

Fault-Tolerant Dynamic Control for Manipulators

Manuel C. Ramos, Jr.*

*Department of Electrical and Electronics Engineering
University of the Philippines Diliman
Quezon City 1101 PHILIPPINES*

ABSTRACT

A fault-tolerant manipulator can be expected to perform a desired task even if a joint actuator fails. When a manipulator experiences a joint actuator failure, it can be considered as an underactuated mechanical system. An underactuated serial-link manipulator can be controlled by using the coupling effects between links. The control goal to be studied in this paper is to move the end-effector from a given initial pose to a specified final pose, after a single joint failure has occurred, where the given robot has at least two degrees of redundancy. The proposed controller is designed based on a reduced-order model. The control scheme first moves the actuated joints to the desired positions. An iterative control input is then applied to an actuated joint so that the unactuated joint is asymptotically driven to the desired joint position. The control approach is illustrated by simulating a PUMA 600 Arm.

Key Words: fault-tolerance, redundant, reduced-order, underactuated, dynamic, manipulator.

1. INTRODUCTION

A serial-link manipulator can be termed underactuated if the number of actuators is less than the number of joints. Thus, an underactuated manipulator system has a fewer number of control inputs than the number of joint position variables.

The dynamic model of an underactuated robotic manipulator can be decomposed to the motion equations with input and those without input. The latter equations represent second-order nonholonomic constraints [9], which make it difficult to directly reduce the number of the generalized coordinates.

The nonlinear manipulator dynamical model can be written in the form $\dot{x} = f(x) + g(x)u$, where x is the system state, and u the input to the system. The differentiable functions $f(x)$ and $g(x)$ are the drift term and the control matrix, respectively. Furthermore, one of the basic properties of the overall system model, the controllability, cannot be guaranteed [6]. The accessibility of the overall system can be shown using Lie algebra [8]. When the system model contains a drift term, the accessibility does not imply controllability [13]. As a consequence [2], the underactuated manipulator system is not asymptotically stabilizable to an equilibrium point using a time-invariant continuous state feedback.

*Correspondence to: Department of Electrical and Electronics Engineering, University of the Philippines Diliman, Quezon City 1101 PHILIPPINES. email:manuel@eee.upd.edu.ph

The controller design of an underactuated manipulator system indeed poses a challenge : even though a manipulator has one or more of the joints unoperational, i.e., it is underactuated, the robotic manipulator should be able to perform a specified task. Several approaches to control an underactuated manipulator have been proposed [6, 15, 7, 1]. These approaches include energy based schemes [15], the use of nonlinear control models [7], nilpotent approximations [6, 1], and kinematics with path planning [14].

An energy based approach uses saturation functions and energy shaping; it is applied to the control of a two-link Acrobot and a three-link Gymnast robot in [15]. The presence of gravity terms in the Acrobot and the Gymnast robot models make the linearizations of the system dynamical model around their equilibrium points locally controllable. On the other hand, in the absence of the gravity term, the linearization of the system dynamics around an equilibrium point contains uncontrollable modes, and the system cannot be asymptotically stabilized by a constant gain linear state feedback [9]. However, an iterative control can be applied to achieve convergence to the desired state [6]. To design iterative control schemes, nonlinear approximations of an underactuated manipulator model can be used : an example of this approach is a nilpotent approximation used in the design of a control for a planar two-link revolute robot whose model contains a drift term [6]. In [6], the control scheme is designed for a two-link manipulator. The approach described in this paper uses a reduced-order model to achieve the control of an n -link manipulator.

When a redundant manipulator has an unactuated joint due to an actuator failure, it can be operated as a locked-joint arm or as a free-swinging arm. It still should be able to accomplish its task, and thus be fault-tolerant. One approach to fault-tolerance is based on planning a fault-tolerant trajectory for the manipulator [10]. A fault-tolerant trajectory is defined as a trajectory $q(t)$ such that there exists an alternate trajectory $q^*(t)$ for every time instant t . An algorithm can be constructed that iteratively searches for a fault-tolerant trajectory among all the acceptable postures at each time instant; an acceptable fault-tolerant trajectory is generated. For a slow-moving manipulator, an approach to a fault-tolerant operation of a manipulator can be based on the kinematics of the redundant manipulators; In this framework, a fault-tolerant configuration is determined in [3], but the dynamics of the manipulator are not taken into account.

This paper will discuss the dynamic control for a serial n -link underactuated manipulator. The underactuated operation can result from the failure of a joint actuator. The dynamics of the underactuated manipulator will be described by means of a reduced-order model [5, 12]. This model is used to design a controller, which moves the end-effector from a given initial pose to a specified final pose.

The paper is organized as follows. The reduced-order model for an n -link underactuated manipulator is first described. The control problem and the proposed two-phase control scheme are outlined. The motion of a general n -link manipulator after a certain time is constrained on a horizontal plane. The conditions on such a manipulator motion are discussed. Simulation results for a PUMA 600 manipulator illustrate the proposed control approach.

2. REDUCED-ORDER DYNAMICAL MODEL

The dynamical model of an n -link manipulator can be expressed as

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$$

where the i th component q_i of the n -vector q is the position of joint i , and $\tau = [\tau_1, \dots, \tau_n]^T$ the (generalized) input torque. If joint i is unactuated, the equations of the model can be rearranged so that the last component of the n -vector $\bar{q}$ is q_i . Then, the model is

$$\bar{\tau} = \bar{D}(\bar{q})\ddot{\bar{q}} + \bar{C}(\bar{q}, \dot{\bar{q}})\dot{\bar{q}} + \bar{G}(\bar{q}) \quad (1)$$

where the n -vectors $\bar{\tau}$ and $\bar{G}(\bar{q})$, and matrices $\bar{D}(\bar{q})$ and $\bar{C}(\bar{q}, \dot{\bar{q}})$ result from τ , $G(q)$, $D(q)$ and $C(q, \dot{q})$, respectively, after the rearrangement of the equations. To simplify notations, define the set $S_A = \{i \mid \text{joint } i \text{ is actuated}\}$ and $S_B = \{i \mid \text{joint } i \text{ is unactuated}\}$. It is now convenient to write $\bar{q} = [q_A^T q_B]^T$ and $\bar{\tau} = [\tau_A^T \tau_B]^T$, where q_A , τ_A are $(n-1)$ -dimensional vectors associated with the actuated joints, and q_B , τ_B are scalar variables of the unactuated joint. Thus, the model in equation (1) for a manipulator with a free-swinging arm can be expressed as

$$\begin{bmatrix} \tau_A \\ 0 \end{bmatrix} = \begin{bmatrix} D_{AA} & D_{AB} \\ D_{BA} & D_{BB} \end{bmatrix} \begin{bmatrix} \ddot{q}_A \\ \ddot{q}_B \end{bmatrix} + \begin{bmatrix} C_{AA} & C_{AB} \\ C_{BA} & C_{BB} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_B \end{bmatrix} + \begin{bmatrix} G_A \\ G_B \end{bmatrix} \quad (2)$$

The submatrices D_{AA} , D_{AB} , D_{BA} and D_{BB} of $\bar{D}(\bar{q})$ have dimensions $(n-1) \times (n-1)$, $(n-1) \times 1$, $1 \times (n-1)$ and 1×1 , respectively. Similarly, the submatrices C_{AA} , C_{AB} , C_{BA} and C_{BB} of $\bar{C}(\bar{q}, \dot{\bar{q}})$ have dimensions $(n-1) \times (n-1)$, $(n-1) \times 1$, $1 \times (n-1)$ and 1×1 , respectively. The $(n-1)$ -vector G_A , and the scalar G_B represent the gravity effects on the links corresponding to the actuated joints and the unactuated joint, respectively.

The n -dimensional vector $\dot{\bar{q}}(t)$ can be transformed to an $(n-1)$ -dimensional pseudovelocity vector $\nu(t)$ by means of the transformation matrix $\Pi = [\Pi_A^T \Pi_B^T]^T$, where Π_A and Π_B are $(n-1) \times (n-1)$ and $1 \times (n-1)$ submatrices of Π , respectively. Thus,

$$\begin{bmatrix} \dot{q}_A \\ \dot{q}_B \end{bmatrix} = \begin{bmatrix} \Pi_A \\ \Pi_B \end{bmatrix} \nu \quad (3)$$

The reduced-order model for the n -link underactuated manipulator can then be expressed as

$$\tau_\nu = D_\nu \dot{\nu} + C_\nu \nu + G_\nu \quad (4)$$

where $\tau_\nu = \Pi^T \bar{\tau}$, $D_\nu = \Pi^T \bar{D} \Pi$, $C_\nu = \Pi^T (\bar{D} \dot{\Pi} + \bar{C} \Pi)$ and $G_\nu = \Pi^T \bar{G}$. Equation (4) is the set of m ordinary differential equations representing the reduced-order dynamical model of an underactuated serial-link manipulator in which the joint i actuator is not functioning.

The number of links n is equal as the DOF of the joint space of the manipulator. The m variable can be regarded as the DOF of the task space. As such, m varies according to the task to be performed. In this paper, m is not necessarily restricted to the DOF of the entire task but may be set depending on the definition of a particular subtask.

3. STATEMENT OF THE CONTROL PROBLEM

The end-effector pose of a manipulator will be denoted by the vector $p \in \mathbf{R}^m$, where m is the degree of freedom (DOF) of the task, i.e. m is the number of variables specifying the task. The

DOF of a manipulator n is the number of joint position variables. A manipulator is redundant for a specified task if n , the DOF of the manipulator, is greater than m , the DOF of the task. The degree of redundancy (DOR) is defined as $r = n - m$.

The control problem is as follows : the end-effector of a redundant manipulator with DOR r is to be moved from an initial pose $p(0)$ to a specified final pose $p^d(t_f)$, where t_f is a specified time signifying the end of the motion. The redundancy allows the manipulator to perform the desired task even if it is underactuated due a joint actuator failure. The initial pose $p(0)$ determines the initial joint position $q(0)$. Assuming a solution exists to the kinematic equations, the final pose $p^d(t_f)$ specifies the final joint position $q^d(t_f)$.

4. CONTROL STRUCTURE

For a two-link planar revolute manipulator moving on a horizontal plane and having an unactuated (second) joint, the second link can be driven to the desired joint position by an appropriate iterative control input applied to the actuated first joint [6]. In order to constrain the motion of the unactuated link to a horizontal plane in an n -link underactuated manipulator, certain conditions must be satisfied. These conditions represent constraint equations. The manipulator forward kinematic equation, in general, can be written as

$$p = f(q) \quad (5)$$

where the function $f : q \rightarrow p$ relates the joint position q to the end-effector pose p . Thus, the desired manipulator configuration $q^d(t_f)$ must satisfy the forward kinematic equations $p^d(t_f) = f(q^d)$. The additional constraint equation will first be determined in the form $h(q) = 0_{2 \times 1}$. Assuming a solution exists, then the desired configuration $q_d(t_f)$ satisfying the kinematic and the constraint equations can be calculated.

The control scheme will be constructed in two parts [6]; namely, phase 1 and phase 2. The approach is hinged on the dynamical control of the manipulator. Furthermore, the failed joint will be free-swinging as opposed to being locked [14].

During phase 1, the actuated joints are moved to the desired joint position $q_A^d(t_f)$. At the end of phase 1 at time $t = T_1$, $q_A(T_1) = q_A^d(t_f)$, $\dot{q}_A(T_1) = 0_{(n-1) \times 1}$, where T_1 is a specified duration of phase 1, and $0_{(n-1) \times 1}$ denotes the $(n-1) \times 1$ zero matrix. In general $q_B(T_1) \neq q_B^d(t_f)$ and $\dot{q}_B(T_1) \neq 0$.

In phase 2, the control is chosen to be cyclic. The motion of the actuated joint (link) is such that at times $t = T_1 + T$, $T_1 + 2T$, ..., signifying the end of the cycles, vector $q_A(t)$ has the value of $q_A^d(t_f)$ and $\dot{q}_A(t) = 0_{(n-1) \times 1}$, where T is the length of a cycle. The chosen control drives the unactuated joint (link) is cyclically by the coupling effect so that $q_B(t)$ moves towards $q_B^d(t_f)$ and $\dot{q}_B(t)$ to zero, for $t > T_1$.

The constraint equations necessary to achieve fault-tolerance for an n -link redundant manipulator will be derived next. Then, the details of the two-phase control algorithm will be presented.

4.1. Constraint Equations

The constraint equations on the motion of a serial n -link underactuated manipulator will be derived by considering that the unactuated link will be constrained to move on a horizontal

plane $z = p_z^d(t_f)$ for $t \geq T_1$, where $p_z^d(t_f)$ is the desired z -coordinate of the end-effector position.

In a two-link underactuated manipulator, an appropriate input to the actuated joint (joint one) can drive the unactuated second joint iteratively to the desired position. In order to apply a similar approach to an n -link underactuated serial-link manipulator ($n \geq 3$), the underactuated link, after time T_1 , should be moved in the horizontal plane $z = p_z^d(t_f)$. It is accomplished by satisfying two conditions on the motion during phase 2.

Condition (i) : Only one actuated joint and the unactuated revolute joint of the manipulator are moved during phase 2; the other joints are maintained at zero velocities with the appropriate joint torque inputs.

Condition (ii) : The axes of the motion of the two joints with nonzero velocities must be in the vertical direction. The second condition guarantees that motion of the unactuated link is constrained in the horizontal plane. The constraint equation for condition (ii) is derived next.

In Figure 1, the axes of the world coordinate frame are the x_0 -axis, y_0 -axis and z_0 -axis, and the axes of the $(i-1)$ th coordinate frame are the x_{i-1} -axis, y_{i-1} -axis and z_{i-1} -axis, where joint i is the unactuated joint.

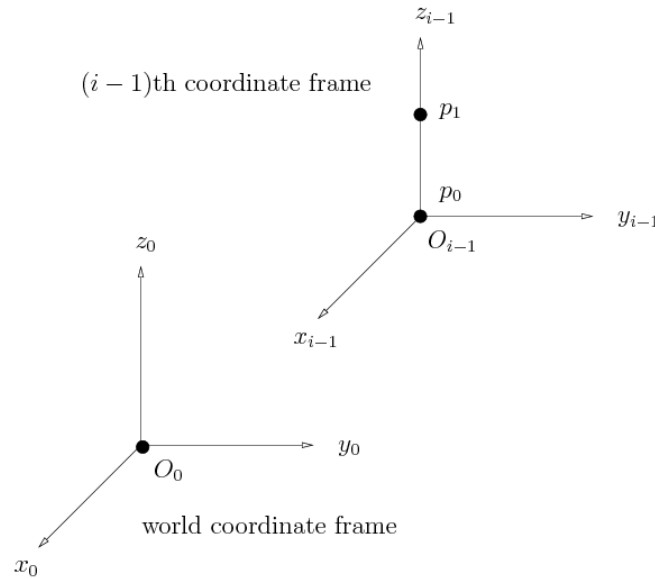


Figure 1. Selected Points on the z_{i-1} -axis

The rotation of link i is about the z_{i-1} -axis. Assuming the z_0 -axis is oriented in the vertical direction, the axis of motion of the unactuated joint will be in the vertical direction if the z_0 -axis and z_{i-1} -axis are parallel. The z_{i-1} -axis will be parallel to the z_0 -axis if two distinct points p_0^{i-1} and p_1^{i-1} on the z_{i-1} -axis have the same x -coordinates and the same y -coordinates in the world coordinate frame. First, two points on the z_{i-1} -axis are selected:

$$p_0^{i-1} = [0 \ 0 \ 0 \ 1]^T, \quad p_1^{i-1} = [0 \ 0 \ 1 \ 1]^T \quad (6)$$

It is noted that the superscript $(i-1)$ signifies the reference coordinate frame. Then, p_0^{i-1} is the origin, and p_1^{i-1} is a point on the z_{i-1} -axis in the $(i-1)$ th coordinate frame.

The transformation matrix $A_j^i(q)$ converts the coordinates of a point expressed in the i th coordinate frame to the coordinates of the same point in the j th coordinate frame [4]. Points p_0^{i-1} and p_1^{i-1} can be expressed in the world coordinate frame (i.e., in the zeroth coordinate frame) as follows:

$$p_0^w = A_0^{i-1}(q)p_0^{i-1}, \quad p_1^w = A_0^{i-1}(q)p_1^{i-1} \quad (7)$$

where the superscript w denotes the world coordinate frame. If $p_0^w = [x_0^w \ y_0^w \ z_0^w \ 1]^T$ and $p_1^w = [x_1^w \ y_1^w \ z_1^w \ 1]^T$, then the z_{i-1} -axis is oriented vertically when

$$x_1^w - x_0^w = 0, \quad y_1^w - y_0^w = 0 \quad (8)$$

From equation (6), $p_1^{i-1} - p_0^{i-1} = [0 \ 0 \ 1 \ 0]^T$. Using equation (7) gives

$$p_1^w - p_0^w = A_0^{i-1}(q) [p_1^{i-1} - p_0^{i-1}] \quad (9)$$

In order to satisfy the equations in (8), the x and y components of $(p_1^w - p_0^w)$ are set to the zero. Thus, equation (9) yields

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} A_0^{i-1}(q) [p_1^{i-1} - p_0^{i-1}] \quad (10)$$

Equation (10) represents the motion constraint that forces the z_{i-1} -axis to be in the vertical direction. Alternatively, the constraint equation (10) can be written as

$$h(q) = \begin{bmatrix} h_1(q) \\ h_2(q) \end{bmatrix} = 0_{2 \times 1} \quad (11)$$

where $0_{2 \times 1}$ denotes the zero vector, $h_1(q)$ and $h_2(q)$ represent the $(1,3)$ and $(2,3)$ elements, respectively, of matrix $A_0^{i-1}(q)$.

Since the task is specified by the m -vector p , and the manipulator motion must satisfy two additional constraint equations, determining the joint position q from $p = f(q)$ and $h(q) = 0_{2 \times 1}$ requires solving $(m+2)$ equations. In order to have a well-posed problem, the DOF n of the manipulator must be $n \geq m+2$. Therefore, the DOR should then be $r \geq 2$.

Assuming that DOR $r \geq 2$, solving the manipulator kinematic equation (5) subject to the constraint equation (11) will yield the desired manipulator configuration that forces the unactuated link to move on a horizontal plane. Thus, the solution to $p^d(t_f) = f[q^d(t_f)]$ and $h[q^d(t_f)] = 0_{2 \times 1}$ gives the desired joint position $q^d(t_f)$.

The control problem of moving the end-effector to the desired pose $p^d(t_f)$ will be solved in two phases. The next sections will describe phase 1 and phase 2 of the control scheme. During phase 1, the actuated joints are moved to the desired joint positions $q_A^d(T_1)$. In phase 2, the unactuated link is driven to the desired joint position $q_B^d(t_f)$ by means of the coupling effects.

4.2. Control in Phase 1

The motion during phase 1 consists of moving the actuated joints from position $q_A(0)$ to the desired position $q_A(T_1) = q_A^d(t_f)$ in time T_1 . The initial and the final joint velocities are $\dot{q}_A(0) = 0_{(n-1) \times 1}$ and $\dot{q}_A^d(T_1) = 0_{(n-1) \times 1}$, respectively. The value of $q_B(T_1)$ will be arbitrarily in the allowable range.

The control steps in phase 1 are :

1. The joint velocity profile $\dot{q}_A^d(t)$ is first chosen for the actuated joints.
2. Let the ν -space velocity represent the joint velocity of the actuated joints, i.e., $\nu(t) = \dot{q}_A(t)$.
3. $\Pi_A(t)$ in matrix $\Pi = [\Pi_A^T \Pi_B^T]^T$ is calculated by the first $n - 1$ rows in equation (3).
4. $\Pi_B(t)$ is determined using equation (3) and the last row of equation (2).
5. τ_ν is computed from the inverse dynamics by means of the reduced-order model in equation (4).
6. Using $\Pi^T \tau = \tau_\nu$, the joint torque τ is calculated; τ must then be generated by the joint actuators.

The details of the phase 1 method will now be discussed.

For the specified motion of the actuated joints, the desired velocity profile $\dot{q}_i^d(t)$ for joint i is chosen as shown in Figure 2, where $h_i = [q_i^d - q_i(0)]/(0.8T_1)$, $i \in S_A$. The corresponding desired joint acceleration profile $\ddot{q}_i^d(t)$ for $i \in S_A$ is then also specified.

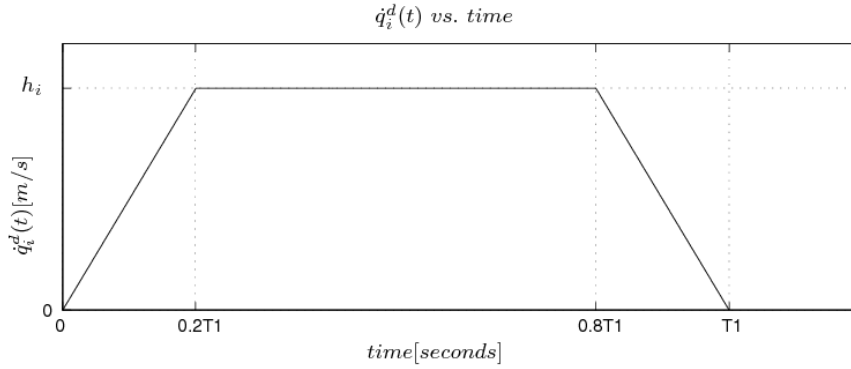


Figure 2. Desired Joint i Velocity for Phase 1, $i \in S_A$

Since the immediate concern during phase 1 is the motion of link i , $\forall i \in S_A$, the dimension of the task space in phase 1 is the number of actuated joints, i.e., the DOF of the task space is $m = n - 1$. Therefore, the order of the dynamical model can be reduced for the controller design. The resulting reduced-order model will then be used to design the control for the manipulator motion.

The $(n-1)$ -dimensional vector $\nu(t)$ is chosen so that $\nu(t) = \dot{q}_A(t)$. Hence, the corresponding desired velocities satisfy

$$\nu^d(t) = \dot{q}_A^d(t) \quad (12)$$

The desired pseudospace acceleration $\dot{\nu}^d(t)$ and position $\mu^d(t)$ can be computed by differentiating $\nu^d(t)$ and integrating $\nu^d(t)$, respectively.

Matrix $\Pi_A(t)$ is determined from equation (3). Consequently, $\Pi_A(t) = I_{(n-1) \times (n-1)}$, where $I_{(n-1) \times (n-1)}$ denotes the $(n-1) \times (n-1)$ identity matrix. The next step is to specify submatrix $\Pi_B(t)$.

Using equation (3), $\ddot{q}_A(t)$ and $\ddot{q}_B(t)$ can be expressed as,

$$\ddot{q}_A(t) = \frac{d}{dt}[\Pi_A(t)\nu(t)], \quad \ddot{q}_B(t) = \frac{d}{dt}[\Pi_B(t)\nu(t)] \quad (13)$$

The last row of equation (2) gives

$$0 = D_{BA} \frac{d}{dt}[\Pi_A(t)\nu(t)] + D_{BB} \frac{d}{dt}[\Pi_B(t)\nu(t)] + C_{BA}\Pi_A(t)\nu(t) + G_B \quad (14)$$

Assuming that D_{BA} , D_{BB} , C_{BA} , and G_B do not change much within a sampling interval, they may be considered constant in this interval. Then, by integrating equation (14) with respect to time, and solving for Π_B , one obtains for t within the sampling interval

$$\Pi_B(t) = \frac{[\nu^d(t)]^T}{[\nu^d(t)]^T \nu^d(t)} \int_0^t \left[\frac{D_{BA}\Pi_A \dot{\nu}^d(\sigma) + C_{BA}\Pi_A \nu^d(\sigma) + G_B}{-D_{BB}} \right] d\sigma \quad (15)$$

Having specified $\Pi_A(t) = I_{(n-1) \times (n-1)}$ and $\Pi_B(t)$ by equation (15), the reduced-order model in equation (4) can now be used to compute pseudospace torque τ_ν (the inverse dynamics). The joint torque τ for an underactuated manipulator is then obtained from $\tau_\nu = \Pi^T \tau$:

$$\begin{bmatrix} \tau_A \\ \tau_B \end{bmatrix} = \begin{bmatrix} \Pi_A^{-T} \tau_\nu \\ 0 \end{bmatrix} \quad (16)$$

where Π_A^{-T} denotes the inverse of Π_A^T . The calculated τ_A is then applied to the actuated joints of the manipulator to realize the tracking of the desired motion.

At the final time T_1 of phase 1, $q_A(T_1) = q_A^d(t_f)$, $\dot{q}_A(T_1) = 0_{(n-1) \times 1}$, $q_B(T_1) \neq q_B^d(t_f)$ and $\dot{q}_B(T_1) \neq 0$. The determination of the control for the underactuated manipulator during phase 2 will next be described.

4.3. Control in Phase 2

In order to produce a motion which is similar to the motion of the two-link planar underactuated manipulator, the n -link manipulator motion must satisfy the conditions outlined in Section 4.1. Therefore, one of the actuated joints and the unactuated joint have nonzero joint velocities during phase 2; the rest of the joints are maintained at zero velocities by appropriate joint torque inputs. It is noted, however, that the joints with zero velocities are still considered in the algorithm, since these joints require a generalized input torque to maintain the corresponding velocities at zero.

During phase 2, the actuated joint is driven by a cyclic input which causes the unactuated joint to move towards the desired joint position $q_B^d(t_f)$ by the mutual coupling effect between the two joints. Let the subscript nz denote the variables corresponding to the actuated link with a nonzero joint velocity; thus, for the actuated joint i_{nz} , velocity $\dot{q}_{nz}(t) \neq 0$, where $i_{nz} \in S_A$.

The motion during phase 2 consists of moving the joint (link) i_{nz} iteratively in a cyclic manner so that after each cycle, $q_B[T_1 + kT]$ is closer to $q_B^d(t_f)$ and $\dot{q}_B[T_1 + kT]$ approaches to zero, while satisfying $q_A[T_1 + kT] = q_A^d(t_f)$ and $\dot{q}_A[T_1 + kT] = 0_{(n-1) \times 1}$, for $k = 0, 1, 2, \dots$

The control steps in phase 2 are :

1. The velocity profile $\dot{q}_{nz}^d(t)$ is chosen for the actuated joint i_{nz} .
2. The desired ν -space velocity $\nu^d(t)$ is calculated.
3. $\Pi_A(t)$ in matrix $\Pi = [\Pi_A^T \Pi_B^T]^T$ is determined by the first $n - 1$ rows in equation (3).
4. $\Pi_B(t)$ is computed by equation (3) and the last row of equation (2).
5. By the inverse dynamics in reduced-order space, τ_ν is calculated.
6. Using $\Pi^T \tau = \tau_\nu$, the joint torque τ obtained; τ is then generated by the joint actuators to track the desired trajectory.

The details of the phase 2 method will now be discussed.

Let $t = T_{phase2}^{start}$ be the time at the beginning of phase 2; then, $q_A(T_{phase2}^{start}) = q_A^d(t_f)$ and $\dot{q}_A(T_{phase2}^{start}) = 0_{(n-1) \times 1}$. The velocity $\dot{q}_A^d(t)$ during phase 2 is chosen such that at the end of each iteration, the joint position $q_A(T_{phase2}^{start} + kT) = q_A^d(t_f)$ and velocity $\dot{q}_A(T_{phase2}^{start} + kT) = 0_{(n-1) \times 1}$. The velocity $\dot{q}_{nz}^d(t)$ for the actuated joint i_{nz} is shown in Figure 3, where $\dot{q}_{nz}^{max}$ is the maximum velocity of joint i_{nz} . Since only one of the actuated joints have a nonzero velocity, the joint position $q_i(t) = q_i^d(t_f)$ and the velocity $\dot{q}_i(t) = 0$, $T_{phase2}^{start} \leq t < t_f$ for every $i \in S_A$, $i \neq i_{nz}$. For convenience, the notation T_{phase2}^{start} will be dropped in the sequel.

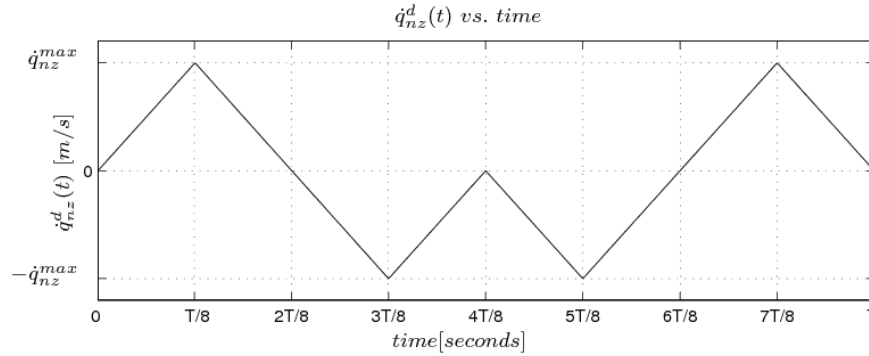


Figure 3. Desired Joint i_{nz} Velocity for Phase 2

The velocity of the unactuated joint of a two-link manipulator is derived in Appendix I when the joint one velocity is that shown in Figure 3. The relationship between the velocity of joint two at the beginning and at the end of an iteration cycle is given in equation (32). Since the conditions presented in Section 4.1 are satisfied, the n -link underactuated manipulator during phase 2 can be treated as a two-link underactuated manipulator. Equation (32) is rewritten here as

$$\dot{q}_B[(k+1)T] = \dot{q}_B(kT) + \Delta \dot{q}_B(kT) \quad (17)$$

$$\Delta \dot{q}_B(kT) = \frac{\bar{a}_2}{4\bar{a}_3} \sin[\bar{q}_B(kT)] (\dot{q}_{nz}^{max})^2 T \left[\frac{\bar{a}_2}{\bar{a}_3} \cos[\bar{q}_B(kT)] - \frac{1}{3} \right] \quad (18)$$

where $k = 0, 1, \dots$, and constants $\bar{a}_2$ and $\bar{a}_3$ are determined by the link properties and $q_A^d(t_f)$. It is noted that for a two-link manipulator the joint two position q_2 is the angle between the axes of links one and two. Furthermore, the links between joint i_{nz} and the unactuated joint of an n -link manipulator can be treated as one virtual link analogous to the link one of the two-link manipulator. Now, $\bar{q}_B(t)$ can be defined as

$$\bar{q}_B(t) = q_B(t) + q_B^{offset} \quad (19)$$

where q_B^{offset} is a constant dependent on the link properties and $q_A^d(t_f)$. The requirement that $q_B(kT)$ approaches $q_B^d(t_f)$ is equivalent to the requirement that $\bar{q}_B(kT)$ approaches $\bar{q}_B^d(t_f) = q_B^d(t_f) + q_B^{offset}$. In order to make $\bar{q}_B(kT)$ approach $\bar{q}_B^d(t_f)$ and $\dot{q}_B(kT)$ approach zero at the end of each iteration, the following penalty function approach can be used. If $x(t) = [\bar{q}_B(t) \dot{q}_B(t)]^T$ and $x^d(t_f) = [\bar{q}_B^d(t_f) \dot{q}_B^d(t_f)]^T$, then one can define a criterion (a penalty function $\mathcal{C}$) which is to be minimized with respect to $x(kT)$.

$$\mathcal{C}[x(kT)] = \frac{1}{2}[x^d(t_f) - x(kT)]^T W [x^d(t_f) - x(kT)] \quad (20)$$

where $k = 1, \dots$, $W = \text{diag}(W_1, W_2)$, and W_1, W_2 are nonnegative weighting factors. Forming the first-order differential of equation (20) using $\Delta x = [\Delta \bar{q}_B \Delta \dot{q}_B]^T$, one obtains

$$\tilde{\mathcal{C}} = \left. \frac{\partial \mathcal{C}}{\partial x} \right|_{x(kT)} \Delta x(kT) + \frac{1}{2} \Delta x^T(kT) \left. \frac{\partial^2 \mathcal{C}}{\partial x^2} \right|_{x(kT)} \Delta x(kT) \quad (21)$$

Since $\Delta \bar{q}_B(kT) = \bar{q}_B[(k+1)T] - \bar{q}_B(kT)$, one can write an approximation as

$$\Delta \bar{q}_B(kT) = \dot{q}_B(kT)T + \frac{1}{2} \Delta \dot{q}_B(kT)T^2 \quad (22)$$

Using equation (22) in equation (21), and minimizing $\tilde{\mathcal{C}}$ with respect to $\Delta \dot{q}_B(kT)$ results in

$$\Delta \dot{q}_B(kT) = -\frac{\beta_k}{2\alpha_k} \quad (23)$$

where $\beta_k = -\frac{1}{2}W_1[\bar{q}_B^d(t_f) - \bar{q}_B(kT)]T^2 - W_2[\dot{q}_B^d(t_f) - \dot{q}_B(kT)] + \frac{1}{2}W_1\dot{q}_B(kT)T^3$ and $\alpha_k = \frac{1}{2}(\frac{1}{4}W_1T^4 + W_2)$.

After $\Delta \dot{q}_B(kT)$ is computed by equation (23), $\dot{q}_{nz}^{max}$ can be calculated from equation (18). Since the sign of $\Delta \dot{q}_B(kT)$ in equation (18) depends on the angle $\bar{q}_B(kT)$, the minimizing value of $\Delta \dot{q}_B(kT)$ in equation (23) may not be feasible due to the sign constraint in equation (18). In this case, $\dot{q}_{nz}^{max}$ is selected as zero. It follows that $\Delta \dot{q}_B(kT) = 0$. Thus, $\dot{q}_B[(k+1)T] = \dot{q}_B(kT) \neq 0$ which allows the unactuated joint to move to a different position. The value $\dot{q}_{nz}^{max}$ is maintained at zero until the unactuated joint moves to a position, where $\Delta \dot{q}_B(kT)$ by equation (23) is compatible with equation (18).

After $\dot{q}_{nz}^{max}$ has been calculated, the velocity profile $\dot{q}_{nz}^d(t)$ in Figure 3 is now specified. Furthermore, the desired velocity for joint i , $i \in S_A$, $i \neq i_{nz}$ is chosen as $\dot{q}_i^d(t) = 0$.

The next step following the determination $\dot{q}_A^d(t)$ is to specify the pseudospace velocity $\nu^d(t)$. Since only one actuated joint has a nonzero velocity, $\nu_i^d(t) = \dot{q}_i^d(t) = 0$ for every $i \in S_A$, $i \neq i_{nz}$. However, from equation (3), in order to have $\dot{q}_{nz}^d(t) \neq 0$, the desired ν -space velocity $\nu^d(t)$ must be nonzero. It can be chosen as a straight line satisfying $\nu_{nz}^d(kT) = \dot{q}_B(kT)$

at the start of the iteration, and $\nu_{nz}^d[(k+1)T] = \varepsilon \dot{q}_B(kT)$ at the end of the iteration, where ε is a small positive number. After $\dot{q}_A^d(t)$ and $\nu^d(t)$ have been specified, submatrix $\Pi_A(t)$ can be determined to satisfy equation (3). Since the inverse transpose of $\Pi_A(t)$ is needed in equation (16), matrix $\Pi_A(t)$ must be nonsingular. $\Pi_A(t)$ can be selected as a diagonal matrix with nonzero diagonal elements satisfying equation (3). Based on $\dot{q}_{nz}^d(t)$ and $\nu_{nz}^d(t)$, and $\nu_i^d(t) = \dot{q}_i^d(t)$ for every $i \in S_A$, $i \neq i_{nz}$,

$$\begin{aligned} \Pi_A &= \text{diag}(d_1, d_2, \dots, d_{n-1}) \\ d_i &= \begin{cases} \dot{q}_{nz}^d(t)/\nu_{nz}^d(t) & i = i_{nz} \\ 1 & i = 1, \dots, (n-1), i \neq i_{nz} \end{cases} \end{aligned} \quad (24)$$

$\Pi_B(t)$ is then computed by equation (14) during phase 2. It is noted that $\Pi_A(t)$ varies with time. Assuming that D_{BA} , D_{BB} , C_{BA} , and G_B are approximately constant within a sampling interval, integrating equation (14) with respect to time and then solving for Π_B gives for t within the sampling interval,

$$\Pi_B(t) = \frac{[\nu^d(t)]^T}{[\nu^d(t)]^T \nu^d(t)} \int_0^t \left[\frac{D_{BA} \Pi_A \dot{\nu}^d(\sigma) + D_{BA} \dot{\Pi}_A \nu^d(\sigma) + C_{BA} \Pi_A \nu^d(\sigma) + G_B}{-D_{BB}} \right] d\sigma \quad (25)$$

After $\Pi(t)$ has been specified, the ν -space torque τ_ν can be computed using the reduced-order model in equation (4). The underactuated manipulator torque τ is computed using equation (16). Then, the calculated value of τ_A is applied to the manipulator.

The application of the proposed two-phase method to the control of a PUMA 600 manipulator will next be described.

5. PUMA 600 ARM SIMULATIONS

The proposed two-phase approach will be applied to the motion control of the PUMA 600 manipulator. The control problem is as follows : the end-effector position $p = [p_x \ p_y \ p_z]^T$ of a PUMA 600 Arm is to be moved from an initial position $p(0) = [0.125 \ -0.235 \ 0.806]^T m$ to a desired position $p^d(t_f) = [0.594 \ 0.288 \ -0.393]^T m$, in time $t_f = 35 \ s$. At the start of the motion, three kinematic equations relating the end-effector position to the joint position are considered when solving for the joint variables. The initial joint position $q(0)$ is calculated from the inverse kinematics; thus, $q(0) = [-1.57 \ -1.05 \ 1.05 \ 0.20 \ 3.00 \ 0.52]^T rad$. The initial joint velocity $\dot{q}(0) = 0_{6 \times 1} rad/s$. Before the failure, the joints are driven to the desired position $\tilde{q}(t_f)$, where $\tilde{q}(t_f)$ satisfies the kinematic equation $p(t_f) = f[\tilde{q}(t_f)]$. Then, $\tilde{q}^d(t_f) = [0.19 \ 1.07 \ 0.44 \ 1.11 \ 1.64 \ 0.70]^T rad$. At time $t_1 = 2.5 \ s$, the fifth joint actuator fails (i.e. it stops producing a torque), while the other joints are actuated during the entire duration of the motion. After the failure of the joint five actuator ($t \geq t_1$), the conditions of Section 4.1 must be taken into account in order to design a controller for the specified task. Specifically, $q^d(t_f)$ is computed such that the unactuated link five moves in the horizontal plane $z = p_z^d(t_f)$. The six components of the desired joint position $q^d(t_f)$ is calculated using the three kinematic equations $p^d(t_f) = f[q^d(t_f)]$ and the two constraint equations $h[q^d(t_f)] = 0_{2 \times 1}$ in equation (11). Thus, $q^d(t_f) = [0.26 \ 1.05 \ 0.52 \ 1.57 \ 1.96 \ 0.78]^T rad$.

The application of the proposed control algorithm is illustrated by simulations on a PUMA 600 manipulator. The numerical values of the link parameters of the model used in the simulations are given in [11]. A sampling period of 10 *ms* is used in the simulation runs, and $T_1 = 5$ s and $T = 1$ s. The weighting factors are $W_1 = 2$, $W_2 = 1$, and $\varepsilon = 10^{-3}$. Furthermore, $\bar{a}_2/\bar{a}_3 = 1.0869$ and $q_B^{offset} = -0.1910$ rad from Appendix II.

At the start of the manipulator motion, all joints are operational. The joints are driven to the desired position $\tilde{q}(t_f)$. However, at time t_1 , the joint five actuator fails; thus $\tau_5(t) = 0$ for $t \geq t_1$ s. For phase 1, before the joint actuator failure, the graphs of the joint positions, velocities and torques for joints one and five are shown in Figure 4. From the graphs of $\dot{q}_1(t)$ and $\dot{q}_5(t)$ for $t \in [0, t_1]$ s, it can be seen that joint velocities track the desired trajectory shown in Figure 2. However, for time $t \geq t_1$ s, $\tau_5(t) = 0$; then, the fifth link is a free-swinging arm. The objective is to design a controller that allows the manipulator to perform the specified motion in spite of the failure.

Since the $p \in \mathbf{R}^3$, the DOF of the specified task is $m = 3$. The DOF of the manipulator is $n = 6$; thus, the DOR $r = 3$. Therefore, the manipulator should be able to perform the desired task even though the actuator on joint five is nonfunctional. It is accomplished by using the two-phase controller described.

In this case, $q_A = [q_1 \ q_2 \ q_3 \ q_4 \ q_6]^T$, $q_B = q_5$, and $\tau_A = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \ \tau_6]^T$. The submatrices in equation (2) are specified accordingly.

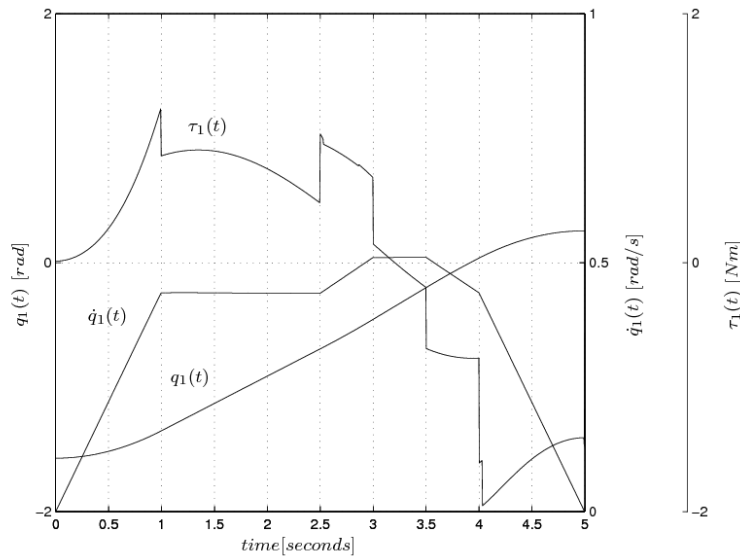
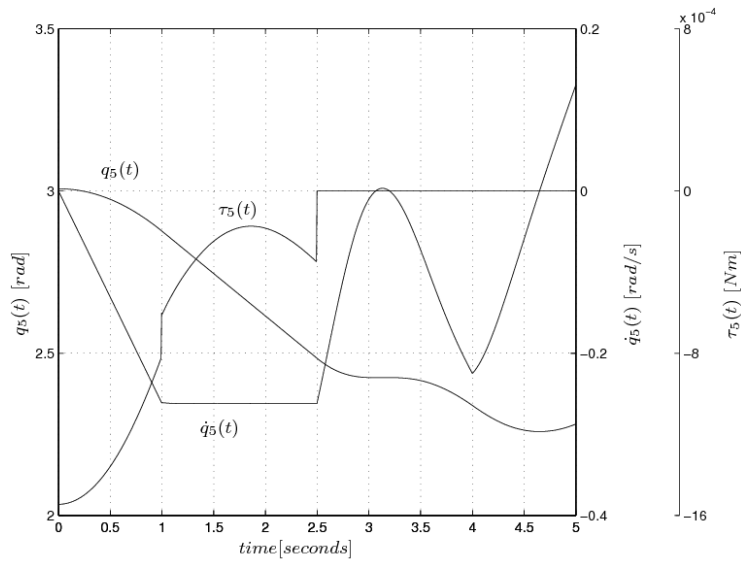
After the failure, the actuated joints are moved in phase 1 to the desired position $q_A^d(T_1) = q_A^d(t_f)$ using the method described in Section 4.2. Figure 4 graphs $q_1(t)$, $\dot{q}_1(t)$ and $\tau_1(t)$ for phase 1. It can be seen that the position and velocity of joint one at the end of phase 1 are $q_1(T_1) = q_1^d(t_f)$ and $\dot{q}_1(T_1) = 0$, respectively. The graphs of the positions, velocities and torques for joints two, three, four and six are similar to those obtained for joint one. The graphs of $q_5(t)$, $\dot{q}_5(t)$ and $\tau_5(t)$ for the unactuated joint five are shown in Figure 5.

During phase 2, the joint one has a nonzero joint velocity, while the other actuated joints have zero velocities. Using the method described in Section 4.3, joint five is driven towards the desired position $q_B^d(t_f)$ in phase 2. The joint positions, velocities, and torques for joints one and two are graphed in Figures 6 and 7, respectively. It can be observed from Figure 6 that $q_1(5) = q_1(6) = q_1^d(t_f)$ and $\dot{q}_1(5) = \dot{q}_1(6) = 0$. Figure 7 displays the torque $\tau_2(t)$ required to keep the velocity $\dot{q}_5(t) = 0$ for phase 2. Graphs similar to those in Figure 7 are obtained for the positions, velocities and torques of joints three, four and six. Figure 8 graphs $q_5(t)$, $\dot{q}_5(t)$ and $\tau_5(t)$; it shows the motion of the unactuated joint due to the coupling effect.

Figures 9 and 10 display the joint position $q(t)$ and velocity $\dot{q}(t)$, respectively, over the duration of the motion. They show that the unactuated joint position $q_5(t)$ approaches to $q_5^d(t_f)$ and $\dot{q}_5(t)$ to zero, while the actuated joints are at the desired positions $q_A^d(t_f)$, and the velocities are zero, at the end of each iteration.

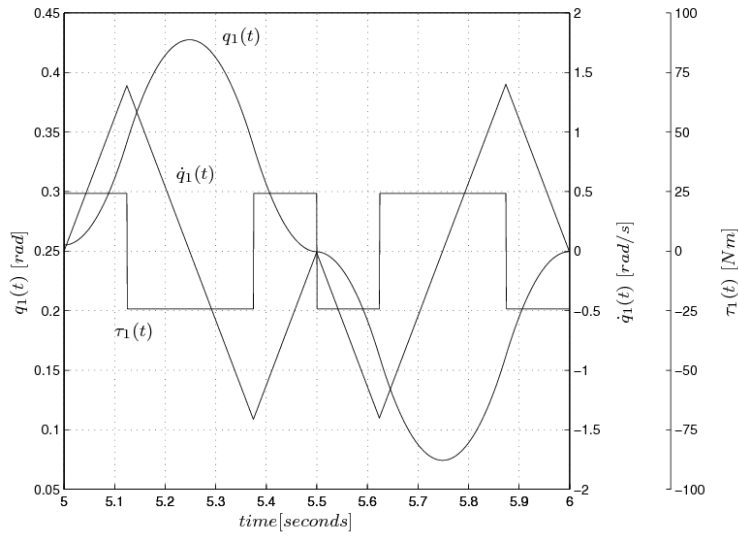
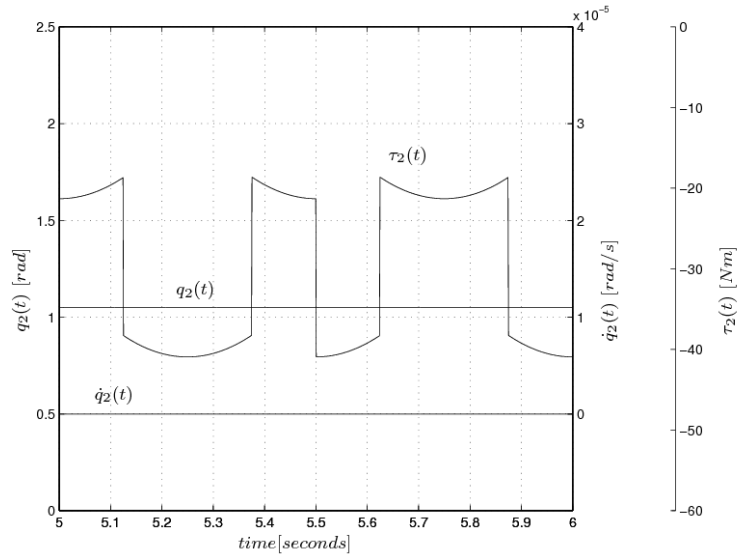
6. DISCUSSION

The control of the underactuated redundant n -link serial manipulator is accomplished by using a two-phase method (assuming there are no brakes on the joints). The redundancy permits the manipulator to perform the desired task even if a free-swinging joint failure occurs during the motion. A constraint equation is derived which forces the unactuated link to move in a horizontal plane. Then, the desired joint position $q^d(t_f)$ is solved from the kinematic equation

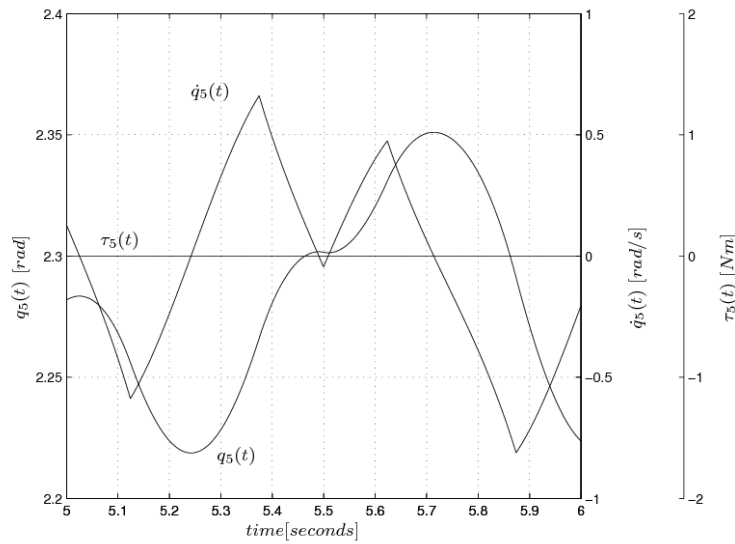
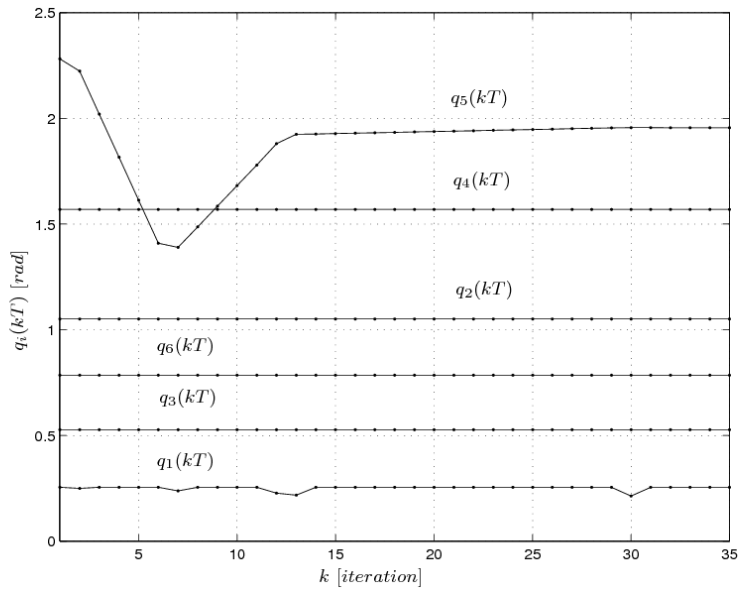

 Figure 4. Graph of $q_1(t)$, $\dot{q}_1(t)$, $\tau_1(t)$ vs. time for Phase 1

 Figure 5. Graph of $q_5(t)$, $\dot{q}_5(t)$, $\tau_5(t)$ vs. time for Phase 1

and the constraint equation. Furthermore, the redundancy allows the derivation of a low-order dynamical model (reduced-order model) for the manipulator.

Using a two-phase control scheme, the actuated joints and the unactuated joint are driven

Figure 6. Graph of $q_1(t)$, $\dot{q}_1(t)$, $\tau_1(t)$ vs. time for Phase 2Figure 7. Graph of $q_2(t)$, $\dot{q}_2(t)$, $\tau_2(t)$ vs. time for Phase 1

to the desired positions in two independent stages. The separate control of the actuated joints and unactuated joint simplifies the overall control approach. In the control of the actuated joints in phase 1, the motion of the unactuated joint is not considered. When controlling the unactuated joint in phase 2, the control problem for the n -link underactuated manipulator is


 Figure 8. Graph of $q_5(t)$, $\dot{q}_5(t)$, $\tau_5(t)$ vs. time for Phase 2

 Figure 9. Graph of $q_i(kT)$, $i = 1, \dots, 6$

approached from the perspective of a two-link manipulator.

In phase 1, the actuated joints are moved to the desired position $q_A^d(t_f)$. During phase 2, the unactuated link is driven to the desired position $q_B^d(t_f)$ by the coupling effect. The phase

1 and phase 2 methods use the reduced-order model of a manipulator to determine the control inputs to the actuated joints. The transformation matrix Π is determined from the desired joint and reduced-order space velocities, and the nonholonomic dynamic constraint imposed by the unactuated joint. In the previous reports [6, 13] on the control of underactuated planar manipulators, partial feedback linearization is used as an initial step in the analysis of the systems. Although this is a straightforward approach, the derivation of the partial feedback linearized system often requires non-trivial symbolic manipulations of the dynamic equations. The use of the reduced-order model avoids the need for partial feedback linearization. Also, the reduced-order model simplifies the extension of the algorithm to the six-link manipulator case.

Furthermore, the expression for $\Delta\dot{q}_2$ in equation (32) is derived based on the dynamics with $\dot{q}_1$ treated as the input. In [6], a similar expression for $\Delta\dot{q}_2$ is derived using nilpotent approximation. However, to obtain a nilpotent approximation for a system, a set of privilege coordinates must be constructed to transform the original system to the nilpotent approximation form. The construction of a set of privilege coordinates for a system without drift is outlined in [1] and applied to mobile robots. It is felt that the approach presented here is easier to use. Thus, the control algorithm presented offers an advantage in controlling an underactuated manipulator and it is an attractive method to achieve a dynamically fault-tolerant operation.

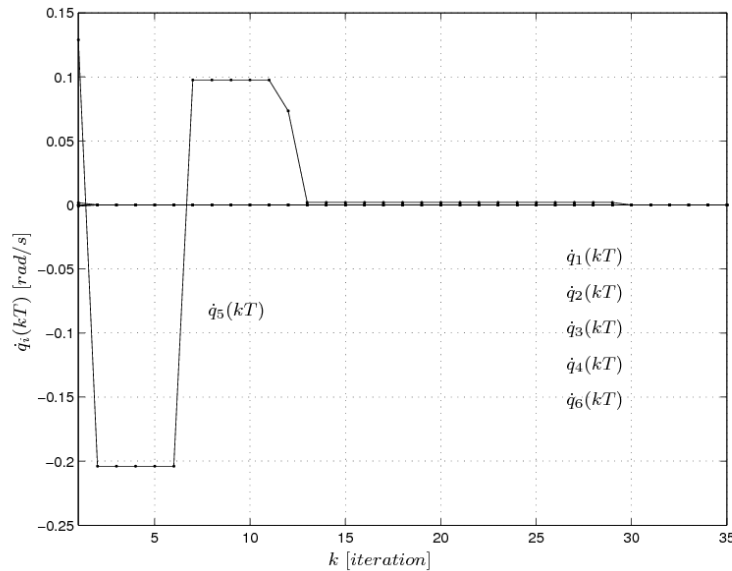


Figure 10. Graph of $\dot{q}_i(kT)$, $i = 1, \dots, 6$

7. SUMMARY AND CONCLUSION

The control objective is to move the end-effector from an initial pose to specified final pose. The redundant manipulator is required to accomplish the desired task even if a joint failure occurs during the motion. The dynamics of an underactuated manipulator is described using the reduced-order model. The conditions and degree of redundancy necessary for the manipulator to achieve fault tolerance are discussed. The reduced-order model is then used to design controllers for an underactuated n -link serial manipulator. The manipulator end-effector is driven to the desired pose in two control phases. During phase 1, the actuated joints are moved to their desired joint positions. In phase 2, a cyclic control input is applied to one of the actuated joints; the coupling effect iteratively drives the unactuated link to the desired joint position. The proposed two-phase method allows the manipulator to achieve the desired goal in the case that an actuator of a redundant manipulator fails. The control method is illustrated by simulations of a PUMA 600 manipulator.

REFERENCES

1. A. Bellaiche, J-P. Laumond, M. Chyba, Canonical Nilpotent Approximation of Control Systems : Application to Nonholonomic Motion Planning. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2694-2699, San Antonio, Texas, 1993.
2. R.W. Brockett, R.S. Millman, and H.J. Sussmann, editors. *Differential Geometric Control Theory*, pages 181-191, Birkhauser, Boston, MA, 1983.
3. J.D. English and A.A. Maciejewski, Fault Tolerance for Kinematically Redundant Manipulators : Anticipating Free-Swinging Joint Failures, *IEEE Transactions on Robotics and Automation*, 14(4):566-575, Aug 1998.
4. A.J. Koivo, *Fundamentals for Control Robotic Manipulators*, J. Wiley & Sons, New York, NY, 1989.
5. A.J. Koivo and S.H. Arnaudovic, Control of Redundant Manipulators with Constraints Using a Reduced-Order Model, *Automatica*, 30(4):665-677, 1994.
6. A. De Luca, R. Mattone, and G. Oriolo, Control of Underactuated Mechanical Systems : Application to the Planar 2R Robot, In *Proceedings of the 35th Conference on Decision and Control*, pages 1455-1460, Kobe, Japan, December 1996.
7. P. Lucibello and G. Oriolo, Stabilization via Iterative State Steering with Application to Chained-Form Systems, *Proceedings of the 35th Conference on Decision and Control*, pages 2614-2619, Kobe, Japan, December 1996.
8. H. Nijmeijer and A.J. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, New York, NY, 1990.
9. G. Oriolo and Y. Nakamura, Control of Mechanical Systems with Second-Order Nonholonomic Constraints : Underactuated Manipulators, In *Proceedings of 30th IEEE Conference on Decision and Control*, pages 2398-2403, Brighton, UK, 1991.
10. C.J. Paredis and P.K. Khosla, Global Trajectory Planning for Fault Tolerant Manipulators, In *Proceedings IEEE Conference on Intelligent Robots and Systems*, Vol. 2, pp. 428-434, Aug. 1995.
11. R.P. Paul. *Robot Manipulators* MIT Press, 1981.
12. M.C. Ramos and A.J. Koivo. Fuzzy logic based optimization for redundant manipulators. *IEEE Transactions on Fuzzy Systems*, 10(4):498-509, August 2002.
13. M. Reyhanoglu, A. van der Schaft, N.H. McClamroch and I. Kolmanovsky, Nonlinear Control of a Class of Underactuated Systems, *Proceedings of the 35th Conference on Decision and Control*, pages 1682-1687, Kobe, Japan, December 1996.
14. R.G. Roberts, R.S. Jamisola, Jr, A.A. Maciejewski. Failure-Tolerant Path Planning for Kinematically Redundant Manipulators Anticipating Locked-Joint Failures, *IEEE Transactions on Robotics*, 22(4):603-612, Aug. 2006.
15. M.W. Spong, Energy Based Control of a Class of Underactuated Mechanical Systems. In *Proceedings of the 13th IFAC World Congress*, pages 431-435, San Francisco, CA, 1996.

APPENDIX

I. Derivation of Equation 9

The dynamical model of the two-link underactuated manipulator moving horizontally is given as

$$\begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (26)$$

where

$$\begin{aligned} D_{11} &= a_1 + 2a_2 \cos(q_2) & C_{11} &= -2a_2 \sin(q_2)\dot{q}_2 \\ D_{12} &= a_3 + a_2 \cos(q_2) & C_{12} &= -a_2 \sin(q_2)\dot{q}_2 \\ D_{21} &= D_{12} & C_{21} &= a_2 \sin(q_2)\dot{q}_1 \\ D_{22} &= a_3 & C_{22} &= 0 \\ a_1 &= m_1 l_1^2/4 + m_2(l_1^2 + l_2^2/4) + I_1 + I_2 \\ a_2 &= m_2 l_1 l_2/2 \\ a_3 &= m_2 l_2^2/4 + I_2 \end{aligned}$$

The joint position variable is defined as $q = [q_1 \ q_2]^T$ and the torque applied to the first joint is represented by τ_1 . The constants m_i , l_i and I_i are link parameters with $i = 1, 2$. The accessibility property of the system represented in equation (26) can be checked using Lie algebra [8].

The derivation of $\Delta \dot{q}_2$ is based on the second equation of the dynamical model in equation (26). Writing out the terms and solving for $\ddot{q}_2$ gives

$$\ddot{q}_2(t) = -\frac{a_3 + a_2 \cos(q_2(t))}{a_3} \ddot{q}_1(t) - \frac{a_2 \sin[q_2(t)]}{a_3} \dot{q}_1^2(t) \quad (27)$$

Using the joint one velocity profile in Figure 3, $\dot{q}_1(t) = \pm \dot{q}_1^{max}/(T/8)$. To simplify the integration of equation (27), the joint one velocity profile is divided into eight time segments of length $T/8$. The contribution of each segment is computed as follows. Integrating equation (27) for one time segment gives,

$$\begin{aligned} \dot{q}_2[(n+1)T/8] &= \dot{q}_2(nT/8) - \frac{a_3 + a_2 \cos[q_2(nT/8)]}{a_3} (\pm \dot{q}_1^{max}) \\ &\quad - \frac{a_2 \sin[q_2(nT/8)]}{a_3} \frac{(\dot{q}_1^{max})^2}{3} \frac{T}{8} \end{aligned} \quad (28)$$

Integrating again yields

$$\begin{aligned} q_2[(n+1)T/8] &= q_2(nT/8) + \dot{q}_2(nT/8) \frac{T}{8} \\ &\quad - \frac{a_3 + a_2 \cos[q_2(nT/8)]}{a_3} \frac{(\pm \dot{q}_1^{max})}{2} \frac{T}{8} \\ &\quad - \frac{a_2 \sin[q_2(nT/8)]}{a_3} \frac{(\dot{q}_1^{max})^2}{12} \left(\frac{T}{8}\right)^2 \end{aligned} \quad (29)$$

To simplify notation let $q_2^n = q_2(nT/8)$, $q_2^{n+1} = q_2[(n+1)T/8]$, $\dot{q}_2^n = \dot{q}_2(nT/8)$, $\dot{q}_2^{n+1} = \dot{q}_2[(n+1)T/8]$. Dropping the higher order term in equation (29) and rewriting the equations,

$$\dot{q}_2^{n+1} = \dot{q}_2^n - \frac{a_3 + a_2 \cos(q_2^n)}{a_3} (\pm \dot{q}_1^{max}) - \frac{a_2 \sin(q_2^n)}{a_3} \frac{(\dot{q}_1^{max})^2}{3} \frac{T}{8} \quad (30)$$

$$q_2^{n+1} = q_2^n + \dot{q}_2^n \frac{T}{8} - \frac{a_3 + a_2 \cos(q_2^n)}{a_3} \frac{(\pm \dot{q}_1^{max})}{2} \frac{T}{8} \quad (31)$$

Approximately, $\cos(\theta + \Delta\theta) = \cos(\theta) - \sin(\theta)\Delta\theta$ and $\sin(\theta + \Delta\theta) = \sin(\theta) + \cos(\theta)\Delta\theta$. Using equations (30) and (31) iteratively for each time segment yields the following expression

$$\dot{q}_2^8 = \dot{q}_2^0 + \frac{a_2}{4a_3} \sin(q_2^0) (\dot{q}_1^{max})^2 T \left[\frac{a_2}{a_3} \cos(q_2^0) - \frac{1}{3} \right] \quad (32)$$

where q_2^0 and $\dot{q}_2^0$ correspond to initial joint two position and initial joint two velocity at the start of the phase II iteration, respectively. The variable $\dot{q}_2^8$ corresponds to the final joint two velocity at the end of each phase II iteration.

II. Constants : $\bar{a}_2$, $\bar{a}_3$ and q_B^{offset}

In order to implement the phase 2 control scheme, the constants $\bar{a}_2$, $\bar{a}_3$ and q_B^{offset} in equations (18) and (19) are computed. These equations determine the motion of the unactuated link when a cyclic input is applied to an actuated joint. Before proceeding, the following notations will be used to simplify the discussion.

1. The rotation of link i_{nz} is about the z_e -axis, where $e = i_{nz} - 1$.
2. The variables associated with the unactuated joint are denoted with the subscript ua ; then, $q_{ua}(t) = q_B(t)$ is the position of the unactuated joint i_{ua} at time t . The rotation of link i_{ua} is about the z_f -axis, where $f = i_{ua} - 1$.

The constant $\bar{a}_2$ in equation (18) will be determined based on the definition of a_2 in equation (26). Thus,

$$\bar{a}_2 = l_A \bar{l}_B \sum_{j=i_{ua}}^n m_j \quad (33)$$

where m_j is the mass of link j .

Constant l_A in equation (33) is the perpendicular distance between the z_e -axis and the z_f -axis, with $r < s$, as shown in Figure 11.

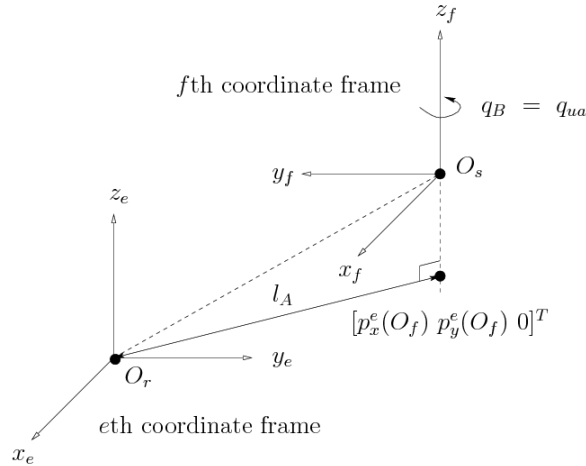


Figure 11. Coordinate Frames Used for Determining $\bar{a}_2$ and $\bar{a}_3$

Then, expressing the origin of the f th coordinate frame O_f in e th coordinate frame as $p^e(O_f)$,

$$p^e(O_f) = [p_x^e(O_f) p_y^e(O_f) p_z^e(O_f) 1]^T = A_e^f(q) [0 \ 0 \ 0 \ 1]^T \quad (34)$$

Now the length l_A can be calculated as $l_A = \sqrt{[p_x^e(O_f)]^2 + [p_y^e(O_f)]^2}$.

Constant $\bar{l}_B$ in equation (33) is the perpendicular distance from the z_f -axis to the composite center of gravity of the links between the unactuated joint and the end-effector. Let $\bar{c}_i = [\bar{x}_i \ \bar{y}_i \ \bar{z}_i]^T$ be the center of gravity (CG) of the i th link in i th coordinate frame for $i_{ua} \leq i \leq n$. Using the transformation matrix $A_f^i(q)$ to express $\bar{c}_i$ in the f th coordinate frame yields for $i_{ua} \leq i \leq n$,

$$\begin{bmatrix} \bar{c}_i^f \\ 1 \end{bmatrix} = A_f^i(q) \begin{bmatrix} \bar{c}_i \\ 1 \end{bmatrix} \quad (35)$$

where $\bar{c}_i^f$ is the CG of the i th link in the f th coordinate frame. Then, the composite center of gravity $\bar{c}_B^f = [\bar{x}_B^f \ \bar{y}_B^f \ \bar{z}_B^f]^T$ is given by

$$\bar{c}_B^f = \frac{1}{m_B} \sum_{i_{ua}}^n \bar{c}_i^f$$

Thus, $\bar{l}_B$ in equation (33) is calculated as $\bar{l}_B = \sqrt{(\bar{x}_B^f)^2 + (\bar{y}_B^f)^2}$.

After calculating the constants l_A and $\bar{l}_B$, $\bar{a}_2$ in equation (33) can now be obtained.

Constant $\bar{a}_3$ in equation (18) will be determined based on the definition of a_3 in equation (26). Thus, $\bar{a}_3 = D_{ii}$, $i = i_{ua}$,

where D_{ii} is the (i, i) element of the pseudo-inertia matrix $D(q)$.

Constant q_B^{offset} used in equation (19) will next be derived. In a two-link planar manipulator, The joint two position q_2 is the angle between the axes of links one and two in a two-link planar manipulator. Thus, $\bar{q}_B$ in equation (18) is the angle between the line $O_e O_f$ and the axis of link i_{ua} as shown in Figure 12. Then,

$$\bar{q}_B = \theta_f + q_{ua} - \theta_{offset} \quad (36)$$

where θ_{offset} is the angle between the x_i -axis, $i = i_{ua}$, and the axis of link i_{ua} . Furthermore, θ_f is the angle between the line $O_e O_f$ and the x_f -axis. It is noted that θ_{offset} is dependent on how the coordinate axes are defined for a specific manipulator. A comparison of $\bar{q}_B$ in equations (19) and (36), and noting that $q_B = q_{ua}$, gives

$$q_B^{offset} = \theta_f - \theta_{offset} \quad (37)$$

To determine θ_f , a point $\hat{x}_f = [1 \ 0 \ 0]^T$ on the x_f -axis is expressed in the e th coordinate frame. Thus,

$$[p_x^e(\hat{x}_f) \ p_y^e(\hat{x}_f) \ p_z^e(\hat{x}_f) \ 1]^T = A_e^f(q) \hat{x}_f \quad (38)$$

Then, θ_f can be computed as

$$\theta_f = \tan^{-1} \left[\frac{p_y^e(\hat{x}_f) - p_y^e(O_f)}{p_x^e(\hat{x}_f) - p_x^e(O_f)} \right] - \tan^{-1} \left[\frac{p_y^e(O_f)}{p_x^e(O_f)} \right] \quad (39)$$

where $p_x^e(O_f)$ and $p_y^e(O_f)$ are specified in equation (34).

This section determines the general expressions for the constants $\bar{a}_2$, $\bar{a}_3$ and q_B^{offset} used in equations (18) and (19). During implementation, the coordinate transformation matrices

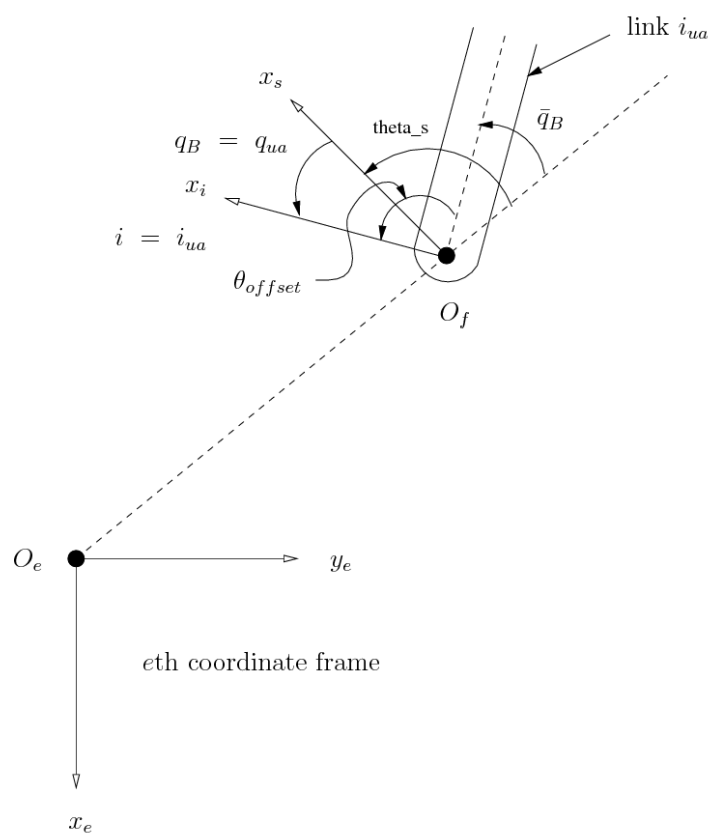


Figure 12. Coordinate Frames Used for Determining q_B^{offset}

$A_i^j(q)$ appearing in the derivations in this section are evaluated the using the desired joint position $q^d(t_f)$; then, $\bar{a}_2$, $\bar{a}_3$ and q_B^{offset} are calculated.