ROBUST CONTROL OF UNDERACTUATED MANIPULATORS: ANALYSIS AND IMPLEMENTATION

Marcel Bergerman Yangsheng Xu

CMU-RI-TR-94-12

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

May, 1994

This research is partially sponsored by the Brazilian National Council for Research and Development (CNPq). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of CNPq or Carnegie Mellon University.

Table of Contents

List of Figures

List of Tables

Abstract

Underactuated manipulators are robot manipulators composed of both active and passive joints. The advantages of using such systems reside in the fact that they weight less and consume less energy than their fully-actuated counterparts, thus being useful for applications such as space robotics. Another interest reside in the reliability or fault-tolerant design of fully-actuated manipulators. If any of the joint actuators of such a device fails, an entire operation may have to be aborted because of the loss of one or more degrees of freedom. The methodology proposed in this paper 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. Therefore, the control law and the performance of the system are completely dependent on the dynamic model. Since it is difficult to obtain the exact dynamic model of the system in general, considerable position errors and even instability can result in some cases. In this paper, we propose a variable structure controller to provide the system with the robustness necessary to perform tasks regardless of the modelling errors. Case studies are provided as a mean of illustration.

1 Introduction

In this work, the authors deal with the problem of robust position control of underactuated manipulators. Here, the word *underactuated* refers to the fact that the manipulator has less actuators than joints. The advantages of using such systems reside in the fact 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. Another interest reside in 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 usual in this situation to simply brake the failed joint and try to resume the task with less degrees of freedom available [8]. Following the methodology proposed in this work, the passive (failed) joint can still be controlled via the dynamic coupling with the active joints, and so the system can still make use of all of its degrees of freedom originally planned. For example, in the extreme case of a two DOF manipulator performing pick-and-place operations, the loss of one DOF would be fatal for the planned task. The results in section 5, however, show that it is possible to control the failed joint and continue with the task successfully.

Not until recently researchers started to work with the control problem of underactuated manipulators. In [1], Arai and Tachi proved that, at least 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. They also developed a controller which is very similar to the classical computed torque controller, to bring all joints to their desired set-points. The drawback existent in this approach is that a very accurate model of the manipulator must be provided to the controller. The approach by Papadopoulos and Dubowsky [12], considered the failure recovery control of a space robotic manipulator. In this work also the authors used a computed torque controller, which requires an accurate model of the system. Oriolo and Nakamura [11] 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 PD controller is able to bring the system to a stable configuration over an equilibrium manifold. They also present a detailed consideration about the conditions for integrability of the nonholonomic constraint present in underactuated manipulators.

Finally, Jain and Rodriguez [7] 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.

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 [1], our method requires that at least half of the joints be actuated. Also following the assumptions in [1], [12], we assume that there is enough dynamic coupling between the passive and the 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 Cartesian 3-link manipulator. Theoretically, there is no coupling at all between the links, and a passive joint could never be controlled using the active ones.

The control methodology proposed in this paper 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 are braked, and the active ones can be controlled to their desired position via any of a series of controllers fully developed in the literature for mechanical manipulators. Because the control of such a system 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 between the joints.

Because modelling error is so critical to the system's performance, and because there has not been much work addressing this issue, we present a variable structure controller in order to provide the system with the robustness necessary to perform tasks regardless of the modelling errors.

We address the computational procedures, simulation results and implementation issues of the proposed scheme. In comparison to other approaches, this work demonstrates robustness to dynamic parameters uncertainty and efficiency in implementation.

2 Model Partition

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. By using either the Newton-Euler method or the Lagrangian formulation [4], one can easily 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* x *n* inertia tensor matrix of the manipulator, and $b(q, \dot{q})$ is the *n* x 1 vector representing Coriolis, centrifugal and gravitational torques. The vector τ represents the torques applied at the active joints, and it is a *n* x 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:

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

The submatrices of *M* in (2) receive their indexes according to the variables they relate. For example, M_{pa} relates the (null) torques at the passive joints to the acceleration of the active ones. The same reasoning holds for the other three submatrices. This partition is very useful to understand 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 $\ddot{q}_p$:

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

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. Current work is being done on the characterization of this transmission, with a possible extension to optimal actuator placement.

As was demonstrated in [1], the number of joints that can be controlled at any moment is equal to the number of active joints. Thus, 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 then 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. At this point, one could choose either q_1 or q_2 as the active joint to be controlled first. Since the joints can be arranged arbitrarily in q_c, we end up with four possible combinations for this rather simple manipulator. We opted here to stack the passive joints at the end of the vector q_c, and to choose the active joints closer to the base to be controlled first. This choice is based on the fact that, in general, the joints closer to the base are larger and more massive, and thus slower. If they are controlled to their set-points from the beginning of the operation, and not only after the brakes are applied, one can expect reduced settling times. With the above rationale, the choice of q_c for the manipulator in figure 1 is:

$$q_c = \begin{bmatrix} q_1 & q_3 \end{bmatrix}^T \qquad (4)$$

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:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{array}{c c} \begin{array}{c} r \\ p \end{array} & \begin{bmatrix} M_{ad} & M_{ac} \\ M_{pd} & M_{pc} \end{bmatrix} \\ & \begin{array}{cc} p & r \end{array} \end{array} \begin{bmatrix} \ddot{q}_d \\ \ddot{q}_c \end{bmatrix} + \begin{bmatrix} b_a \\ b_p \end{bmatrix} \qquad (5)$$

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.

It is necessary to come up with a systematic way to choose q_c and q_d for a generic n-link manipulator with r actuators. We propose here a set of four matrices that perform this selection, based on the rationale given before. Consider a general n-link manipulator with joints numbered $1, \ldots, n$ from base to the tip, and form the matrices $M_1 \in \Re^{r \times n}$, $M_2 \in \Re^{p \times n}$, $\overline{M}_1 \in \Re^{n \times p}$, $\overline{M}_2 \in \Re^{n \times r}$ according to the following algorithms:

```
i = 1; j = n;
while i ≤ r do
        if joint j is active then
                M_1 (i, j) = 1; i = i + 1;
        j = j - 1;

i = 1; j = n;
while i ≤ p do
        if joint j is passive then
                M_2 (i, j) = 1; i = i + 1;
        j = j - 1;

i = 1; j = n;
while i ≤ p do
        if joint j is active then
                M̄_1 (j, i) = 1; i = i + 1;
        j = j - 1;

i = 1; j = n;
while i ≤ p do
        if joint j is passive then
                M̄_2 (j, i) = 1; i = i + 1;
        j = j - 1;
i = 1; j = 1;
while i ≤ (r - p) do
        if joint j is active
                M̄_2 (j, i) = 1; i = i + 1;
        j = j - 1;
```

Note that the matrices thus formed consist only of zeros and ones, and that their dimensions depend only on n and r. The desired partition is given by:

$$
\begin{gathered}
H_{ad} = M_1 M \overline{M}_1 \qquad H_{ac} = M_1 M \overline{M}_2 \qquad H_{pd} = M_2 M \overline{M}_1 \qquad H_{pc} = M_2 M \overline{M}_2 \\
b_a = M_1 b \qquad b_p = M_2 b \\
q_d = \overline{M}_1^T q \qquad q_c = \overline{M}_2^T q \\
\tau_a = M_1 \tau
\end{gathered}
\tag{6}
$$

In section 4 two examples that make use of these matrices will be presented, specifically for 2- and 3-link manipulators.

3 Robust Control

3.1 Variable Structure Controller

In this section we develop a variable structure controller (VSC) that will guarantee convergence of the joint angles to a desired position, despite possible modelling errors.

The idea behind the VSC is to force the system's state trajectory to converge to a predefined 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* [5]. Once the system is sliding over this surface, its dynamics are described by the equation of the 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 accordingly to the desired system's performance once the sliding takes place. Linear surfaces are the most common, given their design simplicity. For example, a linear sliding surface for a second order system could be:

$$s = ce + \dot{e} \tag{7}$$

where e would be the error between the desired and the actual state, and c 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 second requirement is guaranteed in a region "close" to the sliding surface if, in this region,

$$s\dot{s} < 0 \tag{8}$$

where $s(x) = 0$ is the description of the sliding surface s as a function of the state x.

3.2 VSC Design

Variable structure controllers have been applied to standard manipulators, and very good results have been obtained regarding trajectory tracking and disturbance rejection (see, for example, [2], [3], [5], [9]). Here, we propose a VSC for the underactuated case.

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 [1]. The other p accelerations will depend on the r controlled ones. To see this, from the second line of (5) we have:

$$\ddot{q}_d = -M_{pd}^{-1}(M_{pc}\ddot{q}_c + b_p) \tag{9}$$

Thus, if we try to control $\ddot{q}_c$, then $\ddot{q}_d$ is fixed and cannot be arbitrarily chosen. Only after the brakes are engaged can the driving joints' accelerations be controlled.

Define the following r-dimensional sliding surface:

$$s_c = c_c\tilde{q}_c + \dot{\tilde{q}}_c \tag{10}$$

where $\tilde{x} = x_d - x$ is the error on variable x, and c_c is an r x r matrix containing the time-constants of each surface. If the VSC can make $\ddot{q}_c$ to be equal to the following computed acceleration:

$$\ddot{\bar{q}}_c = c_c\dot{\tilde{q}}_c + \ddot{q}_{c,d} + P_c \operatorname{sgn}(s_c) \tag{11}$$

where P_c is an r x r matrix, then the time derivative of s_c will become:

$$\dot{s}_c = c_c\dot{\tilde{q}}_c + \ddot{q}_{c,d} - \ddot{\bar{q}}_c = -P_c \operatorname{sgn}(s_c) \tag{12}$$

Equation (12) then guarantees that (8) is satisfied for s_c, if one chooses appropriate values for the entries of P_c; and this in turn guarantees that the joint errors will converge to zero exponentially, the convergence rate being determined by the elements of c_c. Finally, in order to obtain the computed acceleration $\ddot{\bar{q}}_c$, we compute $\ddot{q}_d$ in (9) using $\ddot{\bar{q}}_c$ instead of $\ddot{q}_c$, and then substitute the obtained value of $\ddot{q}_d$ in the first line of (2):

$$\tau_a = (M_{ac} - M_{ad} M_{pd}^{-1} M_{pc}) \ddot{\bar{q}}_c - M_{ad} M_{pd}^{-1} b_p + b_a \tag{13}$$

The torque given by (13) guarantees that $\ddot{q}_c = \ddot{\bar{q}}_c$, 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 formulation 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. In order to implement the second methodology, we first note that after all passive joints are braked, the new dynamic equation of the system is simply:

$$\tau_a = M_{aa} \ddot{q}_a + b_a \tag{14}$$

where now M_{aa} and q_a can be obtained as:

$$M_{aa} = M_1 M M_1^T, \qquad q_a = M_1 q \tag{15}$$

Following the same reasoning as above, let's define:

$$s_a = c_a \tilde{q}_a + \dot{\tilde{q}}_a \tag{16}$$

where c_a is an r x r matrix. If we can make the acceleration of the active joints, $\ddot{q}_a$, to be equal to the following computed acceleration:

$$\ddot{\bar{q}}_a = c_a \dot{\tilde{q}}_a + \ddot{q}_{a,d} + P_a \operatorname{sgn}(s_a) \tag{17}$$

where P_a is an r x r matrix, then:

$$\dot{s}_a = c_a \dot{\tilde{q}}_a + \ddot{q}_{a,d} - \ddot{\bar{q}}_a = -P_a \operatorname{sgn}(s_a) \tag{18}$$

Using $\ddot{\bar{q}}_a$ in (14) instead of $\ddot{q}_a$ will then guarantee that (18) is satisfied. 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 (13) and (14) introduce an undesired chattering into the system, because of the *sgn(.)* functions in (11) and (17). This problem can be solved with the addition of a boundary layer around the sliding surface. For this sake, we substitute the function *sgn(.)* for the following one:

$$\operatorname{sgn}(x) \rightarrow \begin{cases} \operatorname{sgn}(x) & \text{if } x \geq \varepsilon \\ \dfrac{x}{\varepsilon} & \text{if } x < \varepsilon \end{cases} \tag{19}$$

In this expression, ε is the “thickness” of the boundary layer, pre-defined by the user.

3.3 Robustness Issues

The control methodology presented above provides the system with a great deal of robustness, because the system is forced to slide along the sliding surface. Therefore, modelling errors will not deteriorate the performance once the sliding begins. However, the methodology makes full use of the system model in order to guarantee that the sliding occurs. Consequently, modelling errors may affect the performance by inhibiting the state trajectories to reach the sliding surface.

To overcome this problem, we consider a very simple model of the manipulator for control purposes, which takes into account only the inertia matrix [3], [9]:

$$\tau(t) = M(q)\ddot{q} + f(t) \tag{20}$$

The quantity *f(t)* represents the uncertainty and modelling errors in the dynamic model. This "disturbance" *f(t)* can be estimated via:

$$f(t) = \tau(t - \Delta t) - M[q(t)]\ddot{q}(t) \tag{21}$$

The value obtained above may be low-pass filtered in order to eliminate high-frequency components derived from numerical differentiation.

If (20) is to be used instead of (1) for control purposes, then one should substitute every occurrence of $b(q, \dot{q})$ in the control algorithm for *f(t)*. If the system stabilizes with the use of this new control law, then we can affirm that robustness to parameter uncertainty was obtained.

4 Case Study

Before presenting the results of the simulations, we will present here the partitioning of two- and three-link manipulators. These results will be used in the next section.

Figures 1 and 2 present the manipulators to be used in the simulations. Considering first the 2-link manipulator, the desired matrices are:

$$\begin{aligned} M_1 &= \overline{M}_1^T = \begin{bmatrix} 1 & 0 \end{bmatrix} \\ M_2 &= \overline{M}_2^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{22}$$

which lead to the following partitioning:

$$M_{ad} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = M_{11} \tag{23a}$$

$$M_{ac} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = M_{12} \tag{23b}$$

$$M_{pd} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = M_{21} \tag{23c}$$

$$M_{pc} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = M_{22} \tag{23d}$$

$$M_{aa} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = M_{11} \tag{23e}$$

$$\begin{bmatrix} b_a \\ --- \\ b_p \end{bmatrix} = \begin{bmatrix} b_1 \\ --- \\ b_2 \end{bmatrix}, \begin{bmatrix} q_d \\ --- \\ q_c \end{bmatrix} = \begin{bmatrix} q_1 \\ --- \\ q_2 \end{bmatrix}, \begin{bmatrix} \tau_a \\ --- \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ --- \\ 0 \end{bmatrix} \tag{23f}$$

As for the 3-link manipulator, with actuators at joints 1 and 2, we get:

$$\begin{gathered} M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\ \overline{M}_1^T = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \overline{M}_2^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{gathered} \tag{24}$$

$$M_{ad} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} M_{22} \\ M_{12} \end{bmatrix} \tag{25a}$$

$$M_{ac} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} M_{21} & M_{23} \\ M_{11} & M_{13} \end{bmatrix} \tag{25b}$$

$$M_{pd} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = M_{32} \tag{25c}$$

$$M_{pc} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} M_{31} & M_{33} \end{bmatrix} \tag{25d}$$

$$M_{aa} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M_{22} & M_{21} \\ M_{12} & M_{11} \end{bmatrix} \tag{25e}$$

$$\begin{bmatrix} b_a \\ --- \\ b_p \end{bmatrix} = \begin{bmatrix} b_2 \\ b_1 \\ --- \\ b_3 \end{bmatrix}, \begin{bmatrix} q_d \\ --- \\ q_c \end{bmatrix} = \begin{bmatrix} q_2 \\ --- \\ q_1 \\ q_3 \end{bmatrix}, \begin{bmatrix} \tau_a \\ --- \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_1 \\ --- \\ 0 \end{bmatrix} \tag{25f}$$

5 Simulation Results

For simulation purposes, the following values were adopted for the dynamic parameters of the manipulators.

Table 1: Numerical values of the dynamic parameters.

PARAMETER	2-LINK	3-LINK
m_1 (Kg)	2.0	2.0
m_2 (Kg)	1.0	1.0
m_3 (Kg)	-----	1.0
I_1 (Kg m^2)	0.2	0.2
I_2 (Kg m^2)	0.1	0.1
I_3 (Kg m^2)	-----	0.1
l_1 (m)	0.3	0.3
l_2 (m)	0.3	0.3
l_3 (m)	-----	0.3
l_{c_1} (m)	0.15	0.15
l_{c_2} (m)	0.15	0.15
l_{c_3} (m)	-----	0.15

Using these values and the partitions presented on section 4, we can easily perform a pre-analysis of the dynamical singularities of both manipulators. The idea is to compute the determinant of the matrix M_{pd} and check to see if it is below some specified threshold over all possible values of q. In our case, the sub-matrix M_{pd} for the 2-link manipulator is a function of θ_2, and for the 3-link one, a function of θ_3. Figures 3 and 4 show the value of

$\det(M_{pd})$ over the entire possible range of the mentioned angles. As we can see, there are no dynamical singular points in both cases, i.e.,

$$\det(M_{pd}) \neq 0, \forall q \in \Re^n \tag{26}$$

In other words, we can make the manipulator go from any initial to any final position without concerning with the inversion of the sub-matrix M_{pd}.

In the cases where these dynamic singularities may occur, the performance of the system is compromised. One solution that the authors are investigating is the use of redundant control techniques, in order to drive a redundant underactuated manipulator away from dynamically singular points. The redundancy could further be used to minimize/maximize a performance criterion, such as manipulability or energy consumption, and to account for obstacle avoidance.

In order to demonstrate the robustness of the proposed controller, we are going to present here experiments using the full dynamic model in (13) and (14), and then following the methodology presented in section 3.3.

The objective of the simulations presented here was to make the manipulator achieve a final desired position, i.e., we were interested in the step response of the joint angles. We chose to brake the passive joints whenever they reached a joint angle error of less than 0.0015 *rd* (approximately 0.08 degrees), with a joint velocity of less than 0.001 *rd/s*. This ensured that the passive joints were braked at a point where they were practically at rest and with negligible steady-state error.

Table 2 summarizes the set of experiments, along with the gains used at each one. Table 3 presents the results obtained for each experiment, and also refers the reader to the appropriate figures illustrating the results. All angles are in degrees.

The first experiment consisted of driving the joints over a 90 degree excursion. Note in figure 5 that the active joint initially moves towards negative angles in order to bring the passive joint to its desired set-point with zero velocity. This is achieved at t = 0.7392 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 3.6804 s. In this first experiment the full dynamic model was used.

Table 2: Summary of experiments performed.

EXP.	Robot	Full model	c_c	c_a	P_c	P_a	ε_c	ε_a
1	2-link	yes	10.0	4.1	200.0	200.0	0.03	0.06
2	2-link	no	10.0	4.1	200.0	200.0	0.03	0.06
3	2-link	no	10.0	4.1	200.0	200.0	0.03	0.06
4	3-link	no	$\begin{bmatrix} 3 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 3.7 & 0 \\ 0 & 3.7 \end{bmatrix}$	$\begin{bmatrix} 0 & 100 \\ 0 & 90 \end{bmatrix}$	$\begin{bmatrix} 100 & 100 \\ 0 & 100 \end{bmatrix}$	0.03	0.06

Table 3: Summary of results obtained.

EXPERIMENT	Initial angle	Final angle desired	Final angle error	Brake applied at (s)	Figure
1	$\begin{bmatrix} 0 \\ 90 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.0001 \\ 0.0856 \end{bmatrix}$	0.7392	5
2	$\begin{bmatrix} 0 \\ 90 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.0001 \\ 0.0856 \end{bmatrix}$	0.7122	6, 7
3	$\begin{bmatrix} 90 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 90 \\ -70 \end{bmatrix}$	$\begin{bmatrix} 0.0804 \\ 0.0776 \end{bmatrix}$	0.8610	8
4	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 90 \\ 45 \\ 45 \end{bmatrix}$	$\begin{bmatrix} -0.0152 \\ -0.0675 \\ 0.0777 \end{bmatrix}$	0.7926	9, 10

In order to test the robustness of the VSC, the same experiment was performed following section 3.3, i.e., only the inertia matrix was considered for control purposes. Comparing the first and second lines of table 3, and figures 5 and 6, we can affirm that robustness to uncertainties (in this case, modelling errors) is guaranteed. For this experiment, figure 7 shows the 2D animation of the links of the manipulator. From this figure, one can understand how the dynamic coupling is used by the active joint to control the passive one. Namely, the active moves down until the passive reaches its set-point. Then the brake is engaged and the active joint can move up to reach its own set-point.

Experiment #3 consisted of controlling the passive joint in a region of great instability for the active joint, around $\theta_1 = \pi/2$. Note in figure 8 how the active joint move to accomplish the proposed objective. In this experiment also the reduced dynamic model was used, illustrating once more the robustness of the VSC.

Finally, experiment #4 was performed with the 3-link manipulator. Note in figure 9 how the active joints act combined in order to bring the passive one to rest. Note also that after the passive joint is braked, the active ones converge smoothly and exponentially to their set-points. The animation in figure 10 shows various stages of the movement. In this experiment also the full dynamic model was not used by the controller.

It can be inferred from the above discussion that the control law proposed here not only controls effectively manipulators with passive joints, it also accounts for the uncertainties in the dynamic model. Thus, this control law is robust enough for the problem in hand.

As a matter of comparison with previous works in this area, we also ran simulations using the control law proposed in [1], but without using the pre-acceleration phase in order to obtain comparable results. Namely, we repeated experiments #1 and #2, which required a 90^{o} excursion of both joints, with and without the use of the full dynamic model, respectively. In order to have a fair comparison, we adopted the same saturation levels for the torque at the active joint. The results are shown in figures 11 and 12. The first one shows that when the full dynamic model is used, this control law provides a performance that is comparable to that provided by the VSC. However, it lacks the robustness necessary in this kind of system, as we can see in figure 12. The passive joint cannot reach its set-point within a reasonable time, and the active joint continues to bounce trying to drive the passive joint to rest.

If we compare now the present method with the one presented in [10], we can affirm that our formulation is much simpler, and that the braking and settling times are much smaller.

As for implementation purposes, the present control law requires the computation of the inertia matrix of the manipulator, $M(q)$, which is a symmetric matrix. Thus, our scheme is certainly faster in computational terms than the computed torque controller, which requires the computation of the full dynamic model, i.e., the computation of $M(q)$ and of $b(q, \dot{q})$.

6 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 scheme 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 [6].

We compared the proposed control law with the one presented in [1], 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.

7 Acknowledgments

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

8 References

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

[2] BAILEY, E.; ARAPOSTATHIS, A. Simple sliding mode control scheme applied to robot manipulators. *Int. Journal of Control*, vol. 45, no. 4, 1987, pp. 1197-1209.

[3] BERGERMAN, M.; CRUZ, J. J. Robust position control of mechanical manipulators. *3rd International Workshop on Advanced Motion Control*, Berkeley, CA, Mar. 1994.

[4] CRAIG, J.J. *Introduction to robotics*: mechanics and control. Addison-Wesley, Reading, 2 ed.

[5] DeCARLO, R.A.; ZAK, S.H.; MATTHEWS, G.P. Variable structure control of nonlinear multivariable systems: a tutorial. *Proceedings of the IEEE*, vol. 76, no. 3, Mar.1988, pp. 212-232.

[6] GU, Y.-L.; XU, Y. Under-actuated robot systems: dynamic interaction and adaptive control. Proceedings of the IEEE Systems, Man and Cybernetics Conference, Oct. 1994.

[7] JAIN, A.; RODRIGUEZ, G. An analysis of the kinematics and dynamics of underactuated manipulators. *IEEE Trans. on Robotics and Automation*, vol. 9, no. 4, Aug. 1993, pp. 411-422.

[8] LEWIS, C.L.; MACIEJEWSKI, A.A. Failure tolerant operation of kinematically redundant manipulators. *Conference on Intelligent Robotics in Field, Factory, Service, and Space*, Mar. 1994, pp. 837-841.

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

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

[11] ORIOLO, G.; NAKAMURA, Y. Control of mechanical systems with second-order

nonholonomic constraints: underactuated manipulators. *Proc. of the 30th Conference on Decision and Control*, Dec. 1991, pp. 2398-2403.

[12] PAPADOPOULOS, E.; DUBOWSKY, S. Failure recovery control for space robotic systems. *Proceedings of the 1991 American Control Conference*, vol. 2, 1991, pp. 1485-1490.

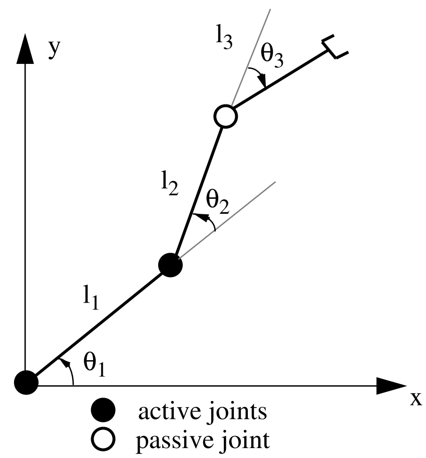

Figure 1: Three-link manipulator with one passive joint.

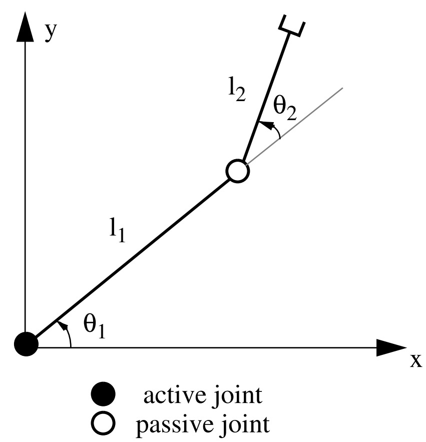

Figure 2: Two-link manipulator with one passive joint.

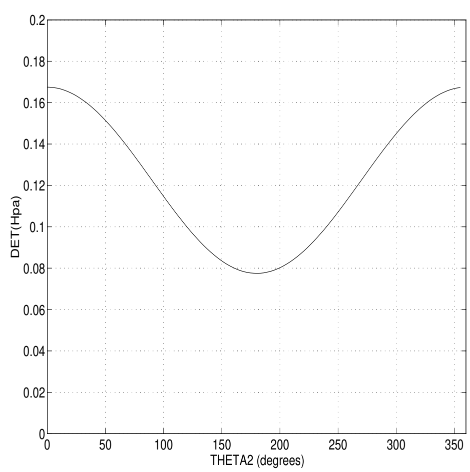

Figure 3: Determinant of M_{pd} for the 2-link manipulator.

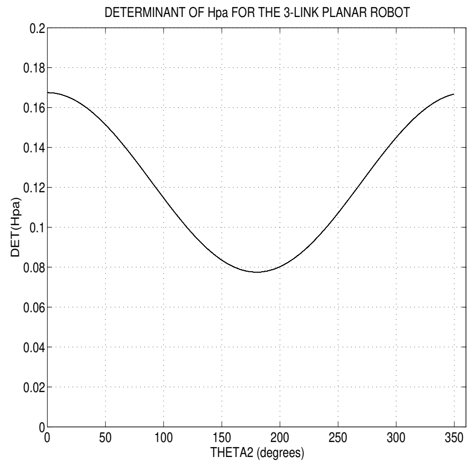

Figure 4: Determinant of M_{pd} for the 3-link manipulator.

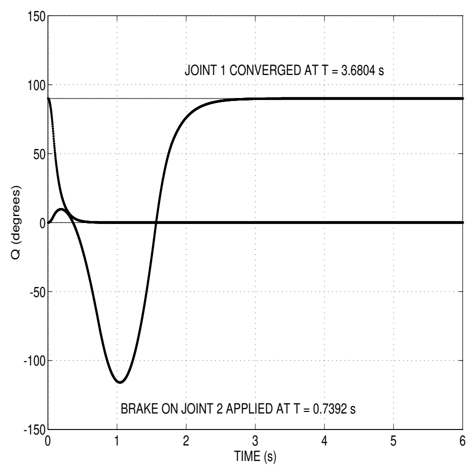

Figure 5: Response of the 2-link manipulator. Experiment #1.

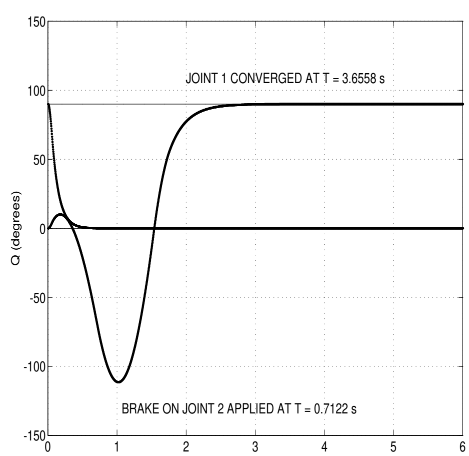

Figure 6: Response of the 2-link manipulator. Experiment #2.

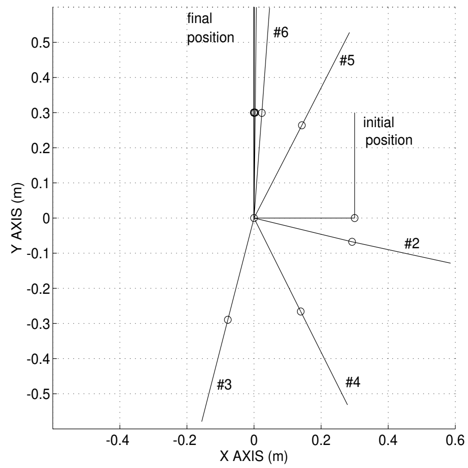

Figure 7: Animation of the 2-link manipulator. Experiment #2.

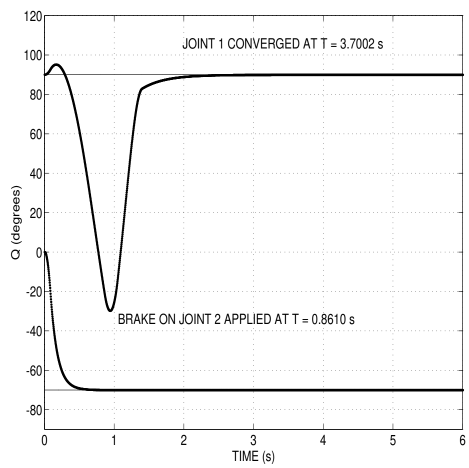

Figure 8: Response of the 2-link manipulator. Experiment #3.

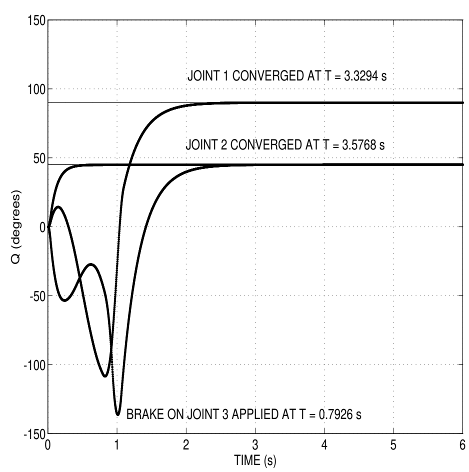

Figure 9: Response of the 3-link manipulator. Experiment #4.

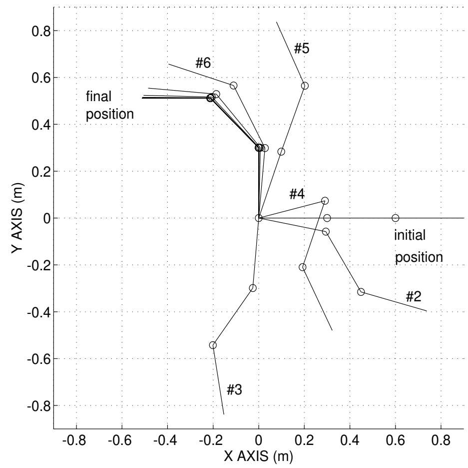

Figure 10: Animation of the 3-link manipulator. Experiment #5.

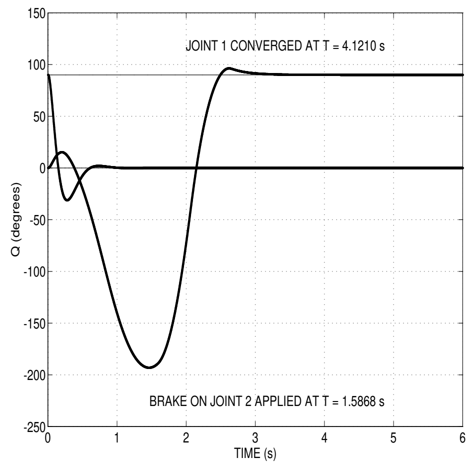

Figure 11: Response of the 2-link manipulator. Experiment #1 with control law from [1].

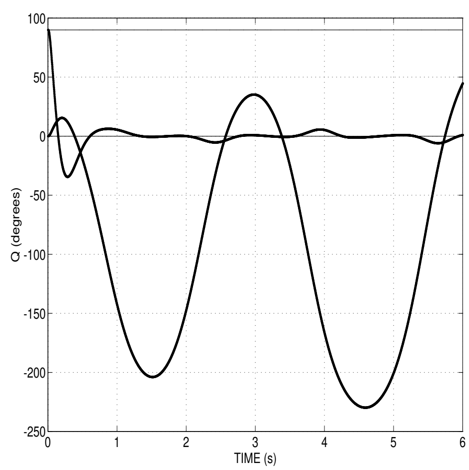

Figure 12: Response of the 2-link manipulator. Experiment #2 with control law from [1].