Available online at www.sciencedirect.com

ScienceDirect

Control Engineering Practice 15 (2007) 615–625

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

A fault tolerance framework for cooperative robotic manipulators

Renato Tinós[a,*], Marco Henrique Terra[b], Marcel Bergerman[c]

[a]Departamento de Física e Matemática, FFCLRP, Universidade de São Paulo (USP), 14040-901, Ribeirão Preto, SP, Brazil
[b]Departamento de Engenharia Elétrica, EESC, Universidade de São Paulo (USP), 13560-970, São Carlos, SP, Brazil
[c]Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, USA

Received 15 July 2005; accepted 30 October 2006
Available online 26 December 2006

Abstract

The problem of fault tolerance in cooperative manipulators rigidly connected to an undeformable load is addressed in this paper. Four categories of faults are considered: free-swinging joint faults (FSJFs), locked joint faults (LJFs), incorrectly measured joint position faults (JPFs), and incorrectly measured joint velocity faults (JVFs). Free-swinging and locked joint faults are detected via artificial neural networks (ANNs). Incorrectly measured joint position and velocity faults are detected by considering the kinematic constraints of the cooperative system. When a fault is detected, the control system is reconfigured according to the nature of the isolated fault and the task is resumed to the largest extent possible. The fault tolerance framework is applied to an actual system composed of two cooperative robotic manipulators. The results presented demonstrate the feasibility and performance of the methodology.

Keywords: Fault tolerance; Robotics; Co-operation; Fault detection and isolation; Neural networks

1. Introduction

Robotic manipulators have been deployed in an ever growing number of unstructured and/or hazardous environments, such as in outer space and in deep sea. Robots are used in these environments to limit or eliminate the presence of human beings in such dangerous places, or due to their capability to execute repetitive tasks very reliably. Faults, however, can put at risk the robots, their task, the working environment, and any humans present there.

Faults in robots are mainly due to their inherent complexity. There are several sources of faults in robots, such as electrical, mechanical, and hydraulic (Visinsky et al., 1994). In fact, the mean-time-to-failure of industrial robots can be considered small for their intended life expectancy and cost. According to studies published in the decade of 1990, the recorded mean-time-to-failure of industrial robots was only 500–2500 h (Dhillon and Fashandi, 1997). This number is probably smaller in unstructured or hazardous environments due to external factors, such as extreme temperatures, moving obstacles, and radiation. Therefore, there are good reasons to research and develop fault detection and isolation (FDI) systems for robots.

In most environments, a robot can be repaired after the FDI. There are some environments, however, where human beings cannot be sent to make the necessary repairs, and, thus, fault tolerance must be provided to the robot. This is the case of robots operating in hazardous or distant places. Fault tolerance is also necessary when the robot must be kept continuously operating even with a fault, such as when robots are used to disarm explosives.

Robotic systems with actuation redundancy are interesting in applications where fault tolerance is needed because the number of degrees of freedom (dof) in these systems is generally higher than the dof required to execute the task. Furthermore, as in the human case where the use of two arms means an advantage over the use of only one arm in several cases, two or more robots can execute tasks that are difficult or even impossible for only one robot (Vukobratovic and Tuneski, 1998). For example,

[*]Corresponding author.
E-mail addresses: rtinos@ffclrp.usp.br (R. Tinós), terra@sel.eesc.usp.br (M.H. Terra), marcel@cmu.edu (M. Bergerman).

0967-0661/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2006.10.018

cooperative robots can be used to manipulate heavy, large, or flexible loads, assemble structures, and carry objects that can slide from only one robot. Humanoid robots are examples of robotic systems with cooperative manipulation capability. Actuation redundancy makes the use of cooperative robots in unstructured and/or hazardous environments very appealing. On the other hand, as in the case for single manipulators, fault tolerance is crucial for cooperative arms to operate reliably in these environments.

Because of the dynamic coupling of joints, inertia, and gravitational torques, faulty arms can quickly accelerate into wild motions that can cause serious damage (Visinsky et al., 1994). If the cooperative system's controller is not designed to detect and isolate the faults, the internal forces increase and cause damage to the load or instability to the system. While the problem of faults in single robotic arms has been studied by several researchers in the recent years, e.g. (English and Maciejewski, 1998; Goel et al., 2004; McIntyre et al., 2005; Miyagi and Riascos, 2006; Vemuri and Polycarpou, 2004; Visinsky et al., 1995), only a few studied the corresponding problem in cooperative manipulators (Liu et al., 1999; Monteverde and Tosunoglu, 1999; Tinós et al., 2006) or parallel manipulators (Hassan and Notash, 2004, 2005). To the best of the authors' knowledge, fault tolerance frameworks, i.e., fault tolerance systems with fault detection and control reconfiguration, were proposed only for single manipulators (English and Maciejewski, 1998; Visinsky et al., 1995).

In this work, a fault tolerance framework for cooperative manipulators is proposed. Section 2 describes the kinematic and the dynamic models of cooperative manipulators, and Section 3 describes the fault tolerance methodology. Four categories of faults are considered: free-swinging joint faults (FSJFs), locked joint faults (LJFs), incorrectly measured joint position faults (JPFs), and incorrectly measured joint velocity faults (JVFs). First, the faults are detected by an FDI system using neural networks and knowledge of the kinematic constraints of the cooperative system (Section 4). Then, the control system is reconfigured according to the nature of the isolated fault (Section 5). Section 6 presents the results of the proposed fault tolerance framework applied to a real system composed of two cooperative manipulators. Finally, the conclusions are presented in Section 7.

2. Cooperative manipulators

The equation of motion for the ith arm of a fault-free multi-robot system with m arms rigidly connected to an undeformable object (load) is

$$\ddot{\mathbf{q}}_i = \mathbf{M}_i^{-1}(\mathbf{q}_i)[\tau_i + \mathbf{J}_i^{\mathrm{T}}(\mathbf{q}_i)\mathbf{h}_i - \mathbf{g}_i(\mathbf{q}_i) - \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i], \tag{1}$$

where $\mathbf{q}_i$ is the vector of joint positions of arm i, $i = 1, \ldots, m$, τ_i is the vector of torques applied at the joints of arm i, $\mathbf{M}_i(\mathbf{q}_i)$ is its inertia matrix, $\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ is its matrix of centrifugal and Coriolis terms, $\mathbf{g}_i(\mathbf{q}_i)$ is its vector of gravitational terms, $\mathbf{J}_i(\mathbf{q}_i)$ is its geometric Jacobian (from joint velocity to end-effector velocity), and $\mathbf{h}_i$ is the force vector at the end-effector of arm i; friction terms are not shown for simplicity. The combined dynamics of all arms can be written as

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})[\tau + \mathbf{J}^{\mathrm{T}}(\mathbf{q})\mathbf{h} - \mathbf{g}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}], \tag{2}$$

where $\mathbf{q} = [\mathbf{q}_1^{\mathrm{T}}\, \mathbf{q}_2^{\mathrm{T}} \ldots \mathbf{q}_m^{\mathrm{T}}]^{\mathrm{T}}$, $\tau = [\tau_1^{\mathrm{T}}\, \tau_2^{\mathrm{T}} \ldots \tau_m^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{h} = [\mathbf{h}_1^{\mathrm{T}}\, \mathbf{h}_2^{\mathrm{T}} \ldots \mathbf{h}_m^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{g} = [\mathbf{g}_1^{\mathrm{T}}\, \mathbf{g}_2^{\mathrm{T}} \ldots \mathbf{g}_m^{\mathrm{T}}]^{\mathrm{T}}$,

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} \mathbf{M}_1(\mathbf{q}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_m(\mathbf{q}_m) \end{bmatrix},$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m) \end{bmatrix}$$

and

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_1(\mathbf{q}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{J}_m(\mathbf{q}_m) \end{bmatrix}.$$

In the cooperative system, the equation of motion for the load is

$$\mathbf{M}_{\mathbf{o}}\dot{\mathbf{v}}_{\mathbf{o}} + \mathbf{b}_{\mathbf{o}}(\mathbf{x}_{\mathbf{o}}, \mathbf{v}_{\mathbf{o}}) = \mathbf{J}_{\mathbf{o}}^{\mathrm{T}}(\mathbf{x}_{\mathbf{o}})\mathbf{h}, \tag{3}$$

where $\mathbf{x}_{\mathbf{o}}$ is the k-dimensional vector of position and orientation at the origin of the frame attached to the center of mass of the load, $\mathbf{v}_{\mathbf{o}}$ is the vector of linear and angular velocities of the load, $\mathbf{b}_{\mathbf{o}}$ is the vector of centrifugal, Coriolis, and gravitational terms, $\mathbf{M}_{\mathbf{o}}$ is the load inertia matrix, and $\mathbf{J}_{\mathbf{o}}(\mathbf{x}_{\mathbf{o}}) = [\mathbf{J}_{\mathbf{o}1}^{\mathrm{T}}(\mathbf{x}_{\mathbf{o}}) \ldots \mathbf{J}_{\mathbf{o}m}^{\mathrm{T}}(\mathbf{x}_{\mathbf{o}})]^{\mathrm{T}}$, where $\mathbf{J}_{\mathbf{o}i}(\mathbf{x}_{\mathbf{o}})$ converts velocities of the load into velocities of the end-effector of arm i. The velocities $\mathbf{v}_{\mathbf{o}}$ can be computed by $\mathbf{v}_{\mathbf{o}} = \mathbf{T}(\mathbf{x}_{\mathbf{o}})\dot{\mathbf{x}}_{\mathbf{o}}$ (Sciavicco and Siciliano, 1996), where $\mathbf{T}(\mathbf{x}_{\mathbf{o}})$ is a transformation matrix that relates the angular velocities to the derivative of the minimal representation of the orientation (Euler angles or RPY angles) in the three-dimensional space ($\mathbf{T}(\mathbf{x}_{\mathbf{o}}) = \mathbf{I}$ for planar manipulators).

As it is possible to compute the positions and orientations of the load using the positions of the joints of any arm of the cooperative system, the following kinematic constraint appears

$$\mathbf{x}_{\mathbf{o}} = \varphi_1(\mathbf{q}_1) = \varphi_2(\mathbf{q}_2) = \cdots = \varphi_m(\mathbf{q}_m), \tag{4}$$

where $\varphi_i(\mathbf{q}_i)$ is the vector of the position and orientation of the load computed via the joint positions of arm i, i.e., the direct kinematics of arm i. The velocities of the load are constrained by

$$\mathbf{v}_{\mathbf{o}} = \mathbf{D}_1(\mathbf{q}_1)\dot{\mathbf{q}}_1 = \mathbf{D}_2(\mathbf{q}_2)\dot{\mathbf{q}}_2 = \cdots = \mathbf{D}_m(\mathbf{q}_m)\dot{\mathbf{q}}_m, \tag{5}$$

where $\mathbf{D}_i(\mathbf{q}_i) = \mathbf{J}_{\mathbf{o}i}^{-1}(\mathbf{x}_{\mathbf{o}})\mathbf{J}_i(\mathbf{q}_i)$ is the Jacobian relating joint velocities of arm i and load velocities.

The control of cooperative manipulators is a complex task due to the interaction among the arms caused by the force constraints. The control should be coordinated and the squeeze forces at the load should be minimized to avoid damage to the system. The squeeze forces are the components of the forces in the object that are in the squeeze subspace (i.e., the ones that do not contribute to the motion of the load). The forces that contribute to the motion of the load are called motion forces; they lie in the motion subspace and are orthogonal to the squeeze forces (Wen and Kreutz-Delgado, 1992).

Several solutions were reported in the literature to deal with the control problem of fault-free cooperative manipulators rigidly connected to an undeformable object, e.g. Luh and Zheng (1987), Carignan and Akin (1988), Wen and Kreutz-Delgado (1992), Bonitz and Hsia (1996). When force and motion must be controlled in robots, hybrid control can be employed (Jatta et al., 2006). The hybrid control method for fault-free cooperative manipulators developed in Wen and Kreutz-Delgado (1992) is particularly interesting because motion and squeeze control are independently dealt with, and because it does not utilize the inertia matrix of the robots in the control law, which, in general, reduces the effect of modeling errors. In the cooperative system, joint torques in the form $\mathbf{D}^{\mathrm{T}}(\mathbf{q})\mathbf{h}_{\mathbf{osc}}$, where $\mathbf{D}(\mathbf{q}) = [\mathbf{D}_1(\mathbf{q}_1)\mathbf{D}_2(\mathbf{q}_2)\ldots\mathbf{D}_m(\mathbf{q}_m)]$ and $\mathbf{h}_{\mathbf{osc}}$ is in the squeeze subspace, do not affect the motion if there are no singular configuration of the arms. However, the motion of the arms affects the squeeze forces due to the squeeze components of the d'Alembert (inertial) forces. Thus, in Wen and Kreutz-Delgado (1992), a stable motion control with compensation of the gravitational torques is firstly designed ignoring the squeeze terms. Then, the squeeze control is addressed considering the component of the squeeze forces caused by the motion as a disturbance.

In this work, the fault-free system is controlled by the hybrid control proposed in Wen and Kreutz-Delgado (1992). The controller used for the fault-free system, however, cannot be used for the cooperative system with FSJFs or LJFs. As the controller was designed considering that all joints are directly actuated, the torques applied to control the squeeze in the object generally do not project forces only in the squeeze subspace when there are FSJFs or LJFs. In this way, the squeeze control influences the motion control and can make the system unstable or drastically increase the squeeze forces.

3. Fault tolerance system

The fault tolerance system addresses the following categories of faults: FSJFs, where an actuation loss occurs in a joint of one arm; LJFs, where a joint of one arm is locked; JPFs, where the measurement of one joint position is not correct; and JVFs, where the measurement of one joint velocity is not correct. JPFs and JVFs can occur due to sensor faults, for example. The fault tolerance system proposed here is designed for cooperative manipulators rigidly connected to an undeformable load. When the manipulators are not connected to a load, traditional fault tolerant methods designed for individual manipulators can be used (Visinsky et al., 1995).

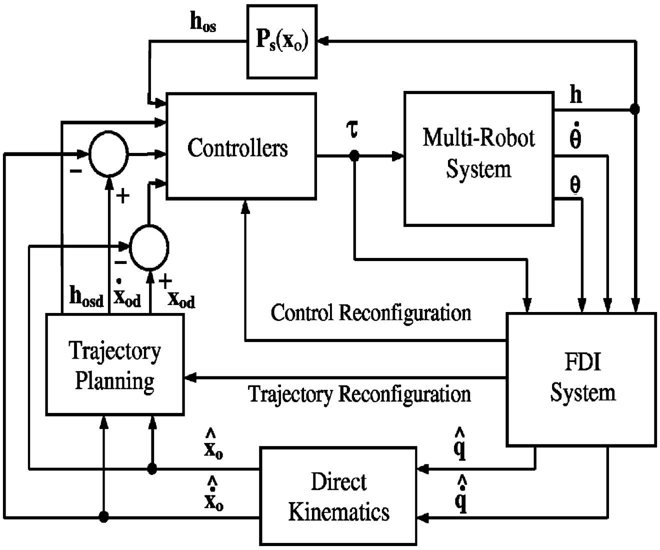

Fig. 1. Fault tolerance system. $\boldsymbol{\theta}$ is the vector of measured joint positions,^ indicates that the vector is estimated, the subscript d refers to the desired vector, and the matrix $\mathbf{P}_\mathbf{s}(\mathbf{x}_\mathbf{o})$ converts the forces in the end-effectors to squeeze forces.

The fault tolerance scheme proposed is displayed in Fig. 1. The faults are firstly detected and isolated by an FDI system. When a fault is detected, the arms may be locked by brakes and the trajectory planning may be reconfigured starting with zero velocities. Other option is to reconfigure the trajectory planning starting with the current load velocity without applying the brakes. The choice depends on the joint configurations, joint velocities, and the parameters of the cooperative system, such as maximum torque allowed and joint limits.

Roughly speaking, in the proposed framework when an FSJF or an LJF is detected, the control system is reconfigured (Section 5). If a JPF or a JVF is detected, the positions or velocities of the faulty joint are estimated using the positions and velocities of the other joints and the same controller used for the fault-free system is utilized.

4. FDI system

In this section, an FDI system composed of three steps is proposed. First, JPFs are detected by analyzing the position constraints (4). Then, JVFs are detected by analyzing the velocity constraints (5). The last step is the detection of FSJFs and LJFs via two artificial neural networks (ANNs). This sequence is important because undetected JPFs can cause false detection of other faults as joints position readings are used to compute the velocities of the load in (5) and as inputs of the first ANN. The same occurs for undetected JVFs in FSJFs and LJFs, as joint velocity readings are used as inputs of the first ANN.

This sequence is also important because joint velocities are very often reconstructed from joint position measurements. In each sample time, the FDI system indicates if the robots are working normally or with a fault. The occurrence of only one fault at a time is considered in this work, although multiple faults can coexist as long as they do not occur at the same instant in time.

4.1. Joint position and joint velocity faults

In Notash (2000), joint position sensor faults are detected in parallel manipulators by using the kinematic constraints imposed by the closed kinematic chain. The direct kinematics problem (given the joint positions, identify the position and orientation of the load) is not trivial in parallel manipulators because they usually have one or more unsensed joints. The direct kinematics problem is however easier in cooperative manipulators because all joints are assumed to be equipped with sensors.

As the load position $\mathbf{x_o}$ can be computed using the joint positions of any arm (4), it is possible to identify the arm f with the wrong joint position readings if $m>2$. The arm with the wrong reading gives an estimate of $\mathbf{x_o}$ that is different from the estimate of the other $m-1$ arms. Therefore, a JPF is detected at arm f if

$$\|\hat{\mathbf{x}}_{\mathbf{o}f}(\boldsymbol{\theta}_f) - \hat{\mathbf{x}}_{\mathbf{o}i}(\boldsymbol{\theta}_i)\| > \gamma_{p1}$$
$$\text{for all } i = 1,\ldots,m \text{ and } i \neq f, \tag{6}$$

where $\hat{\mathbf{x}}_{\mathbf{o}i}$ is the estimate of $\mathbf{x_o}$ using the measured positions of the joints ($\boldsymbol{\theta}_i$) at arm i (forward kinematics), $\boldsymbol{\theta}_f$ is the vector of the measured positions at the joints of arm f, $\|.\|$ represents the Euclidean norm, and γ_{p1} is a positive real constant. The next step is to estimate the position of each joint $j = 1,\ldots,n_f$ of arm f

$$\hat{q}_{fj} = \psi_j(\boldsymbol{\theta}_f, \hat{\mathbf{x}}_{\mathbf{o}}), \tag{7}$$

where

$$\hat{\mathbf{x}}_{\mathbf{o}} = \frac{1}{m-1} \sum_{i=1, i\neq f}^{m} \hat{\mathbf{x}}_{\mathbf{o}i}(\boldsymbol{\theta}_i)$$

and ψ_j is a kinematic function used to estimate the position of joint j. In a planar system with only revolute joints, ψ_j can be written as the difference between the orientation of the load and the sum of the measured positions of the joints $k \neq j$ of arm f. In a three-dimensional space, an inverse kinematics method can be employed to estimate $\hat{q}_{fj}$ (in this case, the measured values of the joints $i \neq j$ can be used to eliminate the possible redundancy of the solutions).

Computing again the estimate of vector $\mathbf{x_o}$ for arm f for each new estimate $\hat{q}_{fj}$, the JPF at joint j of arm f is isolated when

$$\|\hat{\mathbf{x}}_{\mathbf{o}} - \hat{\mathbf{x}}_{\mathbf{o}f}(\boldsymbol{\theta}_f, \hat{q}_{fj})\| < \gamma_{p2}, \tag{8}$$

where $\hat{\mathbf{x}}_{\mathbf{o}f}(\boldsymbol{\theta}_f, \hat{q}_{fj})$ is the vector of positions and orientations of the load estimated for arm f substituting the measured position of joint j by its estimate $\hat{q}_{fj}$ and using the measured positions of the other joints, and γ_{p2} is a positive real constant.

The choice of the thresholds γ_{p1} and γ_{p2} has a strong influence in the performance of the FDI system. If the values of γ_{p1} and γ_{p2} are too small, false alarms may occur due to the presence of noise in the joint position readings. If the thresholds are too large, JPFs can go undetected. If the distribution of the noise in the joint position readings is normal and its statistical properties are known, γ_{p1} and γ_{p2} can be computed as linear functions of the variance of the noise in the joint position readings. In this way, the thresholds are proportional to the variance of the noise, and larger values of the variance imply in larger values of the thresholds.

The procedure to detect and isolate JPFs when $m>2$ can be summarized as follows: compare the estimate of $\mathbf{x_o}$ for all arms (6); if all values are close, a JPF is not declared; otherwise, compute, for all joints of the faulty arm, the estimate of the joint positions (7) and test (8) for all joints; if the test is satisfied for joint j, declare a JPF at this joint.

If $m = 2$, the arm with the fault cannot be identified just by analyzing the estimate of $\mathbf{x_o}$ for each arm. In this case, the joint positions estimation (7) should be performed for the two arms using, instead of the value of $\hat{\mathbf{x}}_{\mathbf{o}}$, the estimate obtained using the joint positions of the other arm. The same should be done in (8), which can be used to detect the JPF.

As it is possible to compute the velocity of the load by using the joint velocities of any arm (5), JVFs can be detected in a way similar to JPFs. Considering the occurrence of only one fault, the JVF at joint j of arm f is detected for $m>2$ when

$$\|\hat{\mathbf{v}}_{\mathbf{o}} - \hat{\mathbf{v}}_{\mathbf{o}f}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f, \hat{\dot{q}}_{fj})\| < \gamma_{v2}, \tag{9}$$

where $\hat{\mathbf{v}}_{\mathbf{o}f}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f, \hat{\dot{q}}_{fj})$ is the velocity of the load estimated for arm f substituting the measured velocity of joint j by its estimate $\hat{\dot{q}}_{fj}$ and using the measured velocities ($\dot{\boldsymbol{\theta}}_f$) of the other joints (forward kinematics), $\hat{\mathbf{v}}_{\mathbf{o}}$ is the estimate of the load velocities using the measured joint velocities of the other arms, and γ_{v2} is the threshold used to avoid that faults be hidden due to the presence of noise in the joint readings. In this work, γ_{v2} is computed as a linear function of the variance of the noise in the joint velocity readings. When $m = 2$, $\mathbf{v_o}$ should be substituted by the estimated velocity obtained using the joint velocities of the other arm.

4.2. Free-swinging and locked joint faults

FDI systems for single manipulators generally employ the residual generation scheme (Visinsky et al., 1994). The residual vector is generated by comparing the measured states of the arm with their estimates obtained by a mathematical model of the fault-free arm. This method, however, does not work well in the presence of modeling errors, generating false alarms or hiding the fault effects. Robust techniques (McIntyre, Dixon, Dawson & Walker

2005) and Artificial Intelligence techniques (Schneider and Frank, 1996) have been used to avoid these problems. In the approach presented in Vemuri and Polycarpou (2004), the off-nominal behavior due to faults is mapped utilizing an ANN trained using a robust observer, based on the robot's physical model. Overall, one problem with FDI methods which rely on the system mathematical model is that, for some real robots, detailed modeling is difficult.

To overcome this problem, Terra and Tinós (2001) proposed a method where the mathematical model of the robot is not necessary. A multilayer perceptron (MLP) is used to map the dynamics of the arm and a radial basis function network (RBFN) classifies the residual vector. The MLP mapping is static, which is possible because the states of the system are measurable, the sample time is small, and control signals are used in the MLP inputs.

FDI for cooperative manipulators has been studied only recently (Tinós et al., 2001). In Tinós et al. (2001), following the line of Terra and Tinós (2001), one MLP is trained to reproduce the dynamics of all arms and the load of the fault-free cooperative system (2). The inputs of the MLP are the joint positions, velocities, and torques in the arms at instant t. The outputs of the MLP are the estimated joint velocities of the fault-free system at instant $t+\Delta t$, which are compared to the measured joint velocities at instant $t+\Delta t$ in order to generate the residual vector. The residual vector is then classified by a RBFN that gives the fault information. The use of only one MLP in Tinós et al. (2001) is an interesting approach. However, the mapping of the MLP is dependent on the load parameters such as the load mass. If the system has to manipulate a different object, the ANNs have to be trained again.

Here, the dynamic model of each arm is mapped by one MLP. Thus, the mapping is not dependent on the object parameters. If the sample period Δt is sufficiently small, the dynamics of the fault-free arm i (1) can be written as

$$\dot{\mathbf{q}}_i(t+\Delta t) = \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)), \tag{10}$$

where $\mathbf{f}(.)$ is a nonlinear function vector representing the dynamics of the fault-free arm i.

Each MLP i ($i = 1,\ldots,m$) maps the dynamic behavior of one fault-free arm (10). The inputs of the ith MLP are the joint positions, velocities, torques, and end-effector forces of arm i at instant t. The output vector of the ith MLP, which should reproduce the joint velocities of the fault-free arm i at time $t+\Delta t$, can be written as

$$\hat{\dot{\mathbf{q}}}_i(t+\Delta t) = \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)) + \mathbf{e}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)), \tag{11}$$

where $\mathbf{e}(.)$ is the vector of mapping errors. The residual of arm i is defined as

$$\hat{\mathbf{r}}_i(t+\Delta t) = \dot{\mathbf{q}}_i(t+\Delta t) - \hat{\dot{\mathbf{q}}}_i(t+\Delta t). \tag{12}$$

From (10)–(12), it can be observed that the residual vector of arm i in the fault-free case is equal to the vector of mapping errors, which must be sufficiently small when compared to the fault function vector in order to allow the fault detection. The residual vector $\hat{\mathbf{r}}(t+\Delta t) = [\hat{\mathbf{r}}_1^{\mathrm{T}}(t+\Delta t),\ldots,\hat{\mathbf{r}}_m^{\mathrm{T}}(t+\Delta t)]^{\mathrm{T}}$ is then classified by an RBFN that gives the fault information. As the residual vector of FSJFs and LJFs occurring at the same joint can occupy the same region in the input space of the RBFN, an auxiliary input $\boldsymbol{\zeta}$ that gives information about the velocity of the joints is used. As there is noise in the measurement of the joint velocities, the $i = 1,\ldots,n$ component (n is the sum of the number of joints of all arms) of $\boldsymbol{\zeta}$ is defined as

$$\zeta_i(t) = \begin{cases} 1 & \text{if } |\dot{q}_i(t)| < \delta_i, \\ 0 & \text{otherwise,} \end{cases}$$

where δ_i is a threshold that can be chosen based on the measurement noise. In this work, the RBFN is trained by the Kohonen's self organizing map (Terra and Tinós, 2001). The fault criteria, which is used to avoid false alarms due to misclassified individual patterns, is defined as

$$\begin{cases} \text{fault } k = 1 & \text{if } \alpha_k = \max_{j=1}^{q}(\alpha_j) \text{ for } d \text{ consecutive samples,} \\ \text{fault } k = 0 & \text{otherwise,} \end{cases}$$

where α_k is the output $k = 1,\ldots,q-1$ of the RBFN (output q refers to the normal operation).

5. Control reconfiguration

5.1. Joint velocity and position faults

When a JPF or JVF is isolated, the sensor readings of the faulty joint are ignored and the joint position or velocity is estimated based on the kinematic constraints. As the joint positions of the faulty arm f were already estimated by the FDI system (Section 4), the component $j = 1,\ldots,n$ of the new joint position vector is defined as

$$\hat{\mathbf{q}}[j] = \begin{cases} \hat{q}_j & \text{if a JPF is isolated at joint } j, \\ \boldsymbol{\theta}[j] & \text{otherwise,} \end{cases}$$

where $\hat{q}_j$ is the estimate of the joint j position based on the other joint readings (7), and $\boldsymbol{\theta}[j]$ is the measured position of joint j. In a similar way, the component j of the new joint velocity vector is defined as

$$\hat{\dot{\mathbf{q}}}[j] = \begin{cases} \hat{\dot{q}}_j & \text{if a JVF is isolated at joint } j, \\ \dot{\boldsymbol{\theta}}[j] & \text{otherwise,} \end{cases}$$

where $\hat{\dot{q}}_j$ is the estimate of the joint j velocity based on the other joint readings, and $\dot{\boldsymbol{\theta}}[j]$ is the measured velocity of joint j.

5.2. Free-swinging joint faults

In this section, the control problem of cooperative systems with passive joints is addressed. Passive joints, which are the joints without actuation, appear in robotic manipulators as a result of free-swinging joint failures or as an inherent characteristic of the project. A single manipulator with passive joints is subject to acceleration constraints usually not integrable and, for that reason, it falls in the category or nonholonomic systems with second order nonintegrable constraints, or nonholonomic dynamic systems, which cannot be controlled by a smooth feedback controller based on Brockett's work (Liu et al., 1999).

Though a single manipulator with passive joints is, in general, a nonholonomic system, cooperative manipulators, rigidly connected to an undeformable load, with passive joints have the holonomic property (Liu et al., 1999). Thus, in Liu et al. (1999), a smooth control law based on the classical PD plus gravity compensation scheme is proposed to control the position of two cooperative arms ($m = 2$) with a number of actuated joints $n_a \geqslant k$, where k is the number of coordinates of motion in the load.

In Tinós et al. (2006), a controller for the system with passive joints based on the decomposed control of the motion and squeeze (Wen and Kreutz-Delgado, 1992) is proposed for the general case $m > 1$. A stable motion control with compensation of the gravitational torques is firstly projected ignoring the squeeze forces when $n_a \geqslant k$. For this purpose, a Jacobian matrix $\mathbf{Q}(\mathbf{q})$, which relates the velocities in the active joints to the load velocities (or the torques in the actuated joints to the resulting forces in the load by using the virtual work principle), is computed. Then, if the number of actuated joints is greater than the number of coordinates of motion in the load ($n_a > k$), the squeeze control is projected considering the component of the squeeze forces caused by the motion as a disturbance.

In this way, the control law applied in the actuated joints for the system with passive joints is given by

$$\boldsymbol{\tau}_{\mathbf{a}} = \boldsymbol{\tau}_{\mathbf{amg}} + \boldsymbol{\tau}_{\mathbf{as}}, \tag{13}$$

where $\boldsymbol{\tau}_{\mathbf{amg}}$ is the motion control law with compensation for the gravitational torques and $\boldsymbol{\tau}_{\mathbf{as}}$ is the squeeze control law.

The method to calculate the Jacobian matrix $\mathbf{Q}(\mathbf{q})$ for the cooperative system with $m > 1$ is presented in the following.

From (5)

$$m\mathbf{v_o} = \mathbf{D}_1(\mathbf{q}_1)\dot{\mathbf{q}}_1 + \mathbf{D}_2(\mathbf{q}_2)\dot{\mathbf{q}}_2 + \cdots + \mathbf{D}_m(\mathbf{q}_m)\dot{\mathbf{q}}_m. \tag{14}$$

Partitioning (14) in the quantities related to the passive and actuated joints

$$\begin{aligned} m\mathbf{v_o} &= \sum_{i=1}^{m} \mathbf{D}_{\mathbf{a}i}(\mathbf{q}_i)\dot{\mathbf{q}}_\mathbf{a} + \sum_{i=1}^{m} \mathbf{D}_{\mathbf{p}i}(\mathbf{q}_i)\dot{\mathbf{q}}_\mathbf{p} \\ &= \mathbf{D_a}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{a} + \mathbf{D_p}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{p}, \end{aligned} \tag{15}$$

where a refers to the quantities related to actuated joints and p to the quantities related to passive joints, $\mathbf{q_a}$ is the vector of angular positions of the actuated joints, and $\mathbf{q_p}$ is the vector of angular positions of the passive joints. Two cases can be considered from (5). When m is even

$$\sum_{i=1}^{m} (-1)^{i+1}\mathbf{D}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i = \mathbf{0}. \tag{16}$$

Partitioning (16)

$$\begin{aligned} &\sum_{i=1}^{m} (-1)^{i+1}\mathbf{D}_{\mathbf{a}i}(\mathbf{q}_i)\dot{\mathbf{q}}_\mathbf{a} + \sum_{i=1}^{m} (-1)^{i+1}\mathbf{D}_{\mathbf{p}i}(\mathbf{q}_i)\dot{\mathbf{q}}_\mathbf{p} \\ &\quad = \mathbf{R_a}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{a} + \mathbf{R_p}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{p} = \mathbf{0} \end{aligned} \tag{17}$$

which relates actuated and passive joint velocities when m is even. It is interesting to observe that such relationship cannot usually be found for single manipulators with passive joints (Liu et al., 1999). When m is odd

$$\sum_{i=1}^{m} (-1)^{i+1}\mathbf{D}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i = \mathbf{v_o}. \tag{18}$$

Partitioning (18)

$$\mathbf{R_a}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{a} + \mathbf{R_p}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{p} = \mathbf{v_o} \tag{19}$$

which relates actuated and passive joints velocities when m is odd. Using (15), (17), and (19),

$$\mathbf{v_o} = \mathbf{Q}(\mathbf{q})\dot{\mathbf{q}}_\mathbf{a}, \tag{20}$$

where, when m is even

$$\mathbf{Q}(\mathbf{q}) = \frac{1}{m}(\mathbf{D_a}(\mathbf{q}) - \mathbf{D_p}(\mathbf{q})\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q})\mathbf{R_a}(\mathbf{q})), \tag{21}$$

where $\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q})$ is the pseudo-inverse of matrix $\mathbf{R_p}(\mathbf{q})$, and, when m is odd,

$$\mathbf{Q}(\mathbf{q}) = (m\mathbf{I} - \mathbf{D_p}(\mathbf{q})\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q}))^{-1}(\mathbf{D_a}(\mathbf{q}) - \mathbf{D_p}(\mathbf{q})\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q})\mathbf{R_a}(\mathbf{q})). \tag{22}$$

One can observe that $\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q})$ must exist in the last equation (the robot configurations where $\mathbf{R}_\mathbf{p}^{\#}(\mathbf{q})$ does not exist are discussed in Liu et al., 1999).

The motion control of the system with passive joints when $n_a \geqslant k$ is given by

$$\boldsymbol{\tau}_{\mathbf{amg}} = \boldsymbol{\tau}_{\mathbf{am}} + \boldsymbol{\tau}_{\mathbf{ag}}, \tag{23}$$

where the motion component is given by

$$\boldsymbol{\tau}_{\mathbf{am}} = \mathbf{Q}^\mathrm{T}(\mathbf{T}^{-1}\mathbf{K_p}\Delta\mathbf{x_o} + \mathbf{K_v}\Delta\mathbf{v_o}), \tag{24}$$

where $\Delta\mathbf{x}_o = (\mathbf{x_{od}} - \mathbf{x_o})$ is the load position error, $\mathbf{x_{od}}$ is the desired position of the load, $\mathbf{K_p}$ and $\mathbf{K_v}$ are positive definite diagonal matrices, and $\Delta\mathbf{v_o} = (\mathbf{v_{od}} - \mathbf{v_o})$ is the load velocity error. The singularity of $\mathbf{T}$ depends on the minimal representation of the orientation chosen. The compensation for the gravitational, centrifugal, and Coriolis terms is given by

$$\boldsymbol{\tau}_{\mathbf{ag}} = \mathbf{g_a} - (\mathbf{R}_\mathbf{p}^{\#}\mathbf{R_a})^\mathrm{T}\mathbf{g_p} + \mathbf{Q}^\mathrm{T}\mathbf{b_o} \tag{25}$$

when m is even, and

$$\boldsymbol{\tau}_{\mathbf{ag}} = \mathbf{g_a} - (\mathbf{R}_\mathbf{p}^{\#}(\mathbf{Q} - \mathbf{R_a}))^\mathrm{T}\mathbf{g_p} + \mathbf{Q}^\mathrm{T}\mathbf{b_o} \tag{26}$$

when m is odd.

The squeeze control problem when the number of actuated joints (n_a) is greater than the number of coordinates of motion in the load (k) is now addressed. The squeeze forces in the load can be written as (Wen and

Kreutz-Delgado, 1992)

$$\mathbf{h}_{os} = \mathbf{h}_{osc} + \mathbf{h}_{osm}, \tag{27}$$

where $\mathbf{h}_{osm}$ is the term of the squeeze forces induced by the motion due to the squeeze components of the d'Alembert forces, and $\mathbf{h}_{osc}$ is the squeeze component that is not affected by the motion. An important property of $\mathbf{h}_{osm}$ is that it is not affected by joint torques in the form $\mathbf{D}^T\mathbf{h}_{osc}$ due to the motion and squeeze decomposition. Hence, $\mathbf{h}_{osm}$ can be treated as a disturbance.

The term $\mathbf{h}_{osm}$ goes to zero as $t \to \infty$ if an asymptotically stable motion control law is used. As the transient performance and convergence rate of $\mathbf{h}_{os}$ are influenced by $\mathbf{h}_{osm}$ in a feedback control approach, Wen and Kreutz-Delgado (1992) suggests a pre-processing of the squeeze forces by a strictly proper linear filter, such as an integrator, for the fault-free system.

The following squeeze control law for the system with ($n_a > k$) at time t is utilized (Tinós et al., 2006)

$$\boldsymbol{\tau}_{as} = -\mathbf{D}_{sa}^T(\mathbf{q})\boldsymbol{\beta}_s, \tag{28}$$

where

$$\mathbf{D}_{sa}(\mathbf{q}) = \begin{bmatrix} \mathbf{D}_{a1}(\mathbf{q}_1) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_{am}(\mathbf{q}_m) \end{bmatrix}$$

and $\mathbf{D}_{ai}(\mathbf{q}_i)$ relates velocities of the actuated joints at arm i and load velocities. The vector $\boldsymbol{\beta}_s$ gives the squeeze forces that should be applied at the load by the squeeze forces controller when there are passive joints in the arms of the cooperative system. For the cooperative system with passive joints, if the arms are not kinematically redundant, it is not possible to independently control all components of the squeeze forces. In (28), the components of $\boldsymbol{\beta}_s$ related to the squeeze forces that are not directly controlled are computed as a function of the components that are directly controlled. The components of the desired squeeze forces that should be applied by the other arms are then computed based on the components computed for the arm f and on the geometry of the grasping. In the case of two manipulators grasping the object, the magnitude of the desired squeeze forces are equal for both arms. The integrator suggested in Wen and Kreutz-Delgado (1992) to guarantee the stability of the force loop in the fault free system is employed for the system with passive joints.

In summary, when an FSJF is isolated and the trajectory is reconfigured (see Section 3), the controller employed for the fault-free system is switched to a new controller defined by (13), (23), (25), (26), and (28). This new controller, which is decomposed in control of motion and control of squeeze force, is applied only in the actuated joints.

5.3. Locked joint faults

A similar control system can be used to control the system with locked joints. However, the problem of the locked joints must be considered in the trajectory planning. Here, this problem will not be considered and only the control problem for the system with locked joints performing a feasible trajectory will be considered. As in the case of the system with passive joints, a Jacobian matrix relating the actuated joints, which are the joints that are not locked, and the load velocities can be defined. As the velocities of the locked joints are equal to zero, then

$$\dot{\mathbf{x}}_o = \mathbf{Q}_l\dot{\mathbf{q}}_a, \tag{29}$$

where

$$\mathbf{Q}_l = \frac{1}{m}\mathbf{D}_a. \tag{30}$$

Using the same procedure described in the last section, the control law given by (13) and (23) is utilized, where

$$\boldsymbol{\tau}_{am} = \mathbf{Q}_l^T(\mathbf{T}^{-1}\mathbf{K}_p\Delta\mathbf{x}_o + \mathbf{K}_v\Delta\mathbf{v}_o), \tag{31}$$

$$\boldsymbol{\tau}_{ag} = \mathbf{g}_a + \mathbf{Q}_l^T\mathbf{b}_o \tag{32}$$

and $\boldsymbol{\tau}_s$ is given by (28).

6. Results

The fault tolerance system was applied in a real cooperative system with two UARMII arms (Fig. 2). Each UARMII is a 3-joint, planar manipulator that floats on a thin air film on an "air table"; the armś bases are fixed to the table. The two arms are equal and the axes of all joints are parallel to the gravity force. The mass of links 1 and 2 is 0.85 kg, the mass of link 3 is 0.625 kg, and the link length is 0.203 m. The cooperative system is controlled by a PC running Matlab. This is possible because the drivers for the UARMII servo board are written as Matlab mex-files. Each joint of the UARMII contains a brushless DC direct-drive motor, an encoder, and a pneumatic brake, which allows one to simulate all faults discussed here. The use of

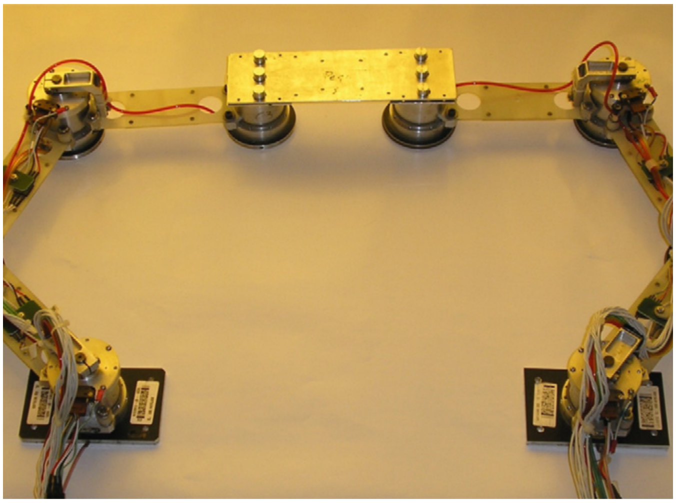

Fig. 2. Cooperative manipulator system composed of two UARMII robots.

direct-drive motors and the fact that the arm floats in an air table significantly reduce the friction in the joints. The cooperative manipulators control environment (CMCE), which allows one to perform simulations or to control directly the actual cooperative system with the same graphical user interface (GUI), is used to control and to monitor the real system. In the experiments presented here, the sampling period adopted is 0.05 s and the joint velocities are derived from the encoder readings using the adaptive filter presented in Wijngaard (1996).

Two MLPs (one for each robot) were utilized, each one with 12 inputs, 37 neurons in the hidden layer, and 3 outputs. The MLPs were trained with 3250 patterns obtained in 50 random trajectories of the fault-free cooperative system. The RBFN had 12 inputs and 13 outputs (6 FSJFs, 6 LJFs, and normal operation) and it was trained with 2506 patterns obtained in 240 random trajectories of the cooperative system with FSJFs and LJFs at different joints and 20 without faults. The parameters of the FDI system are $d = 4$ samples, $\gamma_{p1} = \gamma_{p2} = 0.05$, $\gamma_{v1} = \gamma_{v2} = 1.5$, and $\delta_i = 4 \times 10^{-3}$.

Table 1
Results of the FDI system applied to a team of two cooperative manipulators

Set	Detected faults	Isolated faults	False alarms	MTD(s)
1	337 (93.61%)	260 (72.22%)	1 (6.67%)	0.469
2	333 (92.50%)	247 (68.61%)	0 (0.00%)	0.419
3	325 (90.28%)	268 (74.44%)	0 (0.00%)	0.458

MTD means mean time to detection.

The FDI system was tested considering three trajectory sets, each of them with 360 random trajectories of the cooperative system with FSJFs and LJFs at different joints and 15 without faults. The second and third sets had the same desired trajectories but an object of mass of 0.025 kg was manipulated in the second set and an object of 0.45 kg in the third set. The desired trajectories of the first set were different from the ones of the other sets and the mass of load is equal to 0.45 kg. The results of the FDI system considering the four faults described here is summarized in Table 1.

Both strategies discussed in Section 3, namely, reconfiguration of the system starting with zero velocities and with the current velocities, were tested. In the last case, the new desired trajectory was a third order polynomial with initial velocities of the load equal to the current ones. In the real system, reconfiguration of the system starting with the current velocities should not be applied in cases where the velocities of the load are high and the resulting desired trajectory requires joint positions outside of their physical limits.

Figs. 3–5 show a trajectory of the real system where an FSJF at joint 1 of arm 1 was isolated at $t = 3.8$ s and the trajectory and control laws were reconfigured. One can observe that the time necessary to detect the fault in this trajectory (2.8 s) was higher than the mean time to detection (MTD) for the test sets presented in Table 1. In this trajectory, the time elapsed until the fault is detected was high because the velocity of the faulty joint did not increase

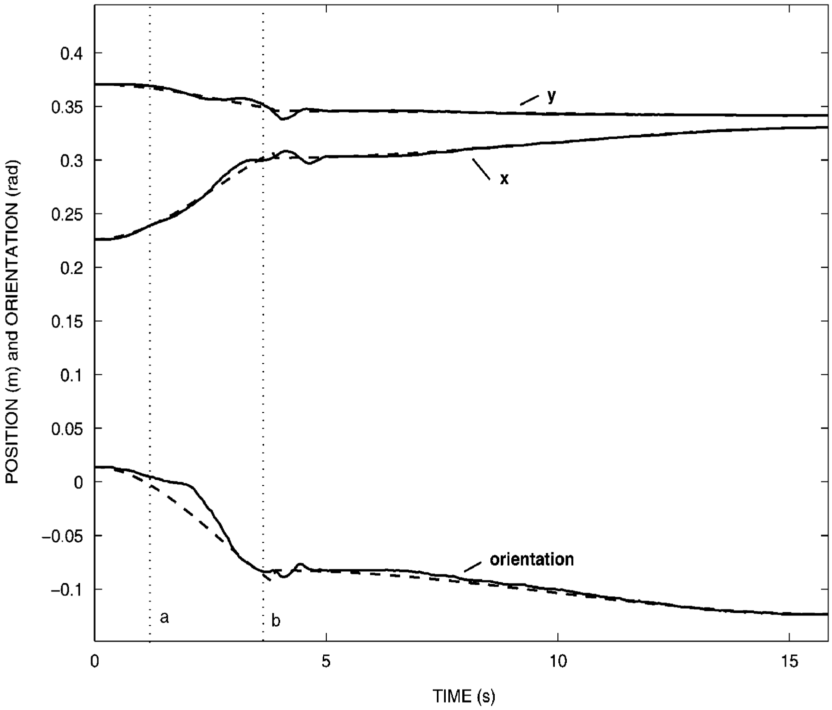

Fig. 3. Position and orientation of the object in a trajectory with an FSJF at joint 1 of arm 1. The dashed lines show the desired trajectory. The FSJF started at $t = 1$ s (dotted line "a") and was detected at $t = 3.8$ s (dotted line "b").

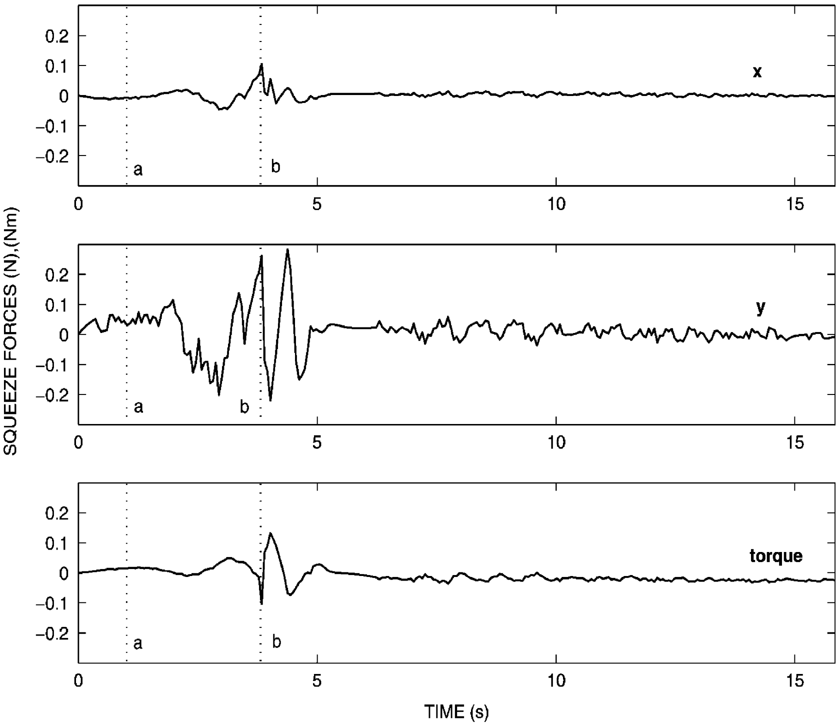

Fig. 4. Squeeze forces in the trajectory shown in Fig. 3. The torque component of the squeeze forces was not controlled after the control reconfiguration.

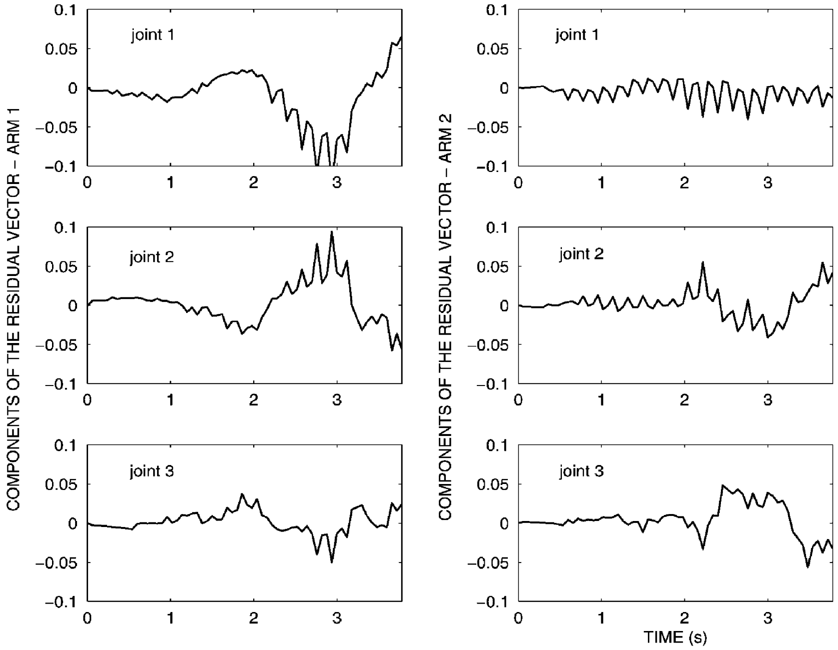

Fig. 5. Components of the residual vector in the trajectory with an FSJF at joint 1 of arm 1. The components are shown from $t = 0$ s to $t = 3.8$ s (when the fault was detected).

abruptly, due to the fact that the load kept moving in a trajectory close to the desired trajectory (Fig. 3). This fact happened in the real system because the gravitational terms did not influence the velocity of the joint and because the load was not heavy. One can also observe that the components of the squeeze forces in the y-axis increased (Fig. 4). As a result, some components of the residual vector increased (Fig. 5), making possible the detection of the fault. After the fault was detected in this trajectory, the brakes were not applied and the desired trajectory started with the actual velocities. It is possible to observe that the system was controlled even with the presence of the passive joint.

Fig. 6 shows a trajectory of the real system where a JPF at joint 2 of arm 1 started at $t = 1.0$ s and it was isolated at

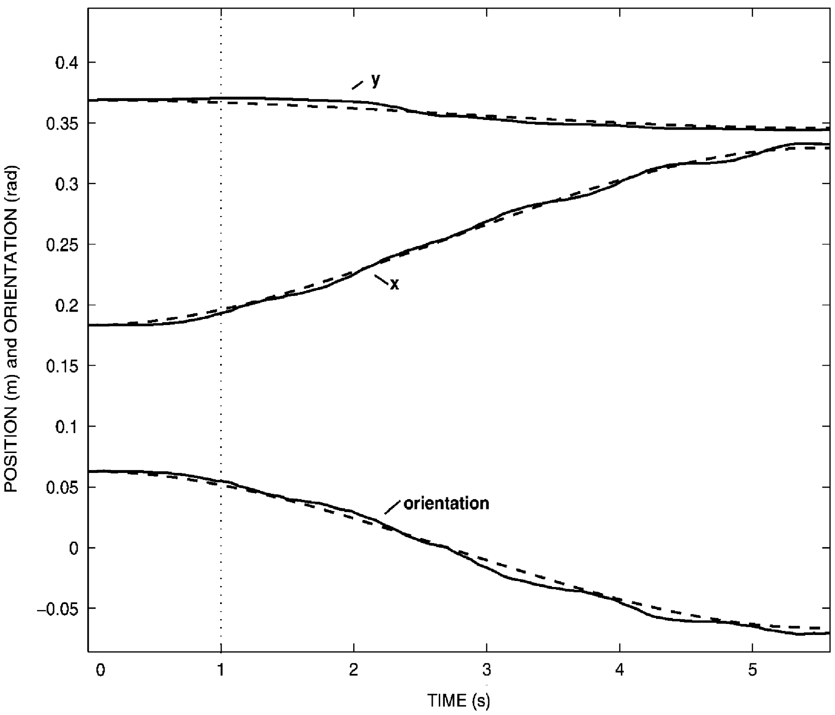

Fig. 6. Position and orientation of the object in a trajectory with JPF at joint 2 (arm 1) for the real system. The JPF started at $t = 1$ s and was detected at $t = 1.05$ s (vertical dotted line).

$t = 1.05$ s. After the fault was detected, the controller ignored the measure produced by the sensor of position of the joint 2 of arm 1, and utilized its estimate produced by (7).

The number of correctly isolated faults was small in the system mainly because FSJFs were sometimes mistaken with LJFs. This occurs because sometimes the velocities of the faulty vectors were small due to the small gravitational torques at the joints. However, in these cases, even with FSJFs, the load converged to the desired positions and the fault did not compromise the performance of the system. This can occur, for example, if it is not necessary to apply high torques at the faulty joint in a given trajectory.

7. Conclusions

This work presents a fault tolerance framework for cooperative manipulators. The faults are firstly detected and isolated and, then, the control system and the desired trajectory are reconfigured. Tests of the fault tolerance system applied to two cooperative robots were presented.

The performance of the fault detection and isolation (FDI) system can be improved with additional tests. When a fault is detected, the brakes can be activated and tests can be performed in order to verify if the fault was correctly isolated. For example, if a locked joint fault (LJF) is declared, the controller can try to move this joint in order to confirm the fault isolation. Similar tests can be made to confirm the isolation of other faults. This strategy can also be used to isolate multiple faults, which can be detected but cannot be correctly isolated by the proposed FDI system. When multiple faults are considered, additional tests to isolate all faults must be made in all joints after the detection of a fault. A relevant future work is to study additional tests to be made to isolate multiple faults.

Acknowledgment

The authors would like to thank FAPESP (Proc. 99/10031-1 and 04/04289-6) for the financial support to this project.

References

Bonitz, R., & Hsia, T. (1996). Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, *12*, 78–89.

Carignan, C. R., & Akin, D. L. (1988). Cooperative control of two arms in the transport of an inertial load in zero gravity. *IEEE Journal of Robotics and Automation*, *4*(4), 414–419.

Dhillon, B. S., & Fashandi, A. R. M. (1997). Robotic systems probabilistic analysis. *Microelectronics and Reliability*, *37*(2), 211–224.

English, J. D., & Maciejewski, A. A. (1998). Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures. *IEEE Transactions on Robotics and Automation*, *14*(4), 566–575.

Goel, M., Maciejewski, A. A., & Balakrishnan, V. (2004). Analyzing unidentified locked-joint failures in kinematically redundant manipulators. *Journal of Robotic Systems*, *22*(1), 15–29.

Hassan, M., & Notash, L. (2004). Analysis of active joint failure in parallel robot manipulators. *Journal of Mechanical Design*, *126*(6), 959–968.

Hassan, M., & Notash, L. (2005). Design modification of parallel manipulators for optimum fault tolerance to joint jam. *Mechanism and Machine Theory*, *40*(5), 559–577.

Jatta, F., Legnani, G., Visioli, A., & Ziliani, G. (2006). On the use of velocity feedback in hybrid force/velocity control of industrial manipulators. *Control Engineering Practice*, *14*(9), 1045–1055.

Liu, Y. H., Xu, Y., & Bergerman, M. (1999). Cooperation control of multiple manipulators with passive joints. *IEEE Transactions on Robotics and Automation*, *15*(2), 258–267.

Luh, J. Y. S., & Zheng, Y. F. (1987). Constrained relations between two coordinated industrial robots for robot motion control. *International Journal of Robotics Research*, *6*(3), 60–70.

McIntyre, M. L., Dixon, W. E., Dawson, D. M., & Walker, I. D. (2005). Fault identification for robot manipulators. *IEEE Transactions on Robotics*, *21*(5), 1028–1034.

Miyagi, P. E., & Riascos, L. A. M. (2006). Modeling and analysis of fault-tolerant systems for machining operations based on petri nets. *Control Engineering Practice*, *14*(4), 397–408.

Monteverde, V., & Tosunoglu, S. (1999). Development and application of a fault tolerance measure for serial and parallel robotic structures. *International Journal of Modeling and Simulation*, *19*(1), 45–51.

Notash, L. (2000). Joint sensor fault detection for fault tolerant parallel manipulators. *Journal of Robotic Systems*, *17*(3), 149–157.

Schneider, H., & Frank, P. M. (1996). Observer-based supervision and fault-detection in robots using nonlinear and fuzzy logic residual evaluation. *IEEE Transactions on Control Systems Technology*, *4*(3), 274–282.

Sciavicco, L., & Siciliano, B. (1996). *Modeling and control of robot manipulators*. McGraw-Hill International Editions.

Terra, M. H., & Tinós, R. (2001). Fault detection and isolation in robotic manipulators via neural networks—a comparison among three architectures for residual analysis. *Journal of Robotic Systems*, *18*(7), 357–374.

Tinós, R., Terra, M. H., & Bergerman, M. (2001). Fault detection and isolation in cooperative manipulators via artificial neural networks. In *Proceedings of IEEE conference on control applications* (pp. 988–1003).

Tinós, R., Terra, M. H., & Ishihara, J. Y. (2006). Motion and force control of cooperative robotic manipulators with passive joints. *IEEE Transactions on Control Systems Technology*, *14*(4), 725–734.

Vemuri, A. T., & Polycarpou, M. M. (2004). A methodology for fault diagnosis in robotic systems using neural networks. *Robotica*, *22*(4), 419–438.

Visinsky, M. L., Cavallaro, J. R., & Walker, I. D. (1994). Robotic fault detection and fault tolerance: A survey. *Reliability Engineering & Systems Safety*, *46*, 139–158.

Visinsky, M. L., Cavallaro, J. R., & Walker, I. D. (1995). A dynamic fault tolerance framework for remote robots. *IEEE Transactions on Robotics and Automation*, *11*(4), 477–490.

Vukobratovic, M., & Tuneski, A. (1998). Mathematical model of multiple manipulators: Cooperative compliant manipulation on dynamical environments. *Mechanism and Machine Theory*, *33*, 1211–1239.

Wen, T., & Kreutz-Delgado, K. (1992). Motion and force control for multiple robotics manipulators. *Automatica*, *28*(4), 729–743.

Wijngaard, W. (1996). An adaptive filter for low frequency encoder applications. *Measurement*, *18*(1), 1–7.