International Journal of Robotics and Automation, Vol. 3, No. 1, 1988

ROBUST CONTROL IN OPERATIONAL SPACE FOR GOAL-POSITIONED MANIPULATOR TASKS

Jean-Jacques E. Slotine,* Oussama Khatib** and Douglas Ruth†

Abstract

The operational space formulation has provided a fundamental tool for the description of the dynamic behavior of manipulator end-effectors. Based on this formulation and the artificial potential field concept, a real-time collision avoidance method using goal-positioned control of robot systems has been developed. This paper shows that the desired goal-positioned behavior can be naturally achieved by properly defining a target sliding surface in operational space, and then applying the systematic methodologies of sliding control and the operational space formulation. A robust implementation of goal-positioned control on a PUMA 560 illustrates the development.

1. Introduction

In this paper, we propose and demonstrate a robust implementation of goal-positioned control, using sliding surfaces in operational space. The notion of sliding surface (Filippov, 1960) has been investigated mostly in the Soviet literature (see Utkin, 1977, for a review), where it has been used to stabilize a class of nonlinear systems. Although it theoretically features excellent robustness properties in the face of *parametric* uncertainty, classical sliding mode control presents several important drawbacks that severely limit its practical applicability. In particular, it involves large control *chattering*, as discussed in detail in (Slotine and Sastry, 1983).

Chattering is in general highly undesirable in practice (with a few exceptions, such as the control of electric motors using pulse width modulation), since it implies extremely high control activity, and further may excite high-frequency dynamics neglected in the course of modelling (such as resonant structural modes, neglected actuator time-delays, or sampling effects). These problems are remedied in (Slotine and Sastry, 1983) and (Slotine, 1984) by replacing chattering control by a smooth control interpolation in a *boundary layer* neighboring a *time-varying* sliding surface. (Slotine, 1984) shows how to monitor the boundary layer width so as not to excite high-frequency unmodeled dynamics, and quantifies the corresponding trade-off between modelling effort and controller tracking performance.

Research in dynamics of robot mechanisms has largely focused on developing the equations of joint motions. These joint space dynamic models have been the basis for various approaches to dynamic control of manipulators. However, task specification for motion and contact forces, dynamics, and force sensing feedback, are closely linked to the end-effector. The description, analysis and control of manipulator systems with respect to the dynamic characteristics of their end-effectors has been the basic motivation in the research and development of *the operational space formulation*. The end-effector equations of motion (Khatib, 1980) is a fundamental tool for the analysis and control of manipulator systems.

In conjunction with the artificial potential field concept (Khatib, 1978), the operational space formulation has been used in the development of a real-time obstacle avoidance approach (Khatib, 1986) involving goal-positioned behavior. The applications of the potential field concept to robot collision avoidance have also been investigated in (Kuntze and Schill, 1982; Hogan, 1983; Krogh, 1984; Espiau and Boulic, 1985).

In this paper, we show that the desired goal-positioned behavior can be naturally achieved by properly defining a target sliding surface in operational space, and then applying the systematic methodologies of sliding control and the operational space formulation. Section 2 summarizes the framework of operational space dynamics and control. Section 3 describes the application of sliding control in order to achieve the target behavior in a fashion robust to uncertainties in the system dynamics. In Section 4, the development is demonstrated experimentally on a PUMA 560. Concluding remarks and suggested extensions are offered in Section 5.

2. The Operational Space Formulation

An *operational coordinate system* is a set $\mathbf{x}$ of *m independent* parameters describing the manipulator end-effector position and orientation in a frame of reference $\mathcal{R}_0$. For a non-redundant manipulator, these parameters form a set of configuration parameters in a domain $\mathcal{D}$ of the operational space and constitute, therefore, a system of generalized coordinates. The end-effector equations of motion in operational space can be written (Khatib, 1980; Khatib, 1983) in the form,

$$\Lambda(\mathbf{x})\ddot{\mathbf{x}} + \mu(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{p}(\mathbf{x}) = \mathbf{F}; \qquad (1)$$

where $\Lambda(\mathbf{x})$ designates the kinetic energy matrix, and $\mu(\mathbf{x}, \dot{\mathbf{x}})$ represents the vector of end-effector centrifugal and Coriolis forces. $\mathbf{p}(\mathbf{x})$ and $\mathbf{F}$ are respectively the gravity and the generalized operational force vectors.

*Nonlinear Systems Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology.

**Artificial Intelligence Laboratory, Computer Science Department, Stanford University.

†Advanced Technology Center for Computer Science, Boeing Computer Services.

The components vector $\mu_i(\mathbf{x}, \dot{\mathbf{x}})$ of the vector $\mu(\mathbf{x}, \dot{\mathbf{x}})$ are given (Khatib, 1987) by

$$\mu_i(\mathbf{x}, \dot{\mathbf{x}}) = \dot{\mathbf{x}}^T \Pi_i(\mathbf{x}) \dot{\mathbf{x}}; \qquad (i = 1, \ldots, m); \tag{2}$$

where the elements of the $m \times m$ matrices $\Pi_i(\mathbf{x})$ are the Christoffel symbols $\pi_{i,jk}$ given as a function of the partial derivatives of $\Lambda(\mathbf{x})$ with respect to the generalized coordinates $\mathbf{x}$,

$$\pi_{i,jk} = \frac{1}{2}\left(\frac{\partial \lambda_{ij}}{\partial x_k} + \frac{\partial \lambda_{ik}}{\partial x_j} - \frac{\partial \lambda_{jk}}{\partial x_i} \right). \tag{3}$$

Using equation (2), the vector $\mu(\mathbf{x}, \dot{\mathbf{x}})$ can be written as

$$\mu(\mathbf{x}, \dot{\mathbf{x}}) = \Pi(\mathbf{x}) \mathbf{v}(\dot{\mathbf{x}}); \tag{4}$$

where $\Pi(\mathbf{x})$ is the $m \times m(m+1)/2$ matrix given by

$$\Pi(\mathbf{x}) = \begin{pmatrix} \pi_{1,11} & 2\pi_{1,12} & \cdots & 2\pi_{1,1m} & \pi_{1,22} & 2\pi_{1,23} & \cdots & 2\pi_{1,2m} & \cdots & \pi_{1,mm} \\ \pi_{2,11} & 2\pi_{2,12} & \cdots & 2\pi_{2,1m} & \pi_{2,22} & 2\pi_{2,23} & \cdots & 2\pi_{2,2m} & \cdots & \pi_{2,mm} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_{m,11} & 2\pi_{m,12} & \cdots & 2\pi_{m,1m} & \pi_{m,22} & 2\pi_{m,23} & \cdots & 2\pi_{m,2m} & \cdots & \pi_{m,mm} \end{pmatrix}, \tag{5}$$

and

$$\mathbf{v}(\dot{\mathbf{x}}) = \begin{bmatrix} \dot{x}_1^2 & \dot{x}_1\dot{x}_2 & \dot{x}_1\dot{x}_3 & \cdots & \dot{x}_1\dot{x}_m \\ \dot{x}_2^2 & \dot{x}_2\dot{x}_3 & \cdots & \dot{x}_2\dot{x}_m & \cdots & \dot{x}_m^2 \end{bmatrix}^T. \tag{6}$$

The equations of motion (1) establish the relationships between positions, velocities, and accelerations of the end-effector and the generalized operational forces acting on it. The control of manipulators in operational space is based on the selection of $\mathbf{F}$ as a command vector. These generalized operational forces are generated by joint-based actuators. The generalized joint force vector Γ corresponding to F and consistent with the end-effector and manipulator dynamic equations is given (Khatib, 1987) by

$$\Gamma = J^T(\mathbf{q})\mathbf{F}; \tag{7}$$

where $\mathbf{q}$ is the vector of joint coordinates and $J^T(\mathbf{q})$ is the Jacobian matrix. The dynamic decoupling and motion control of the manipulator in operational space is achieved by selecting the control structure

$$\mathbf{F} = \hat{\Lambda}(\mathbf{x})\mathbf{u} + \hat{\Pi}(\mathbf{x})\mathbf{v}(\dot{\mathbf{x}}) + \hat{\mathbf{p}}(\mathbf{x}); \tag{8}$$

where, $\hat{\Lambda}(\mathbf{x})$, $\hat{\Pi}(\mathbf{x})$, and $\hat{\mathbf{p}}(\mathbf{x})$ represent the estimates of $\Lambda(\mathbf{x})$, $\Pi(\mathbf{x})$, and $\mathbf{p}(\mathbf{x})$. The system (1) under the command (8) can be represented by

$$I_m \ddot{\mathbf{x}} = G(\mathbf{x})\mathbf{u} + \epsilon(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{d}(t); \tag{9}$$

where I_m is the $m \times m$ identity matrix, and

$$G(\mathbf{x}) = \Lambda^{-1}(\mathbf{x})\hat{\Lambda}(\mathbf{x});$$
$$\epsilon(\mathbf{x}, \dot{\mathbf{x}}) = \Lambda^{-1}(\mathbf{x})\left[\tilde{\Pi}(\mathbf{x})\mathbf{v}(\dot{\mathbf{x}}) + \tilde{\mathbf{p}}(\mathbf{x})\right]. \tag{10}$$

where

$$\tilde{\Pi}(\mathbf{x}) = \hat{\Pi}(\mathbf{x}) - \Pi(\mathbf{x});$$
$$\tilde{\mathbf{p}}(\mathbf{x}) = \hat{\mathbf{p}}(\mathbf{x}) - \mathbf{p}(\mathbf{x}). \tag{11}$$

With a perfect nonlinear dynamic decoupling, the end-effector becomes equivalent to a *single unit mass*, I_m, moving in the m-dimensional space. At the level of the decoupled end-effector, various control structures can be selected for $\mathbf{u}$. For tasks where the desired motion of the end-effector is specified, a linear dynamic behavior can be obtained by selecting

$$\mathbf{u} = I_m \ddot{\mathbf{x}}_d - k_v(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - k_p(\mathbf{x} - \mathbf{x}_d); \tag{12}$$

where $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$ and $\ddot{\mathbf{x}}_d$ are the desired position, velocity and acceleration, respectively, of the end-effector. k_p and k_v are the position and velocity gains.

For tasks that involve large motion to a goal position, where a particular trajectory is not required, a PD command vector of the form

$$\mathbf{u} = -k_v\dot{\mathbf{x}} - k_p(\mathbf{x} - \mathbf{x}_d); \tag{13}$$

will result in a poor coordination of the end-effector motions along its degrees of freedom. This is primarily due to actuator saturation, bandwidth, and velocity limitation. A coordination allowing a straight line motion of the end-effector with an upper speed limit has been shown (Khatib, 1986) to be a desirable behavior for this type of tasks.

Equation (13) can be interpreted as a pure velocity servo-control with a velocity gain k_v, and a desired velocity vector $(k_p/k_v)(\mathbf{x}_d - \mathbf{x})$. Let $V_{\max}$ designate the assigned speed limit. The limitation of the end-effector velocity magnitude can then be obtained by selecting $\mathbf{u}$ as

$$\mathbf{u} = -k_v(\dot{\mathbf{x}} - \nu\dot{\mathbf{x}}_d); \tag{14}$$

where

$$\nu = \text{sat}\left(\frac{V_{\max}}{\lambda|\tilde{\mathbf{x}}|} \right);$$
$$\dot{\mathbf{x}}_d = -\lambda\tilde{\mathbf{x}}; \tag{15}$$

with

$$\lambda = \frac{k_p}{k_v};$$

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d. \tag{16}$$

This allows a straight line motion of the end-effector at a given speed V_{max}. The velocity vector $\dot{\mathbf{x}}$ is in effect controlled to be pointed toward the goal position while its magnitude is limited to V_{max}. The end-effector will then travel at V_{max}, in a straight line, except during the acceleration and deceleration segments. This type of command vector is particularly useful when used in conjunction with the gradient of an artificial potential field (Khatib, 1986) for collision avoidance.

3. Robust Control for Goal-Positioned Tasks

In equation (9), the functions G, ϵ, and $\mathbf{d}$ are not precisely known, but the extent of their imprecision can be upper bounded, as detailed later. The desired goal-positioned behavior can be written as $\mathbf{s} = 0$ where

$$\mathbf{s} = \dot{\mathbf{x}} + \lambda \nu \tilde{\mathbf{x}}. \tag{17}$$

This represents the equation of an $m \times m$-dimensional surface in the state-space. Further, $\dot{\mathbf{s}}$ contains the control input $\mathbf{u}$; therefore, from sliding control theory, the target dynamics can be achieved exactly, in the presence of the bounded modelling uncertainty, by using a control law of the form

$$\mathbf{u} = \frac{1}{g_0}[\hat{\mathbf{u}} + \mathbf{u}_r]; \tag{18}$$

where g_0 is the estimate of the control gain. The vector $\hat{\mathbf{u}}$ is the control input that would maintain $\dot{\mathbf{s}} \equiv 0$ if the dynamics were exactly known,

$$\hat{\mathbf{u}} = -\lambda(\nu \dot{\mathbf{x}} + \dot{\nu} \tilde{\mathbf{x}}). \tag{19}$$

The vector $\mathbf{u}_r$ of components

$$u_{r_i} = -k_i \mathrm{sgn}(s_i); \tag{20}$$

where the k_i are positive constants, guarantees that $\mathbf{s} = 0$ remains attractive in the presence of model uncertainty, and more specifically that (sliding condition)

$$\frac{1}{2}\frac{d}{dt}(s_i^2) \leq -\eta_i |s_i|; \tag{21}$$

where the η_i are positive constants. To this effect, the vector $\mathbf{k}$ of components k_i should be such that

$$k_i = (\beta_i - 1)|\hat{u}_i| + \beta_i(|\epsilon_i| + \eta_i); \tag{22}$$

where, for each degree of freedom, β_i and ϵ_i are, respectively, the gain margin and an upper bound on the dynamic uncertainty. Given the structure (9) of the system dynamics and the definitions (10) and (11), adequate choices of ϵ_i, β_i, and g_i are

$$\epsilon_i = \alpha(\dot{\mathbf{x}}) = \alpha_0 + \alpha_1 |\nu(\dot{\mathbf{x}})|; \tag{23}$$

where the constants α_0 and α_1 are selected such that

$$\alpha_0 \geq \|\Lambda^{-1}(\mathbf{x})\| \cdot |\tilde{\mathbf{p}}(\mathbf{x})|, \qquad \forall \mathbf{x} \in \mathscr{D};$$
$$\alpha_1 \geq \|\Lambda^{-1}(\mathbf{x})\| \cdot \|\tilde{\Pi}(\mathbf{x})\|, \qquad \forall \underset{i}{\mathbf{x}} \in \mathscr{D}; \tag{24}$$

and

$$\beta_i = \beta_0 = \sqrt{\sigma_{min}/\sigma_{max}};$$
$$g_i = g_0 = \sqrt{\sigma_{min} \cdot \sigma_{max}}; \tag{25}$$

where the constants σ_{min} and σ_{max} are selected as

$$\sigma_{min} \leq \|G(\mathbf{x})\|, \qquad \forall \mathbf{x} \in \mathscr{D};$$
$$\sigma_{max} \geq \|G(\mathbf{x})\|, \qquad \forall \mathbf{x} \in \mathscr{D}. \tag{26}$$

While control law (18) achieves the target dynamics (17) exactly, the presence of the switching terms $k_i \mathrm{sgn}(s_i)$ implies that in practice undesirable control chattering will occur. To suppress the chattering and obtain a bandwidth-limited controller that best approximates the exact behavior described above, the switching action $k_i \mathrm{sgn}(s_i)$ is replaced by a smooth interpolation in a boundary layer neighbouring the sliding surface:

$$u_{r_i} = -\bar{k}_i \mathrm{sat}\left(\frac{s_i}{\phi_i}\right); \tag{27}$$

where, following (Slotine, 1984), the ϕ_i are defined by

$$\dot{\phi}_i + \lambda \phi_i / \beta_0^2 = k_{d_i}/\beta_0 \quad \text{if } \lambda \phi_i \geq \beta_0 k_{d_i};$$
$$\dot{\phi}_i + \lambda \phi_i = \beta_0 k_{d_i} \quad \text{if } \lambda \phi_i \leq \beta_0 k_{d_i}; \tag{28}$$

and the vector $\mathbf{k}$ is

$$\bar{\mathbf{k}} = \mathbf{k} - \mathbf{k}_d + \frac{\lambda}{\beta_0}\phi \tag{29}$$

with initially

$$\phi(t = 0) = \frac{\beta_0}{\lambda}\mathbf{k}_d(t = 0). \tag{30}$$

The vector $\mathbf{k}_d$ is equal to the vector $\mathbf{k}$ computed along the desired trajectory; since the desired behavior consists here of a specified relationship between position and veloc-

ity, $\mathbf{k}_d$ can be computed by replacing $\dot{\mathbf{x}}$ by $\lambda\nu\tilde{\mathbf{x}}$ in the expression (22) of $\mathbf{k}$. The components of $\mathbf{k}_d$ are

$$k_{d_i} = (\beta_0 - 1)|\hat{u}_i| + \beta_0(|\epsilon_{d_i}| + \eta_i); \tag{31}$$

where

$$\epsilon_{d_i} = \alpha(\tilde{\mathbf{x}}) = \alpha_0 + \alpha_1|\nu(\tilde{\mathbf{x}})|. \tag{32}$$

With these choices of ϕ and $\mathbf{u}$, the parameter λ, already used in the definition (17) of the sliding surface, can then be interpreted as the desired control bandwidth.

Additional considerations have to be taken into account in this application. With the choice (15) of ν, maintaining $\mathbf{s} = 0$ would require that the control input have discontinuities at $(\lambda|\tilde{\mathbf{x}}| = V_{\max})$. These can be eliminated by using a second order digital filter with an adequate cutoff frequency ω_c. Further, convergence has to be monitored outside the boundary layer, where the system lies initially, as well as when the algorithm is used in the context of obstacle avoidance. To this effect, a vector $\mathbf{u}_L$, with components

$$u_{L_i} = -k_{L_i}\mathrm{sgn}(s_i); \tag{33}$$

is added to the control law $\mathbf{u}$ outside the boundary layer. This guarantees that the boundary layer is reached within a finite time smaller than $g_0|s_i(t=0)|/(\beta_0 k_{L_i})$.

Table 1.
Control Parameters.

$V_{\max}$	15.0 in./sec
λ	35.0
α_0	100.0
α_1	2.0
β_0	1.2
g_0	1.0
ω_c	20.0 rad/sec
k_L	300.0
α_L	2.9

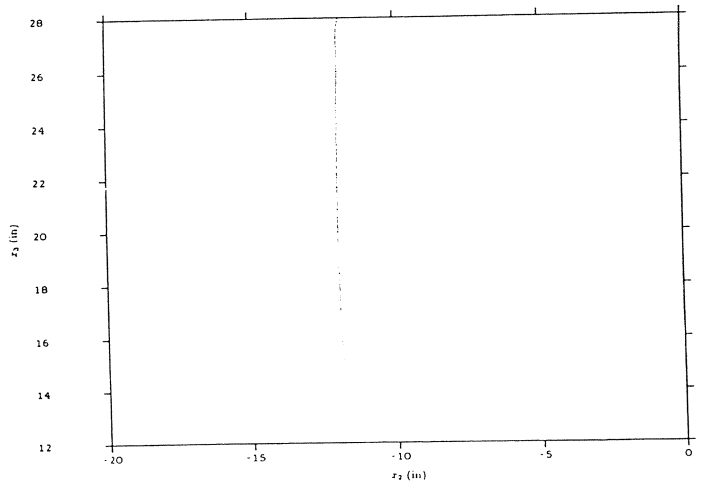


Figure 1. Sliding control–plan view of the trajectory

In practice, to avoid control discontinuities when the system enters the boundary layer, it is desirable to replace $\mathbf{u}_L$ by a function whose derivative is continuous. A computationally efficient choice is

$$u_{L_i} = \begin{cases} -(s_i - \phi_i)k_{L_i}/[\alpha_L + (s_i - \phi_i)] & \text{if } s_i \geq \phi_i; \\ -(s_i + \phi_i)k_{L_i}/[\alpha_L - (s_i + \phi_i)] & \text{if } s_i \leq -\phi_i. \end{cases} \tag{34}$$

Finally, the control vector $\mathbf{u}$ becomes:

$$\mathbf{u} = \frac{1}{g_0}[\hat{\mathbf{u}} + \mathbf{u}_r + \mathbf{u}_L]; \tag{35}$$

with $\hat{\mathbf{u}}$, $\mathbf{u}_r$, and $\mathbf{u}_L$ as given in (19), (27), and (34).

4. Experimental Results

We now illustrate the development with a robust implementation of goal-positioned control on a PUMA 560

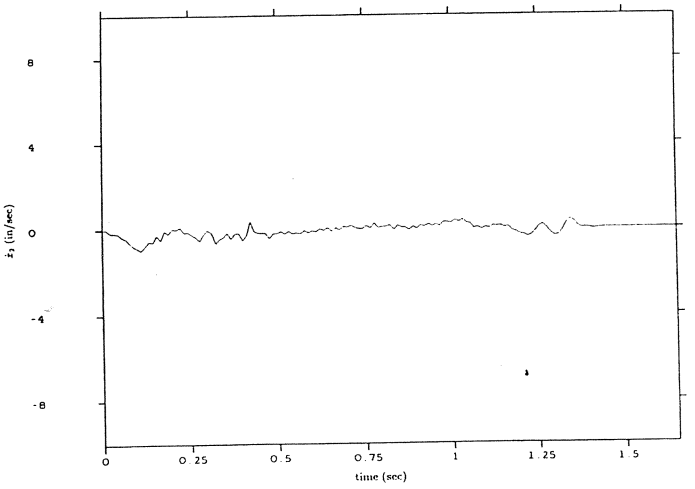


Figure 2. Sliding control–horizontal component of velocity

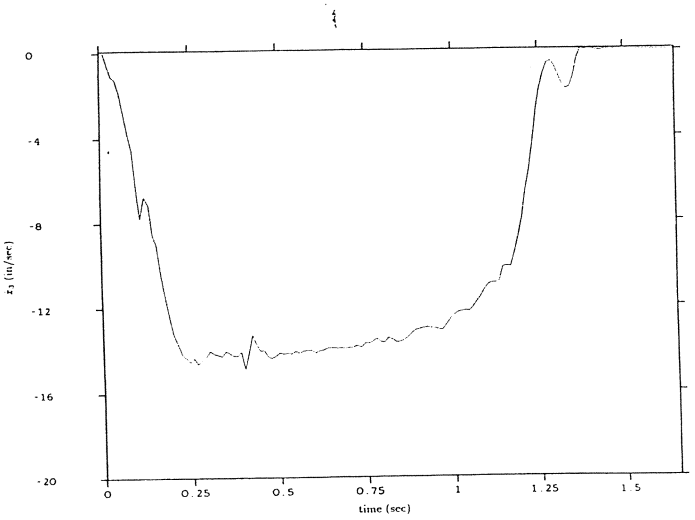


Figure 3. Sliding control–vertical component of velocity

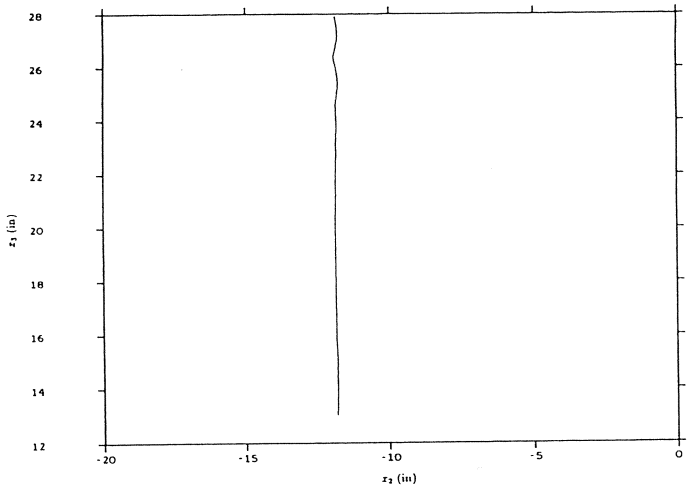


Figure 4. PD control–plan view of the trajectory

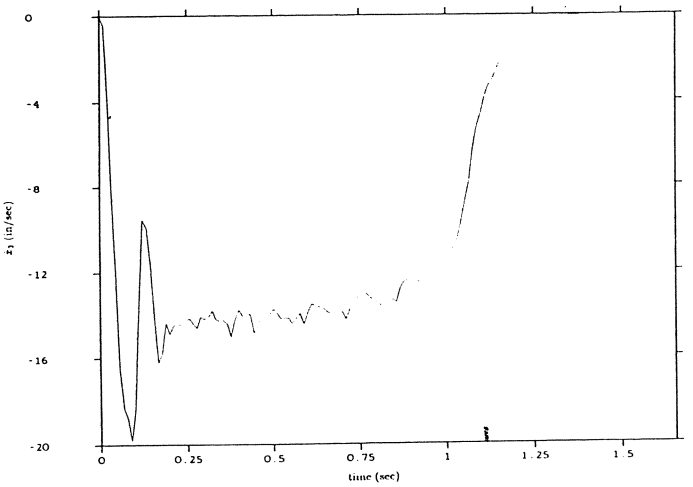


Figure 6. PD control–vertical component of velocity

robot. The above approach has been implemented in ESPRIT/DAPROC, an experimental robotic programming, development and control system developed at Boeing. ESPRIT, residing on a VAX 11/780, provides the high level programming and development environment while DAPROC, the low-level real-time controller, resides on a PDP 11-44. The combined system controls a PUMA 560 robot. At the present time the implementation is limited to a planner motion (position and orientation) using three degree-of-freedom (joints 2, 3 and 5). This is due to the PDP processing power limitations.

A two-level control architecture (Khatib, 1985) is used in the implementation of the operational space control system. These are a low rate parameter evaluation level and a high rate servo-control level. The experimental results were obtained with the parameter evaluation and servo-control levels running at 50 and 100 Hz, respectively. The motion selected in these experiments consisted of a straight line motion along the x_2-axis at a maximum velocity of 15.0 in./sec. The numerical values of the control parameters are given in Table 1.

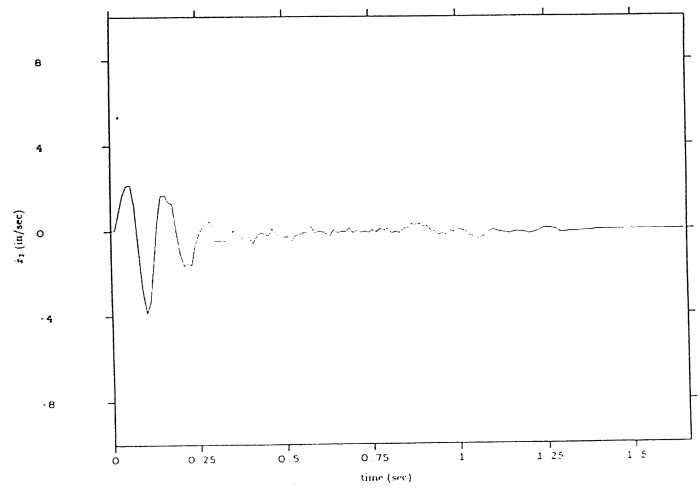


Figure 5. PD control–horizontal component of velocity

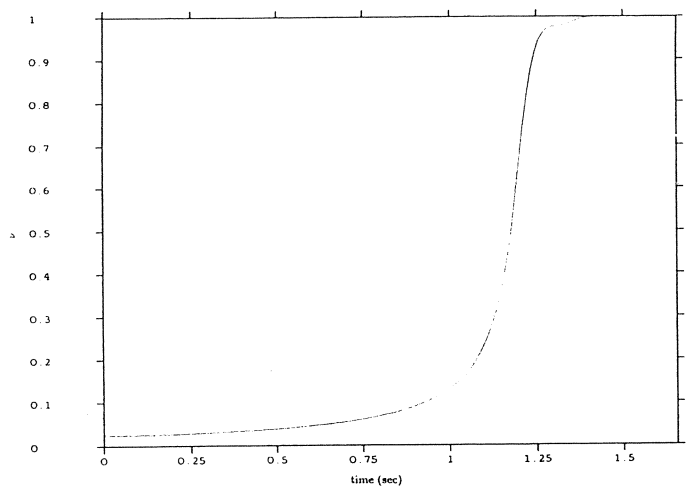


Figure 7. Sliding control–vertical saturation scaling function ν

The sliding surface s_2 and the boundary layer ϕ_2 are shown in Figures 1 and 2 while the three components of the overall control ($\hat{u}_2$, u_{r_2}, and u_{L_2}) are shown in Figures 3, 4, and 5, respectively. The resulting control u_2 is shown in Figure 6. Initially the system lies outside the boundary layer ($s_2 > \phi_2$) resulting in a nonzero u_{L_2} which brings the system within the boundary layer in 0.055 second. The peak in $\hat{u}_2$ reflects the maximum rate of change of ν (Figures 7, and 8) which occurs during the transition from operation with a saturated velocity command to operation with an unsaturated command. The trajectory of x_2 is presented in Figure 9.

5. Concluding Remarks

A robust implementation of goal-positioned control using sliding surfaces in operational space has been presented. In this implementation, a nonlinear decoupling control structure, based on the end-effector dynamic model, is used for the dynamic decoupling of end-effector motion in operational space, while a sliding control structure is used at the level of decoupled end-effector to achieve a

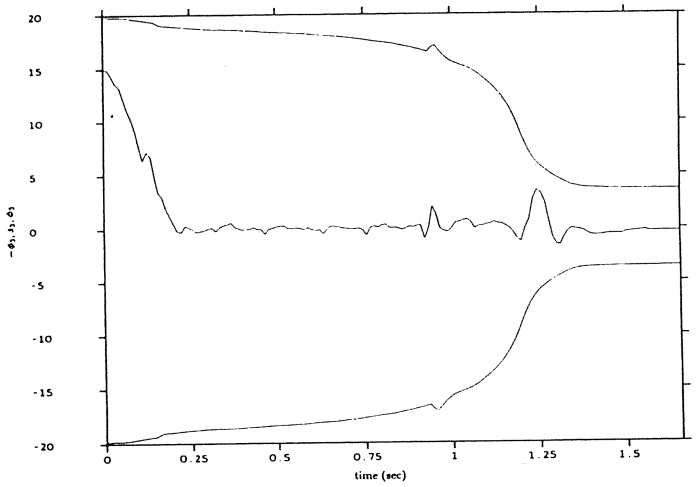


Figure 8. Sliding control–sliding surface s and boundary layer $\pm\phi$

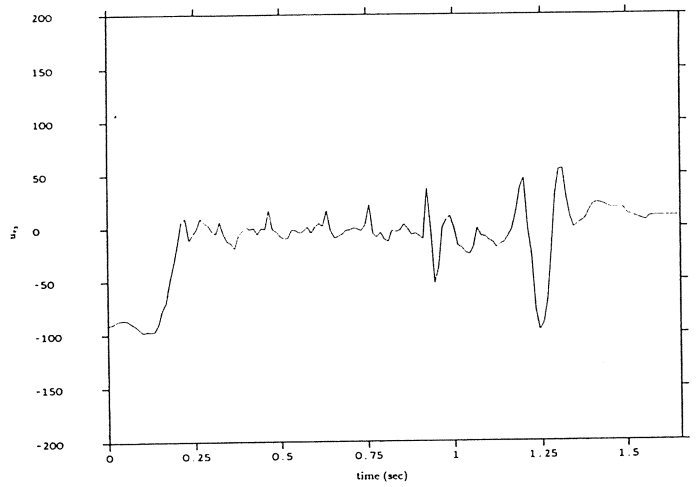


Figure 9. sliding control–robustness component of u_3

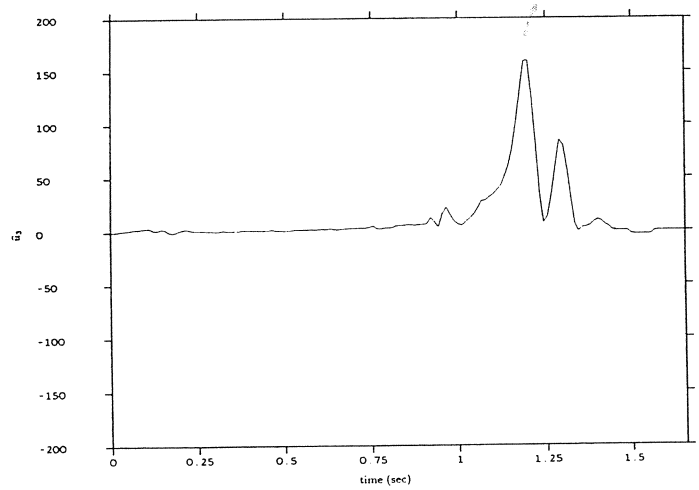


Figure 10. Sliding control–sliding control component of u_3

robust goal-positioned dynamic behavior. The initial experimental results have shown a significant increase of control bandwidth and improvement of the overall dynamic performance of the system. The integration of this implementation in the artificial potential field approach for real-time collision avoidance is being considered.

References

Espiau, B., and Boulic, R. Collision Avoidance for Redundant Robots with Proximity Sensors, Preprints 3rd ISRR, Gouvieux, France, 1985, pp. 95–102.

Filippov, A.F. *Am. Math. Soc. Trans.*, Vol. 62, 1960, p. 199.

Hogan, N. Impedance Control of a Robotic Manipulator, ASME Winter Annual Meeting, Washington, D.C., 1981.

Khatib, O. and Le Maitre, J.F. Dynamic Control of Manipulators Operating in a Complex Environment, *Proc. 3rd CISM-IFToMM Symp.*, Udine, Italy, 1978, pp. 267–282.

Khatib, O. Commande Dynamique dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Obstacles, Thèse de Docteur-Ingénieur, ENSAE, Toulouse, 1980.

Khatib, O. Dynamic Control of Manipulators in Operational Space, *Proc. 6th CISM-IFToMM*, New Delhi, India, 1983, pp. 1128–1131, Wiley, New York.

Khatib, O. The Operational Space Formulation in Robot Manipulator Control, *Proc. 15th ISIR*, Tokyo, 1985, pp. 165–172.

Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *Int. J. Robotic Research*, Vol. 5, No. 1, 1986, pp. 90–98.

Khatib, O. A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE J. Robotics and Automation*, Vol. 3, No. 1, 1987.

Krogh, B. A Generalized Potential Field Approach to Obstacle Avoidance Control, *Proc. SME Conf.*, Bethlehem, Pennsylvania, 1984.

Kuntze, H.B., and Schill, W. Methods for Collision Avoidance in Computer Controlled Industrial Robots, *12th ISIR*, Paris, 1982.

Slotine, J.J.E., and Sastry, S.S. Tracking Control of Nonlinear Systems Using Sliding Surfaces, with Application to Robot Manipulators, *Int. J. Control*, Vol. 38, 1983, p. 465.

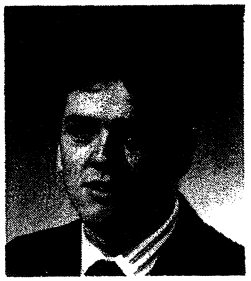
Slotine, J.J.E. Sliding Controller Design for Nonlinear Systems, *Int. J. Control*, Vol. 40, 1984, p. 421.

Slotine, J.J.E., The Robust Control of Robot Manipulators, *Int. J. Robotics Research*, Vol. 4, No. 2, 1985.

Utkin, V.I. Variable Structure Systems with Sliding Mode: A Survey, *IEEE Trans. Autom. Control*, Vol. 22, 1977, p. 212.

Jean-Jacques E. Slotine received his Ph.D. from the Massachusetts Institute of Technology in 1983. He is currently Assistant Professor of Mechanical Engineering, Doherty Professor in Ocean Utilization, and Director of the Nonlinear Systems Laboratory at M.I.T. His research interests include applied nonlinear control, robotics, and learning systems. He is a Technical Editor of the I.E.E.E. *Journal of Robotics and Automation*, an Associate Editor of *The Robotics Review*, and a member of the Editorial Board of the *International Journal of Robotics and Automation*. Dr. Slotine is the coauthor of the textbook "Robot Analysis and Control" (Wiley 1986), and is a frequent consultant to industry and to Woods Hole Oceanographic Institution.

O. Khatib received the degree of "Docteur-Ingénieur en Automatique et Système" in 1980 from "l'École Nationale Supérieure de l'Aéronautique et de l'Espace," Toulouse, France. He is a Senior Research Associate at the Artificial Intelligence Laboratory, Stanford University. He has worked in the field of Robotics since 1976, when he joined "le Centre d'Étude et de Recherche de Toulouse." There, he developed the foundation of the operational space fomulation and the artificial potential field concept for robot manipulator control and collision avoidance. Since 1981, his research at Stanford University was aimed at extending these methodologies to establish a unified framework for motion and active force control of robot manipulators operating in a cluttered and evolving environment. His research interests include issues of task description, constrained motion and force control, force stategies, redundancy, kinematic singularities, real-time obstacle avoidance, sensor fusion and high level programming systems. He is also concerned with the analysis and design of high-performance force-controlled robot manipulator and micro-manipulator systems.



D.E. Ruth received the B.S. degree in Mechanical Engineering from Cornell University, Ithaca, New York, in 1980 and the M.S. degree in Aeronautics and Astronautics from the Stanford University, Palo Alto, California in 1981. He is currently a Research Engineer at the Robotics Laboratory, SRI International, Menlo Park, California where his research interests include hybrid position/force control of manipulators, multi-arm cooperation, sensor based manipulation, and automation of composite parts layup.

From 1981 to 1984 he worked for the Guidance and Navigation Technology group at Boeing Aerospace Company, Seattle, WA, where he worked on several projects involving guidance and control systems for tactical missiles. From 1984 until 1987 he was with the Artificial Intelligence Center at Boeing Computer Services. His interests there included hybrid position/force control and high-level language programming of manipulators. He developed a robotics testbed used to study issues relating to the integration of sensing and control.

Mr. Ruth is a member of Tau Beta Pi, the Institute of Electrical and Electronics Engineers, and the American Institute of Aeronautics and Astronautics.