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, 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 [7]. 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. The methodology proposed in this paper uses the dynamic coupling between the passive and the active joints 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. The word underactuated refers to the manipulator having 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, where large redundancy is available, 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. 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 [7]. 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.

Not until recently researchers started working with the control problem of underactuated manipulators. In [1], Arai and Tachi 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. They also developed a 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 [11], considered the failure recovery of a space manipulator. In this work also the authors used a controller which required an accurate model of the system. Oriolo and Nakamura [10] showed that there does not exist any smooth control law that guarantees that the system will stabilize at an equilibrium point, and that a simple PD controller is able to bring the system to a stable configuration over an equilibrium manifold. Jain and Rodriguez [6] provided an analysis of the kinematic and dynamic issues of this kind of robotic system, using the spatial operator algebra.

Hereafter, we will refer to active joints as the ones which are controlled via an actuator. Passive joints are those who cannot be controlled directly, but are equipped with brakes. It is assumed that all joints are equipped with position encoders. 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 at least half of the joints being actuated. Also following the assumptions in [1], [11], we assume that there is enough dynamic coupling between the passive and the active joints. It is clear that if this coupling is too small, it will be impossible to change the position of the passive joints by simply moving the active ones.

The control methodology proposed in this paper uses the dynamic coupling between the passive joints and the active joints, 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. 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

By using the Lagrangian formulation, one can obtain the following set of differential equations to describe the dynamic behavior of an n-link underactuated manipulator:

\[
\tau = M(q) \ddot{q} + b(q, \dot{q})
\]

In (1), \(M(q)\) is the \(n \times n\) inertia tensor matrix, \(b(q, \dot{q})\) is the \(n \times 1\) vector representing Coriolis, centrifugal and gravitational torques, and \(\tau\) represents the torques applied at the active joints. It has \(p\) components equal to zero.

When dealing with underactuated manipulators it is important to perform a partition of the dynamic equation (1), and it is given as follows:

\[
\begin{bmatrix}
\tau \\
0
\end{bmatrix} = \begin{bmatrix}
r M_{ad} & M_{ap} \\
p M_{pd} & M_{pp}
\end{bmatrix} \begin{bmatrix}
\ddot{q}_a \\
\ddot{q}_p
\end{bmatrix} + \begin{bmatrix}
b_a \\
b_p
\end{bmatrix}
\]

This partition is very useful to understand the dynamic coupling between the passive and active joints of an underactuated manipulator. Namely, from (2):

\[
M_{pa} \ddot{q}_a + M_{pp} \ddot{q}_p + b_p = 0 
\]

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 \(\ddot{q}_a\) 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 shown in [1], \(r\) joints can be controlled at any given instant. 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 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, \(q_c\) is a two-dimensional vector containing the passive joint \(q_2\) and one active joint. 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.

In (1), \(q_d\) stands for “driving joints”), contains \(p\) active joints to be controlled after the joint angles of the joints \(q_c\) reach their set-point.

In section 4 two examples of the partitioned equation (5) 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 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 [4]. On the surface, the system 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.

Two aspects are important in this class of controllers: first, the sliding surface must be designed according to the desired system’s performance. Linear surfaces are most common, given their design simplicity. 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 [4],

\[
s s < 0 .
\]

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], [4], [8]). 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:

\[
\ddot{q}_d = -M^{-1} M_{pd} \ddot{q}_c + b_p
\]

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:

$$s_c = c_c \ddot{q}_c + \dot{q}_c$$  \(8\)

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

$$\ddot{q}_c = c_c \ddot{q}_c + \ddot{\dot{q}}_c + P_c sgn(s_c)$$  \(9\)

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

$$\dot{s}_c = c_c \ddot{q}_c + \ddot{\dot{q}}_c + \ddot{\dot{q}}_c = -P_c sgn(s_c)$$  \(10\)

Equation (10) then guarantees that (6) 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 $\ddot{q}_c$, we compute $\ddot{\dot{q}}_d$ in (7) using $\ddot{q}_c$ instead of $\ddot{q}_c$, and then substitute the obtained value of $\ddot{\dot{q}}_d$ in the first line of (2):

$$\tau_a = (M_{ac} - M_{ad}M_{pb}^{-1}M_{pc})\ddot{q}_c - M_{ad}M_{pb}^{-1}b_p + b_a$$  \(11\)

The torque given by (11) guarantees that $\ddot{q}_c = \ddot{\dot{q}}_c$, therefore the joints grouped in $\ddot{q}_c$ 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 $\ddot{q}_c$ also reach their set-point; control the remaining joints in $\ddot{\dot{q}}_d$;
- 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 $\ddot{q}_c$ 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:

$$\tau_a = M_{aa} \ddot{q}_a + b_a$$  \(12\)

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

$$s_a = c_a \ddot{q}_a + \dot{q}_a$$  \(13\)

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

$$\ddot{q}_a = c_a \ddot{q}_a + \ddot{\dot{q}}_a + P_a sgn(s_a)$$  \(14\)

where $P_a$ is an $r \times r$ matrix, then:

$$\dot{s}_a = c_a \ddot{q}_a + \ddot{\dot{q}}_a, a - \ddot{\dot{q}}_a = -P_a sgn(s_a)$$  \(15\)

Using $\ddot{\dot{q}}_a$ in (12) instead of $\ddot{\dot{q}}_a$ will then guarantee that (15) 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 (11) and (12) introduce an undesired chattering into the system, because of the $sgn(.)$ functions in (9) and (14). 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:

$$sgn(x) \rightarrow \begin{cases} sgn(x) & \text{if } x \geq \varepsilon \\ \frac{x}{\varepsilon} & \text{if } x < \varepsilon \end{cases}$$  \(16\)

In this expression, $\varepsilon$ 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 [2], [8]:

$$\tau(t) = M(q) \ddot{q} + f(t)$$  \(17\)

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

$$f(t) = \tau(t - \Delta t) - M(q(t)) \ddot{q}(t)$$  \(18\)

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

If (17) 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

In this section we present the partitioning of two- and three-link planar 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 partitioning is:

$$M_{aa} = M_{11}, M_{ac} = M_{12}, M_{pd} = M_{21}, M_{pc} = M_{22}$$

$$q_d = q_1, q_c = q_2$$

For the 3-link manipulator shown in figure 1, we get:
SIMULATION RESULTS

For simulation purposes, the values in table 1 were adopted for the dynamic parameters of the manipulators.

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 

\[
M_{pd} = \begin{bmatrix} M_{22} & M_{23} \\ M_{12} & M_{11} \end{bmatrix}, \quad M_{ac} = \begin{bmatrix} M_{21} & M_{22} \\ M_{11} & M_{12} \end{bmatrix}, \quad M_{pd} = M_{32} M_{pc} = \begin{bmatrix} M_{31} & M_{32} \end{bmatrix}
\]

as we can see, there are no dynamical singular points in both cases, for \( \det (M_{pd}) \) is never equal to zero. In other words, we can run the manipulator anywhere in its workspace without concerning with the inversion of \( M_{pd} \).

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 some 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 (11) and (12), and then following the more robust 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 \( \text{rd} \) (approximately 0.08 degrees), with a joint velocity of less than 0.001 \( \text{rd/s} \). 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 illustrating figures. All angles are in degrees.

### Table 1: Numerical values of the dynamic parameters.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>2-LINK</th>
<th>3-LINK</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 ) (Kg)</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>( m_2 ) (Kg)</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>( m_3 ) (Kg)</td>
<td>-----</td>
<td>1.0</td>
</tr>
<tr>
<td>( l_1 ) (Kg m(^2))</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>( l_2 ) (Kg m(^2))</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( l_3 ) (Kg m(^2))</td>
<td>-----</td>
<td>0.1</td>
</tr>
<tr>
<td>( l_1 ) (m)</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>( l_2 ) (m)</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>( l_3 ) (m)</td>
<td>-----</td>
<td>0.3</td>
</tr>
<tr>
<td>( l_{c1} ) (m)</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>( l_{c2} ) (m)</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>( l_{c3} ) (m)</td>
<td>-----</td>
<td>0.15</td>
</tr>
</tbody>
</table>

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, we have:

- 2-link manipulator: \( M_{pd} = 0.1225 + 0.045 \cos \theta_2 \)
- 3-link manipulator: \( M_{pd} = 0.1225 + 0.045 \cos \theta_3 \)

As we can see, there are no dynamical singular points in both cases, for \( \det (M_{pd}) \) is never equal to zero. In other words, we can run the manipulator anywhere in its workspace without concerning with the inversion of \( M_{pd} \).

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 some 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 (11) and (12), and then following the more robust 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 \( \text{rd} \) (approximately 0.08 degrees), with a joint velocity of less than 0.001 \( \text{rd/s} \). 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 illustrating figures. All angles are in degrees.

<table>
<thead>
<tr>
<th>EXP.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot -&gt;</td>
<td>2-link</td>
<td>2-link</td>
<td>2-link</td>
<td>3-link</td>
</tr>
<tr>
<td>Full Model</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>( c_c )</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>[3.0 0 0 10]</td>
</tr>
<tr>
<td>( c_a )</td>
<td>4.1</td>
<td>4.1</td>
<td>4.1</td>
<td>[3.7 0 0 3.7]</td>
</tr>
<tr>
<td>( P_c )</td>
<td>200.0</td>
<td>200.0</td>
<td>200.0</td>
<td>[0 100 0 90]</td>
</tr>
<tr>
<td>( P_a )</td>
<td>200.0</td>
<td>200.0</td>
<td>200.0</td>
<td>[100 100 0 0]</td>
</tr>
<tr>
<td>( \epsilon_c )</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>( \epsilon_a )</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
</tbody>
</table>

The first experiment consisted of driving both joints over a 90\(^{\circ}\) excursion. Note in figure 3 that the active joint initially moves towards negative angles in order to bring the passive joint to its desired set-point with zero velocity at \( t = 0.7392 \) s, when the brake is engaged. Then the active joint is controlled to reach its set-point with zero velocity after a total of 3.6804 s. In this first experiment the full dynamic model was used.

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 1st and 2nd lines of table 3, and figures 3 and 4, we can affirm that robustness to modeling errors is guaranteed. For this experiment, figure 5 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
Table 3: Summary of results obtained.

<table>
<thead>
<tr>
<th>EXP.</th>
<th>Initial angle</th>
<th>Final angle desired</th>
<th>Final angle error</th>
<th>Brake applied at (s)</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0 90]</td>
<td>[90 0]</td>
<td>[-0.0001 0.0856]</td>
<td>0.7392</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>[0 90]</td>
<td>[90 0]</td>
<td>[-0.0001 0.0856]</td>
<td>0.7122</td>
<td>4, 5</td>
</tr>
<tr>
<td>3</td>
<td>[90 0]</td>
<td>[90 -70]</td>
<td>[0.0804 0.0776]</td>
<td>0.8610</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>[0 0]</td>
<td>[0 45]</td>
<td>[-0.0152 -0.0675]</td>
<td>0.7926</td>
<td>7, 8</td>
</tr>
</tbody>
</table>

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 predefined 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 [5].

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 received partial support from the Brazilian National Council for Research and Development (CNPq).

8 REFERENCES


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 7 how the active joint moves up 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 7 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 to their set-points. The animation in figure 8 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, 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 9 and 10. 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 10. 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 [9], 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 reasonably fast in computational terms, for it does not require the computation 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 [5].

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.
Figure 3: Response of the 2-link manipulator (exp. #1).

Figure 4: Response of the 2-link manipulator (exp. #2).

Figure 5: Animation of the 2-link manipulator (exp. #2).

Figure 6: Response of the 2-link manipulator (exp. #3).

Figure 7: Response of the 3-link manipulator (exp. #4).

Figure 8: Animation of the 3-link manipulator (exp. #5).

Figure 9: Exp. #1 repeated with control law from [1].

Figure 10: Exp. #2 repeated with control law from [1].