

## FAULT DETECTION AND ISOLATION FOR MULTIPLE ROBOTIC MANIPULATORS

Renato Tinós\* Marco H. Terra\*

\* *Electrical Eng. Department, EESC, University of São Paulo,  
CP 359, São Carlos, SP, 13560-970, Brazil*  
{tinós} terra@sel.eesc.sc.usp.br

**Abstract:** The problem of fault detection and isolation (FDI) in cooperative manipulators is addressed here. Four faults are considered: free-swinging joint faults, locked joint faults, incorrect measured joint position, and incorrect 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 real joint velocities in order to generate a residual vector that is then classified by an RBF network. The sensor faults are isolated based on the kinematic constraints imposed on the system. Simulations and a real application are presented indicating the effectiveness of the FDI system.

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

### 1. INTRODUCTION

Robots have been used to execute tasks in medicine, outer space, deep sea, and other unstructured or hazardous environments. Furthermore, robots will be usual inside homes as a household or entertainment item (Dhillon and Fashandi, 1997). In these environments, robots are used to avoid the exposition of human beings to danger or because of the reliability of robots in executing repetitive tasks. However, faults can put in risk the robot, its task, and its environment.

Faults in robots have been usual due to the complexity of such systems. There are several sources of faults in robots, such as electrical, mechanical, hydraulic, of software, etc. (Visinsky *et al.*, 1994). In fact, some researches indicate that the mean-time-to-failure in industrial robots are only between 500 and 2500 hours (Dhillon and Fashandi, 1997). If this number is small in structured environments, it is probably smaller in unstructured and hazardous environments due to external factors as extreme temperatures, obsta-

cles, radiation, etc. So, there are good reasons to research fault detection and isolation (FDI) systems for robots.

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 human case, 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). For example, cooperative robots can be used in the manipulation of heavy, large or flexible loads, assembly of structures, and manipulation of objects that can slide from only one robot. 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 causing damage to the load and instability in the system.

In this work, an FDI scheme for cooperative systems is presented. The Section 2 describes the kinematics and dynamics of cooperative manipulators. The Section 3 describes the FDI system. Four faults are considered: free-swinging joint faults (FSJF), locked joint faults (LJF), incorrect measured joint position faults (JPF), and incorrect measured joint velocity faults (JVF). FSJF and LJF are detected by artificial neural networks (ANN): Multilayer Perceptrons (MLP's) are used to reproduce the dynamics of the arms, and an RBF network is utilized to classify the residual vector. JPF and JVF are detected using the kinematic constraints imposed on the cooperative system. The Section 4 presents the results of the FDI system in simulations and in a real application. The conclusions are presented in Section 5.

## 2. COOPERATIVE MANIPULATORS

Considering that  $m$  robotic arms are rigidly connected to a solid object, the dynamics of arm  $i$  in the cooperative system is given by

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

where  $\mathbf{q}_i$  is the vector of joint angles of arm  $i$ ,  $i = 1, \dots, m$ ,  $\tau_i$  is the vector of the torques at joints of arm  $i$ ,  $\mathbf{M}_i$  is the inertia matrix,  $\mathbf{b}_i$  is the vector of centrifugal, Coriolis, and gravitational terms,  $\mathbf{J}_i$  is the Jacobian (from joint velocity to end-effector velocity) of arm  $i$ , and  $\mathbf{h}_i$  is the force vector at the end-effector of arm  $i$ . The dynamics of all arms can be written as

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}[\boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^T \mathbf{h} - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})] \quad (2)$$

where  $\mathbf{q} = [\mathbf{q}_1^T \mathbf{q}_2^T \dots \mathbf{q}_m^T]^T$ ,  $\boldsymbol{\tau} = [\tau_1^T \tau_2^T \dots \tau_m^T]^T$ ,  $\mathbf{h} = [\mathbf{h}_1^T \mathbf{h}_2^T \dots \mathbf{h}_m^T]^T$ ,  $\mathbf{M}$  is formed by the inertia matrices of the arms,  $\mathbf{b}$  is formed by the centrifugal, Coriolis, and gravitational terms of the arms, and  $\mathbf{J}$  is formed by the terms  $\mathbf{J}_i$  for  $i = 1, \dots, m$ .

The dynamics of the load is given by

$$\alpha_o = \mathbf{M}_o^{-1}[-\mathbf{J}_o(\mathbf{x}_o)^T \mathbf{h} - \mathbf{b}_o(\mathbf{x}_o, \nu_o)] \quad (3)$$

where  $\mathbf{x}_o$  is a  $k$ -dimensional vector of the load position and orientation at the center of gravity (CG),  $\alpha_o$  is the spatial acceleration of load CG,  $\nu_o$  is the spatial velocity of load CG,  $\mathbf{b}_o$  is the vector

of centrifugal, Coriolis, and gravitational terms,  $\mathbf{M}_o$  is the 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  $\mathbf{J}_{oi}$  converts velocities at the CG into velocities at the end-effector of arm  $i$ . As it is possible to compute the position and orientation of the object using the positions of the joints of any arm, the following kinematic constraint appears

$$\mathbf{x}_o = \varphi_1(\mathbf{q}_1) = \varphi_2(\mathbf{q}_2) = \dots = \varphi_m(\mathbf{q}_m) \quad (4)$$

where  $\varphi_i(\mathbf{q}_i)$  is a vector containing the position and orientation of the load computed via the joint positions of arm  $i$ . The following velocity constraint is also present

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

where  $\mathbf{D}_i$  is the Jacobian relating joint velocities of arm  $i$  and load velocities.

## 3. FDI SYSTEM

The FDI system is proposed take into account four kinds of faults: FSJF, where an actuation lost occurs in one arm's joint; LJF, where one arm's joint is locked; JPF, where the readings of the joint position are not correct, and JVF, where the readings of the joint velocity are not correct. A three step FDI system is applied here in each sample time. First, JPF are detected by analyzing the position constraints (eq. 4). Then, JVF are detected by analyzing the velocity constraints (eq. 5). The last step is the detection of FSJF and LJF via two ANN. By simplicity, the occurrence of only one fault each time is considered.

### 3.1 Incorrect Measured Joint Position 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 forward displacement problem (knowing the joint displacements, identify the end-effector pose) is not trivial in parallel manipulators because they have one or more unsensed joints. The forward displacement problem is easier in cooperative manipulators with all joints sensed, and it can be used to detect JPF.

As  $\mathbf{x}_o$  can be calculated using the joint positions of any arm (eq. 4), if  $m > 2$ , it is possible to identify the arm  $f$  with the wrong joint position reading. The arm with the wrong reading gives a wrong estimative of  $\mathbf{x}_o$  that is different from the estimates of the other  $m - 1$  arms. A JPF is detected at arm  $f$  if

$$\begin{aligned} \|\hat{\mathbf{x}}_{o_f}(\theta_f) - \hat{\mathbf{x}}_{o_i}(\theta_i)\| &> \gamma_x \\ \text{for all } i &= 1, \dots, m \text{ and } i \neq f \end{aligned} \quad (6)$$

where  $\hat{\mathbf{x}}_{o_i}$  is the estimative of  $\mathbf{x}_o$  using the measured positions of the joints ( $\theta_i$ ) at arm  $i$ ,  $\|\mathbf{a}\|$  is the Euclidean norm of the vector  $\mathbf{a}$ , and the threshold  $\gamma_x$  is a small number used to avoid that false alarms appear due to the presence of noise in the joint readings. It is interesting to choose  $\gamma_x$  as a function of the variance of the noise in the joint position readings. The next step is to estimate the position of each joint  $j = 1, \dots, n_f$  at arm  $f$

$$\hat{q}_{f_j} = v_j(\theta_f, \hat{\mathbf{x}}_o) \quad (7)$$

where  $\hat{q}_{f_j}$  is the estimative of the position of the joint  $j$  at arm  $f$ ,  $v_j$  is the kinematic function used to estimate the position of the joint  $j$ , and

$$\hat{\mathbf{x}}_o = \frac{1}{m-1} \sum_{i=1, i \neq f}^m \hat{\mathbf{x}}_{o_i}(\theta_i).$$

Calculating again the estimative of the vector  $\mathbf{x}_o$  for arm  $f$  for each new estimative  $\hat{q}_{f_j}$ , the fault at joint  $j$  of arm  $f$  is detected when

$$\|\hat{\mathbf{x}}_o - \hat{\mathbf{x}}_{o_f}(\theta_f, \hat{q}_{f_j})\| < \gamma_p \quad (8)$$

where  $\hat{\mathbf{x}}_{o_f}(\theta_f, \hat{q}_{f_j})$  is the vector of the positions and orientations of the load estimated for arm  $f$  substituting the measured position of joint  $j$  by its estimative  $\hat{q}_{f_j}$  and using the measured positions of the other joints. The threshold  $\gamma_p$  is a small number used to avoid that faults be hidden due to the presence of noise in the joint readings. It is interesting to choose  $\gamma_p$  as a function of the variance of the noise in the joint position readings. The procedure to detect and isolate JPF when  $m > 2$  can be summarized as: compare the estimative of  $\mathbf{x}_o$  for all arms (eq. 6); if all values are close, a JPF is not announced, otherwise, calculate for all joints of the faulty arm the estimative of the joint positions (eq. 7) and test eq. (8) for all joints; if the test is satisfied for joint  $j$ , announce a JPF at this joint.

If  $m = 2$ , the arm with the fault can not be identified just looking at the estimates of  $\mathbf{x}_o$  for each arm. In this case, the joint positions estimation (eq. 6) should be done for the two arms using, instead of the the value of  $\hat{\mathbf{x}}_o$ , the estimative obtained using the joint positions of the other arm. The same should be done at eq. (8), which can be used to detect the JPF.

### 3.2 Incorrect Measured Joint Velocity Faults

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

of JPF. The estimated velocities of the joints can be calculated using eq. (5). Considering the occurrence of only one fault, the JVF at joint  $j$  of arm  $f$  is detected if  $m > 2$  when

$$\|\hat{\nu}_o - \hat{\nu}_{o_f}(\hat{\theta}_f, \hat{q}_{f_j})\| < \gamma_v \quad (9)$$

where  $\hat{\nu}_{o_f}(\hat{\theta}_f, \hat{q}_{f_j})$  is the velocity of the load estimated for arm  $f$  substituting the measured velocity of joint  $j$  by its estimative  $\hat{q}_{f_j}$  and using the measured velocities ( $\hat{\theta}_f$ ) of the other joints,  $\hat{\nu}_o$  is the estimative of the load velocities using the measured joint velocities of the other arms, and the threshold  $\gamma_v$  is a small number that can be chosen as a function of the variance of the noise in the readings. When  $m = 2$ ,  $\nu_o$  should be substituted by the estimated velocity obtained using the joint velocities of the other arm.

### 3.3 Free-Swinging Joint and Locked Joint Faults

As FSJF and LJF present effects in the dynamics of the cooperative system, the residual generation scheme can be used to detect these faults. Generally, FDI systems for individual arms utilize the mathematical model of the arm in the residual generation scheme (Visinsky *et al.*, 1994). The residual vector is generated comparing the measured states of the arm with the estimative of them. However, modelling errors can appear generating false alarms or hiding the fault effects. Robust technique (Dixon *et al.*, 2000), fuzzy logic (Schneider and Frank, 1996), and ANN (Vemuri and Polycarpou, 1997) (Terra and Tinós, 2001) have been used to avoid these problems.

To the best of the author's 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 compared with the 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's dynamics. Furthermore, the mapping of the MLP in (Tinós *et al.*, 2001) is dependent on the load parameters. If the system manipulates another object, the ANN have to be trained again.

Here, the dynamics of each arm is mapped by a different MLP. This scheme is interesting because the mapping is not dependent on the load parameters. A Multi-Input Single-Output scheme could be used to reproduce the dynamics of the system instead of the Multi-Input Multi-Output scheme used here. The second approach is adopted because of its smaller time processing.

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 sample period  $\Delta t$  is sufficiently small, the dynamics of the fault-free robot  $i$  (eq. 1) is

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \quad (10)$$

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

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}_\phi(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \quad (11)$$

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

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{f}_\phi(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \\ &\quad - \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)). \end{aligned} \quad (12)$$

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

$$\begin{aligned} \hat{\mathbf{q}}_i(t + \Delta t) &= \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \\ &\quad + \mathbf{e}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \end{aligned} \quad (13)$$

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

$$\hat{\mathbf{r}}_i(t + \Delta t) = \dot{\mathbf{q}}_i(t + \Delta t) - \hat{\mathbf{q}}_i(t + \Delta t). \quad (14)$$

By eq. (10-14), 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 with the fault function vector in order to allow the detection of the fault. The residual vector from all arms ( $\hat{\mathbf{r}}$ ) are then classified by an RBFN trained by the Kohonen's Self Organizing Map (Terra and Tinós, 2001). As the residual vector of FSJF and LJF occurring at the same joint can occupy the same region in the input space of the RBFN, an auxiliary input vector  $\zeta$  that gives information about the velocity of the joints is used (figure 2). The use of  $\zeta$  is motivated by the fact that the velocity of the faulty joint is zero in LJF. In virtue of noise in the measurement of the joint velocity, the  $i$ -th ( $i = 1, \dots, n$  and  $n$  is the sum of the

number of joints of all arms) component of  $\zeta$  is defined as

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

where the threshold  $\delta_i$  is a small number. Here we choose a value of  $\delta_i$  proportional to the variance of the joint velocity measurement noise. 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

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

where  $\psi_i$  is the output  $i = 1, \dots, q-1$  of the RBFN (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.

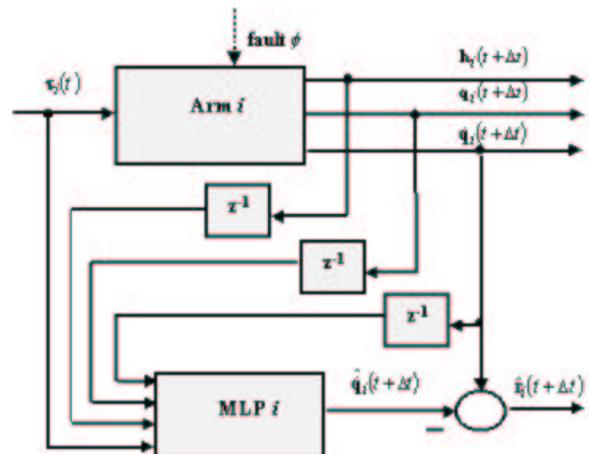


Fig. 1. Residual generation.

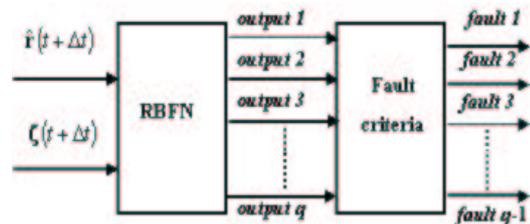


Fig. 2. Residual analysis.

## 4. RESULTS

First, the FDI system is applied in a simulation of two three-dof planar cooperative arms manipulating an object in a x-y plane. The arms are equal and the gravity force is orthogonal to the joint axes. The controller proposed in (Wen and Kreutz-Delgado, 1992) is used to control the cooperative arms. The sample period is 0.008s and measurement noise is added to the joint positions and velocities, and to the end-effector forces.

Two MLP's are utilized: each one has 12 inputs, 27 neurons at the hidden layer, and 3 outputs. The MLP's are trained with 7400 patterns obtained in the simulation of 100 trajectories. The RBFN has 12 inputs and 13 outputs (6 FSJF, 6 LJF, and normal operation) and it is trained with 2691 patterns. The fault criteria uses  $d = 3$  samples. The parameters of the FDI system are  $\gamma_p = 0.05$ ,  $\gamma_v = 1.5$ , and  $\delta_i = 5\sigma_i$ , where  $\sigma_i$  is the standard deviation of the position measurement at joint  $i$ . The FDI system are tested considering eight trajectory sets. Each one of the sets 1-4 has 480 trajectories with faults occurring in different joints and 20 without faults. The first and the second sets have the same initial and final points and faults starting at 0.15 s. and 0.3 s. respectively. The same occurs for the third and fourth sets. The four faults previously presented are simulated. In JPF and JVF for sets 1-4, the correct sensor readings are changed by random numbers. In JPF and JVF for sets 5-8, the correct sensor readings are changed by zeros. The parameters of the sets 5-8 are equals to the sets 1-4, but with only 240 trajectories (for JPF and JVF). The results of the FDI system can be viewed at 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 the 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.

Table 1. *Results: simulation.*

Set	Det. Faults	Is. Faults	False Al.	MTD(s)
1	479 (99.8%)	469 (97.7%)	0 (0%)	0.016
2	480 (100%)	461 (96.0%)	0 (0%)	0.017
3	479 (99.8%)	469 (97.7%)	0 (0%)	0.017
4	480 (100%)	459 (95.6%)	0 (0%)	0.019
5	449 (93.5%)	408 (85.0%)	0 (0%)	0.024
6	457 (95.2%)	408 (85.0%)	0 (0%)	0.018
7	450 (93.7%)	397 (82.7%)	0 (0%)	0.013
8	459 (95.6%)	409 (85.2%)	0 (0%)	0.021

The results for each fault are in table 2. The number of correctly isolated JVF was smaller in sets 5-8 because this fault was confused with LJF. This occurred because the joint velocities readings are changed by zeros. However, in these cases, JVF generally do not present consequences in the control of the trajectory (the load could be controlled even with the joint velocity readings changed to zero). This explain the small number of detected JVF in sets 5-8 and it is a result dependent of the choice of the controller.

The next step is the application of the FDI system in a real cooperative system with two arms UARMII (figure 3). Each UARMII is a 3-joint planar manipulator that floats on a thin air film on an "air table". The two arms are equals and the axis of each joint is parallel to the gravity force. The cooperative system is controlled by a PC running Matlab. Each joint of the UARMII

Table 2. *Results (faults): simulation.*

Set	Fault	Detected Faults	Isolated Faults
1	FSJF	120 (100.0%)	118 (98.33%)
1	LJF	119 (99.17%)	114 (95.00%)
1	JPF	120 (100.0%)	118 (98.33%)
1	JVF	120 (100.0%)	118 (98.33%)
2	FSJF	120 (100.0%)	120 (100.0%)
2	LJF	120 (100.0%)	105 (87.50%)
2	JPF	120 (100.0%)	118 (98.33%)
2	JVF	120 (100.0%)	118 (98.33%)
3	FSJF	120 (100.0%)	118 (98.33%)
3	LJF	119 (99.17%)	114 (95.00%)
3	JPF	120 (100.0%)	117 (97.50%)
3	JVF	120 (100.0%)	120 (100.0%)
4	FSJF	120 (100.0%)	118 (98.33%)
4	LJF	120 (100.0%)	104 (86.67%)
4	JPF	120 (100.0%)	120 (100.0%)
4	JVF	120 (100.0%)	118 (98.33%)
5	JPF	120 (100.0%)	120 (100.0%)
5	JVF	89 (74.17%)	52 (43.33%)
6	JPF	120 (100.0%)	118 (98.33%)
6	JVF	97 (80.83%)	65 (54.17%)
7	JPF	120 (100.0%)	118 (98.33%)
7	JVF	91 (75.83%)	46 (38.33%)
8	JPF	120 (100.0%)	118 (98.33%)
8	JVF	99 (82.50%)	70 (58.33%)

contains a brushless DC direct-drive motor, encoder, and pneumatic brake. Each joint can be actively driven by its motor, locked by its brake, or allowed to move freely with nearly zero torque. The controller proposed in (Wen and Kreutz-Delgado, 1992) is used to control the arms (the sample period is 0.06s). It is important to observe that this system is difficult to be correctly modelled because the flatness of the "air table" is irregular (the gravitational torques change with the position of the joints on the table). Other problem is that the joint velocities are obtained by differentiating the encoder readings, and force sensors are not used (the end-effector forces are estimated using the kinematic and dynamical models).

Two MLP's are utilized: each one has 12 inputs, 37 neurons at the hidden layer, and 3 outputs. The MLP's are trained with 3250 patterns obtained in the simulation of 50 trajectories. The RBFN has 12 inputs and 13 outputs (6 FSJF, 6 LJF, and normal operation) and is trained with 2506 patterns. The fault criteria uses  $d = 4$  samples. The FDI system is tested considering three trajectory sets, each of them with 360 trajectories with faults and 15 without faults. The second and third sets have the same initial and final position but an object of mass equal to 0.025 kg is manipulated in the second set and an object of 0.45 kg in the third set. The first set has different initial and final positions, and mass of load equal to 0.45 kg. The results of the FDI system considering the four faults described here occurring in each joint can be viewed at table 3. The figures 4 and 5 show the torques of arm 1 and the outputs of the RBFN in a trajectory with an FSJF.

The number of correctly isolated faults was smaller in the real system mainly because FSJF are sometimes confused with LJF. This occurs because, as there are small gravitational torques at the joints of the real system, sometimes the ve-

locities of the faulty joints are small. However, in these cases, even with FSJF, the load converges to the desired position and the fault does not present significant effects in the system. This can occur, for example, if it is not necessary to apply high torques at the faulty joint in a given trajectory.

Table 3. *Results: real 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



Fig. 3. *Real system.*

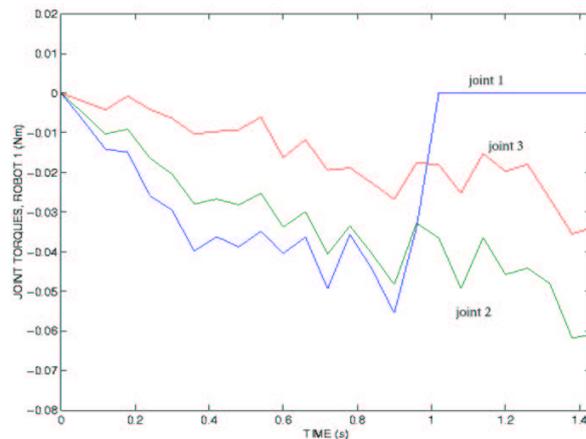


Fig. 4. *Joint torques at arm 1 in a trajectory of the real system with FSJF at joint 1 (arm 1) occurring at  $t=1s$ .*

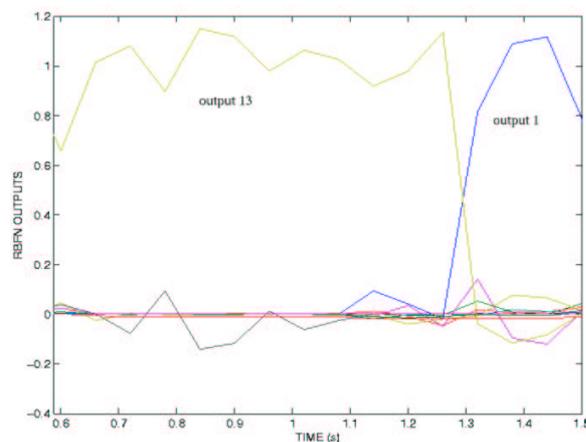


Fig. 5. *Outputs of the RBFN in the same trajectory showed at fig. 4 (Output 1: FSJF at joint 1 of arm 1; Output 13: normal operation).*

## 5. CONCLUSIONS

This work presents a FDI system for cooperative manipulators. Four faults were considered: FSJF, LJF, JPF, and JVF. The first two are detected by ANN: MLP's to reproduce the dynamics of the arms and an RBFN to classify the residual vector. The other faults are detected using the kinematic constraints of the system.

Acknowledgments: This work is supported by FAPESP (grants 98/15732-5 and 99/10031-1).

## 6. REFERENCES

- Dhillon, B. S. and A. R. M. Fashandi (1997). Robotic systems probabilistic analysis. *Microelectronics and Reliability* **37**(2), 211–224.
- Dixon, W. E., I. D. Walker, D. M. Dawson and J. P. Hartranft (2000). Fault detection for robot manipulators with parametric uncertainty: a prediction-error based approach. *IEEE Trans. on Rob. and Aut.* **16**(6), 689–699.
- Nakamura, Y. (1991). *Advanced robotics: redundancy and optimization*. 1st ed.. Addison-Wesley Publishing Company, Inc.. New York.
- Notash, L. (2000). Joint sensor fault detection for fault tolerant parallel manipulators. *Journal of Robotic Systems* **17**(3), 149–157.
- Schneider, H. and P. M. Frank (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.
- Terra, M. H. and R. Tinós (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., M. H. Terra and M. Bergerman (2001). Fault detection and isolation in cooperative manipulators via artificial neural networks. In: *Proc. of IEEE Conf. on Control App.* pp. 988–1003.
- Vemuri, A. T. and M. M. Polycarpou (1997). Neural-network-based robust fault diagnosis in robotic systems. *IEEE Trans. on Neural Networks* **8**(6), 1410–1420.
- Visinsky, M. L., J. R. Cavallaro and I. D. Walker (1994). Robotic fault detection and fault tolerance: a survey. *Reliability Eng. and Syst. Safety* **46**, 139–158.
- Vukobratovic, M. and A. Tuneski (1998). Mathematical model of multiple manipulators: cooperative compliant manipulation on dynamical environments. *Mechanism and Machine Theory* **33**, 1211–1239.
- Wen, T. and K. Kreutz-Delgado (1992). Motion and force control for multiple robotics manipulators. *Automatica* **28**(4), 729–743.