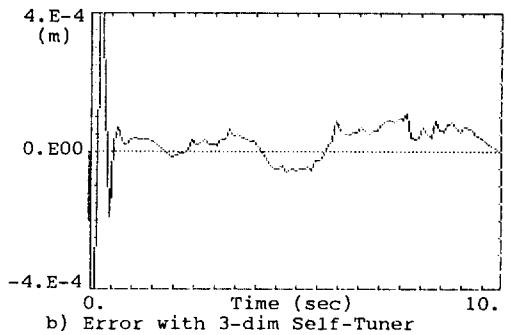
CCSS (Cartesian Coordinate Scalar Self-tuner) was compared with the other cartesian coordinate control, including the RAMC (Resolved Acceleration Motion Control)[3], and the JCSS (Joint Coordinate Scalar Self-tuner)[6]. The resulting errors between the preplanned trajectory and real motion are illustrated in Fig. 5, and the necessary algorithms are summurized in Table 3. The RAMC has the smooth motion property. However, the computation of inverse-dynamics requires a large amount of computing and knowledges of precise dynamic parameters. In using the simplified inverse-dynamics[13], which is obtained from ignoring velocity terms and coupling terms, the response of the SRAMC (Simplified RAMC) is biased as shown Fig. 5. The JCSS, using the joint coordinate model of (3) and neglecting all coupling effects, has similar response to the CCSS, whose response oscillates while trying to follow the desired value and then settles down to the extend of comparing with the RAMC, but it requires slightly more computational burden than the CCSS.

Next, the scalar self-tuner was applied to the complete task of Fig. 2, and its maximum error and error variance are shown in Table 4.

Table 3. The Necessary algorithm for each of controls

RAMC	Inverse-Dynamics, Inverse-Kinematics, Inverse-Jacobian
SRAMC	Simplified Inverse-Dynamics, Inverse-Kinematics, Inverse-Jacobian
JCSS	Inverse-Kinematics, Inverse-Jacobian
CCSS	Kinematics, Jacobian





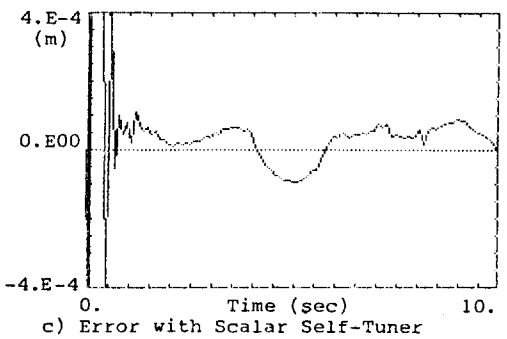
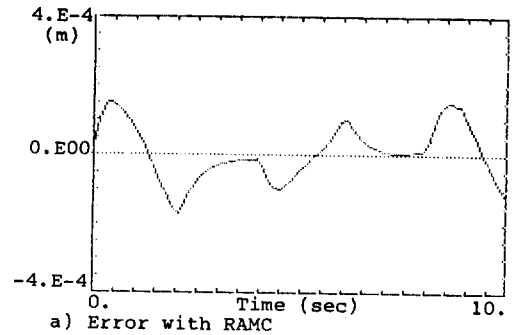
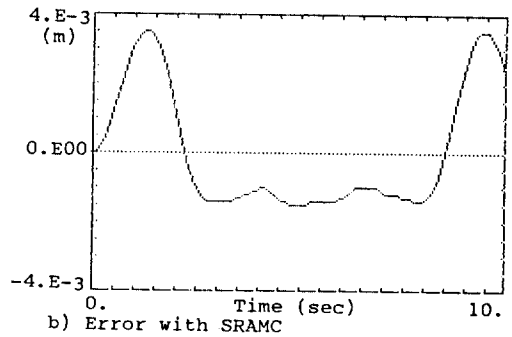


Fig. 4. Error responses of py.





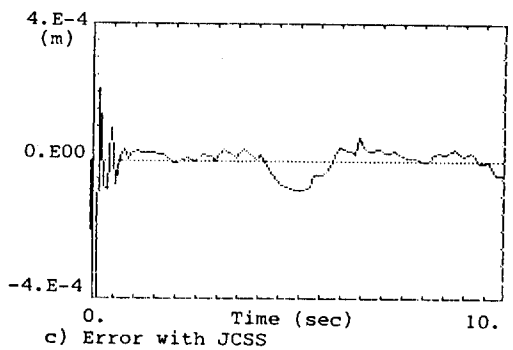


Fig. 5. Error responses of py.

Table 4. The Response errors of scalar self-tuner.

Coordinate Variables		Maximum Position Error	Maximum Velocity Error	Position Error Variance	Velocity Error Variance	Final Position Error	Final Velocity Error
Position (mm)	x	0.6473	8.3970	0.1897	1.4550	−0.0235	0.0005
	y	1.1740	24.7000	0.3130	2.9980	0.3979	−0.0341
	z	0.9152	12.1000	0.2335	1.6800	−0.0965	0.0108
Orientation (deg)	x	0.4236	7.1734	0.1063	0.9534	0.0003	−0.0027
	y	0.2685	11.2987	0.0789	1.2193	−0.0043	−0.0005
	z	0.2329	11.6941	0.0394	1.5476	0.0000	0.0021

The simulation result shows that the scalar self-tuner is well applicable to the control of robot hand in cartesian coordinates, and its operation is comparable to the performance of the systems, based on multivariable models with coupling term, and using the RAMC or the JCSS.

The main advantage of scalar self-tuner is its simplicity, and its adaptability to unmodeled effect and undesired disturbance by adjusting control parameters to compensate directly for the error in cartesian coordinates. The controller can be implemented using microprocessors.

5. Conclusion

An effective cartesian coordinate model and an adaptive controller of self-tuning type, presented in this paper, is for the closed-loop control of the hand of the manipulator along prescribed trajectory in cartesian coordinates.

The nonlinear dynamic model for velocity and force of hand are decomposed into six first-order SISO models assuming that tne coupling effects between the hybrid variable sets are negligible. All the unmodeled effects, resulting from this simplication, are compensated by a bias term which is estimated by recursive least-squares algorithm, based on the fact that it is used with optimal sense in linearization of nonlinear system, reduction of model's order, and decoupling of multivariable model.

The algorithm of the scalar self-tuner using these models is very simple, and numerical examples demonstrate, that it is comparable to the performance of the systems, based on multivariable model with coupling terms, and using the other cartesian coordinate control. The proposed method can be shown to have significant advantages over the other cartesian coordinate controls.

References

1) M. W. Walker, and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," ASME Trans. J. Dyn. Sys. Meas., and Control, vol. 104, pp. 205–211, 1982.
2) H. Asada, "A Characteristics Analysis of Manipulator Dynamics Using Principal Transformation," Proc. Amer. Control Conf., Wachington, D.C., June, 1982.
3) Luh, J., Walker, M., and Paul, R., "Resolved Acceleration Control of Mechanical Manipulators," IEEE Trans. AC-25, No. 3, pp. 468-474, 1980.
4) H. Hemani, and P. C. Camana, "Nonlinear Feedback in Simple Locomotion Systems," IEEE Trans. Automat. Contr., pp. 855-860, 1979.
5) C. S. G. Lee, and M. J. Chung, "An Adaptive Control Strategy for Computer-Based Manipulator," Proceedings of the 21st Conference of Decision and Control, pp. 95-100, 1982.
6) Koivo, A., and Guo, T., "Adaptive Linear Controller for Robotic Manipulators," IEEE Trans. Auto. Control, AC-28, No. 2, pp. 162-171, 1983.
7) S. Dubowsky, and D. T. Desforges, "The Application of Model Referenced Adaptive Control to Robotic Manipulators," Trans, of ASME, J. Dynamic Syst., Meas., and Contr., vol. 101, pp. 193-200, 1979.
8) C. S. G. Lee, M. J. Chung, and B. H. Lee, "Adaptive Control for Robot Manipulators in Joint and Cartesian Coordinates," International Conference on Robotics, Atlanta, March, 1984.
9) G. G. Leininger, "Adaptive Control of Manipulators using Self-tuning Methods," Robotics Research, MIT Press, 1984.
10) Koivo, H. N., "A Multivariable Self-Tuning Controller," Automatica, vol. 16, pp. 351-366, 1980.
11) Aström, K. J., and Wittenmark, B., "On Self-Tuning Regulators," Automatica, 9, pp. 185-199, 1973.
12) Bejczy, A. K., "Robot Arm Dynamics and Control," JPL Technical Report 33-669, Pasadena, CA, 1974.
13) Luh, J., Walker, M., and Paul, R., "On-Line Computational Scheme for Mechanical Manipulators," ASME Trans. J. Dyn. Sys. Meas., and Control, vol. 102, No. 2, pp. 67-76, 1980.
14) R. P. Paul, Robot Manipulators—Mathematics, Programming, and Control, MIT Press, 1981.

Appendix

The Kinematic Relation Between Euler Angle and Hand Orientation.

Orientation of robot hand is represented by 3-by-3 rotation matrix, Euler coordinates, or RPY coordinate, etc. Since the joint structure of Stanford manipulator for hand orientation is similar to the structure of Euler transformation, the Euler angle is frequently used to specify hand orientation. When the set of Euler angle is α, β, and γ, the rotation matrix is as illustrated in [5];

$$
\begin{aligned}
R &= \mathrm{Rot}(z, \alpha)\ \mathrm{Rot}(x, \beta)\ \mathrm{Rot}(z, \gamma) \\
&= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c\alpha c\gamma - s\alpha c\beta s\gamma & -c\alpha s\gamma - s\alpha c\beta c\gamma & s\alpha s\beta \\ s\alpha c\gamma + c\alpha c\beta s\gamma & -s\alpha s\gamma - c\alpha c\beta c\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix} \\
&= [\mathrm{n},\ \mathrm{o},\ \mathrm{a}]
\end{aligned}
$$

where n, o and a are unit vectors of hand coordinate frame as Fig. A-1. Its inverse relation, based on [14' is derived as,

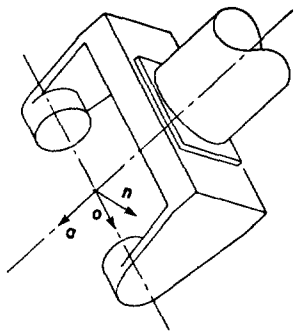


Fig. A-1. n, o, and a vectors.

$$\alpha = \mathrm{atan2}(a_x,\ -a_y)$$

and $\alpha = \alpha + 180$

$$\beta = \mathrm{atan2}(a_x s\alpha - a_y c\alpha,\ a_z)$$

$$\gamma = \mathrm{atan2}(-o_x c\alpha - o_y s\alpha,\ n_x c\alpha + n_y s\alpha)$$

The orientation error between the desired hand coordinates and the real hand coordinates and be specified by the generalized orientation vector.

$$\mathrm{Rd} = \mathrm{Rot}(k, \theta)\ \mathrm{R}$$

where Rd is desired hand rotation matrix, R is real hand rotation matrix, and k is unit generalized orientation error vector with respect to the reference coordinate frame as Fig. A-2. The computation of k and θ refers to [14]. The generalized orientation error e is defined by,

$$e = \theta\ k$$

The relation between angular velocity and Euler angle can be derived from the equation [14]:

$$\frac{d}{df} = \Delta R$$

where Δ is the differential rotation transformation of the hand. Since the inverse of a rotation matrix is equivalent to its transpose, the angular velocity of the hand coordinate frame abut the principal axes of the reference frame is obtained as following;

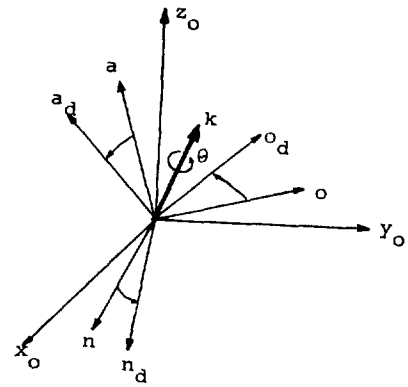


Fig. A-2. Generalized orientation error vector.

$$
\Delta = \frac{d}{dt} R\ R^T = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & -\dot{\alpha} - c\beta\dot{\gamma} & s\alpha\dot{\beta} - c\alpha s\beta\dot{\gamma} \\ \dot{\alpha} + c\beta\dot{\gamma} & 0 & -c\alpha\dot{\beta} - s\alpha s\beta\dot{\gamma} \\ -s\alpha\dot{\beta} + c\alpha s\beta\dot{\gamma} & c\alpha\dot{\beta} + s\alpha s\beta\dot{\gamma} & 0 \end{bmatrix}
$$

From the above equation, the relation between the$(\omega_x,\ \omega_y,\ \omega_z)$ and $(\dot{\alpha},\ \dot{\beta},\ \dot{\gamma})$ can be found by equating the non-zero elements in the matrices;

$$
\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 & c\alpha & s\alpha s\beta \\ 0 & s\alpha & -c\alpha s\beta \\ 1 & 0 & c\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}
$$

and its inverse relation can be found easily;

$$
\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \sec\beta \begin{bmatrix} -s\alpha c\beta & c\alpha c\beta & s\beta \\ c\alpha s\beta & s\alpha s\beta & 0 \\ s\alpha & -c\alpha & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}
$$