Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, vol. 118, no. 3, 09/96.

ROBUST JOINT AND CARTESIAN CONTROL OF UNDERACTUATED MANIPULATORS

Marcel Bergerman[†] Yangsheng Xu[‡]

[†]Department of Electrical and Computer Engineering
[‡]The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213

Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, vol. 118, no. 3, 09/96.

Abstract

Underactuated manipulators are robot manipulators composed of both active and passive joints in serial chain mechanisms. The study of underactuation is significant for the control of a variety of rigid-body systems, such as free-floating robots in space and gymnasts, whose structure include passive joints. For mechanisms with large degrees of freedom, such as hyper-redundant snake-like robots and multi-legged machines, the underactuated structure allows a more compact design, weight decrease and energy saving. Furthermore, when one or more joints of a standard manipulator fail, it becomes an underactuated mechanism; a control technique for such system will increase the reliability and fault-tolerance of current and future robots. The goal of this study is to present a robust control method for the control of underactuated manipulators subject to modelling errors and disturbances. Because an accurate modelling of the underactuated system is more critical for control issues than it is for standard manipulators, this method is significant in practice. Variable structure controllers are proposed in both joint space and Cartesian space, and a comprehensive simulation study is presented to address issues such as computation, robustness, and feasibility of the methods. Experimental results demonstrate the actual applicability of the proposed methods in a real two degrees-of-freedom underactuated manipulator. As it will be shown, the proposed variable structure controller provides robustness against both disturbances and parametric uncertainties, a characteristic not present on previouly proposed PID-based schemes.

1 Introduction

In this work, the authors deal with the problem of robust control of underactuated manipulators, which are mechanisms with less actuators than joints. One of the advantages of using underactuated systems is that they weight less and consume less energy than their fully-actuated counterparts, thus being useful for applications such as space robotics. For hyper-redundant robots, such as snake-like robots or multilegged mobile robots, where large redundancy is available for dexterity and specific task completion, underactuation allows a more compact design and simpler control and communication schemes. The underactuated robot concept is also useful for the reliability or fault-tolerant design of fully-actuated manipulators working in hazardous areas or with dangerous materials. If any of the joint actuators of such a device fails, one degree of freedom of the system is lost. It is important, in these cases, that the passive (failed) joint can still be controlled via the dynamic coupling with the active joints, so the system can still make use of all of its degrees of freedom originally planned.

Not until recently researchers started to work with the control problem of underactuated manipulators. Arai and Tachi (1991) proved that, in a local sense, the number of active joints must be equal or greater than the number of passive ones in order to be possible to control the passive ones—the same result was later established in a simpler way by Bergerman et al. (1994). They also developed a joint space controller which is similar to the classical computed torque controller associated with a proportional-derivative-integral (PID) controller, to bring all joints to their desired set-points. The drawback existent in this approach is that an accurate model of the manipulator must be provided to the controller. Later on this scheme was extended to the Cartesian space control problem (Arai et al., 1993). The approach by Papadopoulos and Dubowsky (1991) considered the failure recovery control of a space robotic manipulator. In this work the authors used both a computed torque controller, which requires an accurate model of the system, and a simple PID one. One of the drawbacks in this work is that the only joints that can be controlled are those for whom the inertia matrix is invariant with respect to the joint angle. Oriolo and Nakamura (1991) showed that there does not exist any smooth control law that guarantees that the system will stabilize at an equilibrium point. Therefore, one must either give up smoothness in the control law, or be satisfied with stabilization to equilibrium manifolds. They showed that a simple proportional-derivative (PD) controller is able to bring the system to a stable configuration over an equilibrium manifold. They also presented a detailed consideration about the conditions for integrability of the nonholonomic constraints present in underactuated manipulators. Jain and Rodriguez (1993) provided an analysis of the kinematic and dynamic issues of this kind of robotic system, using the spatial operator algebra. Control issues were not taken into account in this work. Saito et al. (1994) developed a 2-link underactuated brachiation robot, capable of moving along crossbars using only one actuator.

Hereafter, we will refer to *active joints* as the ones which are fully-controlled via an actuator. Analogously, *passive joints* are those who cannot be controlled directly, but which are equipped with a brake. It is assumed that all joints, passive and active, have position encoders that provide a complete knowledge of the joint angles at every sampling instant. We will denote by n the total number of joints, and by r the number of actuators. The number of passive joints is thus $p = n - r$. Following Arai and Tachi (1991), and Bergerman et al. (1994), our method requires that at least half of the joints be actuated ($r \geq p$); this follows from the fact that, at every time instant, one active joint is necessary to drive one passive joint. Also following the assumptions in Arai and Tachi (1991) and Papadopoulos and Dubowsky (1991), we assume that there is enough dynamic coupling between the passive and active joints, for it is clear that if this coupling is too small (or does not exist at all), it will be impossible to change the position of the passive joints by simply moving the active ones. Consider for example a 3-link Cartesian manipulator; theoretically, there is no coupling between the links, and a passive joint cannot be controlled using the active ones.

Because the control of this kind of mechanism is fully dependent on its dynamic model, modelling must be accurate. The fact that the performance depends on accurate modelling can be understood in terms of the following rationale: first, the control scheme depends on modelling accuracy, and thus modelling errors can result in tracking errors and instability. Second, the coupling between the active and the passive joints depends on the dynamic parameters, and is subject to errors if there are uncertainties on the values of these parameters. Third, one usually specifies the desired motion of the end-effector in Cartesian space, and maps this motion to joint space, where control is executed. This mapping now is related to the dynamic parameters, and becomes uncertain if some parameters are unknown. Last, for conventional robots a local PID scheme without a model-based feedforward controller may provide good trajectory tracking results. For underactuated manipulators, however, it is impossible to control the system with simple PID schemes, because of the coupling existent between the joints.

On the other hand, in most cases it is impossible to obtain an accurate dynamic model for the underactuated system. Joint friction and compliance are difficult to model; the tip weight/mass in transporting tasks is usually unknown; and the inertia of the entire manipulator assembly is usually inaccurate because of the lack of experimental methods to determine it.

Because modelling error is critical to the system's performance, and generally it is difficult to obtain an accurate dynamic model, the goal of this research is to propose a controller that is robust to modelling errors and uncertainties. We present in this paper a variable structure controller for the underactuated system, both in joint space and Cartesian space. The joint space control methodology uses the dynamic coupling between the passive joints and the active joints, and controls the active ones in order to bring the passive joint angles to a desired set-point. After the passive joints reach the set-point, they can be locked, and the active ones can be controlled to their

desired position. The Cartesian space control method uses the dynamic coupling to control a subset of the Cartesian space variables. We address the computational procedures, and present simulation and experimental results of the proposed schemes.

2 Joint Space Modelling

The dynamic modelling of underactuated manipulators differs little from the modelling of fully-actuated ones, the only difference being that the torque applied at the passive joints is constant and equal to zero. Using the Lagrangian formulation, one can obtain the following dynamic description of an open-chain mechanical manipulator, whose links are considered as rigid bodies:

$$\tau = M(q)\ddot{q} + b(q, \dot{q}) \tag{1}$$

In (1), *M(q)* is the $n \times n$ inertia tensor matrix of the manipulator, and $b(q, \dot{q})$ is the $n \times 1$ vector representing Coriolis, centrifugal and gravitational torques. The vector τ represents the torques applied at the active joints, and it is a $n \times 1$ vector with *p* components equal to zero.

One of the important issues when dealing with underactuated manipulators is to correctly perform a partition of the dynamic equation (1). Such a partition is important to show the coupling between the passive and the active joints, and it is given as follows:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{matrix} r \\ p \end{matrix} \begin{bmatrix} M_{aa} & M_{ap} \\ M_{pa} & M_{pp} \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_p \end{bmatrix} + \begin{bmatrix} b_a \\ b_p \end{bmatrix} \tag{2}$$
$$\begin{matrix} & r & p \end{matrix}$$

Here, the subscript *a* is used to represent quantities relative to the active joints, whereas *p* represents those for the passive joints. It is important to note that the inertia matrix *M* in (2) is not equal to the original inertia matrix in (1). This new inertia matrix is obtained from the original one via a sequence of elementary operations that group the joint angles corresponding to q_a and q_p. Nevertheless, it still possesses the important properties of symmetry and positive-definiteness (Bergerman et al., 1994).

Equation (2) is very useful in the understanding of the dynamic coupling between the passive and active joints of an underactuated manipulator. Namely, from the second line of (2), we can find out the relationship between $\ddot{q}_a$ and $\ddot{q}_p$:

$$M_{pa}\ddot{q}_a + M_{pp}\ddot{q}_p + b_p = 0 \tag{3}$$

Since this type of manipulator can only produce torque at the active joints, and thus control $\ddot{q}_a$ directly, equation (3) shows that we can control $\ddot{q}_p$ indirectly, as long as the submatrices M_{pa} and M_{pp} have a structure that "allows" torque to be transmitted in a desired way from the active to the passive joints. In this work, we will assume that this transmission is always possible, i.e., that there is enough dynamic coupling to drive the passive joints. In order to quantify the dynamic coupling before operating the underactuated system, one can use the concept of *coupling index (*Bergerman et al., 1994). In a few words, this index measures either locally or globally the dynamic coupling existent between the active and the passive joints. More details about the coupling index can be found in the reference above.

As was demonstrated by Arai and Tachi (1991) and Bergerman et al. (1994), the number of joints that can be controlled at any instant is equal to the number of active joints. Since we are assuming that $r \geq p$, at the beginning of the operation we can control all p passive joints (via dynamic coupling) and $r - p$ active ones. These r joints to be controlled are grouped in the vector q_c. When $r = p$, this vector contains only passive joints. For example, for the 3-link manipulator shown in Figure 1, with joints 1 and 2 active and joint 3 passive, q_c is a two-dimensional vector containing the passive joint q_3 and one active joint. We opted here to stack the passive joints at the bottom of the vector q_c, and to choose the active joints closer to the base to be controlled first. With this rationale, the choice of q_c for the manipulator in Figure 1 is:

$$q_c = \begin{bmatrix} q_1 & q_3 \end{bmatrix}^T \tag{4}$$

Since we can control r joints at a time, the partitioned equations represented by (2) are not very useful, since they do not show explicitly which joints are grouped into q_c. To this end, we consider a second possible partition of (1), to be used in the control algorithm:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{matrix} r \\ p \end{matrix} \begin{bmatrix} M_{ad} & M_{ac} \\ M_{pd} & M_{pc} \\ p & r \end{bmatrix} \begin{bmatrix} \ddot{q}_d \\ \ddot{q}_c \end{bmatrix} + \begin{bmatrix} b_a \\ b_p \end{bmatrix} \tag{5}$$

Naturally, the vector q_d, where the index d stands for "driving joints", as opposed to c, which stands for "controlled joints", contains p active joints to be controlled after the joint angles of the joints q_c reach their set-point.

3 Cartesian Space Modelling

In order to perform control in Cartesian space, it is necessary to obtain a dynamic model relating the Cartesian space variables' accelerations to the torques applied at the active joints. A natural way is to begin with the dynamic model of the manipulator as given by the Lagrangian formulation, and then use kinematic relationships to map joint space quantities into Cartesian space quantities.

From (2) we can write:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{matrix} r \\ p \end{matrix} \begin{bmatrix} M_a \\ M_p \end{bmatrix} \ddot{q} + \begin{bmatrix} b_a \\ b_p \end{bmatrix} \tag{6}$$

Since control is to be performed in Cartesian space, and the torques commanding the manipulator are applied at the joints, a forward kinematic analysis must be performed in order to map the motion in joint space to that in Cartesian space. This mapping can be described by the Jacobian matrix that transforms velocities in joint space, $\dot{q}$, into velocities in Cartesian space, $\dot{x}$:

$$\dot{x} = J(q)\dot{q} \tag{7}$$

The Cartesian space variables are represented by the $n \times 1$ vector x. Note that, since x and q have the same dimensions, we are considering non-redundant manipulators. Computing the time derivative of (7), we can obtain an expression for the acceleration of the joint angles as:

$$\ddot{q} = K\ddot{x} - K\dot{J}\dot{q} \tag{8}$$

where $K(q)$ represents the inverse of the Jacobian matrix.

Since the underactuated manipulator has only r actuators, it is possible to control r Cartesian variables at a time. These variables will be called the *active Cartesian variables*, x_a. The other p ones cannot be controlled, and will be called the *passive Cartesian variables*, x_p. The vector x can then be partitioned as $x = [x_a^T \;\; x_p^T]^T$. Similarly, the matrix K can be partitioned as $K = [K_a \;\; K_p]$. Equation (8) can then be rewritten as:

$$\ddot{q} = K_a\ddot{x}_a + K_p\ddot{x}_p - K\dot{J}\dot{q} \tag{9}$$

Substitution of (9) into (6) provides the relationship between Cartesian accelerations and joint torques:

$$\tau_a = M_a K_a \ddot{x}_a + M_a K_p \ddot{x}_p - M_a K \dot{J} \dot{q} + b_a \tag{10}$$

$$) = M_p K_a \ddot{x}_a + M_p K_p \ddot{x}_p - M_p K \dot{J} \dot{q} + b_p \tag{11}$$

Recalling that only x_a can be controlled, while x_p cannot, we can isolate $\ddot{x}_p$ in (11) and substitute the result in (10). The final result is:

$$\tau_a = [M_a - M_a K_p (M_p K_p)^{-1} M_p](K_a \ddot{x}_a - K \dot{J} \dot{q}) - M_a K_p (M_p K_p)^{-1} b_p + b_a \tag{12}$$

Equation (12) is the desired relationship between the controlled variables x_a and the torque inputs to the mechanism, τ_a. In the ideal case, one should simply plug in the desired value for $\ddot{x}_a$ into (12) and compute the necessary joint torques to obtain this acceleration. In the real cases, however, imperfections in the dynamic model, measurement noise and other sources of error can make the system perform poorly, and even become unstable. For this reason, it is important to devise a feedback controller that will make $\ddot{x}_a$ to follow a desired $\ddot{x}_a^d$, despite the error sources mentioned.

4 Robust Control Design

4.1 Variable Structure Controller

In this section we develop a variable structure controller (VSC) that will guarantee convergence of the controlled variables—joint angles or Cartesian variables—to a desired set-point, despite possible modelling errors.

The idea behind the VSC is to force the system's state trajectory to converge to a pre-defined surface in the state space. Once the system reaches this surface, it will "slide" along it to the origin; this is the reason for this surface being called *sliding surface*. Once the sliding takes place, the dynamics of the system are described by the equation of the sliding surface and not by its original ones. Thus, modelling errors do not affect the system's performance after the sliding begins. Naturally, two aspects are important in this class of controllers: first, the sliding surface must be designed according to the desired system's performance once the sliding starts. Linear surfaces are the most common, given their design simplicity; for a second order system they are represented by:

$$s = \Gamma \tilde{x} + \dot{\tilde{x}} \tag{13}$$

where $\tilde{x} = x^d - x$ is the error between the desired and the current value of the state vector *x*, and Γ is a (generally diagonal) gain matrix.

Second, a control law must guarantee that the sliding surface is reachable, and that the time it takes for the state trajectory to reach it is finite. This requirement is guaranteed in a region "close" to the sliding surface if, in this region,

$$s\dot{s} < 0 \tag{14}$$

In this paper, we use the *constant rate reaching law* (Hung et al., 1993) in order to guarantee that (14) is satisfied:

$$\dot{s} = -P\,\mathrm{sgn}(s) \tag{15}$$

Equation (15) describes the dynamic behavior of the system while the state trajectories are reaching for the sliding surface. With this choice, the acceleration of the state variables can be found directly from the time derivative of (13):

$$\begin{aligned} \dot{s} &= \Gamma\dot{\tilde{x}} + \ddot{x}^d - \ddot{x} \\ \Rightarrow \ddot{x} &= \Gamma\dot{\tilde{x}} + \ddot{x}^d + P\,\mathrm{sgn}(s) \end{aligned} \tag{16}$$

We will call $\ddot{x}$ above the *required acceleration* of the state variables and denote it by $\ddot{x}^r$.

4.2 Joint Space Controller Design

Variable structure controllers have been applied to standard manipulators, and successful results have been reported regarding trajectory tracking and disturbance rejection—see, for example, Bailey and Arapostathis (1987), Bergerman and Cruz (1994), and Morgan and Özgüner (1985). As we mentioned previously, the dynamic modelling error is more critical for underactuated manipulators than it is for conventional ones. The design of a controller that is robust to modelling errors and disturbances is of significance in practice. Here, we propose a variable structure controller for the underactuated manipulator. In what follows, we will first discuss the controller design in joint space, then that in Cartesian space.

Differently from fully-actuated manipulators, underactuated ones cannot have all their joint accelerations controlled at every time step. Since they have only *r* actuators, it is possible to control *r* accelerations at a time. The other *p* accelerations will depend on the *r* controlled ones. To see this, from the second line of (5) we have:

$$\ddot{q}_d = -M_{pd}^{-1}(M_{pc}\ddot{q}_c + b_p) \tag{17}$$

Thus, if we try to control $\ddot{q}_c$, then $\ddot{q}_d$ is fixed and cannot be arbitrarily chosen. In order to control $\ddot{q}_c$, we define the following *r*-dimensional sliding surface:

$$s_c = \Gamma_c \tilde{q}_c + \dot{\tilde{q}}_c \tag{18}$$

where Γ_c is an $r \times r$ matrix containing the time-constants of each one-dimensional surface. The required acceleration of q_c for the sliding surface to be reached is, according to (16):

$$\ddot{q}_c^r = \Gamma_c \dot{\tilde{q}}_c + \ddot{q}_c^d + P_c \operatorname{sgn}(s_c) \tag{19}$$

where P_c is an $r \times r$ matrix. In order to obtain this required acceleration, we substitute $\ddot{q}_c$ for $\ddot{q}_c^r$ in the first line of (5):

$$\tau_a = (M_{ac} - M_{ad} M_{pd}^{-1} M_{pc}) \ddot{q}_c^r - M_{ad} M_{pd}^{-1} b_p + b_a \tag{20}$$

The torque given by (20) guarantees that $\ddot{q}_c = \ddot{q}_c^r$, therefore the joints grouped in q_c will converge to their desired positions. At this point, one has several options for the braking sequence of the passive joints and controlling of the system, namely:

- brake each passive joint as soon as they reach their set-points with zero velocity; wait until all passive joints are braked *and* the $r - p$ active joints in q_c also reach their set-point; control the remaining joints in q_d;

- brake each passive joint as soon as they reach their set-points with zero velocity; wait until all passive joints are braked; switch to a new control law in order to bring *all* active joints to their desired set-points;

- brake each passive joint as soon as they reach their set-points with zero velocity; after every passive joint is braked, switch to a new control law that includes one more active joint into q_c in substitution for the braked joint.

The first method provides the slowest response, while the third is the most complicate to implement. It is natural, thus, to choose the second one, which is faster than the first and simpler than the third. To implement the second method, we note that after all passive joints are braked, the new dynamic equation of the system is simply:

$$\tau_a = M_{aa} \ddot{q}_a + b_a \tag{21}$$

where M_{aa} is an $r \times r$ matrix. Following the same reasoning as above, the design of a VSC that will bring all the active joints to their desired set-points is straightforward. Namely, we define a sliding surface $s_a = \Gamma_a \tilde{q}_a + \dot{\tilde{q}}_a$ and use $\ddot{q}_a^r = \Gamma_a \dot{\tilde{q}}_a + \ddot{q}_a^d + P_a \operatorname{sgn}(s_a)$ in (21). The design is complete and all joints are guaranteed to converge to their desired position after a finite time.

It should be mentioned here that the control laws (20) and (21) introduce an undesired chattering into the system, because of the *sgn(.)* function used. This problem can be solved with the addition of a boundary layer of width ε around the sliding surface. For this sake, we substitute the function *sgn(.)* for:

$$\mathrm{sgn}(x) \rightarrow \begin{cases} \mathrm{sgn}(x) & \text{if } x \geq \varepsilon \\ \dfrac{x}{\varepsilon} & \text{if } x < \varepsilon \end{cases} \tag{22}$$

4.3 Cartesian Space Controller Design

In this section we extend the design of the joint space VSC to the control of variables in Cartesian space. Define the following *r*-dimensional sliding surface:

$$s_a = \Gamma_a \tilde{x}_a + \dot{\tilde{x}}_a \tag{23}$$

where Γ_a is an $r \times r$ matrix with the time-constants of each one-dimensional sliding surface. As before, we substitute

$$\ddot{x}_a^r = \Gamma_a \dot{\tilde{x}}_a + \ddot{x}_a^d + P_a \mathrm{sgn}(s_a) \tag{24}$$

in (12) in order to guarantee that x_a converges to its desired set-point. Since the sliding surface considered here also takes into account the error in the active components' velocities, this controller allows x_a to follow a time-varying trajectory.

5 Simulation Results

In this section we present a series of simulations with the objective of demonstrating the robustness of the VSC against disturbances and parametric uncertainties. We introduce a multiplicative error δ in the dynamic parameters so that the control laws make use of estimated parameters which are equal to δ times the true ones. In addition, the simulated robot has a frictional component unknown to the controllers.

We first present the results of control in joint space; the objective here is to drive both active and passive joints of the robot from an initial configuration to a given set-point. As explained in section 2, first all passive joints and *r* - *p* active ones will be controlled. After the passive joints reach their set-point they are locked, and the active joints are controlled. We chose to brake the passive joints whenever they reached a joint angle error of less than 0.0015 *rd* (or 0.086º), with a joint velocity

of less than 0.001 *rd/s*. This ensures that the passive joints are braked at a point where they are practically at rest and with negligible steady-state error. In the sequence, we will present the results of control in Cartesian space, where the objective is to make a subset of the Cartesian variables follow a desired time-varying trajectory.

5.1 Joint Space Control

We present here simulations for 2- and 3-link planar rotary manipulators operating on a vertical plane, as shown in Figures 1 and 2. The values in Table 1 were adopted for the dynamic parameters of the manipulators. The symbols m_i, I_i, l_i and l_{c_i} represent, respectively, the *i*-th link's mass, inertia, length and distance from the *i*-th joint to the center of gravity of link *i*.

Table 1: Numerical values of the dynamic parameters.

n	m_1 (Kg)	m_2 (Kg)	m_3 (Kg)	I_1 (Kg m^2)	I_2 (Kg m^2)	I_3 (Kg m^2)	l_1 (m)	l_2 (m)	l_3 (m)	l_{c_1} (m)	l_{c_2} (m)	l_{c_3} (m)
2	1.06	0.29	-----	0.031	0.020	-----	0.26	0.29	-----	0.12	0.25	-----
3	1.06	1.06	0.29	0.031	0.031	0.020	0.26	0.26	0.29	0.12	0.12	0.25

Table 2 summarizes the set of simulations, along with the gains used at each one; Table 3 presents the results obtained in each case. Note that the steady-state error is negligible in all cases.

Table 2: Summary of simulations performed—Joint space control.

SIMULATION	*n*	δ	Γ_c (s^{-1})	Γ_a (s^{-1})	P_c (s^{-1})	P_a (s^{-1})	ε_c (rd/s)	ε_a (rd/s)
J1	2	1.0	12	12	25	25	0.1	0.1
J2	2	1.3	12	12	25	25	0.1	0.1
J3	3	1.3	$\begin{bmatrix} 3 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 100 \\ 0 & 90 \end{bmatrix}$	$\begin{bmatrix} 100 & 100 \\ 0 & 120 \end{bmatrix}$	0.1	0.1

The first simulation consisted of driving the joints over a 135^o and 90^o excursion, respectively. Note in Figure 3 that the active joint initially moves up in order to bring the passive joint to its desired set-point with zero velocity. This is achieved at t = 1.39 s, when the brake is engaged. From this point on, the active joint is self-controlled and reaches its set-point with zero velocity after a total of 3.51 s. In this first simulation the only disturbance present was the frictional torque.

Table 3: Summary of results obtained—Joint space control.

SIMULATION	Initial angles (º)	Set-point (º)	Steady-state errors (º)	Brake applied at (s)	Figure
J1	$\begin{bmatrix} -45 \\ 90 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.000 \\ -0.005 \end{bmatrix}$	1.39	3
J2	$\begin{bmatrix} -45 \\ 90 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.000 \\ -0.053 \end{bmatrix}$	1.06	4
J3	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 45 \\ 45 \end{bmatrix}$	$\begin{bmatrix} 0.032 \\ 0.086 \\ 0.080 \end{bmatrix}$	1.22	5

In order to test the robustness of the VSC, the same simulation was performed using $\delta = 1.3$. Comparing the first and second lines of Table 3, and Figures 3 and 4, we can affirm that robustness to uncertainties (in this case, modelling errors) is guaranteed.

Simulation J3 was performed with the 3-link manipulator. Note in Figure 5 how the active joints act combined in order to bring the passive one to rest. After the passive joint is braked, the active ones converge smoothly and exponentially to their set-points. In this simulation also the parametric uncertainty was set to $\delta = 1.3$.

As a matter of comparison with previous works in this area, we repeated simulations J1 and J2 using the PID-based control law proposed by Arai and Tachi (1991). In order to have a fair comparison, we adopted the same saturation levels for the torque at the active joint. Figure 6 summarizes the results; it presents the state-space trajectory of the passive joint for both the VSC and the PID controller. We can see the sliding mode occurring when the VSC is used; we also see that when $\delta = 1.3$ the PID is not able to bring the passive joint to its set-point, thus not being as robust to the parametric uncertainty as the VSC.

If we compare our results with the ones presented by Mukherjee and Chen (1993), we can affirm that our formulation is much simpler, and that the braking and settling times are greatly reduced.

5.2 Cartesian Space Control

Here, we present a series of simulation studies for the Cartesian variables control. The system to be simulated is the one shown in Figure 2. We assume that the manipulator is lying on an horizontal plane, so that there will be no gravitational torques acting on the joints. The kinematic and dynamic parameters for this manipulator are the same as those used before, and are shown in Table 1.

We consider X as the active component, and Y the passive one. Note that the manipulator's first joint is active, while the second is passive. This choice maximizes the dynamic coupling between the joints, making it easier for the controller to transmit the torque from the active to the passive joints. The desired trajectory to be followed by the X coordinate is a sine wave with amplitude 0.136 m and period 1 s. As a performance index, we measured the maximum error $\tilde{x}_a$ as the manipulator tried to follow the desired path.

In order to present the robustness of the VSC, we will show results using both $\delta = 1.0$ and $\delta = 1.3$. Also, we provide a comparison of this method to the PID-based one presented by Arai et al. (1993). Table 4 presents a summary of the simulations performed, along with the gains used on each of them; and a summary of the results obtained. In order to perform a fair comparison, we tuned the PID gains so as to obtain the same level of torque as the one given by the VSC.

Table 4: Summary of simulations performed—Cartesian space control.

SIMULATION	δ	Γ_a (s^{-1})	P_a (s^{-1})	ε_a (rd/s)	$\max(\tilde{x}_a)$ (m)	Figure
C1	1.0	50.0	100.0	0.1	0.005	7
C2	1.3	50.0	100.0	0.1	0.005	8

As it can be seen in Figures 7 and 8, the active Cartesian variable tracks the desired trajectory with good precision. Figure 6 presents the tracking error for both the VSC and the PID, with and without parametric uncertainty. An analysis of this figure allows us to draw the following conclusions:

- For the same amount of torque, the maximum tracking errors for the VSC are about 5 times smaller than those for the PID.

- The robustness provided by the VSC is much greater than that provided by the PID. Although the PID provides low tracking error for $\delta = 1.0$, it lacks the ability to provide it when the parametric uncertainty is greater.

6 Experimental Results

In order to evaluate the feasibility of the proposed VSC controller, we built a simple 2-DOF planar manipulator, the first joint being active and the second passive. The manipulator lies on a horizontal plane, and so there are no gravitational torques acting on the links. The manipulator is shown in Figure 10, and schematically in Figure 11. We call the robot U-ARM, which stands for

UnderActuated Robot Manipulator. Its dynamic parameters were obtained via measurements and using the data available from the mechanical design. The parameters obtained are those shown in Table 1 for the case $n = 2$.

The control software was written in C, and ran under the real-time operational system Chimera 3.2, developed at CMU (Stewart et al., 1993). Chimera is a multitasking multiprocessor operating system, designed to support software written as a set of modules running independently. Modules share information about the system via a global state variable table. At the beginning of each cycle, modules read the current values of the variables they need from the global table. At the end of the cycle, the values are updated in the global table, so that modules always have access to the most up-to-date information. For U-ARM, the following modules were defined: *encoder* (reads the encoders on the joints of the robot and writes the current joint positions and velocities in the global state table); *actuator* (reads the value of torque applied by the controller and sends it to the motor); and *arm* (reads the current position and velocities of the robot, computes the torque at the active joint using the VSC, and writes this torque to the global table).

The experimental results shown here are for the joint control problem, similar to simulations J1 and J2. Table 5 presents the summary of the results obtained.

Table 5: Summary of experimental results obtained—Joint space control.

EXPERIMENT	Initial angles (º)	Set-point (º)	Steady-state errors (º)	Brake applied at (s)	Figure(s)
EJ1	$\begin{bmatrix} 0 \\ -88 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.057 \\ -0.172 \end{bmatrix}$	1.450	12
EJ2	$\begin{bmatrix} -37 \\ -53 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.057 \\ -0.343 \end{bmatrix}$	1.025	13
EJ3	$\begin{bmatrix} -50 \\ 88 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.057 \\ 0.000 \end{bmatrix}$	3.675	14

As it can be inferred from Figures 12 to 14, the actual behavior of the manipulator under the proposed VSC control scheme is similar to the ones shown in the simulations. Namely, the active joint brings the passive one to its desired set-point with practically zero velocity, and then converge to its own set-point smoothly along the sliding surface. One point to note is that the control of the passive joint in experiment EJ3 was slower than EJ1 and EJ2. This is due to the fact that the table where U-ARM is installed is not perfectly levelled. This creates a small but disturbing gravitational

torque that pulls the manipulator to one side. Nonetheless, the VSC was robust enough to cope with this problem and still control the system.

Another point that deserves comment is the fact that the motor in the first joint of U-ARM has a relatively big static friction coefficient. This friction was not compensated for in our control software, and was considered as a disturbance to be rejected by the controller. As we saw, the disturbance rejection effectively occurred.

7 Conclusion

In this work, the authors demonstrated the feasibility of designing a robust controller for underactuated manipulators. The control of such systems can be extended to the control problem of fault-tolerant robots, space robots and hyper-redundant robot systems, where one or more joints are passive, either because of design considerations or because of a failure. Given the strong dependency of the control system on the dynamic model, uncertainties in the model may result in inaccuracy and loss of stability.

The schemes proposed here consisted of a variable structure controller, used along with the theory of sliding surfaces. This control method makes the system's state trajectory slide over a pre-defined sliding surface in the phase plane, which in turn guarantees tracking and robustness properties. The main point in this work was the demonstration of the controller's robustness to parameter uncertainty. Another possible approach to cope with the uncertainties in the system would be via the use of adaptive control techniques, as done by Gu and Xu (1994).

We compared the proposed control law with the ones proposed by Arai and Tachi (1991), and Arai et al. (1993), and showed that our scheme provides the system with a much greater deal of robustness, an important characteristic in this kind of highly coupled nonlinear dynamic system.

8 Acknowledgments

We would like to thank Christopher Lee, graduate student at the Robotics Institute, CMU, for writing part of the control software in a way compatible with the operational system Chimera 3.2. We would also like to thank Ben Brown, project scientist at CMU, for designing the experimental mechanism.

This work was possible partially due to a grant from the Brazilian National Council for Research and Development (CNPq).

9 References

[1] Arai, H. and Tachi, S., 1991, Position control of a manipulator with passive joints using dynamic coupling, *IEEE Trans. on Robotics and Automation*, vol. 7, no. 4, pp. 528-534.

[2] Arai, H., Tanie, K. and Tachi, S., 1993, Dynamic control of a manipulator with passive joints in operational space, *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 85-93.

[3] Bailey, E. and Arapostathis, A., 1987, Simple sliding mode control scheme applied to robot manipulators, *Int. Journal of Control*, vol. 45, no. 4, pp. 1197-1209.

[4] Bergerman, M. and Cruz, J. J., 1994, Robust position control of mechanical manipulators, *3rd International Workshop on Advanced Motion Control*, Berkeley, CA.

[5] Bergerman, M., Lee, C. and Xu, Y., 1994, Dynamic coupling measure of underactuated manipulators, CMU-TR-RI-94-25, Carnegie Mellon University.

[6] Hung, J.Y., Gao, W. and Hung, J.C., 1993, Variable structure control: a survey, *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 2-22.

[7] Gu, Y.-L. and Xu, Y., 1994, Under-actuated robot systems: dynamic interaction and adaptive control, *Proceedings of the IEEE Systems, Man and Cybernetics Conference*.

[8] Jain, A. and Rodriguez, G., 1993, An analysis of the kinematics and dynamics of underactuated manipulators, *IEEE Trans. on Robotics and Automation*, vol. 9, no. 4, pp. 411-422.

[9] Lewis, C.L. and Maciejewski, A.A., 1994, Failure tolerant operation of kinematically redundant manipulators, *Conference on Intelligent Robotics in Field, Factory, Service, and Space*, pp. 837-841.

[10] Morgan, R.G. and Özgüner, Ü., 1985, A decentralized variable structure control algorithm for robotic manipulators, *IEEE Journal of Robotics and Automation*, vol. RA-1, no.1, pp. 57-65.

[11] Mukherjee, R. and Chen, D., 1993, Control of free-flying underactuated space manipulators to equilibrium manifolds, *IEEE Trans. on Robotics and Automation*, vol. 9, no. 5, pp. 561-570.

[12] Oriolo, G. and Nakamura, Y., 1991, Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators, *Proc. of the 30th Conference on Decision and Control*, pp. 2398-2403.

[13] Papadopoulos, E. and Dubowsky, S., 1991, Failure recovery control for space robotic systems, *Proceedings of the 1991 American Control Conference*, vol. 2, pp. 1485-1490.

[14] Saito, F., Fukuda, T. and Arai, F., 1994, Swing and locomotion control for a two-link brachiation robot, *IEEE Control Systems Magazine*, Feb. 1994, pp. 5-12.

[15] Stewart, D., Volpe, R. and Khosla, P., 1993, *Design of dynamically reconfigurable real-time software using port-based objects*, CMU-TR-RI-93-11, Carnegie Mellon University.

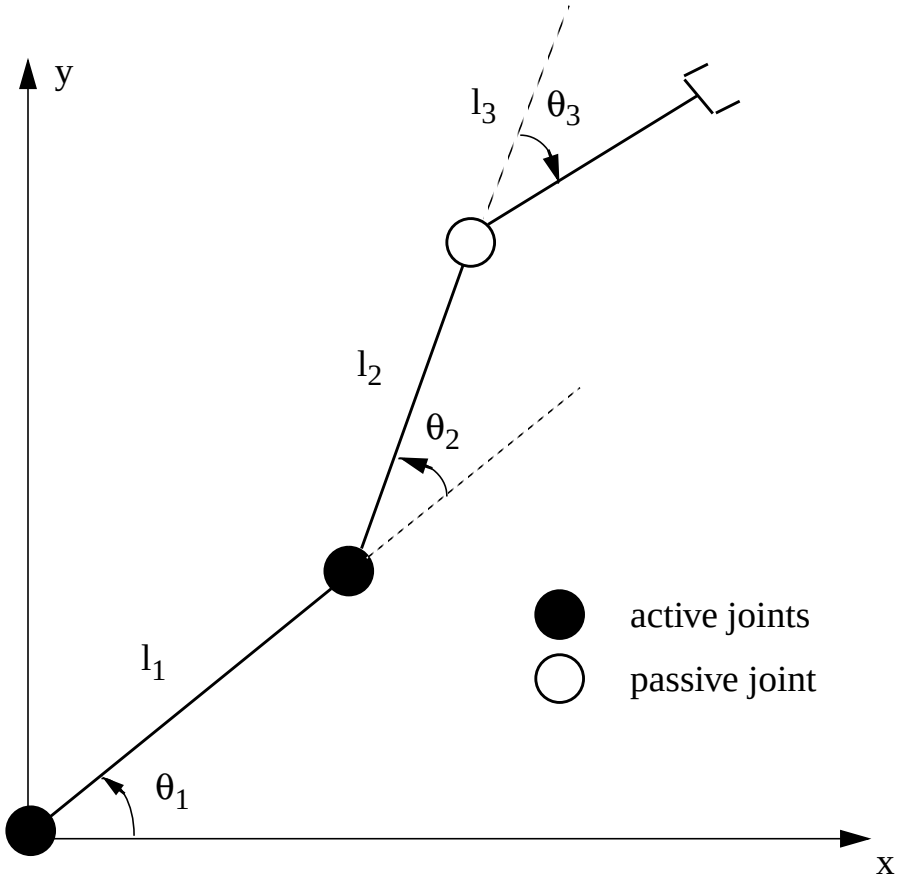

Figure 1: Three-link manipulator with one passive joint.

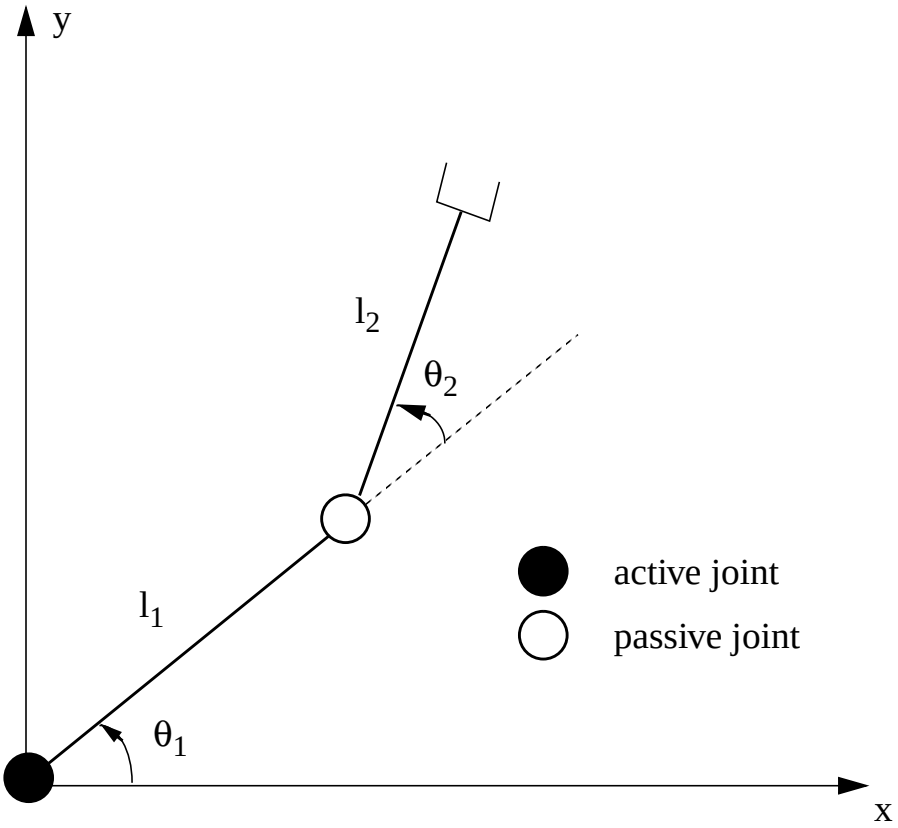

Figure 2: Two-link manipulator with one passive joint.

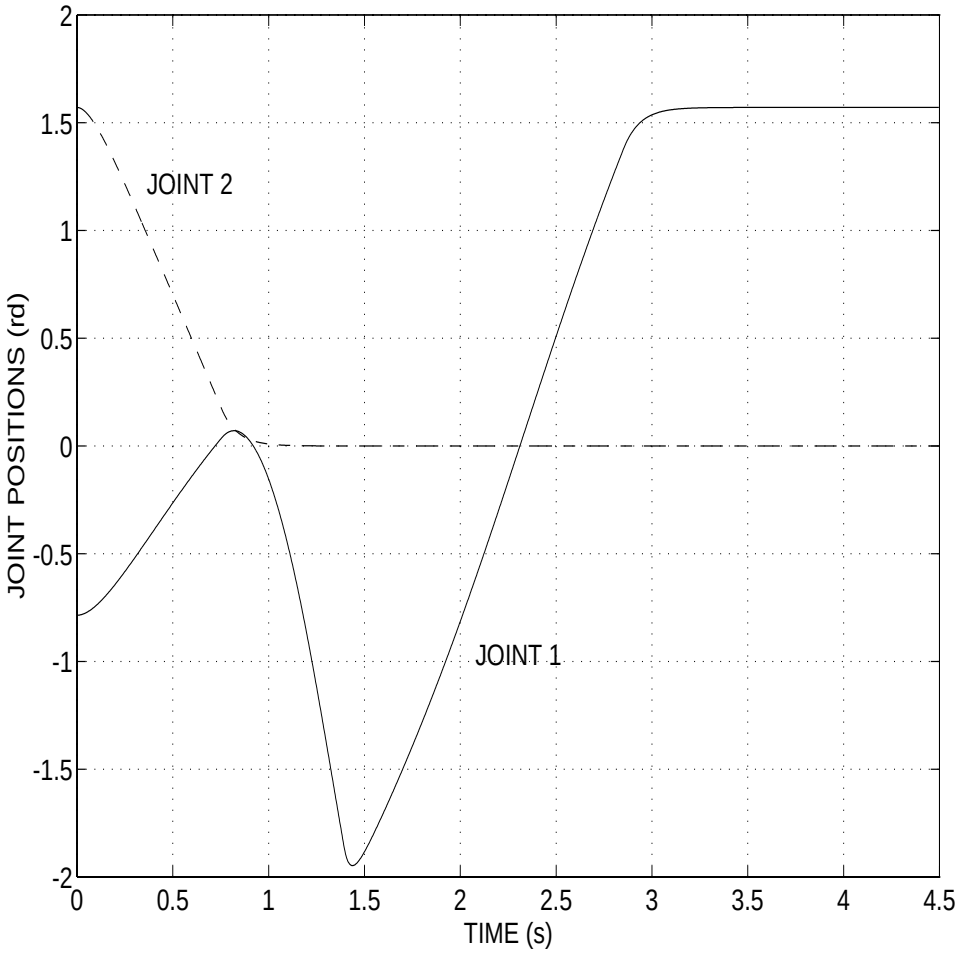

Figure 3: Response of the 2-link manipulator. Simulation J1.

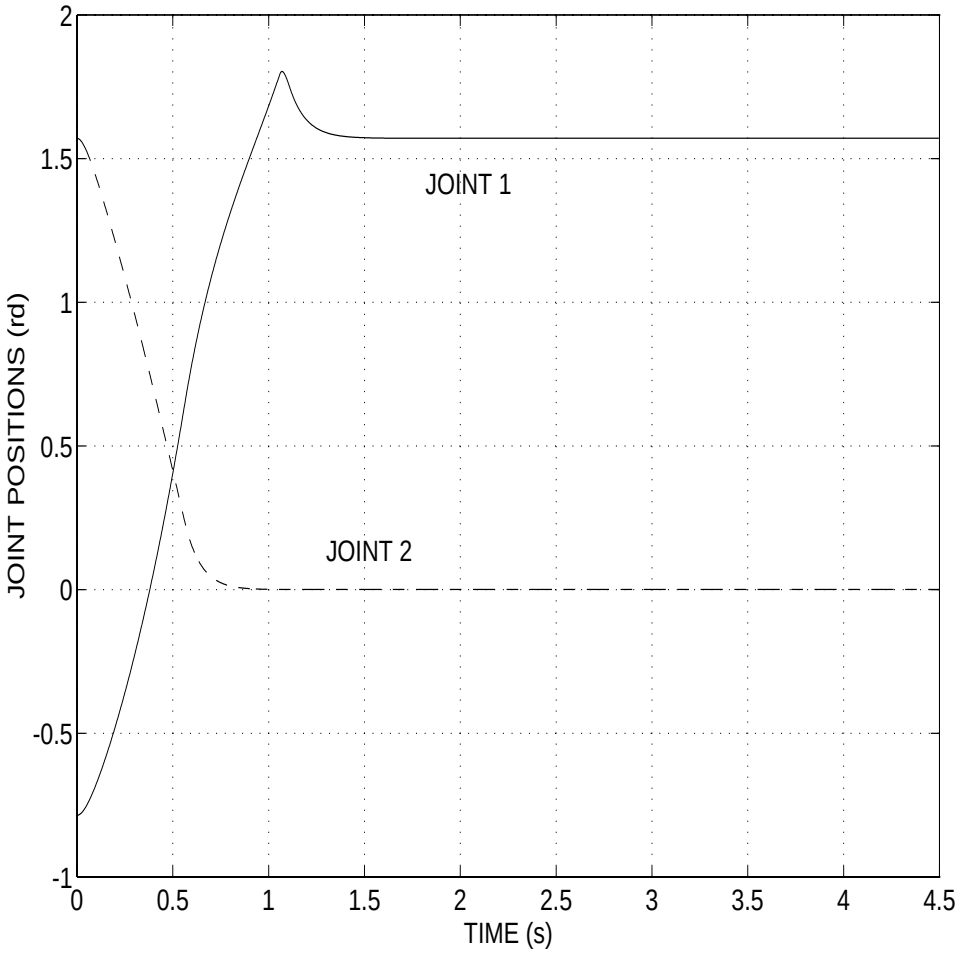

Figure 4: Response of the 2-link manipulator. Simulation J2.

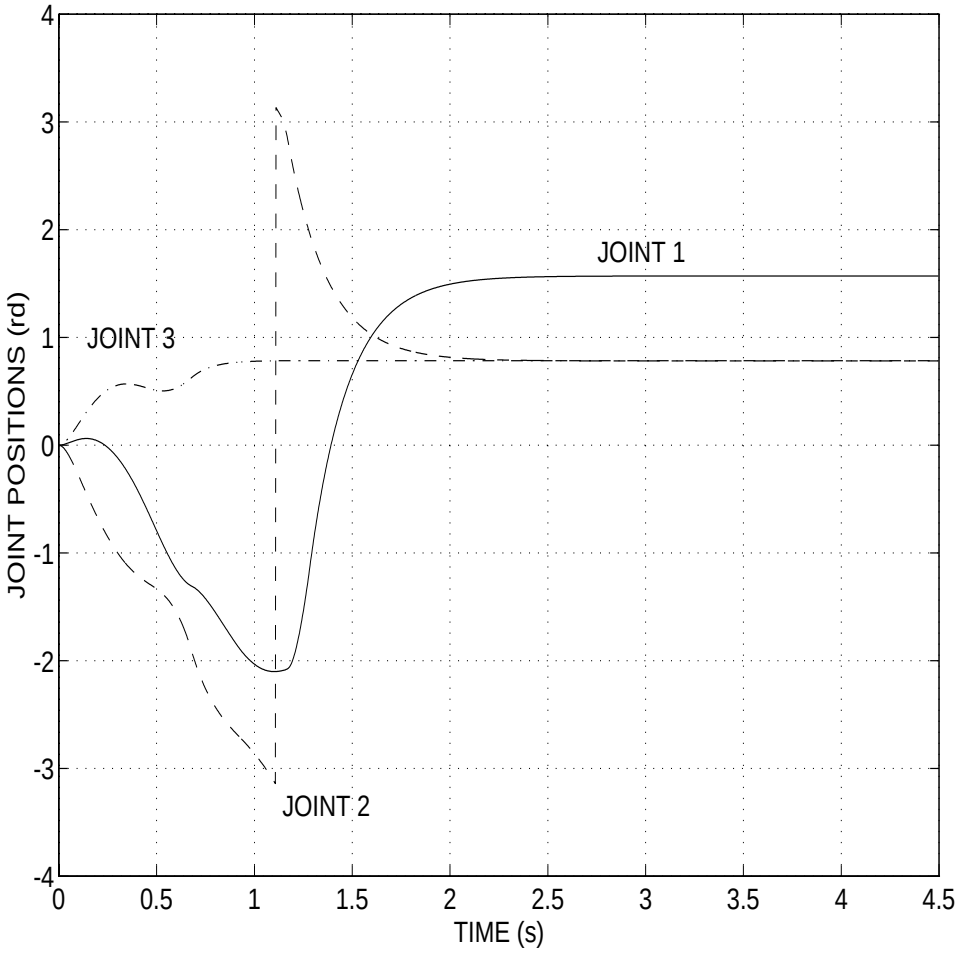

Figure 5: Response of the 3-link manipulator. Simulation J3.

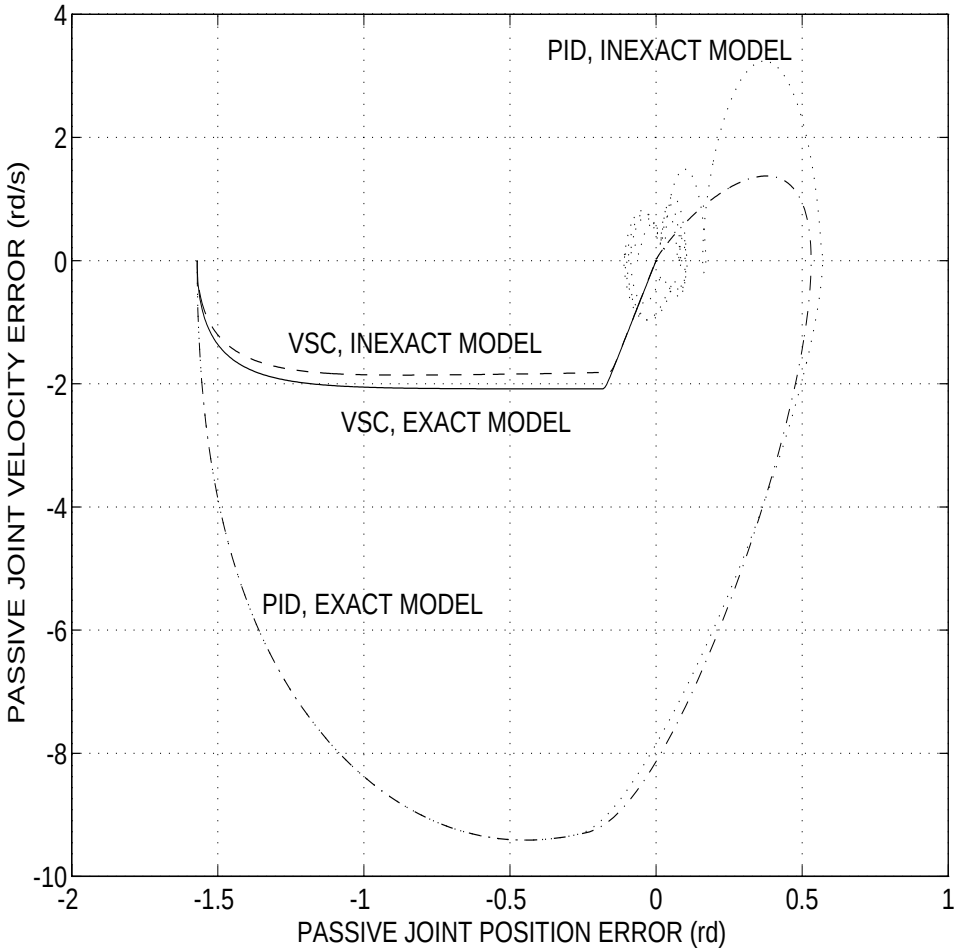

Figure 6: Simulations J1 and J2: comparison of the passive joint's state trajectory under the VSC and the PID.

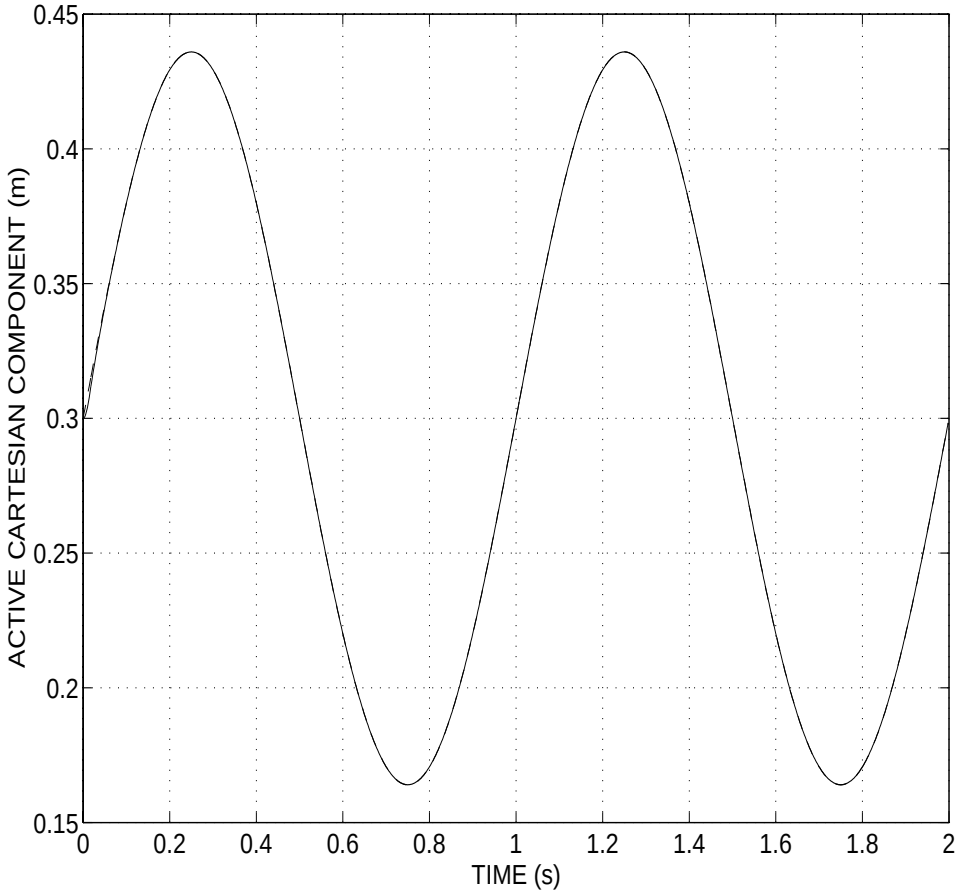

Figure 7: Response of the active Cartesian variable. Simulation C1.

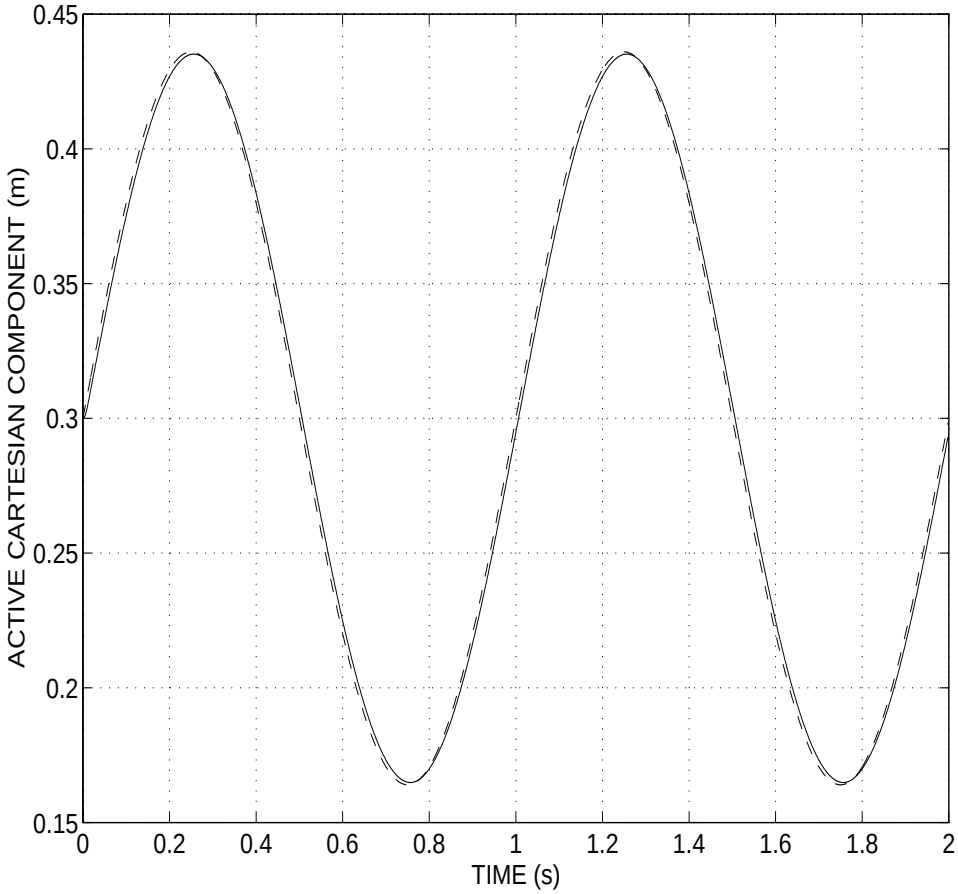

Figure 8: Response of the active Cartesian variable. Simulation C2.

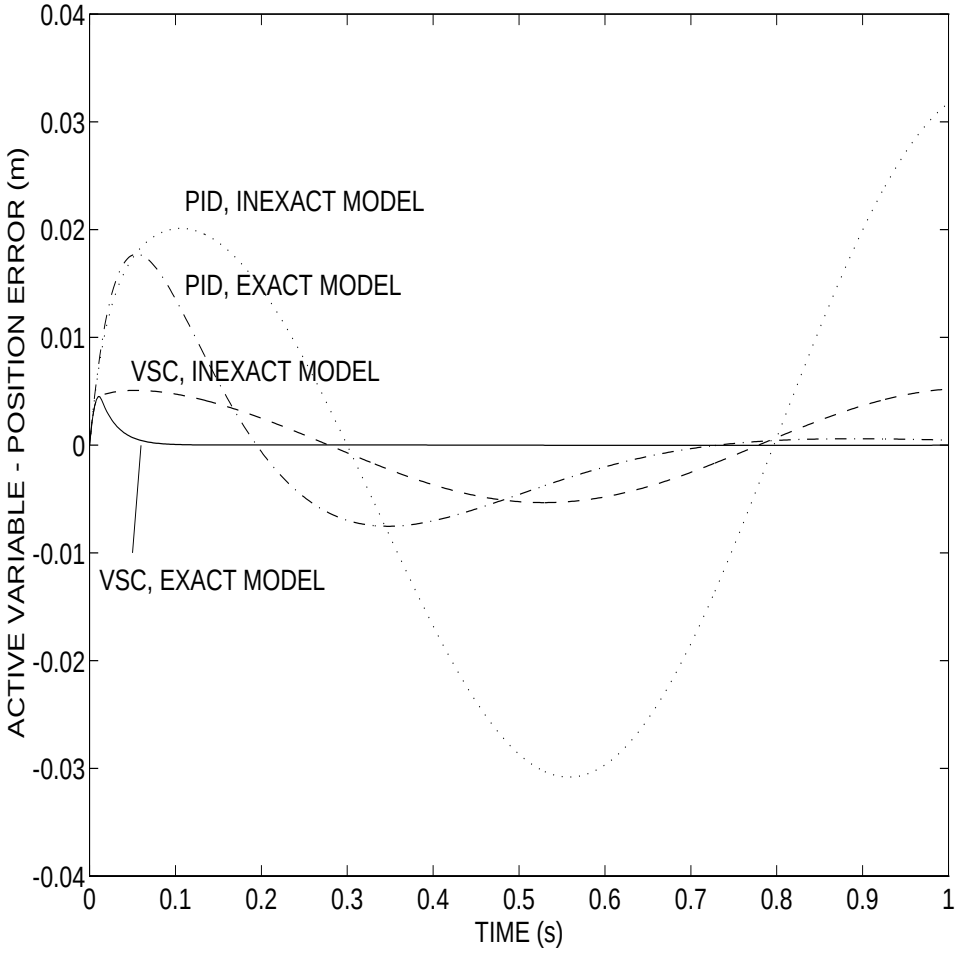

Figure 9: Simulations C1 and C2: comparison of tracking error between the VSC and the PID.

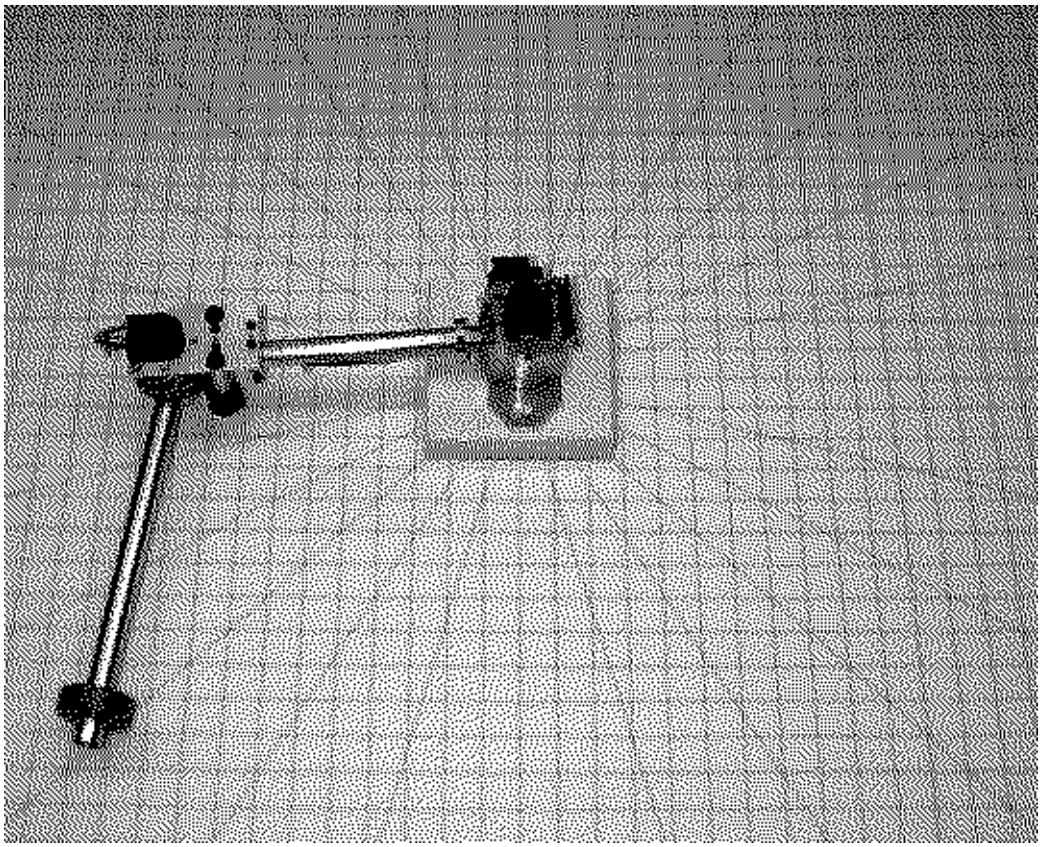

Figure 10: Planar 2-DOF underactuated manipulator: U-ARM.

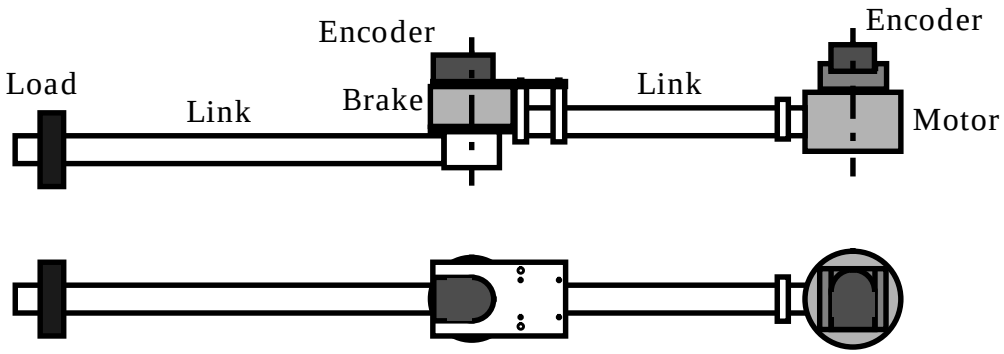

Figure 11: Schematic representation of U-ARM.

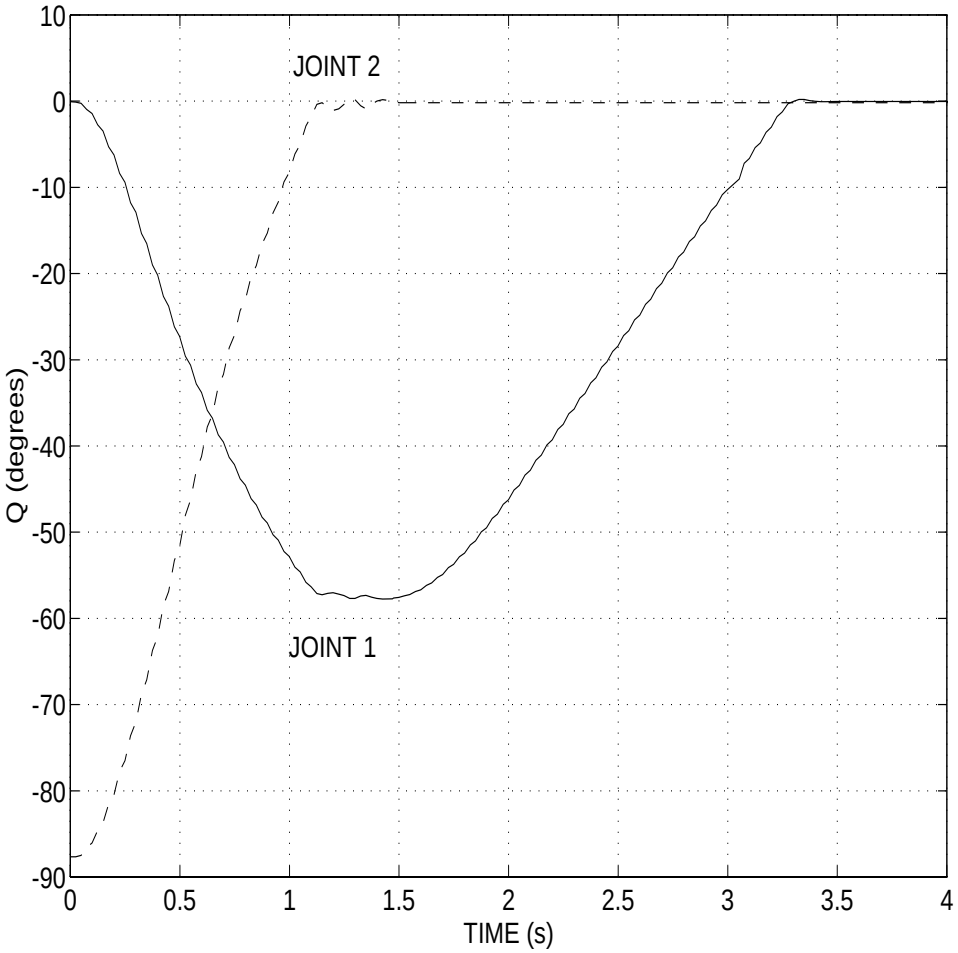

Figure 12: Experimental response. Experiment EJ1.

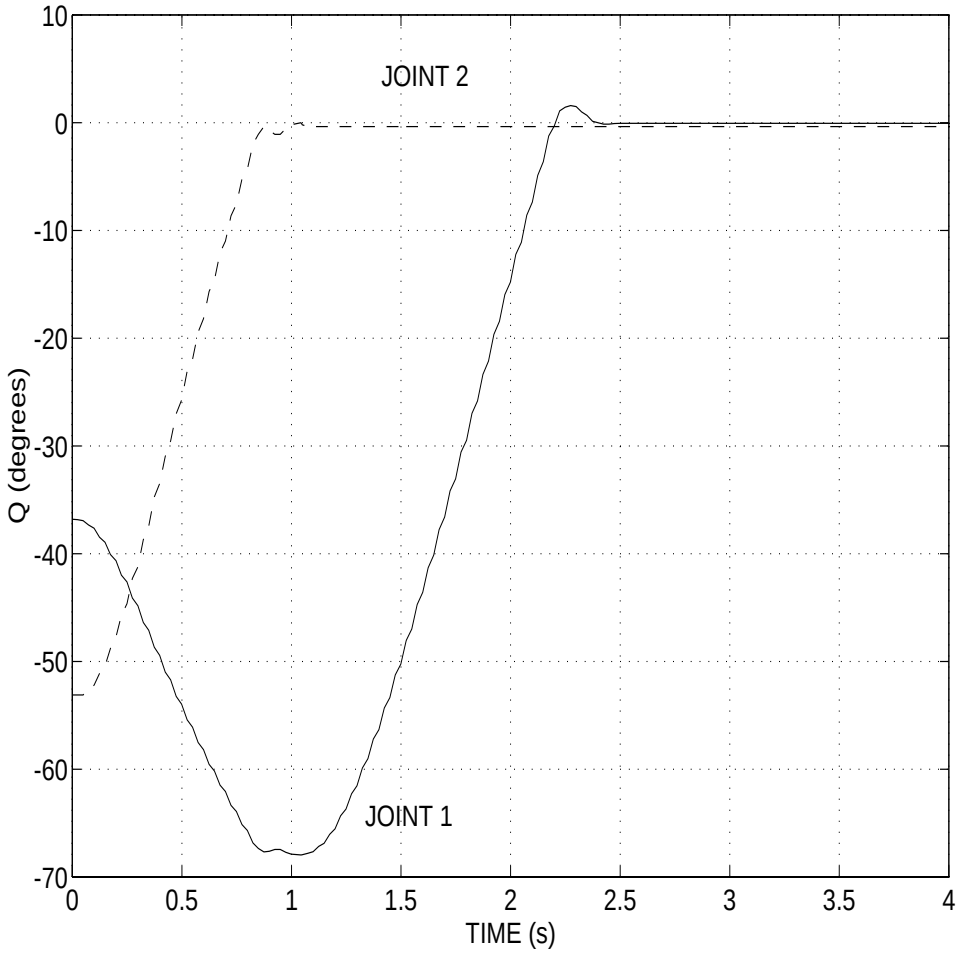

Figure 13: Experimental response. Experiment EJ2.

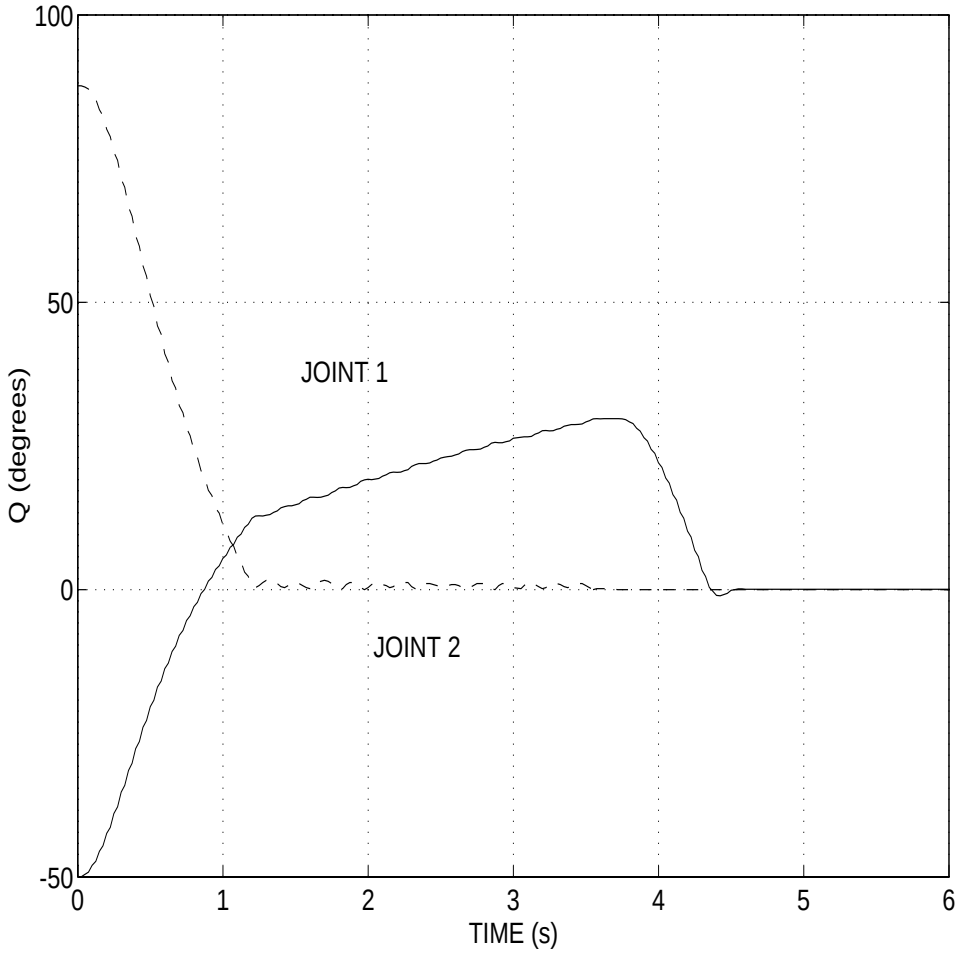

Figure 14: Experimental response. Experiment EJ3.