ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 513-518

Free-Swinging and Locked Joint Fault Detection and Isolation in Cooperative Manipulators

Renato Tinós and Marco Henrique Terra

Electrical Eng. Department, EESC, University of São Paulo
C.P.359, São Carlos, SP, 13560-970, Brazil
tinos@sel.eesc.sc.usp.br , terra@sel.eesc.sc.usp.br

Abstract. The problem of fault detection and isolation (FDI) in cooperative manipulators is addressed. Free-swinging and locked joint faults are detected and isolated by an FDI system based on 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. Simulations and a real application are presented indicating the effectiveness of the FDI system.

1 Introduction

Robots have increasingly been used in unstructured and/or hazardous environments like hospitals, outer space, deep sea, and homes. In unstructured and/or hazardous environments, robots have been used to avoid the exposition of human beings to danger or because of the robot reliability in executing repetitive tasks. Faults in robots can put in risk the robot task, the working environment, and human beings working with the robots.

Unfortunately, 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. [5]. Some research indicate that the mean-time-to-failure in industrial robots are only between 500 and 2500 hours [1]. If this number is small in structured environments, it is probably smaller in unstructured and/or hazardous environments due to external factors as extreme temperatures, obstacles, radiation, etc. So, there are good reasons to research fault detection and isolation (FDI) in 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 closed-link

This work is supported by FAPESP under grants 98/15732-5 and 99/10031-1.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 513-518

mechanisms as cooperative systems formed by two or more arms [2]. 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 [6]. 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 arm. Actuation redundancy makes the use of cooperative robots in unstructured and/or hazardous environments very appealing. However, 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 [5]. Furthermore, as the controller is not projected to operate with faults, the squeeze force in the load can increase and can cause damage to the load and instability to the system.

In this work, an FDI scheme for cooperative systems is presented. The Section 2 describes the dynamics of cooperative manipulators. The Section 3 describes the FDI system. Two kind of faults are considered: free-swinging joint faults (FSJF's), where the applied torque at the faulty joint is always equal to zero, and locked joint faults (LJF's), where the velocity of the faulty joint is always equal to zero. FSJF's and LJF's are detected and isolated via artificial neural networks (ANN's): Multilayer Perceptrons (MLP's) are used to reproduce the dynamics of the arms and an Radial Basis Function Network (RBFN) classifies the residual vector. 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

The equation of motion for the i-th arm in a fault-free cooperative system with m arms rigidly connected to a solid object is

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

where $\mathbf{q}_i$ is the vector of joint angles of arm i, $i = 1, \ldots, m$, τ_i is the vector of the torques at the joints of arm i, $\mathbf{M}_i$ is the inertia matrix, $\mathbf{C}_i$ is the matrix of the centrifugal and Coriolis terms, $\mathbf{g}_i$ is the gravitational torque vector, $\mathbf{J}_i$ is the Jacobian (from joint velocity to end-effector velocity) of arm i, and $\mathbf{h}_i$ is the force vector at end-effector of arm i. If the sample period Δt is sufficiently small, the dynamics of the fault-free arm i (eq. 1) can be written as

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \tag{2}$$

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

$$\dot{\mathbf{q}}_i(t + \Delta t) = \mathbf{f}_\phi(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \tag{3}$$

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 513-518

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

$$\mathbf{r}_i(t + \Delta t) = \mathbf{f}_\phi(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) - \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)). \qquad (4)$$

3 FDI System

To the best of the author's knowledge, only in [4] an FDI system for cooperative manipulators was studied. There, only one MLP is trained to reproduce the dynamics of all arms. As the end-effector force is a function of the joint variables of all arms, the inputs of the MLP are the joint positions, velocities and torques of all 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 an 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 object, and these variables can be very useful to map the system's dynamics. Furthermore, the mapping of the MLP in [4] is dependent on the object parameters. If the manipulated object is changed, the ANN's have to be trained again.

Here, the dynamics of each arm is mapped by one MLP. Thus, the mapping is not dependent on the object parameters. The inputs of the MLP i are the joint positions, velocities, torques, and end-effector forces of arm i at instant t (fig. 1). The output vector of the MLP i, which should reproduce the joint velocities of the fault-free arm i at time $t + \Delta t$, can be written as

$$\hat{\dot{\mathbf{q}}}_i(t + \Delta t) = \mathbf{f}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) + \mathbf{e}(\dot{\mathbf{q}}_i(t), \mathbf{q}_i(t), \mathbf{h}_i(t), \tau_i(t)) \qquad (5)$$

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

$$\hat{\mathbf{r}}_i(t + \Delta t) = \dot{\mathbf{q}}_i(t + \Delta t) - \hat{\dot{\mathbf{q}}}_i(t + \Delta t). \qquad (6)$$

By eq. (2-6), it can be observed that the residual vector of arm i in fault-free case is equal to the vector of mapping errors, which must be sufficiently small when compared with the fault function vector in order to allow the fault detection. The residual vector from all arms at $t + \Delta t$ are then classified by an RBFN that gives the fault information. As the residual vector of FSJF's and LJF's occurring at the same joint can occupy the same region in the input space of the RBFN, an auxiliary input ζ that gives information about the velocity of the joints is used (fig. 1). The use of ζ is motivated by the fact that the velocity of the faulty joint is zero in LJF's. As there is noise in the measurement of the joint velocity, the $i = 1, \ldots, n$ component (n is the sum of the number of joints of all arms) of ζ is defined as

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

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 513-518

where δ_i is a threshold that can be chosen based on the measurement noise. Thus, the inputs of the RBFN at each sample time are the vectors $\hat{\mathbf{r}}(t+\Delta t) = [\hat{\mathbf{r}}_1(t+\Delta t)^T \dots \hat{\mathbf{r}}_m(t+\Delta t)^T]^T$ and $\zeta(t+\Delta t) = [\zeta_1(t+\Delta t)\dots\zeta_m(t+\Delta t)]^T$ (fig. 1). In this work, the RBFN is trained by the Kohonen's Self Organizing Map [3]. The fault criteria shown in fig. 1, which is used to avoid false alarms due to misclassified individual patterns, is defined as

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

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

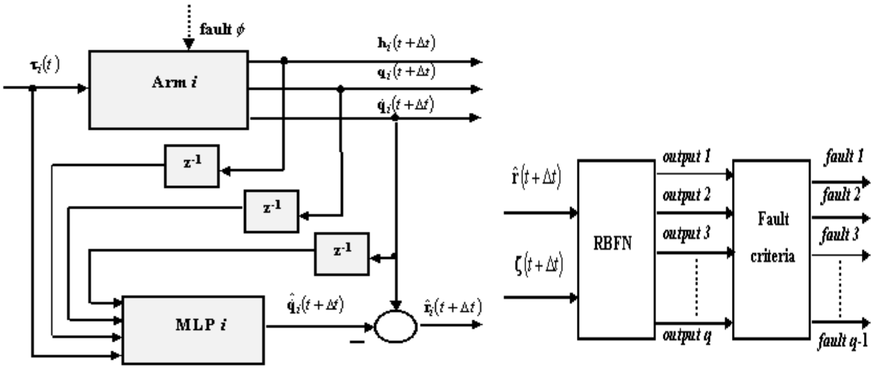

Figure 1: *Left: Residual generation. Right: Residual analysis.*

4 Results

Two three-dof planar cooperative arms manipulating an object in a x-y plane are simulated to test the FDI system. The arms are equal and the gravity force is parallel to the y-axis (the x-axis passes through the bases of the two arms). The controller proposed in [7] is used. The sample period is 0.008s and measurement noise is added to the joint positions and velocities, and to the end-effector forces. The MLP's are trained with 7400 patterns obtained in the simulation of 100 trajectories. The RBFN is trained with 2691 patterns. The FDI system (with $d = 3$ samples) is tested in 960 randomly chosen trajectories with different starting time of the faults and different mass of the object. The faults are detected in 958 trajectories (99.79%) and correctly isolated in 911 (94.9%). More 80 trajectories without faults are used to test the occurrence of false alarms. No false alarm occurs.

The next step is the application of the FDI system in a real cooperative system with two UARMII's (figure 2). Each UARMII is a 3-joint planar manipulator that floats on a thin air film on an "air table". The axis of each joint

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 513-518

is parallel to the gravity force. The cooperative system is controlled by a PC running Matlab. Each joint of the UARMII 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 [7] is used (the sample period is 0.06s). It is important to observe that the modelling of this system is difficult 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 equations of the system).

The MLP's are trained with 3250 patterns obtained in the simulation of 50 trajectories. The RBFN is trained with 2506 patterns. The FDI system (with $d = 4$ samples)is tested in 540 randomly chosen trajectories with different starting time of the faults and different mass of the object. The faults are detected in 457 trajectories (84.63%) and are correctly isolated in 361 (66.85%). More 45 trajectories without faults are used to test the occurrence of false alarms. One false alarm occurs (6.67%). Fig. 3 shows the torques of arm 1 and the RBFN outputs in a trajectory with an FSJF.

The number of correctly isolated faults was smaller in the real system mainly because FSJF's are sometimes confused with LJF's. This occurs because, as there are small gravitational torques at the joints of the real system, sometimes the velocities of the faulty joints are small. However, in these cases, even with FSJF's, 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.

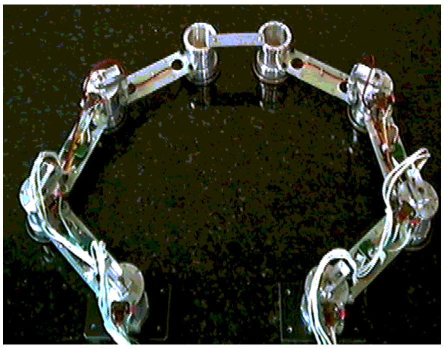

Figure 2: *Real system.*

5 Conclusions

This work presents a FDI system for cooperative manipulators. Two faults were considered: FSJF's and LJF's. The faults are detected by ANN's: MLP's

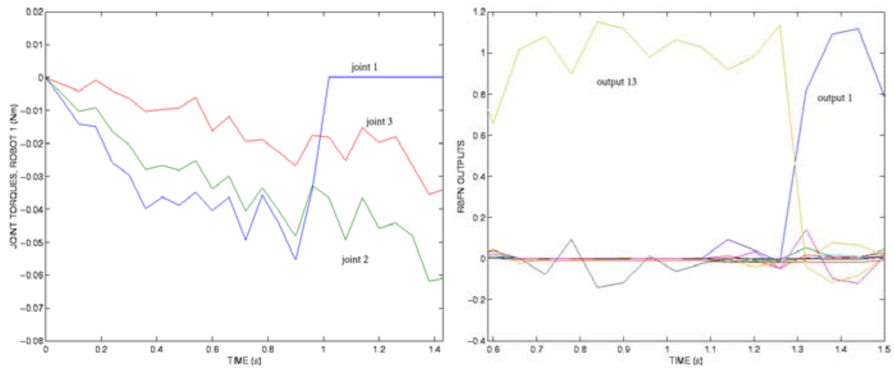

Figure 3: *Joint torques at arm 1 (left), and outputs of the RBFN (right) in a trajectory of the real system with FSJF at joint 1 (arm 1) occurring at t=1s.*

to reproduce the dynamics of the arms and an RBFN to classify the residual vector.

References

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

[2] Y. Nakamura. *Advanced robotics: redundancy and optimization.* Addison-Wesley Publishing Company, Inc., New York, 1st edition, 1991.

[3] M. H. Terra and R. Tinós. 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, 2001.

[4] R. Tinós, M. H. Terra, and M. Bergerman. Fault detection and isolation in cooperative manipulators via artificial neural networks. In *Proc. of IEEE Conf. on Control Applications*, 2001.

[5] M. L. Visinsky, J. R Cavallaro, and I. D. Walker. Robotic fault detection and fault tolerance: a survey. *Reliability Enginnering and Systems Safety*, 46:139–158, 1994.

[6] M. Vukobratovic and A. Tuneski. Mathematical model of multiple manipulators: cooperative compliant manipulation on dynamical environments. *Mechanism and Machine Theory*, 33:1211–1239, 1998.

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