*Reprinted from*

# ROBOT CONTROL 1988 (SYROCO '88)

*Selected Papers from the 2nd IFAC Symposium,*
*Karlsruhe, FRG, 5–7 October 1988*

Edited by
U. REMBOLD
*Institut für Prozeßrechentechnik und Robotik,*
*Universität Karlsruhe, FRG*

Published for the
INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL
by
PERGAMON PRESS
OXFORD · NEW YORK · BEIJING · FRANKFURT
SÃO PAULO · SYDNEY · TOKYO · TORONTO

# A TASK SPACE DECOUPLING APPROACH TO HYBRID CONTROL OF MANIPULATORS

Alessandro De Luca, Costanzo Manes, Fernando Nicolò

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
Via Eudossiana 18, 00184, Roma, Italy

**Abstract.** A scheme for dynamic hybrid control of robot manipulators is presented. The design is directly achieved in task space coordinates. In this way, the inherent orthogonality between force and velocity description of tasks is preserved and overspecification is avoided in the control synthesis. A nonlinear decoupling and linearizing feedback law is obtained which yields invariant control performances over time-varying tasks. The effect of a robustifying integral action is discussed. Simulation results are reported for a two-link arm.

**Keywords.** Robots, Force Control, Velocity control, Nonlinear control systems, Decoupling, Task-space.

## Introduction

In most robotic applications, it is important to deal with contacts between the robot end-effector and the environment. In order to keep the arising contact forces at a desired value, either an accurate position scheme is used or force sensor capability is added to the robot. The first scheme requires exact knowledge of an highly structured environment. The second allows more adaptability to uncertainties.

Usually, the robotic task is divided into two phases: a gross motion, in which only position control is applied, and a fine quasi-static motion, where the major concern is force control. However, certain tasks may ask for an accurate hybrid control both in velocity and force along a dynamic trajectory[1].

Several approaches have been proposed for the hybrid control of manipulators. Raibert and Craig (1981) introduced a scheme which allows to control either the velocity or the force for each direction axis, following the task description due to Mason (1981). The basic idea of the control scheme is to ignore position errors along the force-controlled axes and viceversa. For each axis, a conventional PD/PID force or position controller is developed independently on the basis of the relative error. This error is obtained through the use of kinematic transformations and selection matrices. Interaction effects due to dynamic couplings give rise to disturbances in the feedback loop so that this scheme is useful only at slow speed. Khatib (1987) improved this approach by compensating the full dynamic interactions, which are described directly in the cartesian space. Nonlinear control laws are used in order to decouple and to make linear the behavior of the end-effector along each axis of the associated frame. His implementation of the control law preserves the task compatibility description but destroys the dynamic orthogonal properties between force and velocity. Moreover, an overspecification of the number of controllers leads to a coupled design. Yoshikawa (1987a, 1987b) takes into account the geometric curvature of the object surface to be followed in order to improve the positional part of the hybrid controller. The elasticity of the contact is not included in the control design and this is again a source of disturbance in the closed-loop system.

[1] In the following, by velocity it is intended either a linear or an angular one, while force will denote either a linear force or a torque.

The idea of using dynamic models of the contact for control purposes has been explored by Hogan (1985). In his scheme, the robot end-effector can be given a desired compliant behavior depending on the environment stiffness and the motion control specifications. This also gives rise to interesting stability issues, as shown by An and Hollerbach (1987) and Eppinger and Seering (1987).

In this paper, an hybrid control scheme is proposed *directly in the task space*. The task frame associated with the geometry of the environment at the contact point is generally time-varying. With the chosen strategy, an effective decoupling and linearization is obtained with invariant performances over the whole task definition. *Actual* contact forces are included in the derivation of closed-loop control strategies. A dynamic model of the interaction between end-effector and environment is explicitly considered in the design of the force control loop. End-effector accelerations that guarantee a desired behavior of contact forces and velocities are then computed via an inverse dynamics approach at the task level.

## Dynamic model of robot in contact

As usual, the dynamic model of a rigid robot arm can be written in the joint space as

$$B(q)\,\ddot{q} + n(q,\dot{q}) = \tau + J^T(q)\,F_c^r(q,\dot{q})$$

where $q$ are the generalized coordinates of the system, $B$ is the positive definite inertia matrix, $n$ contains centrifugal, Coriolis, gravitational forces and viscous terms, $\tau$ are the torques supplied by the actuators, $J$ is the Jacobian of the robot direct kinematics, $r = kin(q)$, and $F_c^r$ are the generalized forces acting *from* the environment to the robot end-effector and performing work on $r$. In general, $F_c^r$ is a function of the robot state $(q,\dot{q})$ depending on the contact geometry, on the sensor stiffness and on the friction and compliant properties of the environment, and which is zero for motion in free space. Its functional form may be a very complex one; here, it is not assumed to be known since a sensor will be available for the measure of the contact forces.

It is useful to introduce the dynamic model of the arm in terms of position and orientation of the end effector, as in (Khatib, 1987):

$$B_e(r)\ddot{r} + n_e(r,\dot{r}) = F^r + F_c^r(r,\dot{r})$$

$r$ is expressed in an inertial frame $^0S$ associated with the base

of the robot and can be taken as a set of new generalized coordinates in the non-redundant and nonsingular case. Moreover, $\tau = J^T(r) F^r$. Note that a minimal representation of the end-effector orientation should be used, e.g. any set of Euler angles. In the right hand side of the above equation, the generalized forces have to be defined accordingly.

When the problem involves changes in orientation and is not restricted to a plane, some caution is needed in the transformation of rotational quantities. Usually, the rotational part of a task is described in terms of the angular velocities $\omega_x$, $\omega_y$, $\omega_z$, and of the torques around the same axes (Mason, 1981). Denoting by v the vector of linear and angular velocities, the following relation holds

$$v = T(r)\,\dot{r}$$

with T(r) invertible, except for isolated points. Separating the linear from the angular components, this matrix takes on the form

$$T(r) = \begin{bmatrix} I & 0 \\ 0 & T_A(r) \end{bmatrix}$$

Let F be the vector of forces and moments acting on the infinitesimal linear and angular displacements defined along the axes of $^0S$. Then $F^r$ and F are related by

$$F^r = T^T(r)\,F$$

The robot dynamic model can be rewritten as

$$B_e(r)\ddot{r} + n_e(r,\dot{r}) = T^T(r)\,[\,F + F_c(r,\dot{r})\,]$$

where the contact forces $F_c$ appear in their natural system of measure.

As far as the robot interaction with the environment is concerned, the following modeling assumptions are made:

1. the contact between the end-effector and the environment is point-wise and elastic, and this elasticity can be modeled by a single constant coefficient;
2. a few geometric parameters of the enviroment are known.

The inclusion of elasticity at the point of contact avoids some instabilities intrinsic to a perfectly rigid schematization. In the Appendix, it is described how chattering may occurr for an end-effector in rigid contact with the environment. Note that the first assumption models either a rigid environment with a soft sensor, or a soft environment with a rigid sensor, or the case of sensor and environment with similar elasticity constants but when the mass beyond the point of force measure is negligible. The mechanical behavior of the contact is sketched in Figure 1. The second assumption is used in the derivation of the control law and will be clarified later on. Typically, knowledge of the local normal to the environment at the contact point is a sufficient information.

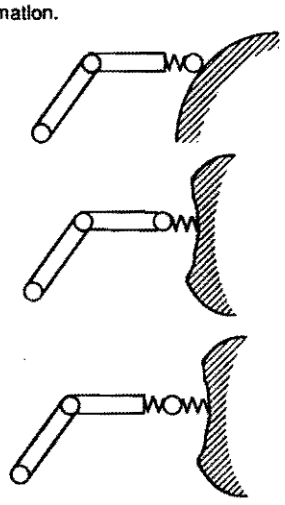


**Figure 1 - Models of elasticity at the contact point**

### Task space description for hybrid control

The control objective is to drive a robot manipulator along a given end-effector trajectory $r_d(t)$, defined on the environment surface, while a contact force is being exerted according to a desired time evolution $F_d(t)$. The hybrid control technique allows to obtain this goal, even in the presence of unavoidable mismatching between the real and the modeled world.

The prescribed motion should be compatible with the constraints imposed by the environment; the concept of C-surfaces (Mason, 1981) is useful for checking this consistency. In the general case, a six-dimensional task frame $^tS$ is associated to the contact point. The orientation of the axes of $^tS$ is time-varying and is such that, at all instants, m scalar forces and 6-m velocities can be specified, one along each of the axes.

The position $r_d$ of the desired contact point is always expressed in $^0S$. Instead, the desired force $F_d$ and velocity $v_d$, both given in $^0S$, can be expressed directly in the task frame $^tS$ as $^tF_d$ and $^tv_d$. Linear velocity and force are transformed from the inertial frame $^0S$ into $^tS$ by means of a 3x3 rotation matrix $R_L(r)$; similarly, angular velocity and torque are transformed by $R_A(r)$. A 6x6 matrix R(r) will be defined as

$$R(r) = \begin{bmatrix} R_L(r) & 0 \\ 0 & R_A(r) \end{bmatrix}$$

with $R^{-1}(r) = R^T(r)$.

For hybrid control, a 6x6 diagonal selection matrix $\Sigma$ with 0/1 elements is used to specify the axes of the task frame which have to be controlled in force. Indeed, $\Sigma^\wedge := I - \Sigma$ selects the remaining axes. Usually, hybrid control schemes (Raibert and Craig, 1981, Asada and Slotine, 1986, Khatib and Burdick, 1986, Whitney, 1987) define force, position and velocity errors used in the controller as

$$F_E = \Omega\,(F_d - F_c) \qquad r_E = \Omega^\wedge\,(r_d - r) \qquad v_E = \Omega^\wedge\,(v_d - v)$$

where

$$\Omega = R^T(r)\,\Sigma\,R(r) \qquad \Omega^\wedge = R^T(r)\,\Sigma^\wedge\,R(r)$$

are matrices which map vectors from $^0S$ in $^tS$, filter out some components and transform the resulting vector back into $^0S$. Using these error signals, various controllers can be designed, with or without compensation of the dynamic terms, either at the cartesian or at the joint level. At least *six* controllers are designed in this way, for *each* of the two main loops on the force error vector $F_E$ and on the position error vector $r_E$ (or velocity error $v_E$). Two criticisms follow naturally:

a. the underlying idea of decoupling between force and position control is lost if the control action is not achieved in the space where these quantities are orthogonal, i.e. in the *task space*;
b. even if the performance of each controller is chosen such as not to vary over time in the cartesian space $^0S$ (e.g. by using constant controller gains), the *weighting* of errors in the task space *varies with the orientation* of the task frame.

As a consequence, integral terms added in the controller could loose their meaning when a significative change of orientation occurs. Consider for instance a planar situation, in which the $^tx$ velocity-controlled axis of $^tS$ is aligned for a certain time interval with the x axis of $^0S$, while the $^ty$ force-controlled axis is aligned with y. Suppose that the end-effector has accumulated a relevant integral error term along the x axis, while the force controller is working properly so that a constantly zero error results on the y axis. At a given instant the surface curvature leads to a change of orientation of the task frame, so that the velocity-controlled axis $^tx$ becomes aligned with the y axis of $^0S$. As a result, the integral of the velocity-error computed on the x axis will start acting along the force-controlled axis of the task frame, while no integral term will correct the error accumulated on the velocity-controlled axis.

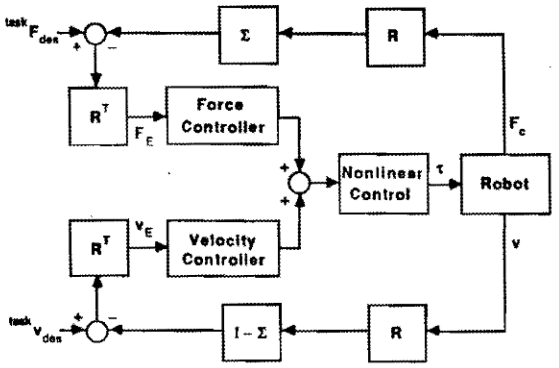
The above considerations clearly suggest that the correct control action should be computed in the task frame. With this choice, *one* single controller can be designed for each axis of $^tS$, with invariant dynamic behavior. In order to develop such a controller, the task description will be formulated in terms of a desired position $r_d(t)$, expressed in $^0S$, and desired velocity $^tv_d(t)$ and force $^tF_d(t)$, both vectors being expressed in $^tS$. Thus, force, position and velocity errors will be defined as:

$$^tF_E = {}^tF_d - \Sigma\, R(r)\, F_c$$

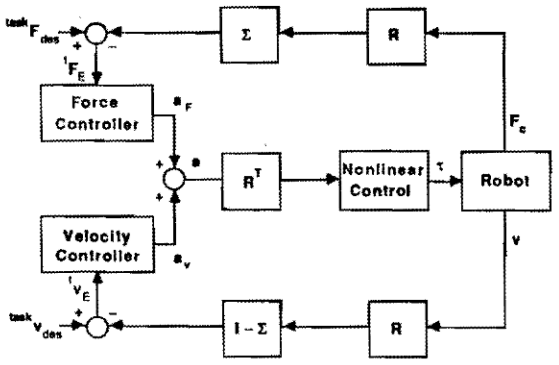
$$^tr_E = \Sigma \wedge R(r)\, T(r)\, (r_d - r)$$

$$^tv_E = {}^tv_d - \Sigma \wedge R(r)\, T(r)\, \dot{r}$$

This approach allows the construction of scalar controllers in a non-redundant way, once a preliminary nonlinear control law has decoupled and linearized the system. Figure 2 shows the block diagram of the proposed task-based hybrid control in comparison with the conventional one (Asada and Slotine, 1986, Khatib, 1987). A similar strategy was used also by West and Asada (1985) but following a purely kinematic approach and thus not obtaining the dynamic decoupling properties.




(a) Khatib and others




(b) Task-based decoupling

Figure 2 - Comparison between hybrid control schemes

Note also that there are two ways of computing the rotation matrix R which is used for the transformation between $^0S$ and $^tS$. As sketched in Figure 3, this matrix can be evaluated at the actual contact point, as $R = R(r)$, or at the desired one, as $R = R(r_d)$. The choice $R(r)$ is the correct one and is used in the above derivation of the errors. However, this choice is expensive in terms of on-line calculations of the controller; at each sampling time, the matrix R has to be computed using the model of the environment and a measure of the cartesian position r. If $R(r_d)$ is adopted, this calculation is moved to the off-line trajectory planner. Similar comments apply for matrix $T(r)$.

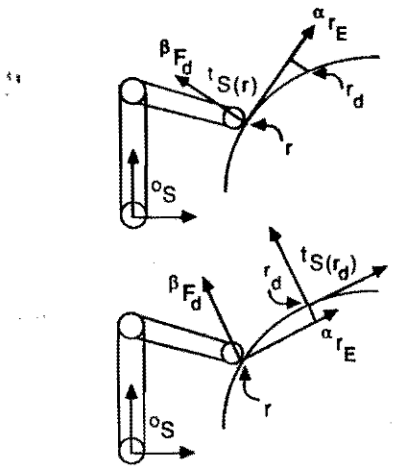



Figure 3 - Comparison between task frame definitions

## Control design in the task space

Nonlinear decoupling

In order to design independent force and velocity controllers, one for each axis of the task space, the nonlinear dynamic interactions arising from robot motion should be compensated first. The dynamic model of the robot in contact with the environment can be solved for the accelerations as

$$\ddot{r} = - B_e^{-1}(r)\, [\, n_e(r,\dot{r}) - T^T(r)\, F_c(r,\dot{r})\, ] + B_e^{-1}(r)\, T^T(r)\, F$$

The feedback force F which gives exact linearization and decoupling can be obtained by applying the standard control results existing for general nonlinear systems (Isidori, 1985). Usually, the system dynamics is expressed in a state-space format with an associated output vector function which defines the variables to be set under control.

For hybrid control of robot velocity and applied force, it would be straightforward to define the output y of interest as the selected components of velocity v and force $F_c$ in the task frame

$$y = \begin{bmatrix} u_{T_i}^T(r)\, v \\ u_{N_j}^T(r)\, F_c(r,\dot{r}) \end{bmatrix}$$

where $u_{Ti}$ are unit vectors relative to the velocity-controlled axes, for i=1,...,6-m, and $u_{Nj}$ are those relative to the force-controlled axes, for j=6-m,...,6[2]. However, this approach has some limitations. In fact, the algorithm that gives the control F which achieves linearization and input-output decoupling in the closed-loop requires to take derivatives of the output function y, until the input F appears explicitly. Thus, with the above choice of outputs, $F_c$ has to be known as *an analytic function* of $(r,\dot{r})$. Uncertainties intrinsic to the modeling of this term result in a poor strategy from the robustness point of view.

Instead, the following output function is taken:

$$y = R(r)\, T(r)\, \dot{r}$$

or $y = {}^tv$, the six velocities expressed in the task frame. Note that, due to the elasticity included at contact point, velocity vectors pointing *into* the object surface are feasible. It will be shown next how this choice allows to control not only velocity but also force. Let a be a desired task acceleration, i.e. we would like to have $^t\dot{v} = a$. Since

$$\dot{y} = R(r)\, T(r)\, \ddot{r} + [\, \dot{R}(r,\dot{r})\, T(r) + R(r)\, \dot{T}(r,\dot{r})\, ]\, \dot{r}$$

[2] After reordering, these vectors are simply the columns of $R^T(r)$. Note that the first or the last three components of $u_{Ti}$ and of $u_{Nj}$ are always zero.

substituting from the dynamic model for the acceleration $\ddot{r}$, using the desired task acceleration, and solving for F gives

$$F = \bar{B}_e(r) R^T(r) \{ a - [\dot{R}T + R\dot{T}](r,\dot{r})\,\dot{r} \} + T^{-T}(r) n_e(r,\dot{r}) - F_c(r,\dot{r})$$

$$= F^*(r,\dot{r},a)$$

where

$$\bar{B}_e(r) = T^{-T}(r)\, B_e(r)\, T^{-1}(r)$$

The resulting input-output relation in the closed-loop system is

$$\dot{y} = a$$

These are six independent integrators; a linear and decoupled behavior is then displayed in the task space. Note that the *measured* $F_c$ is directly used in $F^*$. To implement this control law, some knowledge of the environment geometry should be given to the system. In particular, the gradients of the C-surfaces are needed in R(r), while curvature information is required for R'(r,r'). If not known a priori, the tangent space may also be identified on-line in a efficient way (Manes, 1987). Second order terms becomes relevant for surfaces whose curvature is high with respect to the speed of motion. When this information is not available, disturbances will arise due to kinematic couplings so that further control action is needed. Addition of integral terms in the control may serve to this purpose.

Velocity control loop

Let $\alpha$ denote any of the 6-m velocity-controlled axes in the task frame. To assign a desired dynamic behavior to

$$\ddot{\alpha} = a_\alpha$$

a PI scheme on the velocity error can be used

$$a_\alpha = \ddot{\alpha}_d + kp_v(\dot{\alpha}_d - \dot{\alpha}) + kI_v \int (\dot{\alpha}_d - \dot{\alpha})\, dt$$

where an additional feedforward acceleration term is present. Note that this scheme does not use any measure of the absolute position error. As follows from the previous discussion, using the integral of the task velocity error is more meaningful than resolving in the task space the position error computed in $^0S$ (see Figure 3). The latter may be a sufficient approximation as far as this error is small enough.

Force control loop

Similarly, a scalar loop can be designed for each of the m force-controlled axes in the task space. Let $\beta$ denote any of these axes. The impedance control idea introduced by Hogan (1985) is usefully applied also to force control. The behavior along $\beta$ is chosen so to match with a second-order reference model

$$m_d \ddot{\beta} + d_v \dot{\beta} + k_s(\beta - \beta_s) = -F_{\beta,d}$$

where $-k_s(\beta - \beta_s) = F_{\beta,c}$ is the contact force in the direction $\beta$, $k_s$ is the elasticity coefficient of the contact, $\beta_s$ is the zero reference for the equivalent spring and $F_{\beta,d}$ is the desired force along $\beta$. The positive adopted direction goes from the object surface to the robot. The robot end-effector acceleration along the force-controlled $\beta$ axis is then chosen as

$$\ddot{\beta} = -\frac{1}{m_d}\left[(F_{\beta,d} - F_{\beta,c}) + d_v \dot{\beta}\right]$$

From assumption 2 of the previous section, $k_s$ is specified from the sensor/environment characteristics. Under the hypothesis of linear elasticity, the natural frequency and the damping of the system

$$\omega_n = \sqrt{\frac{k_s}{m_d}} \qquad \zeta = \frac{d_v}{2\sqrt{k_s m_d}}$$

may be selected by modifying $m_d$ and $d_v$. In order to obtain enough damping for different values of the contact stiffness, $d_v$ will be chosen according to the highest $k_s$.

Resulting controller

The various operations to be performed for implementing the task space decoupling controller are summarized here. Starting from measures of position $q$ and velocity $\dot{q}$ at the joints, compute

$$r = kin(q) \qquad \dot{r} = J(q)\,\dot{q} \qquad v = T(r)\,\dot{r}$$

$F_c$ is directly obtained from the force/torque sensor measures. Errors in the task frame are then evaluated as previously shown and used in the scalar force and velocity loops to compute the desired accelerations which have to be imposed. These can be rewritten in vector form as:

$$a_v = {}^t\dot{v}_d + KP_v\, {}^tv_E + KI_v \int {}^tv_E(s)\, ds$$

$$a_F = -M_d^{-1}\left[{}^tF_E + D_v \Sigma\, {}^tv\right]$$

The matrices $KP_v$, $KI_v$, $M_d$, and $D_v$ are diagonal and positive definite. Note that m entries in $a_v$ and the complementary 6-m entries in $a_F$ are zero. These two vectors are merged in

$$a = a_v + a_F$$

which is the acceleration used into the nonlinear decoupling control $F^*(r,r',a)$. The overall feedback from the measured $F_c$ is

$$\left[\bar{B}_e(r)\, R^T(r)\, M_d^{-1}\, \Sigma R(r) - I\right] \cdot F_c$$

Whenever the second order information on the object surface (i.e. R') is missing in $F^*$, an integral term can be added in $a_v$ and in $a_F$. The benefits of this admissible modification will be evaluated by simulation. Moreover, one can rewrite the velocity loop using the position errors $^tr_E$ as an approximation. This leads to:

$$a_v = {}^t\dot{v}_d + KP_v\, {}^tv_E + KP_p\, {}^tr_E + KI_p \int {}^tr_E(s)\, ds$$

$$a_F = -M_d^{-1}\left[{}^tF_E + D_v \Sigma\, {}^tv\right] + KI_F \int {}^tF_E(s)\, ds$$

## Simulation results

The proposed hybrid control strategy has been simulated on a SCARA-type (i.e. without gravity terms) two-link robot arm. A two-dimensional force sensor is located at the tip and is modeled by a damped spring-mass system. Several simulation runs have been performed for tasks requiring to slide on linear or curved surfaces, at various speeds, and using sensors of different stiffness (Manes, 1987). Sampling times have to be chosen with reference to this stiffness, to the mass of the robot, and the apparent mass $M_d$ assigned in the force control loop.

The simulations reported here refer to a task which asks to follow for 2 seconds a circle of radius $\rho = .2$ m, at a given speed while applying a constant normal force. In Figure 4 all the relevant quantities are defined. A Coulomb friction term of the form $F_{frict} = -\gamma\,|F_{norm}|$ affects the motion on the surface and results in a coupling between normal and tangential forces. At 0.4 sec, the desired normal force undergoes a step change from 10 to 20 N, while the desired velocity changes from 0.3 to 0.6 m/sec at the trajectory midpoint.

The given method is compared with the one of Khatib (1987). For fairness, no information on the surface curvature is given to the controllers. In particular, this means that R'(r,r') is set to zero in the expression of $F^*(r,r',a)$. An integral term was added to both control schemes. Note also that here $T(r) = I$, since the motion is a planar one.

Figures 5 and 6 show the behavior of velocity and force using Khatib's method. Figures 7 and 8 are relative to the proposed task-based decoupling controller. The control sampling time is 10 msec while the chosen linear controllers are a PI² on velocity error and a PID on force error, with gains

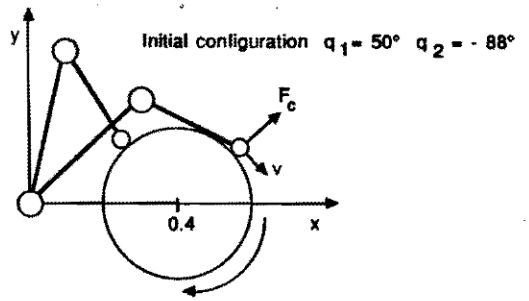
$$KP_v = 40 \qquad KI_v = 200 \qquad KI_v^2 = 1000$$

$KP_F = 2$ $KI_F = 25$ $KD_F = 90$

A remarkable improvement is noticed with the task-based hybrid controller. Both the overshooting and the sensitivity to orthogonal disturbances are reduced, particularly at high speed. Here, the different behavior is due to the way the contact force signal is used. The present scheme yields decoupling based on the actual contact force, while Khatib's one uses instead only a normal force feedforward term.

These results were obtained in the presence of a 10% error in the inertial parameters of the arm. Moreover, the joint viscous friction which is present in the dynamic model is compensated only in part by the controller.

It should be noted that the integral term is not needed when the curvature is zero. However, if this term is not included in the considered case, a steady state error results in the force control loop. This is due to the missing centripetal acceleration that the position control loop is not able to supply in the time-varying normal direction. In fact, if the tip has a constant tangential speed $\alpha'$, a normal acceleration $\beta'' = -\alpha'^2/\rho$ results. This will be provided by the force loop as $F_c = F_d - m_d \alpha'^2/\rho \neq F_d$. Indeed, if the curvature information $R'(r,r')$ is provided to the task-based hybrid controller no such steady state error would result, even without the integral action.




Sensor stiffness $K_s = 1000$ N/m Sensor damping $D_s = 7$ N.sec/m

Sensor mass $m_s = 0.025$ Kg Links $L_1 = 0.45$ m $L_2 = 0.32$ m

Joint viscosity $V_1 = 1.3$ N.m.sec/rad $V_2 = 0.2$ N.m.sec/rad

Mass and center of gravity of link 2 $m_2 = 3.5$ Kg $d_2 = 0.15$ m

Links inertia $I_1 = 0.25$ Kg.m$^2$ $I_2 = 0.16$ Kg.m$^2$

Friction coefficient $\gamma = 0.2$

**Figure 4** - Simulation layout

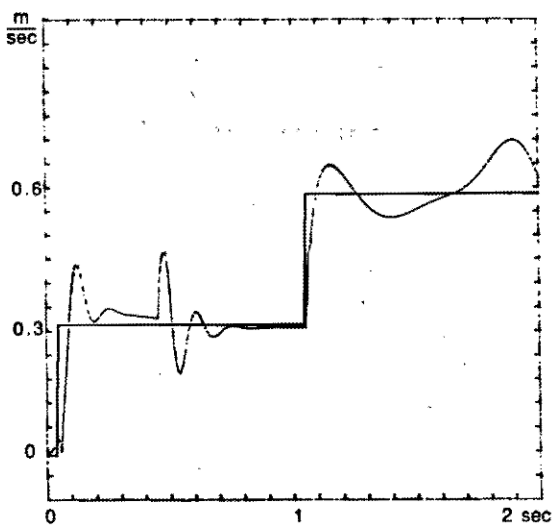



**Figure 5** - Tangential velocity using Khatib's method

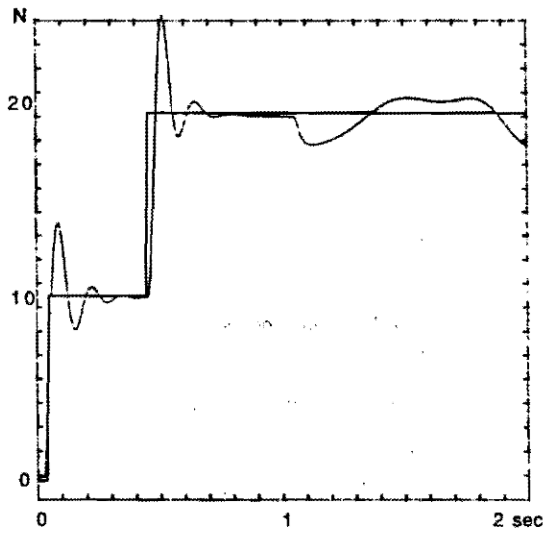



**Figure 6** - Normal force using Khatib's method

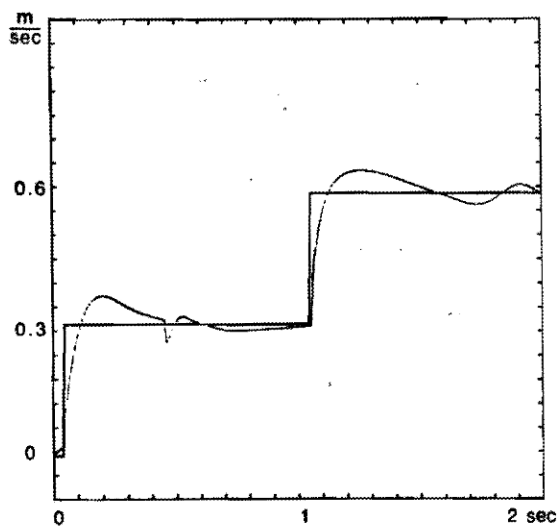



**Figure 7** - Tangential velocity using task decoupling method

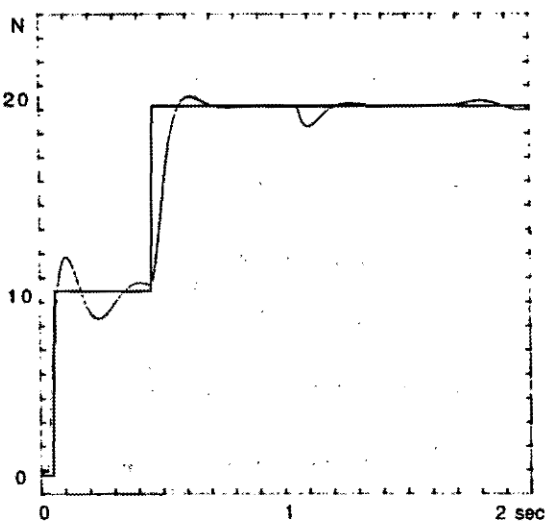



**Figure 8** - Normal force using task decoupling method

## Conclusions

Task-based control is the most natural approach to hybrid force-velocity control for robot manipulators in contact with the environment. Defining both the control objective and the relative control action in terms of task coordinates leads to a decoupled and non-redundant design. In this way, the control performance may be improved by a proper use of force and velocity errors. It was shown that the addition of an integral term in the task-based scheme enables to overcome lacking knowledge of the geometric properties of the environment.

Inclusion of a single dominating *elasticity* at the point of contact avoids instability problems and unrealistic situations. Moreover, using the dynamic model of the robot *in contact* with the environment allows to obtain noninteraction based on the measure of the actual contact forces. Thus, some robustness is achieved with respect to some unmodeled terms like Coulomb friction. This was confirmed by the numerical simulations. However, since part of the nonlinear controller is essentially model-based, the robustness with respect to large uncertainties in robot parameters may still be a critical issue.

The proposed control scheme is also a feasible one for doing on-line identification of the object surface orientation. This topic is currently under investigation.

## References


**An, C. H., and J. M. Hollerbach** (1987). Dynamic stability issues in force controlled manipulators, *4th IEEE Conf. on Robotics and Automation*, Raleigh, 890-896.

**Asada, H., and J. J. Slotine** (1986). *Robot Analysis and Control*, John Wiley.

**Eppinger, S. D., and W. P. Seering** (1987). Introduction to dynamic models for robot force control, *IEEE Control Systems Mag.*, 48-52.

**Hogan, N.** (1985). Impedance control: an approach to manipulation (part I-III), *Trans. ASME J. Dyn. Syst. Meas. and Contr.*, **107**, 1-24.

**Isidori, A.** (1985). *Nonlinear Control Systems: An Introduction*, Springer Verlag.

**Khatib, O., and J. Burdick** (1986). Motion and force control of robot manipulators, *3rd IEEE Conf. on Robotics and Automation*, S.Francisco, 1381-1386.

**Khatib, O.** (1987). A unified approach for motion and force control of robot manipulators: the operational space formulation, *IEEE J. Robotics and Automation*, **RA-3**, 43-53.

**Manes, C.** (1987). On hybrid position/force control (in Italian), Final Project, Università di Roma "La Sapienza".

**Mason, M.T.** (1981). Compliance and force control for computer controlled manipulators, *IEEE Trans. Syst., Man, and Cybern.*, **SMC-11**, 418-432.

**Raibert, M. H., and J. J. Craig** (1981). Hybrid position/force control of manipulators, *Trans. ASME J. Dyn. Syst. Meas. and Contr.*, **103**, 126-133.

**Yoshikawa, T.** (1987a). Dynamic hybrid position/force control of robot manipulators - Description of hand constraints and calculation of joint driving force, *IEEE J. Robotics and Automation*, **RA-3**, 386-392.

**Yoshikawa, T., T. Sugie, and M. Tanaka** (1987b). Dynamic hybrid position/force control of robot manipulators - Controller design and experiment, *4th IEEE Conf. on Robotics and Automation*, Raleigh, 2005-2010.

**West, H., and H. Asada** (1985). Task representation and constraint analysis for hybrid position/force controller design, *1st IFAC Symp. on Robot Control (SYROCO'85)*, Barcelona, 69-74.

**Whitney, D. E.** (1987). Historical perspective and state of the art in robot force control, *Int. J. Robotics Res.*, **6**, 3-14.


## Appendix

Consider a planar robot in contact with a perfectly rigid planar surface and applying a constant force. For simplicity, the surface is assumed to be frictionless so that it can provide only a normal reaction force which counterbalances the normal component of any force applied by the end-effector. Assume also that no gravity is present and that the robot, initially at rest, is required to start moving along the surface. Its dynamic model is then

$$B_e(r)\,\ddot{r} = F + F_c = F_{tot}$$

where $F_{tot}$ is the total force applied at the end-effector. This force is indeed tangent to the surface. The scalar product of vector $F_{tot}$ with the resulting acceleration $\ddot{r}$ yields

$$\ddot{r}^T F_{tot} = \ddot{r}^T B_e(r)\,\ddot{r} \geq 0$$

so that these two vectors form an angle which is always less than $\pi/2$. Moreover, $\ddot{r}$ and $F_{tot}$ will have different directions since in general $B_e(r) \neq \lambda I$. Hence, the acceleration vector will point either inside the surface, producing an inconsistent behavior, or outside, so that the robot leaves the surface. As soon as this happens, the absence of the surface reaction gives a net force which brings back the end-effector to the surface. A chattering behavior results, due to the ideal rigidity assumption.