OPTIMAL CONTROL OF UNDERACTUATED MANIPULATORS VIA ACTUATION REDUNDANCY

Benedito C. O. Maciel*, Marcel Bergerman**, Marco Henrique Terra*

* São Carlos Engineering School, University of São Paulo
São Carlos  SP  Brazil

** Information Technology Institute
Campinas SP Brazil

Abstract

The increased utilization of manipulators in hazardous or hard-to-reach environments have led to a corresponding increase on the study of fault tolerant control methods for these mechanisms, ranging from robot design and trajectory planning to fault detection and isolation and post-failure control algorithms. In this article we focus on post-failure control of a mechanical manipulator from the point of view of optimal control, and present a novel method for controlling the positions of the failed, passive joints, in an optimal way. Although the optimization is performed locally, the results indicate the validity and feasibility of the proposed theory.

1 Introduction

In the last few years the robotics community has witnessed an increased utilization of manipulators in hazardous or hard-to-reach environments, such as inside sewer pipes, in space or in the bottom of the oceans. In such situations, a failed manipulator joint may jeopardize an entire mission, and therefore these mechanisms require sophisticated fault tolerant control methodologies. Fault tolerant manipulator control has traditionally been studied as a combination of separate tasks:

- hardware redundancy [11], [12], [27], [28];
- manipulator design and trajectory planning for fault tolerance [17], [18], [21];
- fault detection and isolation (FDI) methods [8], [9], [10], [15], [19], [20], [22], [25], [26]; and
- post-failure control methods [1], [2], [4], [5], [7], [13], [16], [23], [24].

It is only recently that manipulator FDI has been combined with post-failure control algorithms in a unified method [23]. An extension of this work, which looks at the fault tolerant control problem from an integrated perspective, has been proposed by the authors in [3]. There, the authors present a hybrid systems-based framework consisting of three basic units that guarantee task completion in the presence of any number of failed joints (Figure 1). The first unit is an FDI scheme which continuously monitors the manipulator to detect and identify a joint failure. The second unit is responsible for control reconfiguration; it is modeled as a hybrid automaton with jumps activated by either a joint failure or a joint reaching its set-point. The third unit is composed of control algorithms appropriate for each control mode, based on input from the control reconfiguration unit.

In the current article we focus on the control algorithm unit, and more specifically at the problem of controlling the position of a failed joint to any desired set-point in an optimal way when actuation redundancy is available. This is possible, for example, when one joint of a 3-joint manipulator fails, and the position of the failed joint is controlled by the remaining two joints. Our solution to this problem is based on local redundancy resolution, extensively studied in the context of inverse kinematics [14]. Our main contribution is the utilization of the so-called coupling index as an optimization criterion to minimize the energy spent by the manipulator during the motion of the failed joints.

---

Fig. 1: Fault tolerant manipulator control framework.

---

1. This work is partially supported by FAPESP under grant nos. 97/13384-7, 99/01210-0, and 99/10031-1.
2 Problem Statement and Nomenclature

In this work we address the following problem: Given a serial mechanical manipulator with \( n \) joints, of which \( n_p \) of them lack actuation because of an actuator failure, and the remaining \( n_a \) joints operate normally, where \( n_a > n_p \), find a control method to optimally bring the \( n_p \) passive joints to a desired position, after which the \( n_a \) active joints should also be brought to their desired positions.

To solve this problem we make the following assumptions, valid for most manipulators:

• only free-swinging joint failures are considered;
• all joints are equipped with brakes and encoders;
• joint failures occur one at a time.

With these assumptions, we believe that the work described here is applicable in a variety of scenarios. The ones we have in mind are manipulators operating in harsh or hard-to-reach environments, such as in space or deep-sea, where actuator repair is too costly or impossible.

The nomenclature utilized in this article is as follows.

Consider a manipulator with \( n \) joints. Let \( q \) represent the joint position vector. Its dynamic model can be written as:

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

(1)

where \( M \) represents the \( n \times n \) inertia matrix and \( b \) represents the \( n \times 1 \) vector of Coriolis, centrifugal, gravitational, and frictional torques. Whenever an actuator fails, we analyze the dynamics of the resulting underactuated manipulator by partitioning the joint vector \( q \) in two components, corresponding to the positions of the active and of the passive joints, as:

\[
q = \begin{bmatrix} q_a^T & q_p^T \end{bmatrix}^T
\]

(2)

The dimensions of these vectors are

\[
n_a = \text{dim}(q_a), \quad n_p = \text{dim}(q_p), \quad n = n_a + n_p
\]

(3)

The dynamic model (1) can be partitioned to reflect the contributions of each type of joint to the manipulator’s motion as:

\[
\begin{bmatrix} \dot{q}_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_{aa} & M_{ap} \\ M_{pa} & M_{pp} \end{bmatrix} \begin{bmatrix} \dot{q}_a \\ \dot{q}_p \end{bmatrix} + \begin{bmatrix} b_a \\ b_p \end{bmatrix}
\]

(4)

where we used the fact that \( \tau_p = 0 \) because no torque is applied at the passive joints.

3 PTP Control of Manipulators with Passive Joints

When a manipulator is equipped with both active and passive joints, it is not always possible to make it follow a pre-defined trajectory in Cartesian space. It is possible, however, to bring all joints to a desired position, thus bringing the end-effector to a desired Cartesian configuration. This point-to-point (PTP) control problem was first posed in [2]. The solution consists of driving the joints of the manipulator in two distinct phases (see Figure 2). First, the passive joints are driven to their set-points via their dynamic coupling with the active ones. Each passive joint is locked as it reaches its set-point. After all passive joints reach their desired positions and are locked, the second phase takes place, namely, control of the active joints to their set-points. Because the passive joints are locked during the second phase, the manipulator is as easily controlled as if it were fully actuated. One must note that, in [2], when \( n_a > n_p \) a subset of the active joints can be controlled along with the passive ones in the first phase. For comparison purposes in the results section, we shall denote this by non-redundant control strategy.

Fig. 2: Flow diagram of the PTP control method [2].

Clearly, the problem addressed in this article benefits from the PTP solution of [2], except that no author has ever proposed a method for optimally controlling the position of the passive joints in the first phase. Therefore, from this point on we focus only on this part of our original problem.

4 Coupling Index

Although the positions of the passive joints cannot be directly controlled, they are dynamically coupled to the active joints because of the presence of non-zero off-diagonal elements in the inertia matrix. In this section, we present an approach to quantify this coupling and explain why it is related to energy consumption. Further details about the coupling index concept can be found in [5].

Since the inertia matrix \( M \) is positive definite, we can factor \( \dot{q}_a \) in the first line of (4) to obtain:
\[ \ddot{q}_a = M_{aa}^{-1}(\tau_a - M_{ap}\dot{q}_p - b_a) \]  

(5)

Substituting this expression on the second line of (4), the following relationship between the acceleration of the passive joints and the torques applied at the active ones is obtained:

\[ q_p = -W_{pp}M_{pa}M_{aa}^{-1}\tau_a + W_{pp}(M_{pa}M_{aa}^{-1}b_a - b_p) \]  

(6)

where the \( n_p \times n_p \) matrix \( W_{pp} \) is the inverse of the Schur complement of \( M_{aa} \) in \( M \):

\[ W_{pp} = (M_{pp} - M_{pa}M_{aa}^{-1}M_{ap})^{-1} \]  

(7)

Matrix \( W_{pp} \) is positive definite, because it is equal to the lower diagonal block of the inverse of the inertia matrix.

We focus on the relationship between the accelerations of the passive joints and the active joints’ torques, and rewrite equation (6) as:

\[ \ddot{q}_p = W_{pa}\tau_a \]  

(8)

where

\[ W_{pa} = -W_{pp}M_{pa}M_{aa}^{-1} \]  

(9)

\[ \ddot{q}_p = \dot{q}_p - W_{pp}(M_{pa}M_{aa}^{-1}b_a - b_p) \]  

(10)

The vector \( \ddot{q}_p \) can be considered as a virtual acceleration of the passive joints, generated by the active torques and the nonlinear torques due to velocity and gravitational effects. The coupling index is defined as the product of the singular values of \( W_{pa} \) [5]:

\[ \rho_{\tau} = \prod_{i=1}^{n_p} \sigma_i(W_{pa}) \]  

(11)

The coupling index provides a local measure of how well active joint torques are transmitted to the passive joints, because the elements of \( W_{pa} \) are functions of the manipulator’s position \( q \). Note that the elements of the matrix \( W_{pa} \) must be normalized if the manipulator has both rotary and prismatic joints.

From Equation (8), one can see that the larger the norm of \( W_{pa} \), the smaller the norm of \( \tau_a \) has to be for the same unit-norm acceleration \( \ddot{q}_p \) (after the noninertial and gravitational effects are compensated for). In other words, the larger the norm of \( W_{pa} \) and, consequently, the magnitude of \( \rho_{\tau} \), the smaller the torque we have to supply to the actuators to produce the same motion at the passive joints. Therefore, when the coupling index is maximized, energy consumption for control of the passive joints is reduced.

5 Optimal Control of the Passive Joints

With the definitions and concepts of the previous sections, we are now ready to work out the main result of this article.

We recall that our objective is to control the position of the passive joints via their dynamic coupling with the active ones, taking advantage of actuator redundancy. To this end we start by looking at the second-order nonholonomic equation on the second line of (4):

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

(12)

Since we are considering the case \( n_a > n_p \), Equation (12) represents an underconstrained system of \( n_p \) equations, and the matrix \( M_{pa} \) has dimension \( n_p \times n_a \).

Factoring out \( \ddot{q}_a \) in the second line of (12) we obtain:

\[ \ddot{q}_a = -M_{pa}^\#(M_{pp}\dot{q}_p + b_p) + (I - M_{pa}^\#M_{pa})z \]  

(13)

where the superscript ‘\#' denotes the pseudo-inverse operator and \( z \) is an arbitrary vector [14]. When \( z \) is equal to the null vector, Equation (13) provides the least-norm solution for \( \ddot{q}_a \). On the other hand, \( z \) can be selected as the partial derivative of a suitable potential with respect to the position of the active joints, as in:

\[ z = -k \left( \frac{\partial P}{\partial q_a} \right) \]  

(14)

where \( k \) is a constant. In this case, by following the reasoning in [14], one can expect that the value of the potential function decreases monotonically with time.

We take advantage of this fact and select the potential function \( P \) as:

\[ P = -\rho_{\tau} = -\prod_{i=1}^{n_p} \sigma_i(W_{pa}) \]  

(15)

Equation (15) implies that the potential function is equal to the opposite of the coupling index. Because (14) leads to the minimization of \( P \), it also leads to the maximization of \( \rho_{\tau} \). Now, according to the discussion in Section 4, this implies that the torque needed to drive the passive joints is minimized.

To achieve our goal of controlling the passive joints while minimizing the torque, we substitute (13)-(15) in the first line of Equation (4). The torque thus computed is:
Finally, we design a computed torque-type controller for $q_p$ as:

$$\tau_a = (M_{ap} - M_{aa}M_{pa}^M_{pp})q_p - M_{aa}M_{pa}^b_p + b_a + kM_{aa}(I - M_{pa}^M_{pa})\left(\frac{\partial^2 \varsigma}{\partial q_a}\right)$$

Finally, we design a computed torque-type controller for $q_p$ as:

$$\tau_a = (M_{ap} - M_{aa}M_{pa}^M_{pp})u - M_{aa}M_{pa}^b_p + b_a + kM_{aa}(I - M_{pa}^M_{pa})\left(\frac{\partial^2 \varsigma}{\partial q_a}\right)$$

with

$$u = q^d_p + K_d(q^d_p - q_p) + K_p(q^d_p - q_p)$$

The torque applied at the active joints is then (17)-(18).

To end this section, we note that this optimization method is valid only locally. We do not claim that torque is minimized for all possible initial conditions and setpoints of the passive joints. Rather, we can expect this to occur in the majority of cases, and this has indeed been our experience after several simulated runs. Global optimal control is the subject of our current studies and is beyond the scope of this article.

6 Results

To validate the optimal control strategy proposed in this article, we utilize our simulation and control environment, which consists of a Matlab-based graphical user interface and a planar 3-link manipulator UARM II, designed and built by H. Benjamin Brown and Randal Casciola from Pittsburgh, PA, USA (Figure 3). In this manipulator, all three joints are equipped with an actuator, a brake, and an encoder. Because the joint friction is negligible, the joints can be independently configured as active or passive. This feature allows us to experiment with all possible combinations of joint failures.

We configure the manipulator as AAP, i.e., with joints 1 and 2 active and joint 3 passive. The PTP motion to be performed corresponds to the manipulator starting at $q = [0 \ 0 \ 0]$ and reaching $q^d = [10^\circ \ 20^\circ \ 30^\circ]$. Three strategies were tested: non-redundant control as in [2], redundant control without optimization ($k = 0$), and redundant control with optimization for $k = 0.05$.

Figure 4 (at the end of the article) presents the positions of all joints, including not only the control of the passive joints but the control of the active joints as well, and the torques applied at the active ones. To compare the energy consumption of each strategy, we computed the sum of the absolute value of the torques applied at the active joints during the motion of the manipulator, i.e.:

$$E = \sum_i \tau_1(i\Delta t) + \sum_i \tau_2(i\Delta t)$$

Since the actuators of the UARM II are armature controlled DC motors, for whom torque is directly proportional to current, $E$ provides a fairly accurate measure of the electric power spent during the motion. The results are shown in Table 1, where $E_1$ represents the energy spent during phase 1 of Figure 2 (control of the passive joints) and $E_2$ represents the energy spent during the entire motion (phases 1 and 2 of Figure 2).

Table 1: Energy expenditure for each control strategy.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>$E_1$ (Nm)</th>
<th>$E_2$ (Nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-redundant control</td>
<td>11.31</td>
<td>49.82</td>
</tr>
<tr>
<td>Redundant control with $k = 0$</td>
<td>6.37</td>
<td>14.36</td>
</tr>
<tr>
<td>Redundant control with $k = 0.05$</td>
<td>6.00</td>
<td>13.07</td>
</tr>
</tbody>
</table>

As one can realize, the redundancy-based control method proposed including optimization presents the least energy expenditure of all three methods.
7 Conclusion
As manipulators start to operate in space, in deep sea, and in hazardous environments, fault tolerant control becomes ever more fundamental. The result presented in this article allows one to drive all joints of a manipulator with both active and passive joints to a desired configuration, with the added bonus of energy consumption minimization during the phase when the passive joints are controlled. Extensions of this work that the authors are dealing with include control of passive joints not equipped with brakes, or whose encoders have failed along with their actuators.

8 References
Fig. 4: Control of a 3-joint manipulator with two active and one passive joints.

(Top) Non-redundant control.
(Middle) Redundant control with $k = 0$.
(Bottom) Redundant control with $k = 0.05$. 