
A FAULT DETECTION AND ISOLATION SYSTEM FOR COOPERATIVE MANIPULATORS

Renato Tinós*

rtinos@ffclrp.usp.br

Marco Henrique Terra†

terra@sc.usp.br

* Departamento de Física e Matemática, FFCLRP - Universidade de São Paulo (USP), Ribeirão Preto, SP, 14040-901, Brasil

† Departamento de Engenharia Elétrica, EESC - Universidade de São Paulo (USP), São Carlos, SP, 13560-970, Brasil

ABSTRACT

The problem of fault detection and isolation (FDI) in cooperative manipulators is addressed in this paper. Four FDI procedures are developed to deal with free-swinging joint faults, locked joint faults, incorrectly measured joint position, and incorrectly measured joint velocity. Free-swinging and locked joint faults are isolated via neural networks. For each arm, a Multilayer Perceptron (MLP) is used to reproduce the dynamics of the fault-free robot. The outputs of each MLP are compared to the actual joint velocities in order to generate a residual vector which is then classified by an RBF network. The remaining faults are isolated based on the kinematic constraints imposed on the cooperative system. Results obtained via simulations and via an actual cooperative manipulator robot are presented.

Keywords: Fault Detection, Fault Isolation, Robotic Manipulators, Co-operation, Neural Networks.

1 INTRODUCTION

The actuation areas of Robots have been spreading from industries and laboratories into hospitals, deep sea, outer space, nuclear facilities, and other hazardous, unstructured, or hard-to-reach environments. In these environments, robots are employed to avoid the exposition of human beings to danger or because of the reliability of robots in executing repet-

itive tasks. However, faults in robots, which are not unusual due to their inherent complexity, can put at risk the robot, its mission, and the working environment.

There are several sources of faults in robots, such as electrical, mechanical, hydraulic, and of software (Visinsky et al., 1994). There are good reasons to research and to develop fault detection and isolation (FDI) systems for robots, in particular to improve their safety to work among humans.

Robotic systems with kinematic or actuation redundancy are interesting in applications where the fault problem should be addressed because the number of degrees of freedom (dof) in these systems is greater than the dof required to manipulate the load. Actuation redundancy can be found in only closed-link mechanisms as cooperative systems formed by two or more arms (Nakamura, 1991). As in the humans, where the use of two arms presents 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). Examples of such tasks include the manipulation of heavy, large or flexible loads, assembly of structures, and manipulation of objects that can slide from only one robot end-effector.

Actuation redundancy makes the use of cooperative robots in unstructured or hazardous environments very appealing. However, as cited before, FDI is crucial in these environments. Because of the dynamic coupling of the joints, inertia, and gravitation, the faulty arms can quickly accelerate into wild motions that can cause serious damage (Visinsky et al., 1994). Furthermore, as the controller is not projected to operate with faults, the squeeze forces can increase caus-

Artigo submetido em 27/03/2007

1a. Revisão em 14/10/2008

Aceito sob recomendação do Editor Associado

Prof. José Reinaldo Silva

ing damage to the load and instability in the cooperative system (Tinós et al., 2006).

This work develops an FDI scheme for cooperative robots rigidly connected to an undeformable load based on procedures to isolate four kind of faults: free-swinging joint faults (FSJFs), locked joint faults (LJFs), incorrectly measured joint position faults (JPFs), and incorrectly measured joint velocity faults (JVFs). The faults are detected and isolated as follows:

a) JPFs and JVFs are detected and isolated based on kinematics constraints imposed by the closed kinematic chain. If the number of arms m in the cooperative system is greater than two ($m > 2$) and supposing that only one fault occurs each time, the manipulator with the wrong measurement can be directly detected by checking the estimates of the load position (or velocity for JVFs) obtained for each arm. Then, the estimate of each joint position (or velocity for JVFs) of the faulty arm is computed and compared to its measurement in order to isolate the fault. If $m = 2$, the arm with the wrong measurement cannot be identified just by checking the estimate of position (or velocity) of the load obtained for each arm. In this case, the estimate of the joint position (or velocity) should be done for all arms.

b) FSJFs and LJFs are detected by artificial neural networks (ANNs). The dynamics of the arms are mapped by Multi-layer Perceptrons (MLPs), which produces outputs that are compared to actual velocity measurements in order to generate the residual vector. Then, a Radial Basis Function Network (RBFN) is utilized to classify the residual vector.

This paper is organized as follows: Section 2 describes the kinematics and dynamics of cooperative manipulators; Section 3 describes the FDI system; Section 4 presents the results of the FDI system in simulations and in an actual cooperative robot with two arms; finally, the conclusions are presented in Section 5.

2 COOPERATIVE MANIPULATORS

The equation of motion for the i -th arm in a fault-free multi-robot system with m robots rigidly connected to an undeformable load is given by

$$\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i = \boldsymbol{\tau}_i - \mathbf{J}_i(\mathbf{q}_i)^T \mathbf{h}_i \quad (1)$$

where \mathbf{q}_i is the vector of joint angles of arm i , $i = 1, \dots, m$, $\boldsymbol{\tau}_i$ is the vector of applied torques 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 the geometric Jacobian (from joint velocity to end-effector velocity) of arm i , $\mathbf{h}_i = [\mathbf{f}_i^T \boldsymbol{\eta}_i^T]^T$ is the force vector at the end-effector of arm i , \mathbf{f}_i is the vector of

spatial forces at the end-effector of arm i , and $\boldsymbol{\eta}_i$ is the vector of torques at the end-effector of arm i ; the friction terms were not shown for simplicity. The combined dynamics of all arms can be written in only one equation as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{J}(\mathbf{q})^T \mathbf{h} \quad (2)$$

where $\mathbf{q} = [\mathbf{q}_1^T \mathbf{q}_2^T \dots \mathbf{q}_m^T]^T$, $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T \boldsymbol{\tau}_2^T \dots \boldsymbol{\tau}_m^T]^T$, $\mathbf{h} = [\mathbf{h}_1^T \mathbf{h}_2^T \dots \mathbf{h}_m^T]^T$, $\mathbf{M}(\mathbf{q})$ is formed by the individual inertia matrices of the arms, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is formed by the individual centrifugal and Coriolis matrices of the arms, $\mathbf{g}(\mathbf{q})$ is formed by the individual gravitational terms of the arms, and $\mathbf{J}(\mathbf{q})$ is formed by the terms $\mathbf{J}_i(\mathbf{q}_i)$ for $i = 1, \dots, m$.

The equation of motion for the manipulated object (load) is given by

$$\mathbf{M}_o \dot{\mathbf{v}}_o + \mathbf{b}_o(\mathbf{x}_o, \mathbf{v}_o) = \mathbf{J}_o(\mathbf{x}_o)^T \mathbf{h} \quad (3)$$

where $\mathbf{x}_o = [\mathbf{p}_o^T \boldsymbol{\phi}_o^T]^T$ is the k -dimensional vector of position and orientation at the origin of the frame attached to the center of mass of the load (frame CM), \mathbf{p}_o is the vector of load position, $\boldsymbol{\phi}_o$ is the minimal representation of orientation of the load, $\mathbf{v}_o = [\dot{\mathbf{p}}_o^T \boldsymbol{\omega}_o^T]^T$ is the vector of linear and angular velocities of the load, \mathbf{b}_o is the vector of centrifugal, Coriolis, and gravitational terms, \mathbf{M}_o is the load inertia matrix, and $\mathbf{J}_o(\mathbf{x}_o) = [\mathbf{J}_{o1}(\mathbf{x}_o)^T \dots \mathbf{J}_{om}(\mathbf{x}_o)^T]^T$, where $\mathbf{J}_{oi}(\mathbf{x}_o)$ converts velocities of the load into velocities of the end-effector of arm i . In the three-dimensional space, $\mathbf{p}_o = [x_o \ y_o \ z_o]^T$, and either Euler angles or RPY (Roll-Pitch-Yaw) angles can be chosen as the minimal representation of the load orientation, i.e., $\boldsymbol{\phi}_o = [\varphi_o \ v_o \ \psi_o]^T$. The velocities \mathbf{v}_o can be calculated by $\mathbf{v}_o = \mathbf{T}(\mathbf{x}_o)\dot{\mathbf{x}}_o$ (Sciavicco and Siciliano, 1996), where $\mathbf{T}(\mathbf{x}_o)$ is a transformation matrix used to relate the angular velocities to the derivative of the minimal representation of the orientation (euler angles or RPY angles) in the 3-dimensional space ($\mathbf{T}(\mathbf{x}_o) = \mathbf{I}$ for planar manipulators, where \mathbf{I} is the identity matrix).

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

$$\mathbf{x}_o = \boldsymbol{\varphi}_1(\mathbf{q}_1) = \boldsymbol{\varphi}_2(\mathbf{q}_2) = \dots = \boldsymbol{\varphi}_m(\mathbf{q}_m) \quad (4)$$

where $\boldsymbol{\varphi}_i(\mathbf{q}_i)$ 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

$$\mathbf{v}_o = \mathbf{D}_1(\mathbf{q}_1)\dot{\mathbf{q}}_1 = \mathbf{D}_2(\mathbf{q}_2)\dot{\mathbf{q}}_2 = \dots = \mathbf{D}_m(\mathbf{q}_m)\dot{\mathbf{q}}_m \quad (5)$$

where $\mathbf{D}_i(\mathbf{q}_i) = \mathbf{J}_{oi}(\mathbf{x}_o)^{-1}\mathbf{J}_i(\mathbf{q}_i)$ is the Jacobian relating joint velocities of arm i and load velocities. It is important to observe that the matrix $\mathbf{J}_{oi}(\mathbf{x}_o)$ is nonsingular.

The squeeze forces are given by (Wen and Kreutz-Delgado, 1992)

$$\mathbf{h}_{os} = \mathbf{P}_s(\mathbf{x}_o)\mathbf{h} \quad (6)$$

where the matrix $\mathbf{P}_s(\mathbf{x}_o)$ transforms forces and torques in the end-effectors into squeeze forces in the load.

Several solutions have been proposed to deal with the control problem in fault-free cooperative manipulators rigidly connected to an undeformable load as the master/slave strategy (Luh and Zheng, 1987), the optimal division of the load control (Carignan and Akin, 1988), (Nahon and Angeles, 1992), the definition of new task objectives or variables (Koivo and Unseren, 1991), (Caccavale, 1997), and the hybrid control of motion and squeeze in the object (Uchiyama, 1998), (Wen and Kreutz-Delgado, 1992), (Bonitz and Hsia, 1996).

3 FDI SYSTEM

The FDI system is proposed based on the following faults: FSJF, where an actuation loss occurs in one arm joint (English and Maciejewski, 1998); LJF, where one arm joint is locked (Goel et al., 2004); JPF, where the measurement of the joint position is not correct (Notash, 2000), and JVf, where the measurement of the joint velocity is not correct. JPFs and JVfS generally occur due to sensor faults. By simplicity, the occurrence of only one fault at a time is considered. A three step FDI system is applied here in each sample time. First, JPFs are detected by analyzing the position constraints (Eq. 4). Then, JVfS are detected by analyzing the velocity constraints (Eq. 5). The last step is the detection of FSJFs and LJFs via ANNs. This sequence is important because undetected JPFs can cause the false detection of other faults as joint position measurements are used in Eq. (5) and as inputs of the ANNs. The same occurs for undetected JVfS in FSJFs and LJFs as joint velocity measurements are used as inputs of the ANNs.

3.1 Incorrectly Measured Joint Position Faults (JPFs)

In (Notash, 2000), the direct kinematics is used to detect joint position sensor faults in parallel manipulators. The direct kinematics problem (knowing the joint positions, identify the position and orientation of the load) is not trivial in parallel manipulators because they 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. The problem of fault tolerance in parallel manipulators is still addressed in (Hassan and Notash, 2004) and (Hassan and Notash, 2005).

In this work, the direct kinematics and the kinematic constraints imposed by the closed chain are used to detect JPFs

in cooperative manipulators. Two cases are considered: when $m = 2$ or $m > 2$, i.e., there are two or more than two manipulators in the cooperative system.

3.1.1 JPFs when $m > 2$

As \mathbf{x}_o can be calculated using the joint positions of any arm (Eq. 4), it is possible to identify the arm f with the wrong joint position measurements if $m > 2$. A wrong estimate of the load position \mathbf{x}_o is produced by the arm with the wrong measurements, which differs from the estimate of the other $m - 1$ arms. A JPF is detected in arm f if

$$\|\hat{\mathbf{x}}_{of}(\boldsymbol{\theta}_f) - \hat{\mathbf{x}}_{oi}(\boldsymbol{\theta}_i)\| > \gamma_{p1} \quad (7)$$

for all $i = 1, \dots, m$ and $i \neq f$

where $\hat{\mathbf{x}}_{oi}(\boldsymbol{\theta}_i)$ is the estimate of \mathbf{x}_o using the measurements $\boldsymbol{\theta}_i$ of the joint positions \mathbf{q}_i in arm i , $\boldsymbol{\theta}_f$ is the vector of the measured positions of the joints in arm f , $\|\cdot\|$ represents the Euclidean norm, and the threshold γ_{p1} is adjusted by the designer to avoid that false alarms appear due to the presence of noise in the joint measurements. The next step is to estimate the position of each joint $j = 1, \dots, n_f$ of arm f

$$\hat{q}_{fj} = \psi_{pj}(\boldsymbol{\theta}_f, \hat{\mathbf{x}}_o) \quad (8)$$

where ψ_{pj} is the kinematic function used to estimate the position of joint j , and

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

Calculating again the estimate of vector \mathbf{x}_o for arm f for each new estimate \hat{q}_{fj} , the JPF in joint j of arm f is detected when

$$\|\hat{\mathbf{x}}_o - \hat{\mathbf{x}}_{of}(\boldsymbol{\theta}_f, \hat{q}_{fj})\| < \gamma_{p2} \quad (9)$$

where $\hat{\mathbf{x}}_{of}(\boldsymbol{\theta}_f, \hat{q}_{fj})$ 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. The threshold γ_{p2} is adjusted by the designer to avoid that faults are hidden due to the presence of noise in the joint measurements.

The procedure to detect and to isolate JPFs when $m > 2$ can be summarized as follows: compare the estimate of \mathbf{x}_o for all arms (Eq. 7); if all values are close, a JPF is not announced, otherwise, calculate for all joints of the faulty arm the estimate of the joint positions (Eq. 8), and test Eq. (9) for all joints; if the test is satisfied for joint j , announce a JPF in this joint.

3.1.2 JPFs when $m = 2$

If $m = 2$, the faulty arm cannot be identified just by checking the estimates of \mathbf{x}_o . However, it is possible to detect a JPF

by comparing the two estimates of \mathbf{x}_o . In this way, a JPF is detected when $m = 2$ if

$$\|\hat{\mathbf{x}}_{o1}(\boldsymbol{\theta}_1) - \hat{\mathbf{x}}_{o2}(\boldsymbol{\theta}_2)\| > \gamma_{p1}. \quad (10)$$

As it is not possible to identify the arm with the fault, the joint positions estimate (Eq. 8) should be done for all joints in the two arms using, instead of the the value of $\hat{\mathbf{x}}_o$, the estimate obtained using the joint positions of the other arm. For arm 1, the position of the joint $j = 1, \dots, n_1$ is estimated by

$$\hat{q}_{1j} = \psi_{p_j}(\boldsymbol{\theta}_1, \hat{\mathbf{x}}_{o2}(\boldsymbol{\theta}_2)) \quad (11)$$

and, for arm 2, the position of the joint $j = 1, \dots, n_2$ is estimated by

$$\hat{q}_{2j} = \psi_{p_j}(\boldsymbol{\theta}_2, \hat{\mathbf{x}}_{o1}(\boldsymbol{\theta}_1)). \quad (12)$$

Calculating again the estimate of vector \mathbf{x}_o for each new estimate of joint position (eqs. 11 and 12), the JPF in joint j of arm 1 is detected when

$$\|\hat{\mathbf{x}}_{o1}(\boldsymbol{\theta}_1, \hat{q}_{1j}) - \hat{\mathbf{x}}_{o2}(\boldsymbol{\theta}_2)\| < \gamma_{p2} \quad (13)$$

where $\hat{\mathbf{x}}_{o1}(\boldsymbol{\theta}_1, \hat{q}_{1j})$ is the vector of positions and orientations of the load estimated for arm 1 substituting the measured position of joint j by its estimate \hat{q}_{1j} and using the measured positions of the other joints, and the JPF in joint j of arm 2 is detected when

$$\|\hat{\mathbf{x}}_{o1}(\boldsymbol{\theta}_1) - \hat{\mathbf{x}}_{o2}(\boldsymbol{\theta}_2, \hat{q}_{2j})\| < \gamma_{p2} \quad (14)$$

where $\hat{\mathbf{x}}_{o2}(\boldsymbol{\theta}_2, \hat{q}_{2j})$ is the vector of positions and orientations of the load estimated for arm 2 substituting the measured position of joint j by its estimate \hat{q}_{2j} and using the measured positions of the other joints

3.2 Incorrectly Measured Joint Velocity Faults (JVFs)

As it is possible to calculate the velocity of the load by using the joint velocities of any arm (Eq. 5), JVFs can be detected in a similar way of JPFs.

3.2.1 JVFs when $m > 2$

By using Eq. (5), it is possible to identify the arm f with the wrong joint velocity measurements if $m > 2$. A JVF is detected in arm f if

$$\|\hat{\mathbf{v}}_{of}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f) - \hat{\mathbf{v}}_{oi}(\dot{\boldsymbol{\theta}}_i, \boldsymbol{\theta}_i)\| > \gamma_{v1} \quad (15)$$

for all $i = 1, \dots, m$ and $i \neq f$

where $\hat{\mathbf{v}}_{oi}(\dot{\boldsymbol{\theta}}_i, \boldsymbol{\theta}_i)$ is the estimate of \mathbf{v}_o using the measured velocities $\dot{\boldsymbol{\theta}}_i$ and the measured position $\boldsymbol{\theta}_i$ of the joints in arm

i , $\dot{\boldsymbol{\theta}}_f$ is the vector of the measured velocities of the joints in arm f , and the threshold γ_{v1} is adjusted by the designer to avoid that false alarms appear due to the presence of noise in the joint measurements. The next step is to estimate the velocity of each joint $j = 1, \dots, n_f$ of arm f

$$\hat{q}_{fj} = \psi_{v_j}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f, \hat{\mathbf{v}}_o) \quad (16)$$

where ψ_{v_j} is the kinematic function used to estimate the velocity of joint j , and

$$\hat{\mathbf{v}}_o = \frac{1}{m-1} \sum_{i=1, i \neq f}^m \mathbf{D}_i(\boldsymbol{\theta}_i) \dot{\boldsymbol{\theta}}_i.$$

Calculating again the estimate of vector \mathbf{v}_o for arm f for each new estimate \hat{q}_{fj} , the JVF in joint j of arm f is detected when

$$\|\hat{\mathbf{v}}_o - \hat{\mathbf{v}}_{of}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f, \hat{q}_{fj})\| < \gamma_{v2} \quad (17)$$

where $\hat{\mathbf{v}}_{of}(\dot{\boldsymbol{\theta}}_f, \boldsymbol{\theta}_f, \hat{q}_{fj})$ is the vector of velocities of the load estimated for arm f substituting the measured velocity of joint j by its estimate \hat{q}_{fj} and using the measured velocities of the other joints. The threshold γ_{v2} is adjusted by the designer to avoid that faults are hidden due to the presence of noise in the joint measurements.

The procedure to detect and to isolate JVFs when $m > 2$ can be summarized as follows: compare the estimate of \mathbf{v}_o for all arms (Eq. 15); if all values are close, a JVF is not announced, otherwise, calculate for all joints of the faulty arm the estimate of the joint velocities (Eq. 16), and test Eq. (17) for all joints; if the test is satisfied for joint j , announce a JVF in this joint.

3.2.2 JVFs when $m = 2$

If $m = 2$, the faulty arm cannot be identified just by checking the estimates of $\hat{\mathbf{x}}_o$. However, it is possible to detect a JVF by comparing the two estimates of $\hat{\mathbf{x}}_o$. In this way, a JVF is detected when $m = 2$ if

$$\|\hat{\mathbf{v}}_{o1}(\dot{\boldsymbol{\theta}}_1, \boldsymbol{\theta}_1) - \hat{\mathbf{v}}_{o2}(\dot{\boldsymbol{\theta}}_2, \boldsymbol{\theta}_2)\| > \gamma_{v1}. \quad (18)$$

As it is not possible to identify the arm with the fault, the joint velocities estimate (Eq. 16) should be done for all joints in the two arms using, instead of the the value of $\hat{\mathbf{x}}_o$, the estimate obtained using the joint velocities of the other arm. For arm 1, the velocity of the joint $j = 1, \dots, n_1$ is estimated by

$$\hat{q}_{1j} = \psi_{v_j}(\dot{\boldsymbol{\theta}}_1, \boldsymbol{\theta}_1, \hat{\mathbf{v}}_{o2}(\dot{\boldsymbol{\theta}}_2, \boldsymbol{\theta}_2)) \quad (19)$$

and, for arm 2, the velocity of the joint $j = 1, \dots, n_2$ is estimated by

$$\hat{q}_{2j} = \psi_{v_j}(\dot{\boldsymbol{\theta}}_2, \boldsymbol{\theta}_2, \hat{\mathbf{v}}_{o1}(\dot{\boldsymbol{\theta}}_1, \boldsymbol{\theta}_1)). \quad (20)$$

Calculating again the estimate of vector $\dot{\mathbf{x}}_o$ for each new estimate of joint velocity (eqs. 19 and 20), the JVF in joint j of arm 1 is detected when

$$\|\hat{\mathbf{v}}_{o1}(\dot{\theta}_1, \theta_1, \hat{q}_{1j}) - \hat{\mathbf{v}}_{o2}(\dot{\theta}_2, \theta_2)\| > \gamma_{v2}. \quad (21)$$

where $\hat{\mathbf{v}}_{o1}(\dot{\theta}_1, \theta_1, \hat{q}_{1j})$ is the vector of velocities of the load estimated for arm 1 substituting the measured velocity of joint j by its estimate \hat{q}_{1j} and using the measured velocities of the other joints, and the JVF in joint j of arm 2 is detected when

$$\|\hat{\mathbf{v}}_{o1}(\dot{\theta}_1, \theta_1) - \hat{\mathbf{v}}_{o2}(\dot{\theta}_2, \theta_2, \hat{q}_{2j})\| > \gamma_{v2}. \quad (22)$$

where $\hat{\mathbf{v}}_{o2}(\dot{\theta}_2, \theta_2, \hat{q}_{2j})$ is the vector of velocities of the load estimated for arm 2 substituting the measured velocity of joint j by its estimate \hat{q}_{2j} and using the measured velocities of the other joints.

3.3 Free-Swinging Joint Faults (FSJFs) and Locked Joint Faults (LJFs)

As FSJFs and LJFs introduce dynamic effects in the cooperative system, the residual generation and analysis concept can be used to detect and isolate these faults (Isermann and Ballé, 1997). In the residual generation, the mathematical model is generally used to reproduce the dynamic behavior of the fault-free system. The deviation of the output predicted by the model from actual output measurements forms the so-called residuals which, when properly analyzed, provides valuable information about the failures. Modeling errors, however, may obscure the effects of some faults and can be a source of false alarms (Gertler, 1997). To solve this problem, robust FDI schemes have been proposed (Mangoubi, 1998), (Chen and Patton, 1999).

Alternatively, one may resort to artificial intelligence techniques like knowledge-based systems, fuzzy logic, and ANNs. ANNs have been employed for FDI mainly in static systems, and less intensively in dynamic systems (Korbicz, 1997). In most applications, ANNs are used as classifiers based on measurements of the process output. In dynamic systems, however, the outputs are affected by the inputs and, therefore, this procedure generally is not valid. An alternative is to employ the residual generation concept, where one ANN is used as a classifier based on the residuals produced by the system mathematical model or by another ANN (Köppen-Selinger and Frank, 1996).

FDI for individual manipulators has been typically pursued employing the robot mathematical model for residual generation (Visinsky et al., 1994). For residual analysis, fixed thresholds can be used. However, the effects of modeling errors and sensor noise fluctuate dynamically with the robot motion and with the faults, resulting in false alarms

and detection errors. To avoid this problem, (Visinsky et al., 1995) employs time-varying state-dependent thresholds to achieve robustness in parity relations. In (Schneider and Frank, 1996), a robust observer is used for residual generation and fuzzy logic is used to produce dynamic thresholds to mask the effects of unmodeled friction. In (McIntyre et al., 2005), a nonlinear observer is proposed to identify a class of actuator faults after the detection of the fault by some other method. In (Naughton et al., 1996), a robust observer is employed for residual generation and an MLP classifies the residual vector. In an interesting approach, (Vemuri and Polycarpou, 2004) maps the fault vector employing an ANN trained using a robust observer. Overall, one problem with FDI methods which rely on the system mathematical model is that, for some kinds of robots, detailed modeling is difficult.

In (Terra and Tinós, 2001), the mathematical model of the robot is not used. An MLP is used to map the dynamics of the arm and an RBFN classifies the residual vector. The MLP mapping is static, which is possible because states are considered measurable, the sample time is small, and control signals are used in the MLP inputs. In fact, this procedure is valid only if the states are measurable. If the states are not measurable, but the measured outputs have sufficient information about the states (i.e., if the system is observable), static ANNs with delayed values of process outputs and control signals as inputs (Sorsa and Koivo, 1993) or dynamic ANNs (static ANNs provided with dynamic elements) (Marcu and Mirea, 1997) should be employed. Observe that the Multi-Input Multi-Output scheme is used to map the dynamics of the fault-free system. A Multi-Input Single-Output scheme could be used to reproduce the dynamics of the system instead of the Multi-Input Multi-Output scheme, but the second approach was chosen because it results in a smaller time of processing for this problem.

To the best of the authors knowledge, only in (Tinós et al., 2001) an FDI system for cooperative manipulators was presented. There, only one MLP is trained to reproduce the dynamics of all arms (Eq. 2). As the end-effector forces are functions of the joint variables, 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 at instant $t + \Delta t$, which are compared with the measured joint velocities at instant $t + \Delta t$ in order to generate the residual vector. The residual vector is then classified by a Radial Basis Function Network (RBFN) that gives the fault information. The use of only one MLP is an interesting approach when the end-effector forces are not measured. However, most of the controllers for cooperative manipulators use force sensors to minimize the squeeze forces on the load, and these variables can be very useful to map the system dynamics. Furthermore, the mapping of the MLP in

(Tinós et al., 2001) is dependent on the load parameters, such as the load mass. If the system manipulates another object, the ANN have to be trained again.

Here, the fault-free dynamic behavior of each arm is mapped by a different MLP. This scheme is interesting because the mapping is not dependent on the load parameters. The inputs of the MLP i are the joint positions, velocities, torques, and end-effector forces of arm i at instant t (Figure 1).

If the sampling period Δt is sufficiently small, the dynamics of the fault-free robot i (Eq. 1) can be represented by

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right) \quad (23)$$

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

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}_\phi\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right) \quad (24)$$

where $\mathbf{f}_\phi(\cdot)$ is a nonlinear function vector representing the dynamics of the arm i with the fault ϕ . The function of the fault ϕ is defined as

$$\mathbf{r}_i(t + \Delta t) = \mathbf{f}\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right) + \mathbf{f}_\phi\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right). \quad (25)$$

The outputs of the MLP i should reproduce the joint velocities of the fault-free arm i at time $t + \Delta t$ and can be expressed as

$$\hat{\dot{\mathbf{q}}}_i(t + \Delta t) = \mathbf{f}\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right) + \mathbf{e}\left(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \boldsymbol{\tau}_i(t)\right) \quad (26)$$

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

$$\hat{\mathbf{r}}_i(t + \Delta t) = \dot{\mathbf{q}}_i(t + \Delta t) - \hat{\dot{\mathbf{q}}}_i(t + \Delta t). \quad (27)$$

By Eq. (23-27), it can be observed that the residual vector of arm i is equal to the mapping error vector for the fault-free case. The mapping error vector must be sufficiently small when compared to the fault function vector in order to allow the detection of the fault. The residual vector from all arms $\hat{\mathbf{r}}(t + \Delta t) = [\hat{\mathbf{r}}_1(t + \Delta t)^T \dots \hat{\mathbf{r}}_m(t + \Delta t)^T]^T$ are then classified by an RBFN trained by the Kohonen Self Organizing Map (Terra and Tinós, 2001). As the residual vector of FSJFs and LJFs occurring in the same joint can occupy the same region in the input space of the RBFN, an auxiliary input vector $\boldsymbol{\zeta}$ that gives information about the velocity of the joints is used. The use of $\boldsymbol{\zeta}$ is motivated by the fact that the velocity of the faulty joint is zero in LJFs. Due to the noise in the measurement of the joint velocity, the component j

($j = 1, \dots, n$, where n is the sum of the number of joints of all arms) of $\boldsymbol{\zeta}$ is defined as

$$\zeta_j(t) = \begin{cases} 1 & \text{if } |\dot{\mathbf{q}}(t)[j]| < \delta_j \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

where $\dot{\mathbf{q}}(t)[j]$ is the j -th component of vector $\dot{\mathbf{q}}(t)$ and δ_j is a threshold. The fault criteria, which is shown at Figure 2, is employed to avoid false alarms due to misclassified individual patterns and it is defined as

$$\begin{cases} \text{fault } i = 1 & \text{if } \psi_i(t) = \max_{j=1}^q(\psi_j(t)) \text{ for } d \text{ samples} \\ \text{fault } i = 0 & \text{otherwise} \end{cases}$$

where q is the number of outputs of the RBFN, $\psi_i(t)$ is the output i of the RBFN at time t , and $i = 1, \dots, (q - 1)$ (the output q refers to the normal operation). For example, if the output 2 is higher than the other outputs during d consecutive samples, fault 2 is announced.

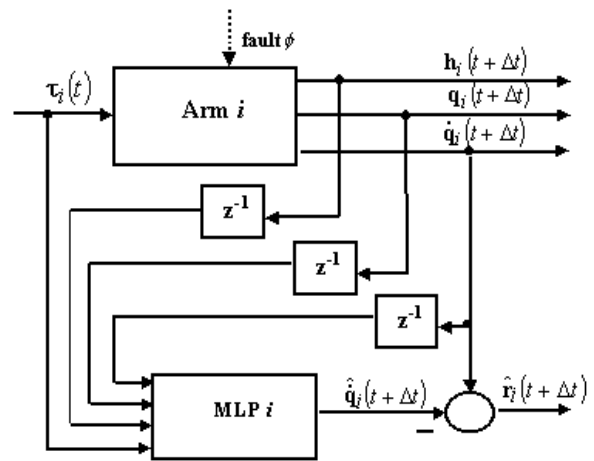


Figure 1: Residual generation.

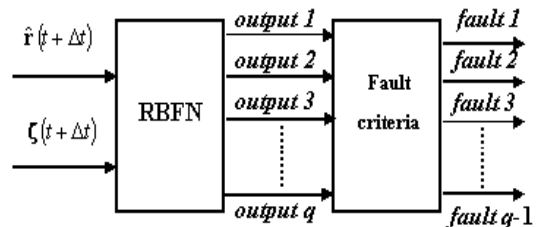


Figure 2: Residual analysis.

4 RESULTS

The FDI system was firstly tested in the simulation of two three-dof planar cooperative arms with passive joints manipulating an object with mass equal to 2.5kg in an x-y

plane. The gravity was parallel to the y-axis (the x-axis passes through the bases of the two arms). The parameters of the simulated system are presented in the Appendix. The controller proposed in (Wen and Kreutz-Delgado, 1992) was used to control the cooperative arms. The simulations were performed in the Cooperative Manipulators Control Environment (CMCE), which runs in Matlab and is still employed to control the actual cooperative system presented later. The CMCE allows one to change the parameters of the system (kind and time of faults, controller parameters, etc.) and to generate graphics with the variables of the robots, load, and faults. The main GUI of the CMCE is shown in Figure 3.

In the simulations, the sampling period adopted was 0.008s and measurement noise with normal distribution was added to joint positions, joint velocities, and end-effector forces.

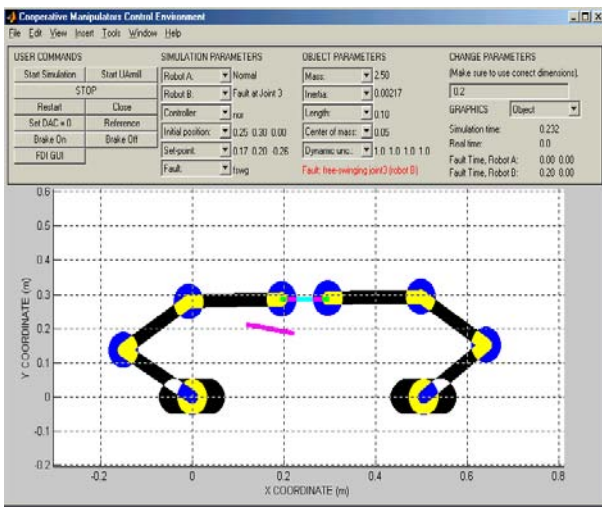


Figure 3: Cooperative Manipulators Control Environment.

Two MLPs were utilized: each one with 12 inputs, 27 neurons in the hidden layer, and 3 outputs. The Backpropagation Algorithm was used to train the MLP's and their weights were initialized by the Nguyen-Widrow-Russo Algorithm (Looney, 1997). The MLPs were trained with 7400 patterns obtained in the simulation of 100 trajectories. The RBFN had 12 inputs and 13 outputs (6 FSJFs, 6 LJFs, and normal operation) and it was trained with 2691 patterns. It was adopted $d = 3$ samples in the fault criteria. The parameters of the FDI system were $\gamma_{p1} = \gamma_{p2} = 0.05$, $\gamma_{v1} = \gamma_{v2} = 1.5$, and $\delta_j = 4 \times 10^{-3}$. Figure 4 shows the detection of a JPF in a trajectory with a fault occurring at $t=0.1s$. The norm $\|\hat{x}_{o1}(\theta_1) - \hat{x}_{o2}(\theta_2)\|$ (Eq. 10) used to detect JPFs is shown on the left (Figure 4), and the norms $\|\hat{x}_{o1}(\theta_1, \hat{q}_{1j}) - \hat{x}_{o2}(\theta_2)\|$ (Eq. 13) and $\|\hat{x}_{o1}(\theta_1) - \hat{x}_{o2}(\theta_2, \hat{q}_{2j})\|$ (Eq. 14) used to isolate JPFs is shown in the right for the joints $j = 1, \dots, n$.

Table 1: Results: simulation of two three-dof arms.

Set	Detected Faults	Isolated Faults	False Alarms	MTD(s)
1	958 (99.79%)	938 (97.71%)	0 (0%)	0.0165
2	960 (100.0%)	920 (95.83%)	0 (0%)	0.0180
3	899 (93.65%)	805 (83.85%)	0 (0%)	0.0185
4	926 (95.42%)	817 (85.10%)	0 (0%)	0.0195

Table 2: Results (faults): simulation of two three-dof arms.

Set	Fault	Detected Faults	Isolated Faults
1	FSJF	240 (100.0%)	236 (98.33%)
1	LJF	238 (99.17%)	228 (95.00%)
1	JPF	240 (100.0%)	235 (97.92%)
1	JVF	240 (100.0%)	238 (99.17%)
2	FSJF	240 (100.0%)	238 (99.17%)
2	LJF	240 (100.0%)	209 (87.08%)
2	JPF	240 (100.0%)	238 (99.17%)
2	JVF	240 (100.0%)	236 (98.33%)
3	JPF	240 (100.0%)	238 (99.17%)
3	JVF	180 (75.00%)	098 (40.83%)
4	JPF	240 (100.0%)	236 (98.33%)
4	JVF	196 (81.67%)	135 (56.25%)

The FDI system was tested considering four trajectory sets, each one with 960 trajectories with faults occurring in different joints and 40 without faults. The first and the second sets had the same desired trajectories, but with faults starting at 0.15 s. and 0.3 s. respectively. The four faults previously presented were simulated. In JPFs and JVFs for sets 1 and 2, the correct sensor reading were changed by random numbers. The desired trajectories and initial time of fault of sets 3 and 4 were the same of sets 1 and 2 respectively, but the correct sensor measurements were changed by zeros in JPFs and JVFs. The results of the FDI system are summarized in table 1. The second and third columns present the number of detected faults and the number of correctly isolated faults respectively. The fourth column shows the number of false alarms in fault-free trajectories. The last column presents the Mean-Time-to-Detection (MTD): the mean time that the FDI system takes to isolate a fault after its occurrence.

The results of the FDI for each fault are summarized in table 2. The results of FSJFs and LJFs for the sets 3 and 4 are not shown (the desired trajectories in sets 3 and 4 were the same of sets 1 and 2). The number of correctly isolated JVFs was smaller in sets 3 and 4 because this fault was mistaken with LJFs. This occurred because, as the joint velocities measurements were changed by zeros in sets 3 and 4, JVFs generally did not present consequences in the control of the load for this controller (the load could be controlled even with the joint velocity measurements equal to zeros). This explain the small number of isolated JVFs in sets 3 and 4.

The following step was the simulation of two Puma560 arms

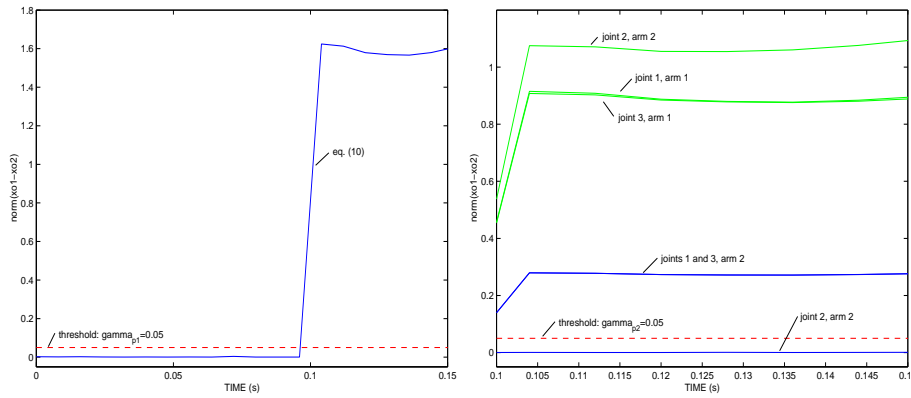


Figure 4: FDI in a trajectory of the simulated system (three-dof planar cooperative arms) with JPF in joint 2 of arm 2 occurring at $t=0.1s$. Left: $\|\hat{\mathbf{x}}_{o1}(\theta_1) - \hat{\mathbf{x}}_{o2}(\theta_2)\|$ (Eq. 10); Right: $\|\hat{\mathbf{x}}_{o1}(\theta_1, \hat{q}_{1j}) - \hat{\mathbf{x}}_{o2}(\theta_2)\|$ (Eq. 13) and $\|\hat{\mathbf{x}}_{o1}(\theta_1) - \hat{\mathbf{x}}_{o2}(\theta_2, \hat{q}_{2j})\|$ (Eq. 14) for the joints $j = 1, \dots, n$. The dashed lines show the threshold γ_{p2} .

Table 3: Results of the FDI system: simulation (Puma560 arms).

Set	Detected Faults	Isolated Faults	False Alarms	MTD(s)
1	718 (99.72%)	689 (95.69%)	0 (0.0%)	0.117
2	715 (99.31%)	656 (91.11%)	0 (0.0%)	0.118

manipulating a cylinder with mass equal to 2.5kg and 0.3m of length in a 3-dimensional space. The Robotics Toolbox for Matlab (Corke, 1996) was used to simulate the terms of Eq. (1) and the friction torques. The sampling period was 0.018s and measurement noise with normal distribution was added to joint positions, joint velocities, and end-effector forces.

Two MLPs were utilized: each one with 24 inputs, 49 neurons in the hidden layer, and 6 outputs. The MLPs were trained with 6804 patterns obtained in the simulation of 50 trajectories. The RBFN had 24 inputs and 25 outputs (12 FSJFs, 12 LJFs, and normal operation) and it was trained with 5291 patterns. The parameters of the FDI system were $d = 3$ samples, $\gamma_{p1} = \gamma_{p2} = 0.01$, $\gamma_{v1} = \gamma_{v2} = 0.8$, and $\delta_j = 4 \times 10^{-3}$.

The FDI system was tested considering two trajectory sets, each of them with 720 trajectories with faults and 15 without faults. The first and the second sets have the same desired trajectories but with the faults starting at 0.3s and 1.3s respectively. The four faults previously presented were simulated. In JPFs and JVF, the correct sensor measurements were changed by random numbers. The results of the FDI system are summarized in table 3.

Finally, the FDI system was applied in an actual cooperative

system with two arms UARMII (Figure 5). Each UARMII is a 3-joint, planar manipulator that floats on a thin air film on an “air table”. The two arms are equal and the longitudinal axis of each joint is parallel to the gravity force. 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 CMCE, which was used to simulate the cooperative system with 3-dof arms, is used to control and to monitor the actual system. The robot parameters are the same of the simulated system and the sampling period was chosen as 0.05s. The joint velocities are obtained by encoder measurements (the adaptive filter presented in (Wijngaard, 1996) is used), and force sensors are not used (the end-effector forces are estimated using the kinematic and dynamic models).

Two MLPs were utilized to reproduce the model of the actual robots: each one with 12 inputs, 37 neurons in the hidden layer, and 3 outputs. The MLPs were trained with 3250 patterns obtained in the simulation of 50 trajectories. The RBFN had 12 inputs and 13 outputs (6 FSJFs, 6 LJFs, and normal operation) and it was trained with 2506 patterns. The parameters of the FDI system were $d = 4$ samples, $\gamma_{p1} = \gamma_{p2} = 0.05$, $\gamma_{v1} = \gamma_{v2} = 1.5$, and $\delta_j = 4 \times 10^{-3}$. The FDI system was tested considering three trajectory sets, each of them with 360 trajectories with faults and 15 without faults. The second and third sets had the same desired trajectories but an object of mass equal to 0.025 kg was manipulated in the second set and an object of 0.45 kg was manipulated in the third set. The first set had different desired trajectories, and the mass of load was equal to 0.45 kg. The results of the FDI system considering the four faults described here

Table 4: Results: actual system.

Set	Det. Faults	Is. Faults	False Al.	MTD(s)
1	337 (93.6%)	260 (72.2%)	1 (6.7%)	0.469
2	333 (92.5%)	247 (68.6%)	0 (0.0%)	0.419
3	325 (90.3%)	268 (74.3%)	0 (0.0%)	0.458

occurring in each joint are summarized in Table 4. Figures 6, 7, and 8 show respectively the torques of arm 1, the residuals, and the outputs of the RBFN in a trajectory with an FSJF.

The number of correctly isolated faults was smaller in the actual 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 does not present significant effects in the system. This occurred, for example, when it was not necessary to apply high torques at the faulty joint during the given trajectory.



Figure 5: Actual system: arm 1 (left); arm 2 (right).

5 CONCLUSIONS

This work presents an FDI system for cooperative manipulators. Four faults were considered: FSJFs, LJFs, JPFs, and JVs. The first two are detected by ANNs: MLPs to reproduce the dynamics of the arms and an RBFN to classify the residual vector. The remaining faults are detected using the kinematic constraints of the system. Tests of the FDI system applied to simulated cooperative arms and to actual robots were presented. As the number of correctly detected faults is greater than the number of correctly isolated faults, additional tests can be done after the detection of a fault to confirm the isolated fault. If a LJF is isolated in the joint i for example, a simple test to confirm this fault is to apply a torque in this joint and to check if the joint i moves after applying the brakes in all joints. Tests to confirm the isolation of the other faults can be done in similar ways.

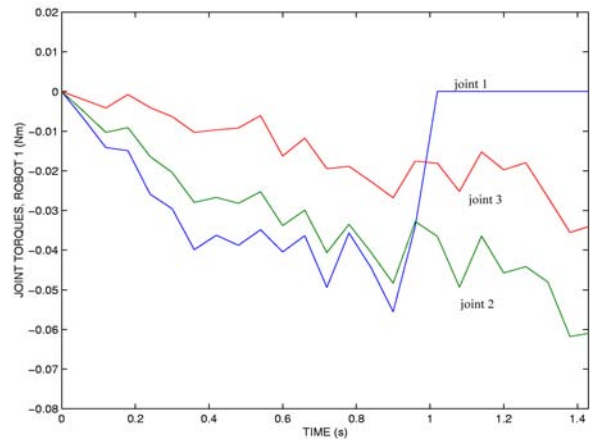


Figure 6: Joint torques of arm 1 in a trajectory of the actual system with FSJF in joint 1 (arm 1) occurring at $t=1s$.

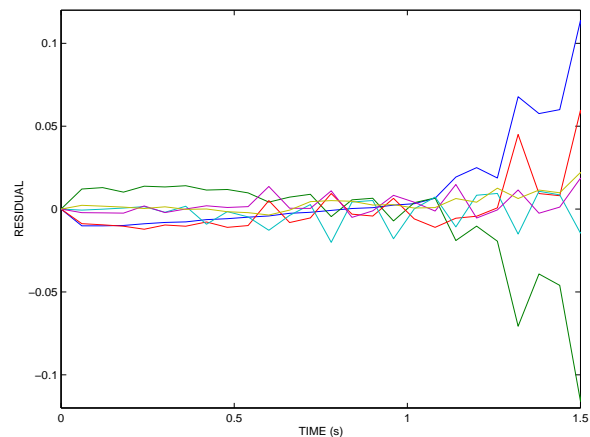


Figure 7: Residuals in a trajectory of the actual system with FSJF in joint 1 (arm 1) occurring at $t=1s$.

Acknowledgments

The authors would like to thank Marcel Bergerman for his contributions. This work was supported by FAPESP under grants 98/15732-5 and 99/10031-1.

REFERENCES

- Bonitz, R. and Hsia, T. (1996). Internal force-based impedance control for cooperating manipulators, *IEEE Trans. on Rob. and Aut.* **12**: 78–89.
- Caccavale, F. (1997). Task-space regulation of cooperative manipulators, *Prisma lab. technical report 97-04*, Università degli Studi di Napoli Federico II, Napoli, Italy.

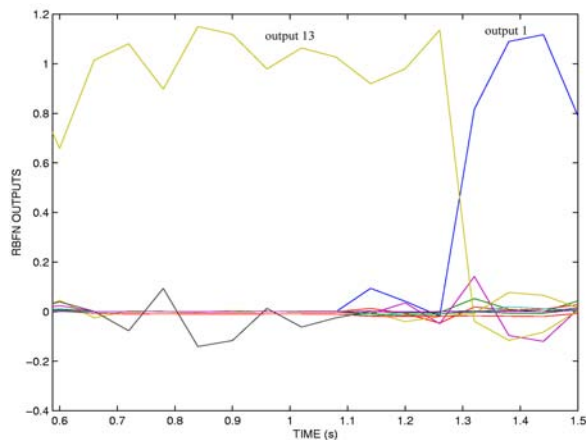


Figure 8: Outputs of the RBFN in the same trajectory shown in Fig. 6 (Output 1: FSJF in joint 1 of arm 1; Output 13: normal operation).

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

Chen, J. and Patton, R. J. (1999). *Robust model-based fault diagnosis for dynamic systems*, Kluwer Academic Publishers.

Corke, P. (1996). A robotics toolbox for MATLAB, *IEEE Robotics and Automation Magazine* **3**(1): 24–32.

English, J. D. and Maciejewski, A. A. (1998). Fault tolerance for kinematically redundant manipulators: anticipating free-swinging joint failures, *IEEE Trans. on Rob. and Aut.* **14**(4): 566–575.

Gertler, J. (1997). A cautious look at robustness in residual generation, *Proc. of the 3rd IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Vol. 1, pp. 133–139.

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

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

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

Isermann, R. and Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes, *Control Engineering Practice* **5**(5): 709–719.

Koivo, A. J. and Unseren, M. A. (1991). Reduced order model and decoupled control architecture for two manipulators holding a rigid object, *Journal of Dyn. Syst., Measurement, and Control (Trans. of the ASME)* **113**: 646–654.

Köppen-Selinger, B. and Frank, P. M. (1996). Neural networks in model-based fault diagnosis, *Proc. of 13th IFAC World Congress*, pp. 67–72.

Korbicz, J. (1997). Neural networks and their application in fault detection and diagnosis, *Proc. of the 3rd IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Vol. 1, pp. 377–382.

Looney, C. G. (1997). *Pattern recognition using neural networks*, Oxford University Press.

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

Mangoubi, R. S. (1998). *Robust estimation and failure detection*, Springer-Verlag.

Marcu, T. and Mirea, L. (1997). Robust detection and isolation of process fault using neural networks, *IEEE Control Systems* **17**(5): 72–79.

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

Nahon, M. and Angeles, J. (1992). Minimization of power losses in cooperating manipulators, *Journal of Dyn. Syst., Measurement, and Control (Trans. of the ASME)* **114**: 213–219.

Nakamura, Y. (1991). *Advanced robotics: redundancy and optimization*, 1st edn, Addison-Wesley Publishing Company, Inc., New York.

Naughton, J. M., Chen, Y. C. and Jiang, J. (1996). A neural network application to fault diagnosis for robotic manipulator, *Proc. of IEEE Int. Conf. on Control App.*, Vol. 1, pp. 988–1003.

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

- Schneider, H. and Frank, P. M. (1996). Observer-based supervision and fault-detection in robots using nonlinear and fuzzy logic residual evaluation, *IEEE Trans. on Control Syst. Tech.* **4**(3): 274–282.
- Sciavicco, L. and Siciliano, B. (1996). *Modeling and control of robot manipulators*, McGraw-Hill International Editions.
- Sorsa, T. and Koivo, H. N. (1993). Application of artificial neural networks in process fault diagnosis, *Automatica* **29**(4): 843–849.
- Terra, M. H. and 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. and Bergerman, M. (2001). Fault detection and isolation in cooperative manipulators via artificial neural networks, *Proc. of IEEE Conf. on Control App.*, pp. 988–1003.
- Tinós, R., Terra, M. H. and Ishihara, J. Y. (2006). Motion and force control of cooperative robotic manipulators with passive joints, *IEEE Trans. on Control Systems Technology* **14**(4): 725–734.
- Uchiyama, M. (1998). Multirobots and cooperative systems, in B. Siciliano and K. P. Valavanis (eds), *Control problems in robotics and automation*, Springer-Verlag, London, U. K.
- Vemuri, A. T. and 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. and Walker, I. D. (1994). Robotic fault detection and fault tolerance: a survey, *Reliability Eng. and Syst. Safety* **46**: 139–158.
- Visinsky, M. L., Cavallaro, J. R. and Walker, I. D. (1995). A dynamic fault tolerance framework for remote robots, *IEEE Trans. on Rob. and Aut.* **11**(4): 477–490.
- Vukobratovic, M. and Tuneski, A. (1998). Mathematical model of multiple manipulators: cooperative compliant manipulation on dynamical environments, *Mechanism and Machine Theory* **33**: 1211–1239.
- Wen, T. and 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.

Table 5: Parameters of the simulated system (3-dof arms).

Mass of links 1 and 2	0.85 kg
Mass of link 3	0.625 kg
Link length	0.203 m
Load length (between contact points)	0.1 m
Load mass	2.5 kg
Load moment of inertia	0.0022 kg m ²
Gravity	9.8 m/s ²

APPENDIX