DESIGN AND IMPLEMENTATION OF A MODULAR FAULT TOLERANT ROBOT CONTROL CONCEPT

G. Milighetti, C.W. Frey, B. Diestel-Feddersen, H.-B. Kuntze

Systems for Measurement, Control and Diagnosis
Fraunhofer Institut IITB
Fraunhoferstraße 1, 76131 Karlsruhe, Germany
surname@iitb.fraunhofer.de

Abstract: There are many products in the automotive or white good industry which are manufactured in a widely distributed network of several production cells in which robots play a dominating role. Faults during automatic production in the plants can result in considerable losses of the product quality or additional production costs e.g. due to maintenance or standstill periods. Moreover, faults in the robot environment of a cell can considerably endanger human health. Thus, around this field of application a very important issue will be the overall multi-sensoric supervision and control of the complete process to achieve goals of higher accuracy and more efficiency as well as to guarantee safety even in the case of component malfunctions or faults of the human operator. Within the framework of the ongoing IFATIS project at the Fraunhofer Institut IITB a multi-sensoric monitoring and control concept that meets the above mentioned requirements is being developed. Some results are presented in this paper.

Keywords: robot control, fault-diagnosis, fault tolerant control, Neuro-Fuzzy supervision

1. INTRODUCTION

The number of smart multi-sensor based robots in complex manufacturing systems is rather small and restricted to especially "tailored" production problems (e.g. welding applications). The inadequate flexibility results on the one hand from the incompatibility of hardware and software interfaces. On the other hand the missing reusability of application specific software components (e.g. control routines) is responsible for this poor state. Thus, due to the high adaptation effort, the acceptance of smart multi-sensor based manufacturing concepts is rather low.

Considerable progress within this area can be achieved by more *intelligence*, *flexibility* and *safety*. More *intelligence* will be achieved both by multi-sensory for smart communication and interaction of the manufacturing modules (e.g. robots) as well as by a closer integration of human operators for cooperative management of complex tasks. More *flexibility* with respect to different manufacturing requirements requires an open dynamic configuration and reconfiguration ("plug and work" features) of different task specific hardware and software modules like sensors, actuators or controller components using field-bus technology and software agent concept. More *safety* will be required because the human operator will be no longer separated from the manufacturing environment but more integrated as before.

In the past, very efficient concepts were developed for the sensor-based monitoring and control of robots. However, all these concepts were realized only for individual subtasks and/or movement phases in the manufacturing environment. For more complex tasks, which consist of a sequence of several different movement phases and which require in each case special sensors and controllers, only few sophisticated generic concepts have been developed until now (cf. e.g. Hayakawa, *et al.*, 2000; Hoover, Olsen, 2000; Buss and Schlegl, 2000).

Within the framework of the ongoing EU-RTD-project IFATIS at the Fraunhofer Institute IITB (cf. IFATIS 2001) a multi-sensoric monitoring and control concept that meets the above mentioned requirements is being developed. Some results are presented in this paper.

2. FAULT TOLERANT SUPERVISORY CONTROL CONCEPT

The proposed two stage hierarchical control concept (cf. Figure 1) is able to cope with manifold scenarios and has already been successfully applied by IITB for the flexible automation of a complex industrial batch process (Frey, Kuntze, 2001). It relies on the decomposition of the complex control problem into a sequence of smaller, more transparent phase specific sub-control problems. In the upper control level the actual process phase is identified with the help of both model based and multi-sensor diagnosis.
The lower level of the control structure consists of different sub-controllers which are optimized with respect to specific process phases. Principally the various low level sub-controllers may have any structure (Fuzzy control, model-based or hybrid).

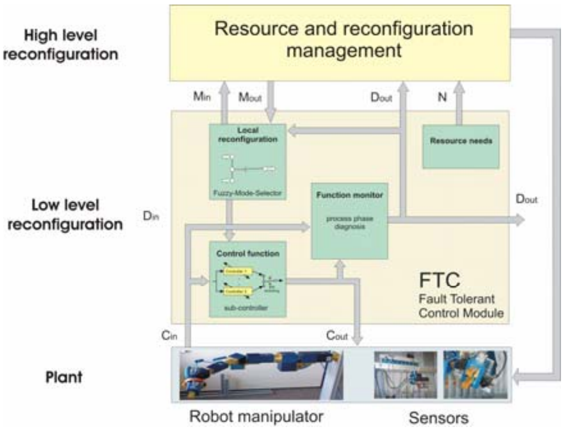

Figure 1. Hierarchical structure of fault tolerant systems

Depending on the identified process phase or fault event a mode selector activates the most appropriate control mode, which can comprise both the adaptation of control parameter sets without structural changes, or the switching of different situation-specific low level controller structures. Of course, structural switching requires necessarily the compatibility of the alternative controller structures. Moreover, shock disturbances have to be avoided by introducing a Fuzzy-based soft switching strategy (cf. Sajidman, *et al.*, 1995 for details). In order to face more critical events when a controller reconfiguration is not sufficient, a completely new strategy can be developed and a new sub-task on line generated.

To diagnose automatically malfunctions and special phases or states of a technical process the successive steps of residual generation and evaluation have to be performed (Patton, *et al.*, 2000). This general abstract model allows a broad variety of different realizations and implementations.

2.1 Residual Generation

Within the residual generation step the available sensor signals of the process are preprocessed for the purpose of extracting relevant signal characteristics. The generated residuals can be regarded as a condensed signal representation, ideally containing all important signal information. Basically residual generation in technical processes is performed by following methods (cf. Figure 2):

- signal based methods
- model based methods

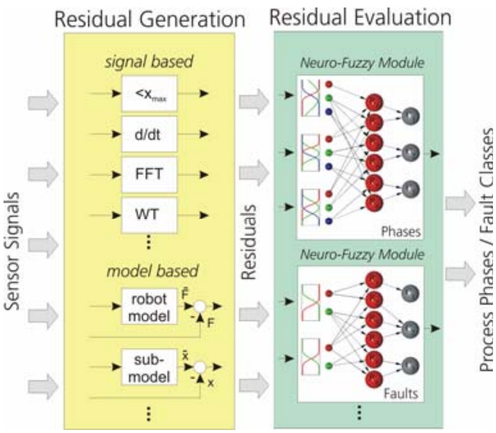

Figure 2. Neuro-Fuzzy process phase and fault diagnosis

At IITB a model based approach is used in order to monitor the robot axes. The model of the robot is given by

$$\tau - \tau_f = M(q)\ddot{q} + C(q,\dot{q}) + g(q) + f(\dot{q}) \qquad (1)$$

where q denotes the robot joint positions, M the inertia matrix, C the Coriolis forces, g the gravity vector, f the various friction terms (viscous and dry friction), τ and τ_f, respectively, the commanded (nominal) and the (unknown) fault torques.
Following the method proposed in De Luca, Mattone (2003) founded on the idea of the generalized momenta, a residual vector with n decoupled components (with n the number of the axes) is generated to supervise the dynamic behavior of each joint.
Being the generalized momenta defined by

$$p = M(q)\dot{q} \qquad (2)$$

every i component of the residual vector can be so defined

$$r_i = K_i\left[\int(\tau_i - \alpha_i - r_i)dt - p_i\right] \qquad (3)$$

with $K_i > 0$ and

$$\alpha_i = -\frac{1}{2}\dot{q}^T\frac{\partial M}{\partial q_i}\dot{q} + g_i(q) + f_i(\dot{q}) \qquad (4)$$

obtaining so a decoupled residual dynamics given by

$$\dot{r} = -Kr + K\tau_f \qquad (5)$$

Every component of r reproduced so the dynamic behavior of the correspondent component of the fault torques.

An adaptation scheme is also implemented in order to cope with the unavoidable inaccuracies of the model (cf. Milighetti, 2003).
At the same time the robot interaction with the environment is monitored through the information coming from the available external sensors (e.g. force sensor, microphone). In this way a hybrid residual vector in the form

$$\mathbf{r} = \left[r_1, \cdots, r_n, r_{sens1}, \ldots, r_{sensn} \right]^T \qquad (6)$$

can be designed.

2.2 Residual Evaluation

Once the diagnostic signals have been generated they must be evaluated. The purpose of this second step is to classify the actual motion phase or fault event according to the available signals.
Desirable is a residual evaluation module which stores the decision knowledge in an interpretable and modifiable form. In this respect Fuzzy Logic provides an ideal tool for realizing residual evaluation modules. Fuzzy Logic gives the possibility to describe knowledge by linguistic rules like e.g.:

if	*Residual 1*	*is …*	*and*
	Residual 2	*is …*	*and*
	Residual n	*is …*	
then	*Process Phase…*		(7)

The implementation of a Fuzzy module for residual evaluation can be very difficult with an increasing number of residuals to be taken into account. The problem of finding appropriate membership functions and rules is often a tiring process of trial and error. Just like linear classifiers, Fuzzy systems require in contrast to Artificial Neural Networks (ANNs) manual tuning to obtain good classification results. In order to automate the design phase of the Fuzzy system in the proposed diagnosis scheme Neuro-Fuzzy (NF) approaches are used for designing Fuzzy residual evaluation modules.
Therefore, the NEFCLASS (NEuro Fuzzy CLASSification) approach proposed by Nauck (1997) has been followed. The NF model is characterized by a three layer topology (cf. Figure 8). The input nodes in the input layer are connected by Fuzzy sets μ with the rule nodes R in the hidden layer. For semantical reasons each rule unit is assigned to a single output node. In the output layer in order to avoid weighted rules the weights are fixed to 1.
To obtain an optimal classification result the learning algorithm creates the rules and adjusts the Fuzzy sets from training examples. After training the system the classification knowledge can be easily accessed and extended by the user (Nauck, 1997).

3. HARDWARE IMPLEMENTATION

The experimental platform available at IITB for the development and investigation of the NF based supervisory control concept consists of two 7DoF AMTEC robot arms and a 2DoF pan-tilt unit (robot head). For communication and interaction with its Cartesian environment the robot is equipped with different visual, acoustic and force sensors (cf. Figure 3).
The head is equipped with a 3D stereo camera and a stereo acoustic sensor (microphone array). Moreover an optical 3D sensor (laser stripes sensor) is integrated in the gripper and a force–torque sensor is mounted on the wrist.

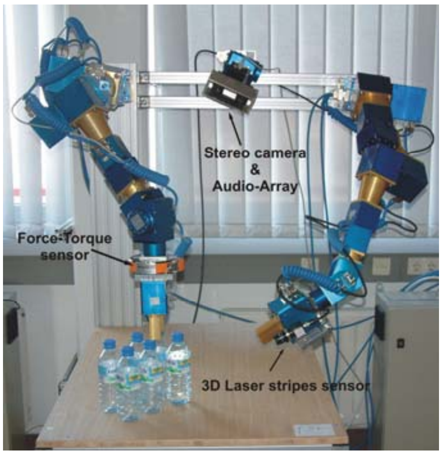

Figure 3. Experimental robot platform

The implementation of the control program is developed in C++ under Microsoft Windows.

For the first investigations a simulation platform is also available. It consists of a robot model realized with ADAMS® and controlled in the MATLAB/Simulink® environment.

4. SOFTWARE IMPLEMENTATION

The software architecture has to take into account the physical (robot hardware and setup) as well as the logical (control concept) structure of the system. To consider both aspects and achieve a transparent and flexible software architecture all data related to physical components (sensors and actuators) are implemented as software objects (cf. Figure 4).
Each of these components is defined only once in the program. The data of these objects is either periodically updated (modules and sensors connected via CANBUS) or asynchronous updated (sensors connected via TCP/IP).

To achieve a flexible control architecture which allows the easy extension of available (and implementation of new) robot tasks, these tasks are formulated as a sequence of basic skills corresponding to different process respectively motion phases. Several available skills are for example “Move Robot to Goal Frame T”, “Move Module *i* to Position P”, “Move till Force F” etc.

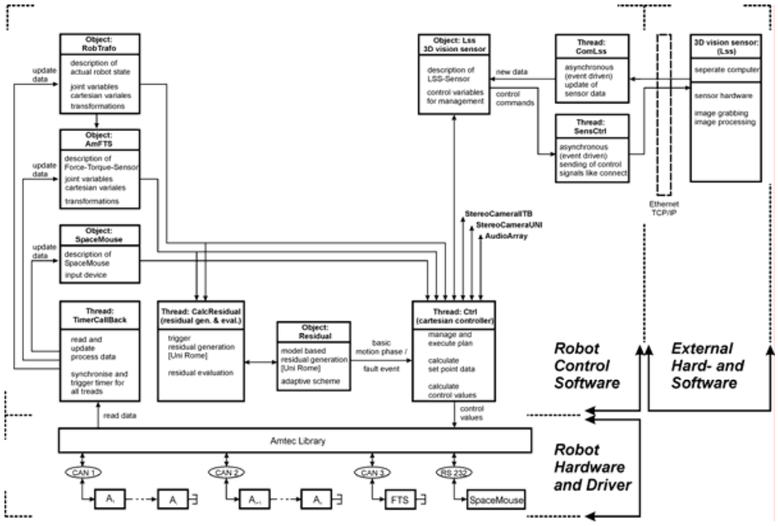

Figure 4. Principal architecture of the robot control software

Every task can so be represented by a sequence of basic skills. In order to pick up an object for example, the task is composed by the following: motion to a position, opening the gripper, move down, close the gripper and so on. It can be seen, that the formulation of plans is very easy by the usage of single skills. Every skill is structured in the same way with the implementation of three main commands:

- Initialisation
- Execution
- Exiting / Leaving

The initialisation routine calculates all missing data that are needed during the execution. These are (if applicable)

- Calculation of a set point trajectory
- Activating of external sensors like stereo camera, 3D-sensor of the gripper

After the command is initialised, every time step the execution command is performed till the leave condition is fulfilled and the program springs to the next planned skill (cf. Figure 5).

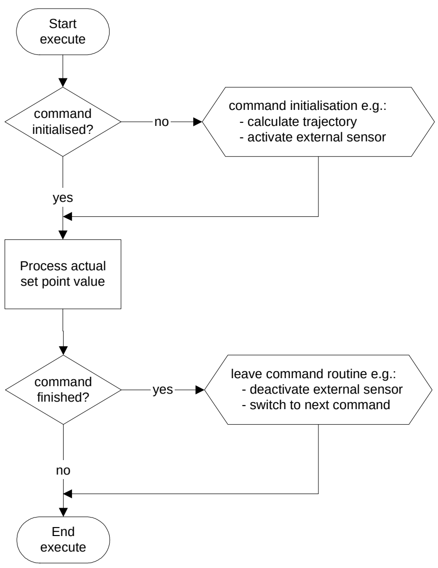

Figure 5. Flow chart of skill execution

In case of detection of unexpected faults or motion phases, the described modular software architecture makes possible an easy and fast on-line reconfiguration of the actual robot motion plan, simply interrupting the actual skill and inserting the most appropriate one for the identified new situation.
The sequence of the skills is controlled by a upper control level (corresponding to the IFATIS resource and reconfiguration management). Moreover, to follow the trajectories generated by the single skills, on the lower control level a set of motion phase specific controllers (position, force, hybrid…) is implemented, such realizing the hierarchical control structure described in paragraph 2.

5. EXPERIMENTAL RESULTS

5.1 Constraint Motion with Collision

One important basic action of robots is a hybrid motion of its arms within a complex constraint environment with different obstacles. The motion is composed of two phases: "free position control motion" and "constraint force control motion" along the obstacle surfaces. The inevitable collision must not damage either objects or robot mechanics.

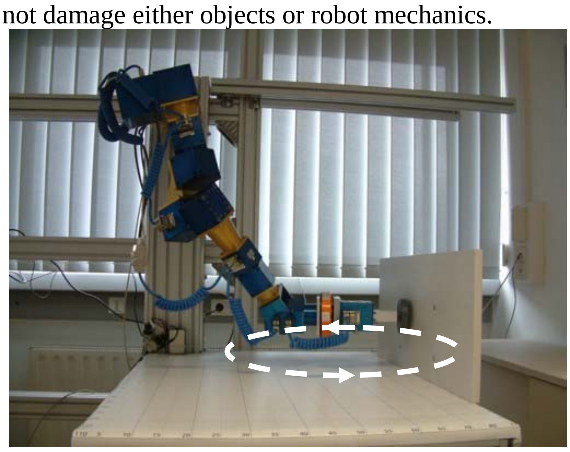

Figure 6. Circular motion constraint by a wall

Therefore this scenario has been thoroughly investigated on the experimental robot platform taking as nominal motion a circular planar motion (Figure 6). In order to verify the effectiveness of the diagnosis method used, an actuator fault of the 4th actuator has been introduced in the interval $t \in [2, 3]$ sec.)
In Figure 7 some calculated residuals are shown; the residual vector in this case includes the seven residuals coming from the seven axes of the robot (generated by the measurement of positions, velocities and commanded torques) and by one more residual coming from the microphone signal filtered by the band pass H(z).

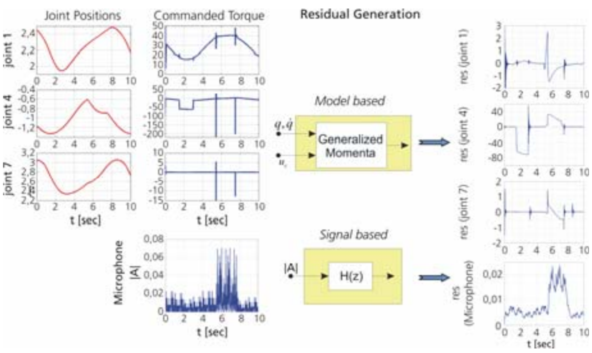

Figure 7. Example of residual generation

From the time responses shown, it can be seen that by naked eye it is very difficult to detect clearly (not to mention to classify) the malfunction in the joint. Moreover the signal disturbances due to collision can be clearly identified but without additional information it might be difficult to distinguish them from other fault classes. Thus, after the generation phase an evaluation phase must anyway be carried out.

The results coming out from the NF evaluation of the residual vector are shown in Figure 8. The output of the neural network is a probability for each possible phase considered. Taking every time step the phase with the highest probability as actual phase, a discrete classification is obtained. For example the transition between the fault free phase (FF) and the fault in the 4th joint (FJ4) at the time t = 2 seconds can be clearly seen.

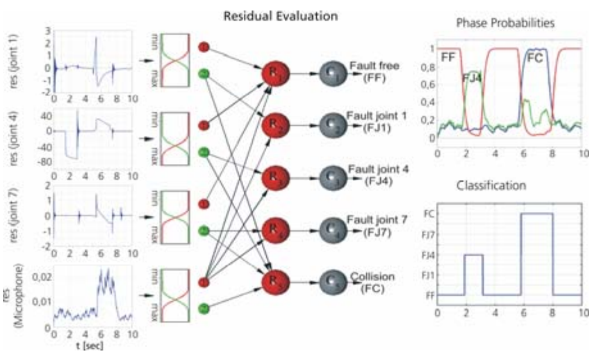

Figure 8. Example of residual evaluation

Depending on this result the most appropriate control strategy is consequently chosen.
Concerning for example the detection of the collision, a switching between position and force control can be performed (cf. Figure 9). In order to avoid shock disturbances and to have a transition between the two different controllers as soft as possible, a fuzzy switching has been implemented.

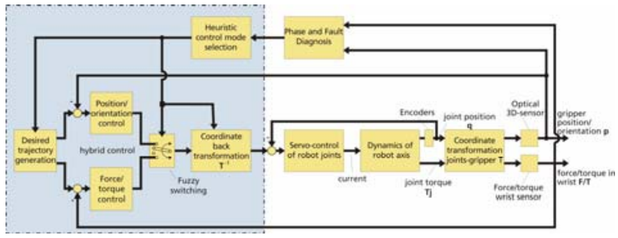

Figure 9. Block diagram showing a force/position switching

5.2. *Fitting an Object into a Hole*

A further basic problem which has frequently to be solved by robots is the so called “peg in hole” problem. Therefore an intelligent concept for performing part mating based on the supervisory control concept has been developed.
The main issue that has to be addressed is that of high contact forces during the insertion process. These can result from uncertainties of the location of the peg relative to the hole caused by positioning errors of the robot or grasping errors of the gripper. Passive and active compliance techniques are used successfully in automated assembly operations. In case of a robot acting in a time variant environment there is no precise a priori knowledge about the location of the hole, in contrast to industrial assembly tasks. The proposed concept solves the “peg in hole” problem in five different phases by combining visual sensor with force-torque sensor data. A constant monitoring of different process parameters is required in order to correctly classify the current state of the insertion process and consequently to activate the appropriate sub-controllers.
First the hole center coordinates are approximately estimated by the stereo camera. An impedance controller or position controller is activated for the first motion phase (phase I) during the approach to the hole’s environment. A detection of contact with the surface initializes the second phase (phase II).
In this case a hybrid force/position control is used to scan the surface in order to locate the exact position of the hole.
Once the peg partially dips into the hole during the scan, the processed torque signal peaks. This peak occurs when the peg lies nearly in the center of the hole. This estimation can thus be used to plan a new trajectory (cf. Newman, *et al.*, 2001 for details).
Phase III, still in hybrid control mode, consists of a tilted insertion of the peg until the characteristic three

points contact configuration described in Bruyninckx, *et al.*, 1995 has been established.
This phase is followed by the execution of a model based motion with predefined velocity profiles, which ensures the stability of the contact with the hole and results in a good pre-insertion alignment (phase IV).
Finally in phase V the peg can easily be inserted without wedging and jamming at the entrance. For the last two phases a switching back to position control takes place.

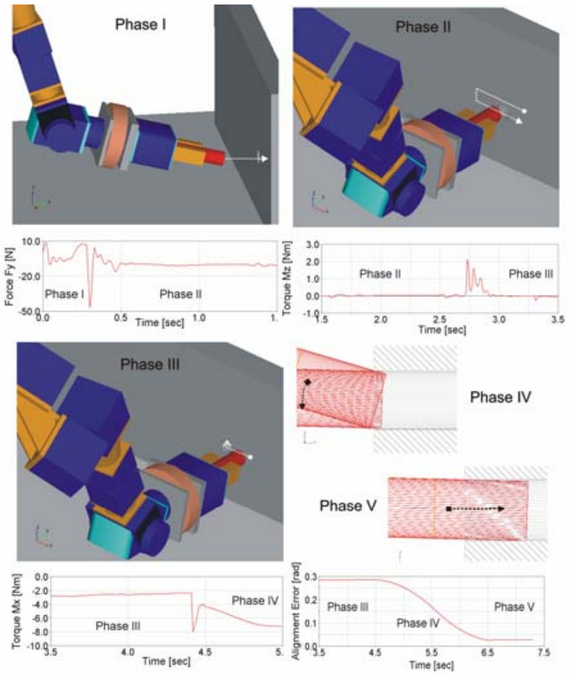

Figure 10. Five phases solution of the “peg in hole” problem

Figure 10 shows the different motion phases and the corresponding diagnosis signals monitored in order to classify them. Force/torque data as well as alignment error are used. It can be clearly seen that the switching to the next phase can be easily detected (cf. Diestel-Feddersen, 2004 for details).
In Figure 11 some pictures taken from the experimental process are shown.

Figure 11. Experimental validation of the proposed concept

6. CONCLUSIONS

In this paper a new modular supervisory robot control concept for robots has been presented. It relies on a Neuro-Fuzzy based diagnosis of the current motion phase or fault event. For each identified situation an appropriate dynamic reconfiguration either of optimal sub-controllers or of actual skills is activated. The efficiency of the proposed concept has been investigated by various experiments and simulations. Different basic actions like constraint motion with collision and fitting an object into a hole have been considered.

REFERENCES

Bruyninckx, H.; Dutré, S. and De Schutter, J.: “Peg-on-Hole: A model based solution to peg and hole alignment”, *IEEE International Conference on Robotics & Automation*, 1995

Buss, M. and Schlegl, T.: “A discrete-continuous control approach to dextrous manipulation”, *IEEE International Conference on Robotics & Automation*, San Francisco, CA, April 2000

De Luca, A. and Mattone, R.: “Actuator fault detection and isolation using generalized momenta”, *IEEE International Conference on Robotics & Automation*, Taipei, Taiwan, 2003

Diestel-Feddersen, B.: “Strukturvariable Regelung eines humanoiden Roboterarmes mit bildgebenden und Kraft-Momenten-Sensoren“, *Master thesis*, University of Karlsruhe, 2004

Frey, C.W. and Kuntze, H.-B.: “A neuro-fuzzy supervisory control system for industrial batch processes”, *IEEE Transactions on Fuzzy Systems*, Vol. 9 (2001) No. 4, August, pp. 570-577

Hayakawa, Y.; Ogata, T. and Sugano, S.: “A robotic co-operation system based on a self-organization approached human work model”. *IEEE International Conference on Robotics & Automation,* San Francisco, CA, April 2000

Hoover, A. and Olsen, B.D.: “Sensor network perception for mobile robotics”, *IEEE International Conference on Robotics & Automation*, San Francisco, CA, April 2000

IFATIS 2001: EU-RTD Projekt: “Intelligent Fault Tolerant Control in Integrated Systems (IFATIS)”, Contract number: EU-IST-2001-32122, Research programme: IST2001, Project duration: 01/01/2002-31/12/2004

Milighetti, G.: “Detection and isolation of actuator faults in robot manipulators”, *Master thesis*, University of Rome “La Sapienza”, 2003

Nauck, D.: “Neuro-Fuzzy systems: review and prospects”, *5. European Congress on Intelligent Techniques and Soft Computing (EUFIT’97)*, Aachen, September 1997

Newman, W.S., Zhao, Y. and Pao, Y.-H.: “Interpretation of force and moment signals for compliant peg-in-hole assembly”, *IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 2001

Patton, R.J., Frank, P. M. and Clark, R.N.: “Issues of fault diagnosis for dynamic systems” Springer pub. London Ltd. 2000, ISBN 3-540-19968-3

Sajidman, M. and Kuntze, H.-B. and Jacubasch, A.: “A fuzzy-logic concept for highly fast and accurate position control of industrial robots”, *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, Mai 1995