

A CONTROL STRUCTURE FOR FAULT-TOLERANT OPERATION OF ROBOTIC MANIPULATORS

Yung Ting, Sabri Tosunoglu, and Delbert Tesar

DOE/NE/37966--2

Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas 78712

DE93 005783

Abstract

Failure of any component of a robotic system during operation is a matter of concern. This work investigates internal shock phenomena due to the failure of joint actuation, and a recovery algorithm for both serial and parallel mechanisms under such circumstances. A control algorithm is studied that consists of a model reference algorithm and computed torque method in the feedforward process, and a simple PID controller in the feedback process. Simulation results illustrate the effectiveness of this recovery algorithm which attempts to reduce the internal shock when failure occurs, and accomplish the tracking of the given end-effector trajectory. The outlined recovery algorithms, which include two stages of robot control, path planning and path tracking, are expected to be applied not only to a case where some joint is fully failed, but also to cases where some joints experience partial failure.

1. Introduction

Fault tolerance has matured as an area of investigation in various fields; particularly those of aerospace systems [1] and computers [2]. In robotics, its implementation to space applications has received considerable interest lately. Some researchers have sought to address specific issues [3-6]. However, fault tolerance being a systems problem, an integrated approach is seen to be necessary. For instance, a framework to support a fault-tolerant robot such that the robot can carry out fault detection, fault diagnosis, and recovery through reconfiguration has been proposed in [7] based on the requirements outlined in [8, 9]. Also, a four-level canonical architecture for fault tolerance is being developed. These levels are listed as

- I. Dual joint actuators
- II. Parallel-structured modules
- III. Redundant manipulators
- IV. Multiple cooperating arms

In each of these levels, redundancies are designed to meet the demanding objectives of fault tolerance. Operating over these architectural levels, an ideal controller is required to be able to overcome the internal shocks due to a sudden internal component failure such as a failed actuator or prime mover sub-system and minimize the disturbance to the system's function. Because of the on-line control function, the control algorithm must necessarily be easily

transformed with new recovery configuration when a fault occurs.

In this work, identification of response phenomena due to the failure of joint actuators, and then definition of the recovery process were studied for both serial and parallel mechanisms. The internal shock problem was simulated by assuming that an actuator failed during manipulation. Simulations revealed that several disturbances occur upon failure. The first one is caused by the shock of failure, which may be of a large magnitude. The second one comes from implementing an emergency function at the start of recovery such as the braking action for the serial robot or torque redistribution for the parallel robot. The third one results from the increase of model (parametric) errors. This disturbance is due to the configuration change following failure. The fourth one is caused by a sudden set point change since the recovery process immediately attempts to drive the robot back to a new desired joint-space trajectory.

Initially, a model-based algorithm using the computed torque method as a feedforward controller, combined with a PID feedback controller was employed for evaluating the system performance. A planar 4-link redundant serial manipulator, and a planar 5-bar parallel manipulator were simulated for various PID controller gains and different recovery times. According to the simulation results, efficient prototypes of control algorithms for fault-tolerant robots were outlined.

The recovery process following the full failure of a joint, which is covered in this paper, can be extended to cases where some joints experience only partial failure. Here, the partial failure means that the robotic system can still sustain the remaining manipulation without locking the failed joints for the serial robot or without switching the minimum set of active joints for the parallel robot. In other words, without isolating the failed joints the demanded manipulation can be maintained by slowing down the manipulation or by redistributing the load via a torque optimization technique. Although simulations in this study focus on only the planar type of both serial and parallel manipulators, these techniques and strategies are valid for any robotic system including the proposed spatial, fault-tolerant serial and parallel architectures mentioned above.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

2. Kinematic, Dynamic Modeling and Control Methodologies

The methodologies of kinematics and dynamics for both serial and parallel manipulators have been developed, and the dynamic model is represented as

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

where

τ denotes the joint input effort,

$M(q)$ is the inertia matrix of the manipulator,

$C(q, \dot{q})$ is the vector of centrifugal and Coriolis terms or velocity related terms,

$G(q)$ is the vector of gravity terms,

$F(q, \dot{q})$ is the vector of friction terms.

To specify the end-effector position and orientation (x, y, ϕ) of a planar serial robot, three active joints is the minimum requirement. For the purpose of fault tolerance, a four-link robot which contains one degree of redundancy, as shown in Figure 1, is employed for investigating the problems of a serial robot at Level III of the canonical architecture. When failure occurs in one of the actuators of a serial robot, the failed joint is locked; thus, a new configuration results with two adjacent links connected through the failed joint. After the failure, the robot is reconfigured with $\phi_1^R, \phi_2^R, \phi_3^R$ instead of $\phi_1, \phi_2, \phi_3, \phi_4$ as shown in Figure 1. The structure of the dynamic equations remains the same before and after failure although the robot's joint space degree-of-freedom is reduced (by one in this case).

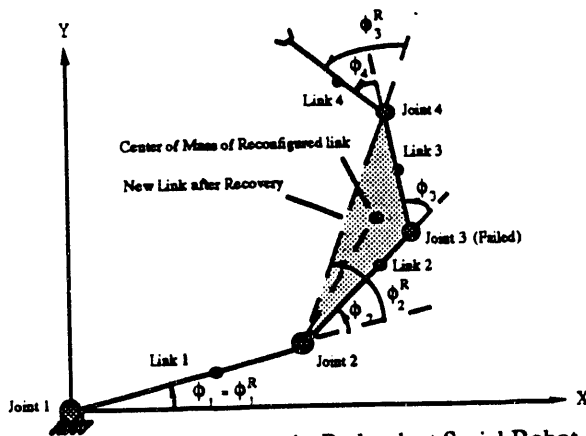


Figure 1: Kinematic-Redundant Serial Robot

In Figure 2, a five-bar parallel robot with two degrees-of-freedom is shown. This configuration can be analyzed in terms of two two-link chains [10]. Thus, to specify the end-effector position (x, y) of a planar parallel robot, the minimum number of active (or independent) joints is two, which are analyzed as active joints (or the independent set of joints). For the purpose of fault tolerance of the parallel robot, additional actuators are assumed to be installed in the dependent joints, that is, every joint of the two legs is

installed with an actuator. Hence, in case of a failure, the failed actuator's task is taken over by one of the so-called force-redundant joints. When this happens, a passive (dependent) joint becomes active (independent), and the failed joint assumes a passive role. In our example, the five bar system has two force redundancies and is employed for investigating the problems of Level II and Level IV structures of the canonical architecture.

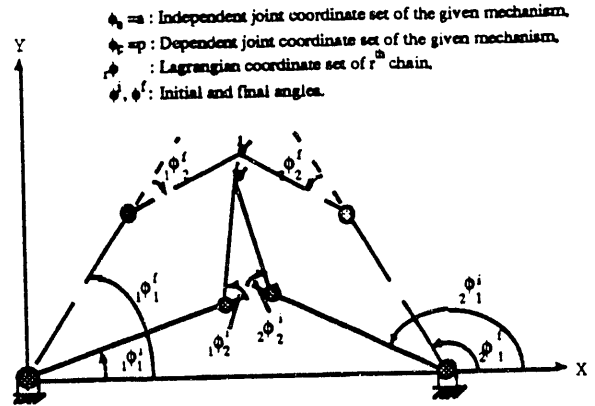


Figure 2: Force-Redundant Parallel Robot

Fault tolerance at Level I has not been directly addressed in this paper; it is assumed that partial failures of one of a pair of dual actuators will be tolerated within Level I. Should the failure exceed recovery capabilities at that level, it may be treated at Levels II and III as considered here as partial or full joint failures depending on failure magnitude.

The feedforward and the feedback controllers of the parallel robot were applied with the similar algorithms as those of the serial robot. Since in this case there are more actuators than required, the independent and dependent sets can share the load. The torque expression in (1) is expressed with a torque distribution method as

$$\tau = \tau_a + [G_a^p]^T \text{ and } \tau_p = [I : [G_a^p]^T] \begin{bmatrix} \tau_a \\ \tau_p \end{bmatrix} \quad (2)$$

where τ_a, τ_p are distributed torques of the independent and dependent sets, respectively. $[G]$ represents a Jacobian function. This equation yields

$$\begin{bmatrix} \tau_a \\ \tau_p \end{bmatrix} = ([G])^+ \tau + (I - [G])^+ [G] \varepsilon \quad (3)$$

where

$$[G] = [I : [G_a^p]^T] \quad (4)$$

$$\text{and } [G]^+ = [W]^{-1} [G]^T ([G] [W]^{-1} [G]^T)^{-1} \quad (5)$$

$[W]$ is a diagonal weighting matrix, which is added to weight the performance of each joint. In general, ε can be selected to yield desired, specified active stiffness. For this study, this homogeneous term can be neglected because of no specified secondary criterion.

Without considering the actuator dynamics, the robot system can be analyzed using a nonlinear second order dynamic model. Since the desired trajectory is defined in terms of position, velocity, and acceleration, a tracking control method such as the well-known computed torque technique can be applied. The computed torque technique is a tracking control algorithm employed to cancel out the nonlinear terms in the controller. The model-based feedforward controller produces a nominal on-line control signal consisting of the desired state ($q_d, \dot{q}_d, \ddot{q}_d$) at every sampling. The feedback PID controller of the second order system attempts to generate a control signal to complement the feedforward control signal by choosing proper gains (K_V, K_P, K_I).

This computed torque law takes the form

$$\tau = \hat{M}(q) (\ddot{q}_d + K_V \dot{E} + K_P E + K_I \int E dt) + \hat{C}(q, \dot{q}) + \hat{G}(q) + \hat{F}(q, \dot{q}) \quad (6)$$

where $\hat{M}(q)$, $\hat{C}(q, \dot{q})$, $\hat{G}(q)$, and $\hat{F}(q, \dot{q})$ represent the on-line measurements; K_V , K_P , K_I are diagonal matrices with respect to each link [5]. E is a matrix with elements of position error, i.e., $[e = (q_d - q) = e_1]_j$; and \dot{E} is a matrix with elements of velocity error, i.e., $[\dot{e}_1 = e_2 = (\dot{q}_d - \dot{q})]_j$; j represents the link counter.

Combining equations (1) and (6), the closed loop system is characterized by the error equation as

$$\begin{aligned} \ddot{E} + K_V \dot{E} + K_P E = \\ E_m = \hat{M}^{-1}(q) \{ [M - \hat{M}(q)] + [C(q, \dot{q}) - \hat{C}(q, \dot{q})] \\ + [G(q) - \hat{G}(q)] + [F(q, \dot{q}) - \hat{F}(q, \dot{q})] \} \end{aligned} \quad (7)$$

If there is no model error ($E_m = 0$) the system exponentially approaches zero error performance as time approaches infinity. However, the dynamic model may not be accurate, or some of the system parameters may not be known precisely. This mismatch ($E_m \neq 0$) between actual and modeled parameters will cause servo errors.

3. Recovery Process - A Case Study on Serial and Parallel Robots

Computer simulations have been performed to identify the phenomena occurring after actuator failures and to define the recovery process. Note that the recovery process so developed is based on the assumption that the task (workspace) in demand after failure is still achievable (reachable). For parallel robots there is no such problem, but for serial robots, the reduced reachability of workspace after reconfiguration by applying the brake to the failed joint is a primary consideration of post-failure operation. These issues are beyond the scope of the current investigation.

In the simulation case study, the entire period of operation is defined as ten seconds for both serial and parallel robots in the absence of a failure. Taking the computation time for computing kinematic and dynamic parameters into account, and from prior experimentation, the total period for detecting the failure and recovering with a new configuration is conservatively set at 200 milliseconds. Thus, it is assumed that the joint failure for both robots occurs at time $t = 4.7$ seconds, and the recovery action starts on the system 0.2 second later. That is, a 200-millisecond duration ($t = 4.7$ to 4.9 seconds; Stage I in Fig. 3) was considered to be the period during which the robot drifts, finds the error, and starts to recover. During the time interval $t = 4.9$ to $t = 5.0$ seconds (Stage II in Fig. 3), different emergency functions are employed respectively to the serial and parallel robots.

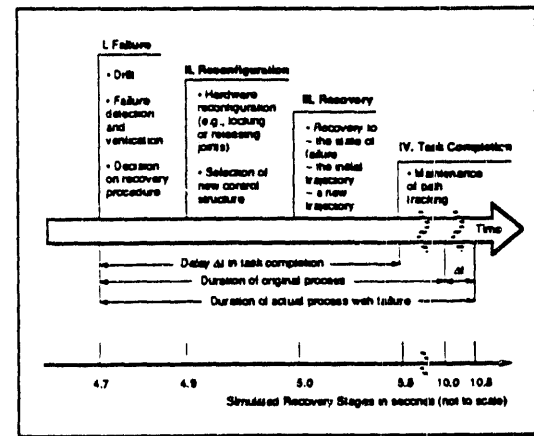


Figure 3 Fault-Tolerant Recovery History

In the case of the serial robot, the brake was applied to the failed joint which reduced its relative velocity and acceleration to zero. For the parallel robot, a new load distribution was employed by setting the torque input of the failed joint to zero to free the failed joint. In Stage II of Figure 3, a reconfiguration scheme was assumed to have been developed and completed in this duration. The new kinematic and dynamic parameters such as link length, center of mass, and moment of inertia are decided according to the configuration shown in Figure 1 for the serial robot. For the parallel robot in Figure 2, there are no changes because the brake is not applied at the failed joint. During the drift period, the system temporarily loses its control. Thus, the actual torque τ must be computed again using equation (2), and the element in the weighting matrix corresponding to the failed joint must be set to zero, which represents zero torque output of the fully failed joint. The diagonal elements of the weighting matrix were set with an equivalent value of one for the normal joints in this simulation, which means no special assignments for any joint.

Several minimum sets can be chosen for the parallel case; for example, one set with active joints $1\phi_1, 1\phi_2$.

and the other one with active joints ${}_1\phi_1, {}_2\phi_1$. If the failure occurs in the dependent set (passive joints, e.g., ${}_2\phi_2$), then there is no change of the independent set (active joints). But if failure occurs in the independent set, the active joints must be re-selected. For instance, when ${}_1\phi_2$ fails, the active joints are switched to be ${}_1\phi_1, {}_2\phi_1$. If ${}_2\phi_1$ fails, then the active joints are switched to be ${}_1\phi_1$ and ${}_1\phi_2$.

The robot begins the recovery at time 5.0 seconds with a smooth motion (e.g., a 4-5-6 polynomial), which moves the end-effector back to its position (at $t = 5.5$ seconds; Stage III in Fig. 3) where the failure occurred. The polynomial function of generating smooth motion of the joints is given as

$$q_d = (q_f - q_i) * (C_p * x^p + C_{p+1} * x^{p+1} + C_{p+2} * x^{p+2} + \dots + C_n * x^n) \quad (8)$$

e.g. $p = 4, n = 6; C_4 = 15.0, C_5 = -24.0, C_6 = 10.0$ for 4-5-6 polynomial, where $x = \frac{t}{t_f}$ is the ratio of the current

time step to the final time, q_f is the final angle position of joint, and q_i is the initial joint displacement. Note that there are several possibilities in the above recovery process. After the robot drifts away from the desired trajectory following the failure, bringing the robot back to the failed state may be desirable ($A \rightarrow B \rightarrow A'$ in Fig. 4). Alternatively, the robot may skip reaching the state of failure and reach some other state along the original trajectory ($A \rightarrow B \rightarrow C$ in Fig. 4), or recover to a new trajectory. This choice depends on the task requirements as well as the structural capabilities of the robot at hand.

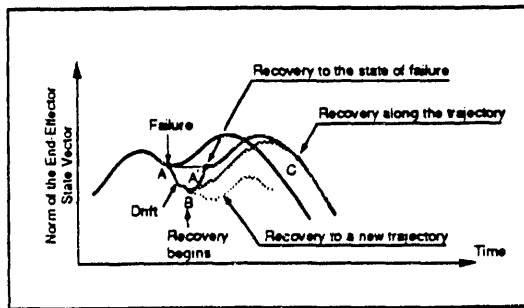


Figure 4 Recovery Strategy of Trajectory Planning

For the robot to return to the position where the failure occurs ($A \rightarrow B \rightarrow A'$ in Fig. 4) or to any new recovery position ($A \rightarrow B \rightarrow C$ in Fig. 4), a 4-5-6 polynomial can be used in which case the polynomial coefficients are determined uniquely by knowing the position, velocity, and acceleration at the beginning and at the end of this period. Of course, it may be advantageous to choose a new end-effector trajectory. However, on-line trajectory

planning requires additional computational tasks and decision making capabilities. In the simulations presented in this work, the end-effector trajectories are not changed after the failure; that is, disabled robots are required to follow the original paths. Ideally, a low-level controller functioning at this level needs to be supervised by a higher-level decision making unit for trajectory planning purposes which is not the focus of the current study. However, in order to find the desired position, velocity and accelerations of joints after failure, inverse kinematics analysis was carried out for the disabled robots.

4. Simulations

The manipulation trajectory and controller implementation for the simulation are described as follows: The initial joint angles of the serial robot are set as $\phi_1(0) = 0^\circ; \phi_2(0) = 0^\circ; \phi_3(0) = 0^\circ; \phi_4(0) = 0^\circ$. The operational range of each joint is assigned as $\phi_1(t): 0^\circ \sim 30^\circ; \phi_2(t): 0^\circ \sim 20^\circ; \phi_3(t): 0^\circ \sim 60^\circ; \phi_4(t): 0^\circ \sim 20^\circ$, governed by a 4-5-6 polynomial function. The initial joint angles of the parallel robot are given by ${}_1\phi_1(0) = 120^\circ; {}_1\phi_2(0) = -90^\circ$ (considering ${}_1\phi_1$ and ${}_1\phi_2$ as the minimum set), and the range of each joint is assigned as

$${}_1\phi_1(t): 120^\circ \sim 60^\circ; {}_1\phi_2(t): -90^\circ \sim -30^\circ,$$

also governed by a 4-5-6 polynomial function. The desired joint velocities for both robots are set to zero at the beginning and at the end of manipulation.

Several sets of feedback gains are assigned for serial and parallel robots according to their pole placements. For example, for the serial case, the first set of gains for K_v, K_p , and K_i are 15, 74, 120, respectively, which locates the system poles at $(-4.0, -5.0, -6.0)$ in the S-plane. The second set of gains are 30, 299, and 990, respectively, locating the system poles at $(-9.0, -10.0, -11.0)$. The kinematic and dynamic parameters (link length L and the moment of inertia w.r.t. to the z-axis I_{zz}) of the serial robot are: $L(\phi_1) = 1.0, L(\phi_2) = 1.0, L(\phi_3) = 1.0, L(\phi_4) = 1.0; I_{zz}(\phi_1) = 0.05, I_{zz}(\phi_2) = 0.03, I_{zz}(\phi_3) = 0.02, I_{zz}(\phi_4) = 0.075$, and of the parallel robot are: $L({}_1\phi_1) = 1.0, L({}_1\phi_2) = 0.8, L({}_2\phi_2) = 0.8, L({}_2\phi_1) = 1.0; I_{zz}({}_1\phi_1) = 0.25, I_{zz}({}_1\phi_2) = 0.1, I_{zz}({}_2\phi_2) = 0.1, I_{zz}({}_2\phi_1) = 0.25$. The center of mass of each link is located in the middle of the link for both cases. Several values of model error of the robots, such as the error in estimating the inertia and the link center of mass locations, were simulated; for example, 1% error for both inertia and center of mass; 5% error for center of mass and 10% for inertia are introduced.

5. Discussion of Simulation Results

Joint-space trajectories, end-effector trajectory, and torque history with 1% error in estimating the inertia and center of mass for a post-failure period of 5.3 seconds are

presented in Figure 5 and Figure 6 for both serial and parallel robots, respectively. In the serial case, the failure is assumed to occur at joint ϕ_3 with gain sets I (15, 74, 120) and II (30, 299, 990). In the parallel case, the failure is assumed to occur at joint ${}_2\phi_2$, and then at ${}_1\phi_2$ with a gain set (15, 74, 120). These plots indicate that the proposed recovery process is able to guide the robots with the same position, velocity, and acceleration of the end-effector as those of normal operation via the proposed controller. There are several important phenomena in recovery processes that have been discovered which are the main issues that need future studies.

The joint drift due to a joint failure cannot be avoided until the failure is detected and recovery measures are taken. Therefore, the development of a fast detection, diagnosis, and reconfiguration algorithm to respond quickly in this period represents a vital stage. When a joint fails and is detected at a certain time (4.7 to 4.9 seconds), there exists an unavoidable torque deviation at the other (unfailed) joints. This is so, because the failed joint, without producing the computed torque, causes all links beyond it to fall drastically under the gravitational effect with an angular velocity which may be largely different from its desired velocity. Thus, the term associated with the Coriolis and centrifugal effects may generate a torque deviation from the nominal torque trajectory in the other joints. Therefore, an emergency function such as locking the failed joint for serial robots or switching the minimum set of active joints for parallel robots must be activated immediately after failure detection. This operation would reduce unwanted torque deviations in other joints caused by the Coriolis and centrifugal effects. Moreover, the reconfigured robot was unable to complete the task in the serial case if the model error was large; for example, 5% error in the center of mass and 10% in inertia parameters.

Figure 7 shows that the torque deviations cause the new configuration to be unable to complete the remaining task no matter what values of gains were employed to the system. This result suggests that good metrology or an adaptive controller (to identify the parameters) is imperative to overcome such a problem. In order to track the same trajectory of the end-effector after recovery, a new joint space trajectory is computed via the inverse kinematics for every joint in the serial case. Because the inverse kinematics of the manipulator is strongly dependent upon the new configuration, the joint space trajectories of the remaining operating joints will be obviously different from the unfailed case, and it may generate vastly different new joint space trajectories. In order to reduce the error as quickly as possible, a high-gain controller seems to be necessary. However, setting a higher gain of the controller; for example, employing the gain set (60, 1199, 7980) for the serial case and (30, 299, 990) for the parallel case, may cause the system to become unstable. Therefore, selecting a suitable set of gains in the controller is critical in the recovery process.

6. Conclusions and Recommendations

According to the simulation results, the parallel robot seems easier to manipulate and have a better performance after a failure in one of its joints. Parallel robots possess a notable characteristic for the application of fault tolerance, which protects the reachability, operates easily without mechanically braking its functionality, and performs better without new architectural configuration after recovery. Nevertheless, the primary criteria for the selection of robots depend upon the task requirements while the robots are manipulating normally. The essential focus for this study in fault tolerance is to investigate what types of mechanism architectures (e.g., serial, parallel or hybrid mechanisms) are better suited for fault tolerance, and how to improve the efficiency and reliability of the recovery process.

The proposed recovery process for fault tolerance is a preliminary study in this area. Based upon this recovery scheme, more sophisticated controllers can be developed to reduce any detrimental effect due to the internal shocks and to keep good tracking performance. Therefore, further refinement of the recovery processes is needed in order to synthesize the control structure of the manipulator prior to and following failure, the degree of synthesis being dependent on the capacity and scheme at the decision-making level. Manipulation with good tracking performance after reconfiguration is not only a control problem, but also a trajectory planning problem. In the future, primary efforts will be focused on developing control algorithms such as parameter adaptation techniques to overcome the problem of model error, and thereby achieving better tracking performance. Additionally, avoidance of the actuator torque saturation which becomes a more important problem under failure will also be addressed within the framework of fault-tolerant robot control. Furthermore, the scheme of fault-tolerant robot operation integrates software and hardware of fault detection, fault diagnosis, and fault recovery. This work provides guidelines for the design of control algorithms towards the full development of fault-tolerant robotic manipulator systems.

Acknowledgments

This work was partially supported by the NASA Johnson Space Center (Grant No. NAG-9-411) and the U.S. Department of Energy's program on Robotics for Advanced Reactors (Grant No. DE-FG-02-86NE37966). The Universities of Florida, Michigan, Tennessee, Texas and Oak Ridge National Laboratory participate to this DOE program.

References

- [1] Athans, M., "On Restructurable Control System Theory," *Proceedings of the NASA Workshop on Restructurable Controls*, NASA CP-2277, NASA Langley Research Center, September 1982.

- [2] Avižienis, A., "Fault-Tolerant Systems," *IEEE Transaction on Computers*, Vol. C-25, No. 12, pp. 1304-1311, December, 1976.
- [3] Maciejewski, A.A., "Fault Tolerance Properties of Kinematically Redundant Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. I, pp. 638-642, September 1982.
- [4] Visinsky, M.L., "Robotic Fault Tolerance: Algorithms and Architectures," *Robotics and Remote Systems in Hazardous Environments*, to appear, Prentice Hall, Englewood, Cliffs, New Jersey, 1992.
- [5] Wu, E., et al., "A Fault-Tolerant Joint Drive System for the space Shuttle Remote Manipulator System," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 2504-2509, 1991.
- [6] Khosla, P., and Paredis, C., "Kinematic Design of Serial Link Manipulators," 1992.
- [7] Sreevijayan, D., "On The Design of Fault Tolerant Robotic Manipulator Systems," Master Thesis, University of Texas at Austin, Austin, Texas, May 1992.
- [8] Chladek, J. T., "Fault Tolerance For Space Based Manipulator Mechanisms And Control System," *First International Symposium On Measurement And Control In Robotics*, Houston, Texas, June 1990.
- [9] Tesar, D., Sreevijayan, D., and Price, C., "Four-Level Fault Tolerance In Manipulator Design For Space Operation," *First International Symposium On Measurement And Control In Robotics*, Houston, Texas, June 1990.
- [10] Freeman, R.A. and Tesar, D., "Dynamic Modeling of Serial and Parallel Mechanisms/ Robotic Systems, Part - I Methodology, Part II - Applications," *Proceedings of the 1988 ASME Design Technical Conference*, Kissimmee, Florida, DE-Vol. 15-2, pp. 7-21, 1988.
- [11] Craig, J.J., *Introduction To Robotics: Mechanics and Control*, Addison-Wesley, 1989.
- [12] Lin, S.H., Tosunoglu, S., and Tesar, D., "Control of A Six-Degree-of-Freedom Flexible Industrial Manipulator," *IEEE Control Systems Magazine*, Vol. 11, No. 3, pp. 24-30, April 1991.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

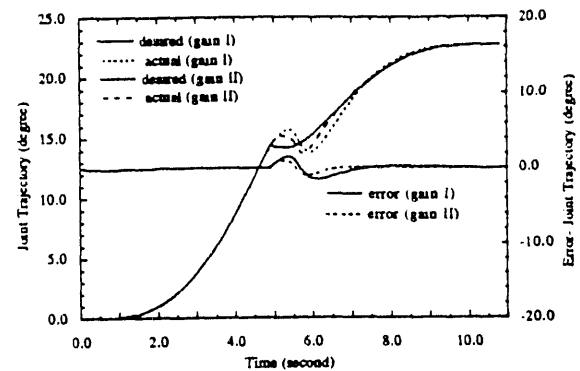


Fig. 5(a) ϕ_1^R Trajectory With Failure in ϕ_2 (Gain Set: I and II)

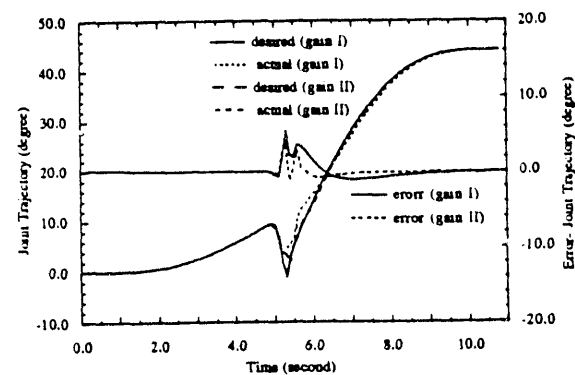


Fig. 5(b) ϕ_2^R Trajectory With Failure in ϕ_3 (Gain Set: I and II)

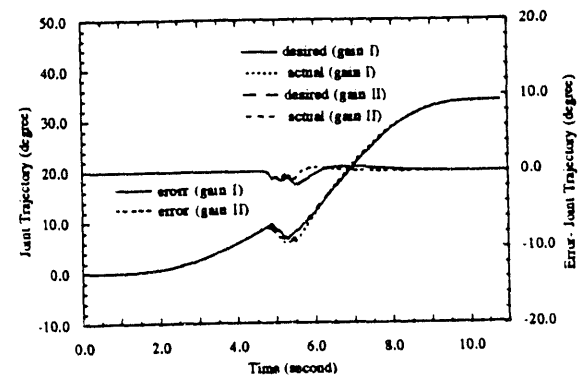


Fig. 5(c) ϕ_3^R Trajectory With Failure in ϕ_4 (Gain Set: I and II)

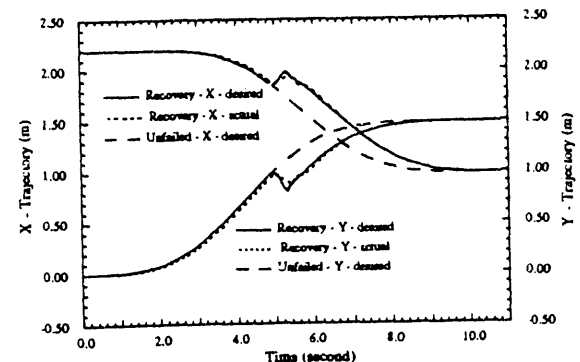
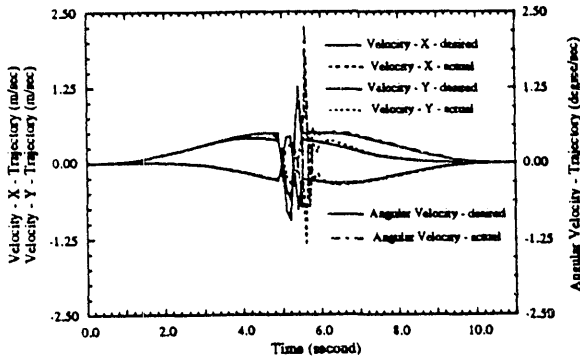
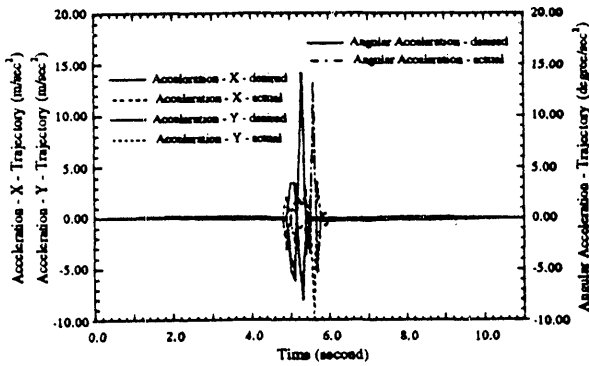
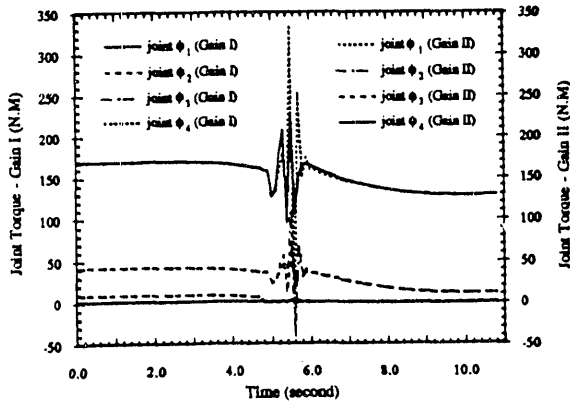
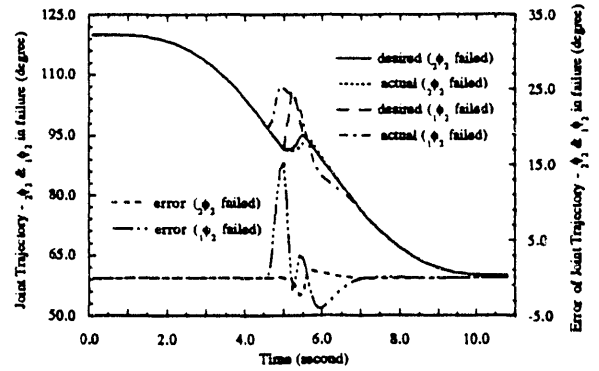
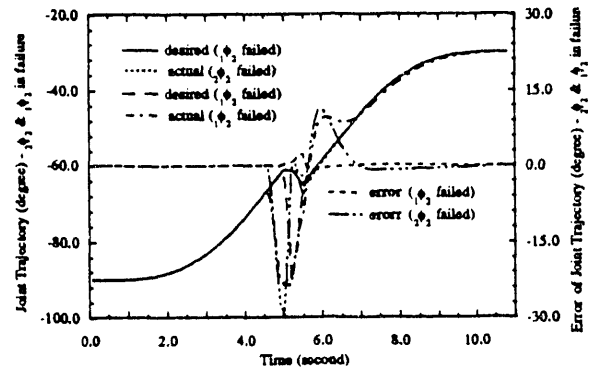
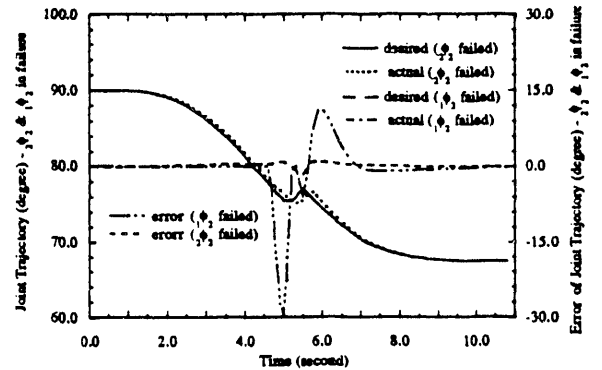
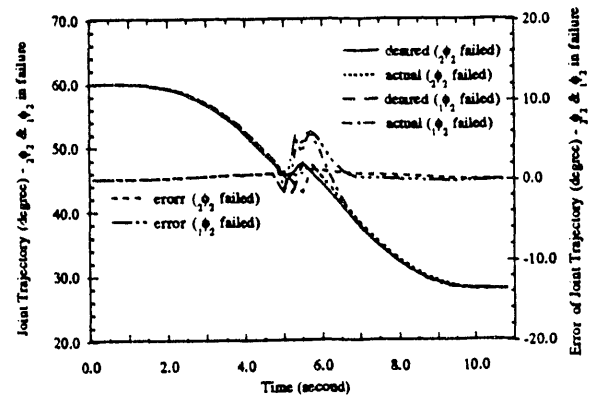


Fig. 5(d) Position Trajectory of End Effector With Failure in ϕ_4

Fig. 5(e) Angular Velocity Trajectory of End Effector With Failure in ϕ_3 Fig. 5(f) Angular Acceleration Trajectory of End Effector With Failure in ϕ_3 Fig. 5(g) Torque Trajectory With Failure in ϕ_3 (Gain Set: I and II)Fig. 6(a) ϕ_1 Trajectory With Failure in ϕ_2 & ϕ_3 (Active Set: ϕ_1 & ϕ_2)Fig. 6(b) ϕ_2 Trajectory With Failure in ϕ_2 & ϕ_3 (Active Set: ϕ_1 & ϕ_2)Fig. 6(c) ϕ_2 Trajectory With Failure in ϕ_2 & ϕ_3 (Active Set: ϕ_1 & ϕ_2)Fig. 6(d) ϕ_1 Trajectory With Failure in ϕ_2 & ϕ_3 (Active Set: ϕ_1 & ϕ_2)

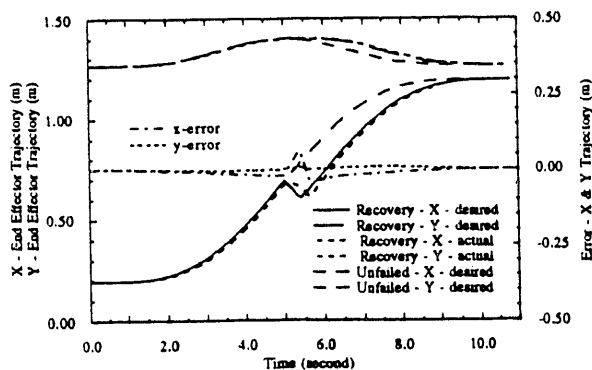
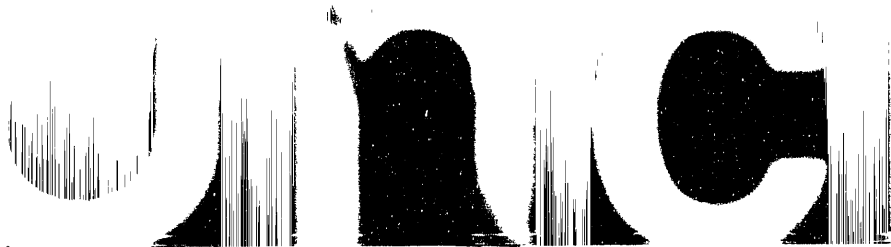


Fig. 6(e) End Effector Trajectory With Failure in ϕ_2 (Active Set: ϕ_1 & ϕ_2)

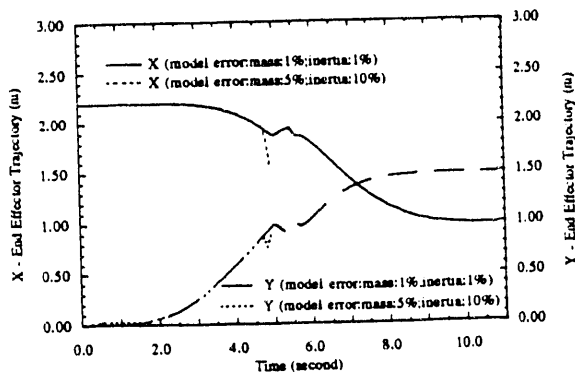


Fig. 7(a) Position Trajectory of End Effector With Failure in ϕ_3 - Serial case (performance of different model error)

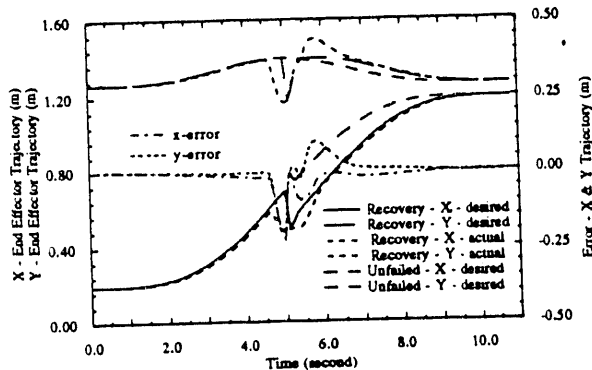


Fig. 6(f) End Effector Trajectory With Failure in ϕ_1 (Active Set: ϕ_1 & ϕ_2)

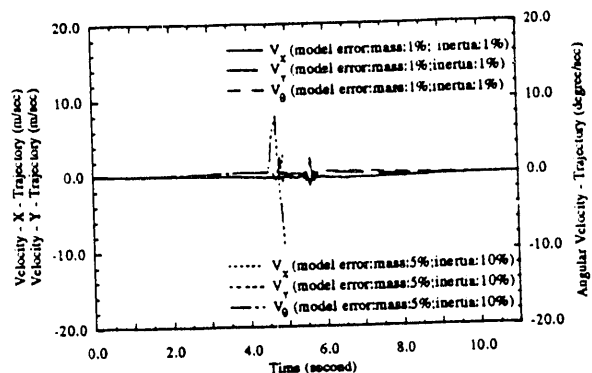


Fig. 7(b) Velocity Trajectory of End Effector With Failure in ϕ_3 - Serial case (performance of different model error)

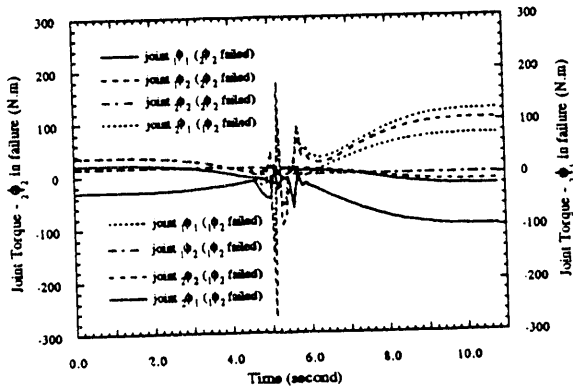


Fig. 6(g) Torque Trajectory With Failure in ϕ_2 & ϕ_1 (Active Set: ϕ_1 & ϕ_2)

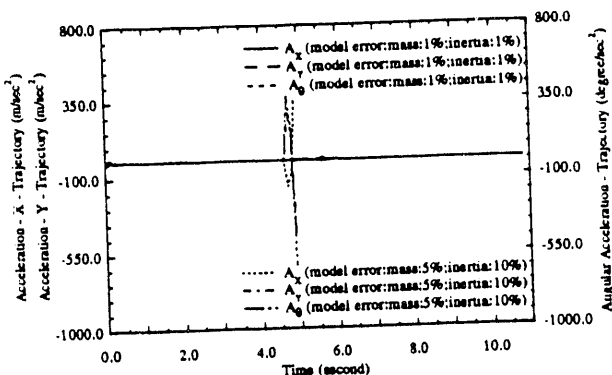


Fig. 7(c) Acceleration Trajectory of End Effector With Failure in ϕ_3 - Serial case (performance of different model error)

END

DATE
FILMED

4 / 7 / 93

