

## ROBUST HYBRID DYNAMIC CONTROL OF ROBOT ARMS

A. De Luca, C. Manes, G. Ulivi

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

### Abstract

A robust hybrid scheme is presented for force and velocity control of robot arms in dynamic contact with the environment. This work develops a recently proposed approach, whose main feature is to preserve in the closed-loop the orthogonality between force and velocity also for time-varying tasks. Such a result can be achieved by designing a nonlinear decoupling control law directly in the space associated with the task. The resulting behavior is linear in terms of task-space forces and velocities and can be shaped using a minimum number of scalar controllers. However, since the control law is model-based, large uncertainties in the robot dynamic model may limit its performance. In this paper the sliding mode technique is used in order to overcome the need of an accurate robot model. Moreover, the independent linear task-space controllers on force and velocity are designed in the discrete-time domain, taking explicitly into account delays due to control computation. Additional integral control may usefully be included to compensate for a reduced knowledge of the environment geometry or to deal with unmodeled terms. The resulting control law is simple enough to be implemented in real-time. The robustness of the present control scheme is confirmed by realistic simulations for a two-link robot arm moving along a circular surface in the vertical plane.

### 1. Introduction

Many robotic applications involve contact between the robot and the environment, with an associated exchange of forces<sup>1</sup>. Typical examples are assembly or deburring operations performed by a single robot, or more advanced tasks in which the cooperation of two robot arms or the use of multi-fingered dextrous hands is needed.

When the constrained motion of the arm is not required to be fast, active compliance or impedance control already gives satisfactory results [1]. Instead, the interest is focused here on those robotic tasks for which the robot dynamics plays a relevant role. This is the case, for instance, when the robot end-effector tracks a high-speed trajectory defined on a smooth surface while exerting a possibly time-varying desired contact force. An accurate dynamic hybrid control of both velocity and force is then needed, and to this aim it is assumed that a multi-dimensional force sensor located near the arm tip is available. With force feedback, the robot controller increases its capability of adaptation to uncertainties in the world model.

The inclusion of any force sensing device implies the presence of some compliance between the robot end-effector and the environment. In the process of modeling the dynamic behavior of the robot in contact, an equivalent compliant element should be used. This may also account for other sources of elastic deformation, like the nonrigid characteristics of the environment and of the arm transmission elements.

Early approaches to robot force control were purely oriented on kinematic transformations (see e.g. the survey of Whitney [2]). The same happened to the first hybrid control scheme of Raibert and Craig [3], in which either velocity or force was controlled along every cartesian direction, ignoring position errors on the force-controlled axes and viceversa. Since dynamic coupling effects are neglected, their PID-based control works well only at slow speed.

Khatib [4,5] improved the hybrid approach using nonlinear state feedback to compensate for dynamic interactions. Control of force and velocity is performed then separately in cartesian coordinates; however, his scheme fails to preserve dynamical decoupling between force and

velocity when the orientation of the task is time-varying [6]. This can be illustrated even in a planar constrained motion, for example when the arm tip follows a circle. The task formulation of Yoshikawa [7,8] correctly takes into account also the dynamic effects which arise due to the geometric curvature of the constraining surface. However, the elasticity at the contact is not considered at all, and this is again a source of disturbance and a limitation for the controller.

An hybrid force-velocity dynamic control approach which copes with the above criticisms has been proposed by the Authors in [9]. At a dynamic level, it closely implements the orthogonality property between force and velocity of Mason's description of hybrid tasks [6]. The basic underlying idea is a proper treatment of the position, velocity and force errors. The operation of the hybrid dynamic controller can be summarized in four basic steps (Fig. 1): i) use rotational transformations to express all relevant vector variables in terms of their *task space* components; ii) close, at the task level, linear independent control loops of the velocity or of the force type, taking into account the contact dynamics; iii) map the obtained signals into an end-effector acceleration and then into a joint acceleration, using the proper inverse kinematic transformations; iv) compute the required actuator torques, based on the robot inverse dynamics.

As a result, linear and decoupled dynamic behavior is obtained in terms of task-space coordinates; therefore, the closed-loop system has an effective invariant performance over the whole task definition. A minimal number  $M$  of scalar controllers is required in this approach, where  $M$  is the dimension of the task space; in fact, only *one* independent velocity or force controller has to be designed for each of the generalized task coordinates, as opposed to previous hybrid control schemes (see the gain matrices in the Appendix of [3], or [10, p.209]). Numerical simulations have shown a better performance of this scheme over Kathib's hybrid controller [9,11]. Moreover, this task-space approach allows to add integral terms on the error in any of the scalar velocity or force controllers, without introducing back dynamic cross-couplings. With this integral action, one may avoid the use of second order (curvature) information about the environment, an improvement over the results of Yoshikawa [7,8]. The extension of this hybrid controller to redundant robot arms has been considered in [12], using a local scheme of torque redundancy resolution, as suggested by Hollerbach and Suh [13].

The main drawback of all the hybrid dynamic control methods [4,5,7-9] is in their intrinsic model-based nature. Therefore, uncertainties in the knowledge of the arm inertial parameters or changes in the end-effector tooling may heavily affect the tracking accuracy. Indeed, these is the same limitation arising in inverse dynamics schemes for free motion robot control, like in the computed torque method [14,15]. A standard way for reducing performance sensitivity of model-based controllers is to add a robustifying discontinuous feedback law, as in the sliding mode technique. More in general, this control strategy is referred to as variable structure control (VSC) (see e.g. [16] and the references therein). In robotic applications, this technique has been used successfully for trajectory control [17-20]. In particular, Slotine [18] has proposed a modified continuous sliding (suction) controller, studying the trade-off between chattering suppression and tracking precision. Experimental results obtained using VSC schemes are described in [19,20].

One of the appealing features of the VSC approach is that it leads to a cheap implementation, once the necessary stability analysis has been performed. The method is robust with respect to the presence of unmodeled phenomena, e.g. Coulomb friction. There are also cases when the overall model of the plant is known exactly, but part of the

<sup>1</sup> Force and velocity will be used to denote shortly both linear and angular quantities.

control computations have to be dropped due to real-time restrictions. By proper adjusting the bounds of the sliding controller, these missing terms are rejected as disturbances.

In this paper the sliding mode technique is applied in order to obtain a robust version of the hybrid dynamic control scheme introduced in [9]. A sliding surface is designed specifying zero-velocity error at the joint-space level, so that analog circuitry can be used for implementation [20], leading to a much faster operation rate and thus to a reduced chattering. This choice overcomes the need of an exact dynamic model of the robot arm. Nevertheless, the kinematic transformations between task, cartesian and joint space have to be known.

Moreover, a discrete-time approach is followed in the design of independent dead-beat linear controllers on task-space force and velocity. Delays due to computations, arising mainly for kinematic transformations and measurement processes, are taken explicitly into account in this framework. By the addition of an integral term the amount of kinematic computations can be reduced since no information on the surface curvature is then used.

A final remark concerns the use of a discontinuous law in a robot force control scheme. If only an instantaneous relationship is assumed between the contact forces and the joint torques, via the transposed Jacobian, the chatter on the control would affect directly the forces exchanged with the environment. Actually, the compliant dynamics of the contact acts as a filtering device on the discontinuous torques. In particular, a simple second-order linear spring-mass system is used here for modeling the robot-environment interaction, similarly to the impedance scheme of Hogan [1].

The paper is organized as follows. The robot and the contact dynamic models are introduced first, and the task-space model-based control strategy is briefly reviewed. Section 4 describes the proposed robust controller, with the sliding mode synthesis and the dead-beat discrete-time design of the scalar force and velocity control laws. Simulation results are finally reported for a two-link arm moving along a circular surface in a vertical plane, thus taking into account also gravity.

## 2. Dynamic model of robot and contact

The dynamic model of a rigid robot arm with  $N$  joints can be written as

$$B(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} + \mathbf{J}^T(\mathbf{q}) \mathbf{F}_c \quad (1)$$

where  $\mathbf{q} \in \mathbb{R}^N$  are the generalized coordinates of the system,  $B$  is the positive definite inertia matrix,  $\mathbf{n}$  contains centrifugal, Coriolis and gravitational forces,  $\mathbf{u}$  are the torques supplied by the actuators, and  $\mathbf{J}$  is the Jacobian of the robot direct kinematic relation  $\mathbf{f}(\mathbf{q})$ . This relation maps the joint space into the  $M$ -dimensional extended space of position and orientation of the robot end-effector, with  $M \leq 6$ . Let  $\mathbf{r}$  be a vector which coordinatizes this space, i.e.  $\mathbf{r} = \mathbf{f}(\mathbf{q})$ . In the general case  $\mathbf{r}$  lives in  $\mathbb{R}^3 \times \text{SO}(3)$ , while for planar motion one can assume  $\mathbf{r} \in \mathbb{R}^3$ . In the following, only "square" systems will be considered, i.e.  $M = N$ , but the approach is applicable also to redundant arms [12]. In (1),  $\mathbf{F}_c$  are the generalized (contact) forces performing work on  $\mathbf{r}$  and acting from the environment to the robot end-effector. An  $M$ -dimensional force sensor is assumed to be available for the direct measure of these contact forces.

Note that a minimal representation of the end-effector orientation should be used in  $\mathbf{r}$ , like the roll-pitch-yaw angles or any proper set of Euler angles [21]. Some caution is needed in the definition and transformation of these rotational quantities. In fact, the rotational part of a robotic task is described in terms of angular velocities and torques around the axes of a common frame [6], e.g. the base frame. Let  $\mathbf{v}$  be the vector of linear and angular velocities and  $\mathbf{F}_m$  be the measured forces and torque<sup>2</sup>, both expressed in this frame. Then, the following general relations hold

<sup>2</sup> In practice, forces and torques are measured w.r.t. a frame which is specific to the particular sensor. Using another transformation matrix which is not considered here, these measures are mapped into  $\mathbf{F}_m$ .

$$\mathbf{v} = \mathbf{T}(\mathbf{r}) \dot{\mathbf{r}}, \quad \mathbf{F}_c = \mathbf{T}^T(\mathbf{r}) \mathbf{F}_m \quad (2)$$

with  $\mathbf{T}(\mathbf{r})$  invertible, except for isolated points. The presence of this orientation-dependent matrix  $\mathbf{T}$  is a way of saying that the angular velocity is *not* an exact differential of any minimal representation of orientation. Note that  $\mathbf{T}(\mathbf{r}) = \mathbf{I}$  when the motion does not involve changes in orientation or is restricted to a plane.

As for the robot interaction with the environment, the following modeling assumptions are made:

- the contact between end-effector and environment is pointwise and elastic, and this compliance can be modeled using a single constant elasticity coefficient;
- the orientation of the tangent plane and the curvature of the surface at the contact point are assumed to be known.

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, i.e. the sensor mass, is negligible.

The assumed knowledge about the environment geometry is required by the complete version of the control law. However, the use of curvature information at the contact point can be avoided by the addition of an integral term in the controller, as shown in [9]. It is also possible to relax the a-priori knowledge of the local normal to the environment by incorporating an on-line identification procedure, based on force measurements processing [11].

## 3. Task-space hybrid dynamic control

Let  ${}^0\mathcal{S}$  be the base reference frame for robot motion analysis (cartesian world coordinates). It is also convenient to define a frame  ${}^0\mathcal{S}^*$ , which is oriented as  ${}^0\mathcal{S}$ , but translated at the contact point between the end-effector and the environment. Following [6], an hybrid force-velocity task description uses another  $M$ -dimensional frame  ${}^1\mathcal{S}$  associated to the task<sup>3</sup>. The orientation of the axes of  ${}^1\mathcal{S}$  is in general time-varying and such that, at all instants,  $p$  scalar forces and  $M-p$  scalar velocities can be specified, one along or around each of the axes. In the following, we set  $M = 6$  for notational convenience. The desired force  ${}^1\mathbf{F}_d$  and velocity  ${}^1\mathbf{v}_d$  associated to the task are naturally expressed in this task frame. Linear velocity and force are transformed from  ${}^0\mathcal{S}^*$  into  ${}^1\mathcal{S}$  by means of a  $3 \times 3$  rotation matrix  $\mathbf{R}_L(\mathbf{r})$ ; similarly, angular velocity and torque are transformed by  $\mathbf{R}_A(\mathbf{r})$ . A  $6 \times 6$  orthonormal matrix  $\mathbf{R}(\mathbf{r})$  will be defined as

$$\mathbf{R}(\mathbf{r}) = \begin{bmatrix} \mathbf{R}_L(\mathbf{r}) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_A(\mathbf{r}) \end{bmatrix} \quad (3)$$

with  $\mathbf{R}^{-1}(\mathbf{r}) = \mathbf{R}^T(\mathbf{r})$ . Note that this matrix is used only for expressing vectors in different frames *without* changing the point of application, thus being a special case of the general transformation mapping. Finally, a  $6 \times 6$  diagonal selection matrix  $\Sigma$  with 0/1 elements is used to specify the axes of the task frame which will be force controlled. Indeed, the matrix  $\Sigma^\wedge := \mathbf{I} - \Sigma$  selects the remaining axes.

In De Luca et al. [9], this description of desired task force  ${}^1\mathbf{F}_d$  and velocity  ${}^1\mathbf{v}_d$  has been used for defining errors as

$${}^t\mathbf{F}_E = {}^t\mathbf{F}_d - \Sigma \mathbf{R}(\mathbf{r}) \mathbf{F}_m \quad (4)$$

$${}^t\mathbf{v}_E = {}^t\mathbf{v}_d - \Sigma^\wedge \mathbf{R}(\mathbf{r}) \mathbf{v} \quad (5)$$

To allow for a decoupled control action which preserves compatibility between force and velocity specifications, these errors should be processed *directly* in the task space (see Fig. 1). In this way only *six* (i.e. a minimal number of) linear controllers are designed for the main loops,  $p$  on the force error vector  $\mathbf{F}_E$  and  $6-p$  on the velocity error

<sup>3</sup> In the general case these frames are the superposition of a three-axes frame for linear quantities and a three-axes frame for angular quantities.

$v_E$ . This treatment of errors is basically different from the classical hybrid kinematic [3] or dynamic [4,5,10] control schemes. The *weighting* of errors becomes *independent* from the *orientation* of the task frame, so that an integral penalization can be consistently introduced.

Each force and velocity controller provides as output a scalar instantaneous acceleration in one different task-space direction. These outputs are then merged componentwise to produce a resulting task acceleration vector  ${}^l a$ . In the force loop, the second-order dynamic model of the contact is included; this allows to design the force control law using an impedance paradigm, although based explicitly on the measured force error.

In order to have one independent controller for each axis of the task frame, the dynamic interactions arising from the robot arm motion should be compensated first. In [9], this was accomplished via a nonlinear state-feedback law obtained as an application of decoupling control theory [22]. In that case, the nonlinear control block of Fig. 1 was formed the inverse dynamic model of the robot arm plus the kinematic transformations needed for computing torques which produce the required task acceleration.

To complete the above analysis, the differential kinematic relation between task and joint acceleration is needed. Taking the time derivative of the task velocity

$${}^l \dot{v} = R(r) T(r) \dot{r} \quad (6)$$

yields

$${}^l a = {}^l \dot{v} = R(r) T(r) \ddot{r} + [\dot{R}(r, \dot{r}) T(r) + R(r) \dot{T}(r, \dot{r})] \dot{r} \quad (7)$$

On the other hand, the end-effector acceleration is

$$\ddot{r} = J(q) \ddot{q} + \dot{J}(q, \dot{q}) \dot{q} \quad (8)$$

which can be substituted in (7). If a value  ${}^l a_d$  is desired for the task acceleration, then the associated joint acceleration becomes

$$\ddot{q} = J^{-1} \{ T^{-1} R^T [{}^l a_d - (\dot{R} T + R \dot{T}) \dot{r}] - \dot{J} \dot{q} \} \quad (9)$$

where dependence of the various matrices is dropped. As expected, the kinematic transformation (9) requires some geometric knowledge about the environment. In particular,  $R(r)$  contains information on the gradient of the surface at the point of contact, while curvature information is represented by  $R'(r, \dot{r})$ . When the term due to surface curvature is not available or is neglected, disturbances will arise in the dynamic behavior. These kinematic couplings become relevant only for surfaces whose curvature is high with respect to the speed of motion.

#### 4. Robust controller design

The proposed general task-based decoupling hybrid control scheme will be specified here with the purpose of achieving robustness with respect to robot parameter uncertainties and computational delays.

##### 4.1. Sliding mode control

A sliding controller is used to determine the torque imposing the required joint acceleration to the arm, as computed from the task acceleration  ${}^l a_d$  using (9) (see also Fig. 2). This will be achieved in face of the nonlinear dynamic interactions due to robot motion and to gravity.

In general, a variable structure control  $u$  can be decomposed in two terms: the so-called equivalent control [16], and the discontinuous part  $u_s$ . If an approximation  $u_0$  of the first one is available, it can be used to reduce the control effort of the discontinuous term. Therefore,  $u_s$  is designed to balance the effects due to differences between the nominal model and the true dynamics. In the robotic case, it is also convenient to weight  $u_s$  with the estimate of the inertia matrix; as a result, the overall torque will be of the form

$$u = u_0 + B_0(q) u_s \quad (10)$$

An  $N$ -dimensional time-varying sliding surface  $\sigma$  is then defined as:

$$\sigma(\dot{q}, t) = \dot{q} - \dot{q}_d(t) \quad (11)$$

The term  $u_0$  is formally obtained by setting  $d\sigma/dt = 0$ ,

$$\dot{\sigma}(q, \dot{q}, t) = \ddot{q} - \ddot{q}_d = B_0^{-1}(q) [u_0 + J^T(q) F_d - n_0(q, \dot{q})] - \ddot{q}_d = 0 \quad (12)$$

which gives

$$u_0 = n_0(q, \dot{q}) - J^T(q) F_d + B_0(q) \ddot{q}_d \quad (13)$$

The subscript 0 denotes a nominal model of the robot dynamics which is used within this computation. Both the degree of accuracy and the complexity of this nominal model may vary according to the specific case. The term  $u_0$  becomes a feedforward whenever the desired  $\ddot{q}_d$  are used in place of the actual  $\ddot{q}$ .

The  $u_s$  part of the control is derived by imposing the attractivity condition for the sliding surface (11) or, equivalently, the non-positivity of the time-derivative of the Lyapunov function  $V = 1/2 \sigma^T \sigma$  [18]:

$$\sigma^T \dot{\sigma} = \sigma^T (\ddot{q} - \ddot{q}_d) = \sigma^T \{ B^{-1}(q) [u_0 + B_0 u_s + J^T(q) F_c - n(q, \dot{q})] - \ddot{q}_d \} \leq 0 \quad (14)$$

After some manipulation, this can be rewritten as

$$\begin{aligned} \sigma^T \dot{\sigma} &= \sigma^T B^{-1}(q) [\Delta B \ddot{q}_d + \Delta n - J^T(q) \Delta F + B_0(q) u_s] = \\ &= \sigma^T B^{-1}(q) B_0(q) [\Delta(q, \dot{q}, \ddot{q}_d, \Delta F) + u_s] \leq 0 \end{aligned} \quad (15)$$

where  $\Delta B = B_0 - B$ ,  $\Delta n = n_0 - n$ , and  $\Delta F = F_d - F_c$ . The perturbation term is  $\Delta = B_0^{-1} (\Delta B \ddot{q}_d + \Delta n - J^T \Delta F)$ .

Various strategies can be followed for ensuring inequality (15). Under the realistic hypothesis of limited desired acceleration,  $|\ddot{q}_d| \leq Q_i$ , knowing lower and upper limits  $b_{\min}$ ,  $b_{\max}$  on the norm of the inverse of the inertia matrix  $\|B^{-1}(q)\|$  [23], and bounds  $\delta_i$  on the single terms  $|\Delta_i|$ , it is possible to specify componentwise  $u_s$  as

$$u_{s,i} = -\rho_i \operatorname{sgn}(\sigma_i), \quad i = 1, \dots, N \quad (16)$$

where  $\rho_i$  is bounded from below by a function of  $Q_i$ ,  $\delta_i$ ,  $b_{\min}$  and  $b_{\max}$ .

From a practical point of view, the  $\rho_i$ 's can be tuned also by off-line simulation, e.g. considering maximum excursions of parametric variations, or directly with real-time experiments over different typical trajectories. Note that the VSC technique has been applied here at the joint level, the more suitable for implementation purposes.

##### 4.2. Discrete-time design of linear compensators

The design of compensators specifying the task acceleration  ${}^l a_d$  vector in (9) completes the control law synthesis. Indeed, if the sliding mode controller (11) works properly, approximate linear and decoupled behavior is ensured in the task-space. Therefore, independent linear controllers can be used for each of the task components. A discrete-time approach will be followed for designing the linear control laws. In fact, unavoidable time lags arise in real-time implementations due to data acquisition and processing. These delays can be explicitly taken into account when working in the discrete-time domain. In particular, a dead-beat philosophy [24] is chosen here.

In the following, the design of the velocity and of the force control laws will be treated separately. Let  $\alpha$  and  $\beta$  denote generic task-space directions controlled in velocity and, respectively, in force. Acceleration signals  $a_\alpha$  and  $a_\beta$  will be computed and organized in two vectors  ${}^l a_v$  and  ${}^l a_f$ . By adding  $p$  zero entries to  ${}^l a_v$  and  $6-p$  complementary zero entries to  ${}^l a_f$ , these will be finally merged into  ${}^l a_d$  and used in (9).

#### 4.2.1. Velocity control

Along a velocity-controlled axis  $\alpha$ , the resulting continuous time closed-loop system is

$$\ddot{\alpha} = a_\alpha \quad (17)$$

Therefore, a simple integration connects the input  $a_\alpha$  to the task velocity. Since the acceleration input is held constant over a sampling period of duration  $T_s$ , the transform into the z-domain yields the model

$$\dot{\alpha}(z) = \frac{T_s}{z-1} a_\alpha(z) \quad (18)$$

The input computed at time  $t = kT_s$ , i.e.  $a_{\alpha,k}$ , will be applied only at the next sampling instant  $t = (k+1)T_s$ , so a one-step delay  $z^{-1}$  is introduced in cascade with (18). The resulting control scheme is shown in Fig.3. To obtain zero-error in the output after a finite number of steps, the two control gains  $K_{v1}$  and  $K_{v2}$  are chosen as

$$K_{v1} = \frac{1}{T_s} \quad K_{v2} = -1. \quad (19)$$

The closed-loop system has two poles in  $z=0$  and guarantees dead-beat response of velocity within two steps. In the presence of external disturbances, this controller may accumulate a finite position error. If position instead of velocity control along the surface is of primary concern, it is to modify this scheme by closing a further feedback loop based on the desired reference position or by adding an integral action on the velocity error, in the way shown below for the force controller.

#### 4.2.2. Force control

The design of the force controller along a direction  $\beta$  is made assuming a second order spring-mass model for the contact dynamics. The impedance scheme introduced by Hogan [1] can be modified and used here for force control. For the continuous-time behavior along  $\beta$

$$\ddot{\beta} = a_\beta \quad (20)$$

a proper control is chosen so to match the model

$$m_d \ddot{\beta} + d_d \dot{\beta} + k \beta = -F_{\beta,d} \quad (21)$$

where  $\beta$  is measured from the rest position of the force sensor. The parameter  $1/k^*$  is an equivalent compliance used to model the contact, assuming that the mass sensor is negligible w.r.t.  $m_d$ . This value results from the parallel of environment compliance  $1/k_e$  and of sensor compliance  $1/k_s$ ; the latter, which is known, provides a lower bound for  $1/k^*$  that can be used in a worst-case design. In (21),  $F_{\beta,d}$  is the desired force along  $\beta$ , while the mass and damping parameters  $m_d$  and  $d_d$  can be chosen freely so to obtain a prescribed transient dynamic behavior. This is somewhat different from Hogan's impedance control, where all the three coefficients on the left side of (21) are design parameters, and the contact force results as a consequence.

With the chosen origin for distances, the measured force  $F_{\beta,c}$  can be rewritten as the product of a displacement times the equivalent stiffness,  $F_{\beta,c} = -k^*\beta$ , so that (21) becomes:

$$m_d \ddot{\beta} + d_d \dot{\beta} + (F_{\beta,d} - F_{\beta,c}) = 0 \quad (22)$$

Note that the force  $F_{\beta,c}$  is a measured quantity; therefore, an imperfect knowledge of  $k^*$  affects only the transient behavior of the system but not the steady-state force error, which is zero for any positive  $m_d$  and  $d_d$ .

A continuous-time law for (20), yielding (21) as closed-loop behavior, is

$$a_\beta = -\frac{1}{m_d} [(F_{\beta,d} - F_{\beta,c}) + d_d \dot{\beta}] \quad (23)$$

The discretization of (23) works fine when using relatively fast sampling

times, as shown by the numerical results in [9]. When this condition cannot be met, a discrete-time design provides better performance.

Assuming a piece-wise constant input  $a_\beta$ , the model of the double integrator (20) becomes in the z-domain:

$$\beta(z) = \frac{T_s^2}{2} \frac{z+1}{(z-1)^2} a_\beta(z) \quad (24)$$

Taking into account as before a one-step computational delay, the following control law provides a dead-beat response:

$$a_\beta(z) = -\frac{5z}{2T_s(z+2)} \beta(z) - \frac{z}{T_s^2 k^*(z+2)} [F_{\beta,d}(z) - F_{\beta,c}(z)] \quad (25)$$

The corresponding difference equation is

$$a_{\beta,k+1} = -2a_{\beta,k} - \frac{5}{2T_s} \dot{\beta}_k - \frac{1}{T_s^2} \left[ \frac{F_{d,k} - F_{c,k}}{k^*} \right] \quad (26)$$

where the subscripts  $k$  and  $k+1$  refer to the values of the variables sampled at the instants  $kT_s$  and  $(k+1)T_s$ . Figure 4 shows the closed-loop system described assuming as input  $\beta_{d,k} = F_{d,k}/k^*$  and as output  $\beta$  itself. From (26), the coefficients  $K_{f1}$ ,  $K_{f2}$ , and  $K_{f3}$  in this scheme are to be taken respectively equal to  $1/T_s^2$ ,  $-5/2T_s$ , and  $-2$ . The dashed branch is relative to a further integral control action not present in (26),  $K_{f4} = 0$ , and whose benefits will be considered in the next section. It is easy to see that, even if the controller itself is unstable, the overall closed-loop transfer function between desired and measured force is stable and is given by:

$$\frac{F_c(z)}{F_d(z)} = \frac{z+1}{2z^3} \quad (27)$$

As expected, the force error is zeroed in three sampling steps.

## 5. Simulation results

The proposed control scheme has been simulated for a two-link robot arm with rotary joints, moving in a vertical plane under the effect of gravity. A two-dimensional force sensor is located at the tip. The numerical results reported here refer to the task shown in Fig. 5, where the values of the relevant quantities can be found. The arm tip has to follow a circle of radius  $r = 0.3$  m, at a given speed while applying a constant normal force. A Coulomb friction term of the form  $F_{\text{frict}} = -\gamma |F_{\text{norm}}|$  opposes the motion on the surface and results in a coupling between normal and tangential forces. The task is simulated for 1.5 seconds. At 0.4 sec, the desired normal force undergoes a step change from 10 to 20 N, while after 1 second the desired tangential velocity changes from 0.2 to 0.4 m/sec.

In the design of the sliding mode law, control  $u_0$  in (13) is based on a very simple approximation of the real robot dynamics. In fact, a constant diagonal inertia matrix  $B_0$  is used, while  $n_0$  and  $F_d$  are both set equal to zero; therefore, the resulting  $u_0$  is a pure feedforward term. For the discontinuous part  $u_s$ , values of 100 Nm for the bounds  $p_i$  in (15) were found to be satisfactory for both joints. A saturation level has been added at 500 Nm to limit the actuator torque  $u$  as a whole.

Since the motion is planar,  $T(r) = I$  in all kinematic expressions. Moreover, no second order information on the curvature is given to the controller, so  $R'(r, r')$  is set to zero in (9). Therefore, an integral term is added to the force control loop ( $K_{f4}$  in Fig. 4). This counterbalances the effect of a missing centripetal acceleration that the velocity controller is not able to supply in the time-varying normal direction. In fact, when the robot arm tip has a constant tangential speed  $v$ , a normal acceleration  $-v^2/r$  results. This will be provided by the force loop as  $F_c = F_d - m_d v^2/r \neq F_d$  and, without integral action, a steady state error would result.

Figures 6 and 7 show the force and velocity time-behavior plotted against the desired one. The sampling time is  $T_s = 10$  msec and a time delay  $T_r = 10$  msec is considered so to let the controller perform all the required computations. The plots show a good response to the step

inputs, both in the velocity and in the force controlled directions. The introduced delay does not affect the performance of the control algorithm. These results give also evidence of the reduced sensitivity to orthogonal disturbances, confirming the decoupled nature of the closed-loop behavior. Note that no jamming phenomena appear due to friction. The computed torque profiles are reported in Fig. 8. The actuator of the first joint saturates for a while at the beginning of the task. The spikes present in the control inputs should be expected, since no boundary layer was used around the sliding surface. Nonetheless, this behavior does not affect the controlled variables.

As a further validation of the discrete-time design, the discretized version of a continuous-time linear control law for force and velocity has been simulated using the same sampling time  $T_s$  and with a smaller delay  $T_r = 5$  msec. The results on the contact force in Fig. 9 clearly show a degradation in performance, with large overshoots.

A final remark on the role of the equivalent elasticity constant  $k^*$  is in order. Sensors with different stiffnesses were used in simulations performed in [11] and the obtained results have shown the relevance of this parameter in the overall dynamic behavior. Roughly speaking, the system under control can be seen as constituted by the robot mass and the sensor spring. Larger values for this stiffness lead to a higher resonant frequency, which implies a reduction of the sampling time; this in turn increases the computational burden. Besides, small sampling times produce also high gains in the dead-beat controllers (see (19) and (26)) and thus larger values of the instantaneous torques. On the other hand, low stiffness sensors provide poor positional accuracy. Therefore, a suitable trade-off between these two opposite requirements should be found when selecting the force/torque sensor.

## 6. Conclusions

An improved method for hybrid dynamic robot control has been presented which guarantees decoupled behavior at the task-space level without explicit knowledge of the robot arm dynamics. This is obtained with a proper treatment of the task-space force and velocity errors and with the use of the sliding mode technique for the synthesis of the joint actuator torques, resulting in robustness with respect to model parametric uncertainties. A digital design of the linear force and velocity task-space controllers provides explicit treatment of computational time delays, which are due to the kinematic transformations involved in the control law. Additional integral control action allows to neglect second order information on the environment geometry at the contact point, without producing steady-state force or position errors. The satisfactory performance obtained in realistic simulations and the very limited complexity of the proposed control law suggest that this hybrid scheme may be successfully realized in practice using available digital/analog hardware.

## References

- [1] Hogan, N., Impedance control: an approach to manipulation (part I-III) *Trans. ASME J. Dyn. Syst. Meas. and Contr.*, **107**, 1-24, 1985.
- [2] Whitney, D.E., Historical perspective and state of the art in robot force control, *Int. J. Robotics Res.*, **6**, 3-14, 1987.
- [3] Raibert, M.H., Craig, J.J., Hybrid position/force control of manipulators, *Trans. ASME J. Dyn. Syst. Meas. and Contr.*, **103**, 126-133, 1981.
- [4] Khatib, O., Burdick, J., Motion and force control of robot manipulators, *3rd IEEE Conf. on Robotics and Automation*, 1381-1386, S. Francisco, 1986.
- [5] Khatib, O., A unified approach for motion and force control of robot manipulators: the operational space formulation, *IEEE J. Robotics and Automation*, **RA-3**, 43-53, 1987.
- [6] Mason, M.T., Compliance and force control for computer controlled manipulators, *IEEE Trans. Syst., Man, and Cybern.*, **SMC-11**, 418-432, 1981.
- [7] Yoshikawa, T., 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, 1987.

- [8] Yoshikawa, T., Sugie, T., Tanaka, M., Dynamic hybrid position/force control of robot manipulators - Controller design and experiment, *4th IEEE Conf. on Robotics and Automation*, 2005-2010, Raleigh, 1987.
- [9] De Luca, A., Manes, C., Nicolò, F., A task space decoupling approach to hybrid control of manipulators, *2nd IFAC Symp. on Robot Control (SYROCO'88)*, Karlsruhe, October 1988.
- [10] Asada, H., Slotine J.J.E., *Robot Analysis and Control*, J.Wiley, 1986.
- [11] Manes, C., On hybrid position/force control (in Italian), Final Project, Università di Roma "La Sapienza", December 1987.
- [12] De Luca, A., Manes, C., Nicolò, F., Hybrid force-velocity control using redundant manipulators, *NATO Advanced Research Workshop on Robots with Redundancy: Design, Sensing and Control*, Salò, June 1988.
- [13] Hollerbach, J.M., Suh, K.C., Redundancy resolution of manipulators through torque optimization, *IEEE J. Robotics and Automation*, **RA-3**, 308-316, 1987.
- [14] Freund, E., Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators, *Int. J. Robotics Res.*, **1**, 65-78, 1982.
- [15] Tarn, T.J., Bejczy, A.K., Isidori, A.K., Chen, Y., Nonlinear feedback in robot arm control, *23rd IEEE Conf. Decision and Control*, Las Vegas, 1984.
- [16] DeCarlo, R.A., Zak, S.H., Matthews, G.P., Variable structure control of nonlinear multivariable systems: a tutorial, *Proceedings of the IEEE*, **76**, 3, 212-232, 1988.
- [17] Young, K.K.D., Controller design for a manipulator using theory of variable structure systems, *IEEE Trans. Syst., Man, and Cybern.*, **SMC-8**, 101-109, 1978.
- [18] Slotine, J.J.E., The robust control of robot manipulators, *Int. J. Robotics Res.*, **4**, 2, 49-64, 1985.
- [19] Harashina, F., Hashimoto, H., Maruyama, K., Practical robust control of robot arm using variable structure systems, *3rd IEEE Conf. on Robotics and Automation*, San Francisco, 1986.
- [20] Bellini, A., Figalli, G., Pinello, P., Ulivi, G., Realization of a control device for a robotic manipulator based on nonlinear decoupling and sliding mode control, *Proc. IEEE Industrial Application Soc. Annual Meeting*, 1288-1294, Atlanta, 1987.
- [21] Goldstein, H., *Classical Mechanics*, Addison Wesley, 1980.
- [22] Isidori, A., *Nonlinear Control Systems: An Introduction*, Springer Verlag, 1985.
- [23] Spong, M.W., Vidyasagar, M., Robust linear compensator design for nonlinear robotic control, *IEEE J. Robotics and Automation*, **RA-3**, 4, 345-351, 1987.
- [24] Isermann, R., *Digital Control Systems*, Springer Verlag, 1977.

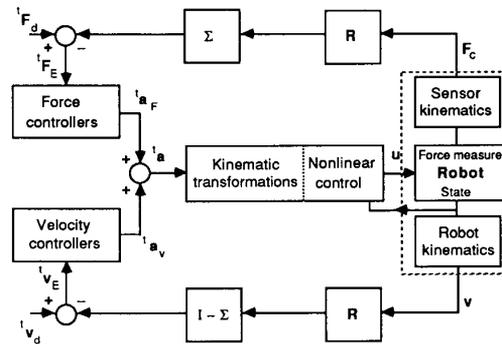


Figure 1 - Task-based decoupling hybrid control

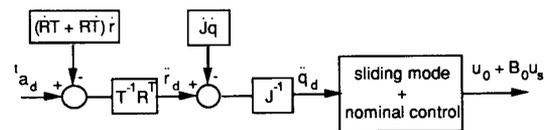


Figure 2 - Block diagram of the decoupling controller and of the kinematic transformations

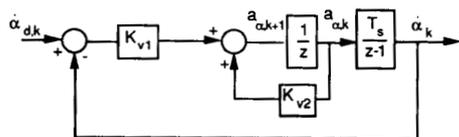


Figure 3 - Digital velocity controller

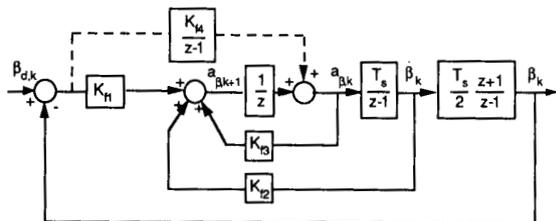
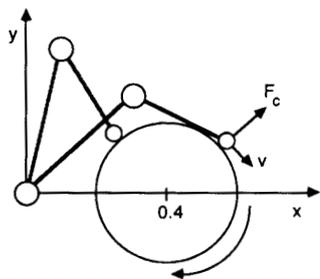


Figure 4 - Digital force controller



Sensor stiffness  $k_s = 10^4 \frac{N}{m}$       Link length  $l_1 = l_2 = 0.5 \text{ m}$   
 Sensor mass  $m_s = 0.025 \text{ Kg}$       Link masses  $m_1 = 25 \text{ Kg}, m_2 = 13.6 \text{ Kg}$   
 Sensor damping  $d_s = 22.13 \frac{N \cdot \text{sec}}{m}$       Friction coefficient  $\gamma = 0.2$

Figure 5 - Simulation set-up of hybrid task

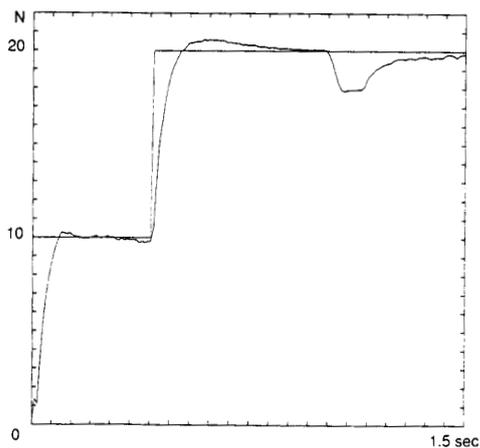


Figure 6 - Normal force using discrete-time controller ( $T_r = 10 \text{ ms}$ )

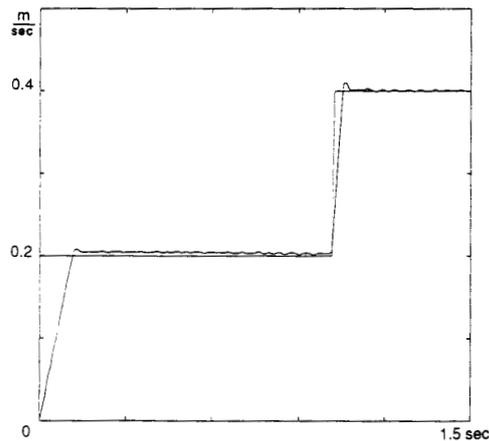


Figure 7 - Tangential velocity using discrete-time controller ( $T_r = 10 \text{ ms}$ )

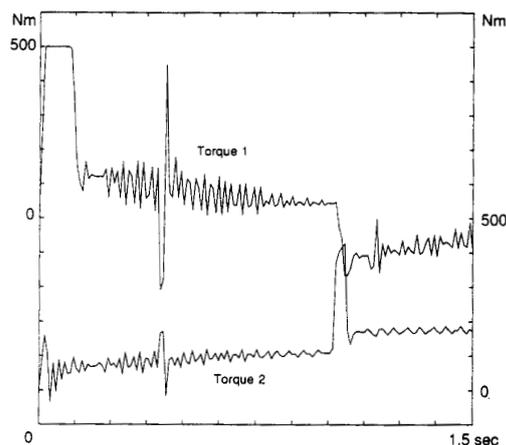


Figure 8 - Joint torques using discrete-time controller ( $T_r = 10 \text{ ms}$ )

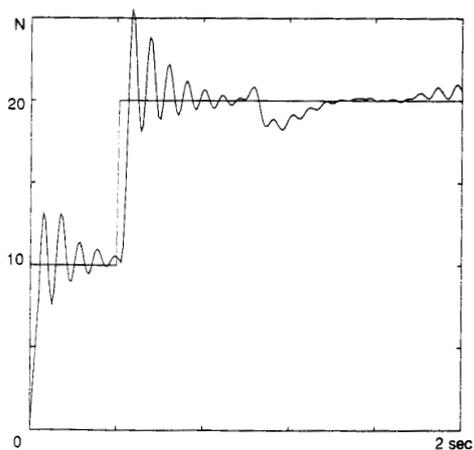


Figure 9 - Normal force using continuous-time controller ( $T_r = 5 \text{ ms}$ )

This work was partially supported by CNR Progetto Finalizzato Robotica