

Off-Line Error Prediction, Diagnosis and Recovery using Virtual Assembly Systems

Cem M. Baydar and Kazuhiro Saitou
({cbaydar, kazu}@umich.edu)

Department of Mechanical Engineering
The University of Michigan, MI, 48109, USA.

ABSTRACT

Automated assembly systems often stop their operation due to the unexpected failures occurred during their assembly process. Since these large-scale systems are composed of many parameters, it is difficult to anticipate all possible types of errors with their likelihood of occurrence. Several systems were developed in the literature, focusing on on-line diagnosing and recovering the assembly process in an intelligent manner based on the predicted error scenarios. However, these systems do not cover all of the possible errors and they are deficient in dealing with the unexpected error situations. The proposed approach uses Monte Carlo simulation of the assembly process with the 3D model of the assembly line to predict the possible errors in an offline manner. After that, these predicted errors can be diagnosed and recovered using Bayesian reasoning and Genetic Programming. A case study composed of a peg-in-hole assembly was performed and the results are discussed. It is expected that with this new approach, errors can be diagnosed and recovered accurately and costly downtime of robotic assembly systems will be reduced.

1. INTRODUCTION

Automation is one of the unavoidable concepts nowadays. The developments in the robotics area enable to use robots in large-scale assembly operations for high productivity. However, robotic assembly systems are so sensitive to any perturbation during their operation and this makes the system open to unexpected failures. The cost for excessive maintenance of these systems to recover from these unexpected failures was identified as 200 billion dollars in the USA in 1990 [15].

The unexpectedness of the errors for these systems arises from the fact that most of the errors are *unforeseen* to human design experts before the operation of the line. This is natural since these systems are composed of many working parameters such as dimensional variations of the products, fixtures, sensor capabilities and robot repeatability and when these working parameters are coupled with the 3D workspace, it is difficult to anticipate the conditions, likelihood of occurrence and 3D state of all of the failures [2].

Previous approaches on the error diagnosis and recovery are focused on either “on-line” investigation of error followed by a manual recovery when an error is detected or providing automated intelligent means (i.e., expert systems) to diagnose and patch the process. However, these systems are deficient to deal with most of the errors because of the following reasons:

- Not all of the error scenarios can be predicted,
- 3D-state of the possible errors is not included [2].
- Most of the systems are deficient in dealing multiple error conditions [17],
- Mapping the sensory domain to failure domain is not easy [13],
- It is not easy planning heuristics to all conditions, so they are not robust [13].

Therefore, the challenge is to predict all possible error conditions as well as their likelihood of occurrence and the associated 3D-state to provide efficient and robust error recovery means.

The proposed approach looks at the problem from a different viewpoint, which has not been used so far. It is called “*off-line error prediction and recovery*”. The method uses a commercial software package to model the assembly environment virtually in 3D. After that, the possible error situations and their likelihood of occurrence are predicted by using Monte Carlo simulation of the assembly process. Having the sensory symptoms and their associated failure type and 3D state, these conditions are stored and used for the diagnosis using Bayesian reasoning. Next step is using an off-line error recovery system to generate *robust* recovery plans [4] that can deal with multiple error conditions of similar nature using Genetic Programming [3,12] and the 3D model of the assembly system. Finally, this offline recovery system can be downloaded to the controller of the robotic system to patch the process.

It is expected that this approach will provide efficient means of gathering information about the probable error situations during an assembly process and use this information correctly to develop robust recovery plans. The outcomes of this approach have impact on the

industry to reduce costly downtime and maintenance expenses.

2. PREVIOUS WORK

Past research on error recovery in automated assembly lines has focused on using failure trees, expert systems or other intelligent reasoning methods. Among these people, Srinivas [18] is one of the earliest researchers who investigated error detection and recovery strategies. His approach was considering the tasks decomposable into a sequence of transformations from the initial state to a goal state. The next step is building a failure tree and generating an error recovery plan.

Expert systems are also one of the most popular tools used in the error diagnosis and recovery in flexible assembly systems. Several systems were developed in the literature [1] to provide diagnosis and recovery. These are knowledge-based systems, which try to provide recovery plans for possible error conditions for multi-station assembly systems. However, since they are at an abstract level and do not include all of the possible scenarios, they are deficient in handling the unexpected error situations.

Manipulating PLC codes is another approach. Zhou and DiCesare [21], proposed four argumentation methods of process control logic code with error recovery codes: input conditioning, alternate path, feedback error recovery and feedforward error recovery. Fuzzy reasoning was also used in conjunction with the Fuzzy Petri-Nets [6,10] or with expert systems [11,19] to provide probabilistic reasoning on the error diagnostics.

However, those approaches are deficient in handling geometric features of the assembly line, which is essential to make predictions of error scenarios. Since those 3D error states are missing, generated recovery plans' robustness are questionable since some unanticipated error states for the same error type (i.e., collision) may require different plan of recovery.

The need for a robust plan was first discussed in [9]. They developed an automated compliant motion planner based on geometric theory of error detection and recovery. However, the model is limited on modeling the configuration space with all of the properties such as kinematics or motion planning of the robots and 3D positional change of the products. Therefore, it is limited to one-station only. Consequently, a gap was formed between this type of concrete approach (i.e., prediction of the 3D error states) and the abstract approach that was followed by the expert systems.

The following illustration in Figure 1 shows the mapping of the two approaches and this gap between these two approaches. Therefore, a different approach is needed to fill this gap and combine the two different types of approaches discussed above in order to provide efficient means of error recovery. The proposed method

discussed in the next section aims to satisfy this need by combining latest developments in the robotics simulation technology with the intelligent reasoning and recovery planning.

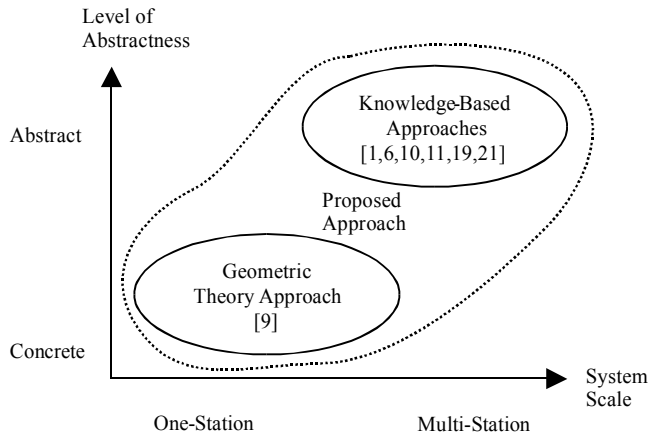


Figure.1: Station Level and System Level Approach

3. PROPOSED METHOD

Recent developments in the computer aided robotic simulation field revealed a new concept called “off-line programming”. In *off-line programming*, any robotic system can be modeled virtually in 3D and the performance of the system can be evaluated accurately from the simulations.

The proposed method takes the advantage of off-line programming to predict the possible error scenarios with their 3D-geometric state. The first step is the three-dimensional geometry-based modeling of an entire assembly line using a commercial modeling software. At this step Workspace [22] from FLOW Software Inc. is used. After the system is modeled virtually, next step is the prediction of probable error scenarios by Monte Carlo simulation of assembly processes with the three-dimensional model, based on the statistical model of the dimensional and functional errors in sensors, actuators, products and fixtures.

After probable symptoms and their likelihood is obtained, the next step is the *off-line* logic synthesis for error diagnosis and recovery from the predicted error scenarios based on the three-dimensional model using Bayesian reasoning and Genetic Algorithms. The use of Genetic Algorithms to generate recovery logic is discussed in our previous works in detail [2,3,4].

The final step is building a library of recovery logic and implementing this library to the robot controller in the assembly system to patch the process against unexpected error situations. The following sections give information on the details of the each step.

The following figure summarizes the logic of the proposed approach.

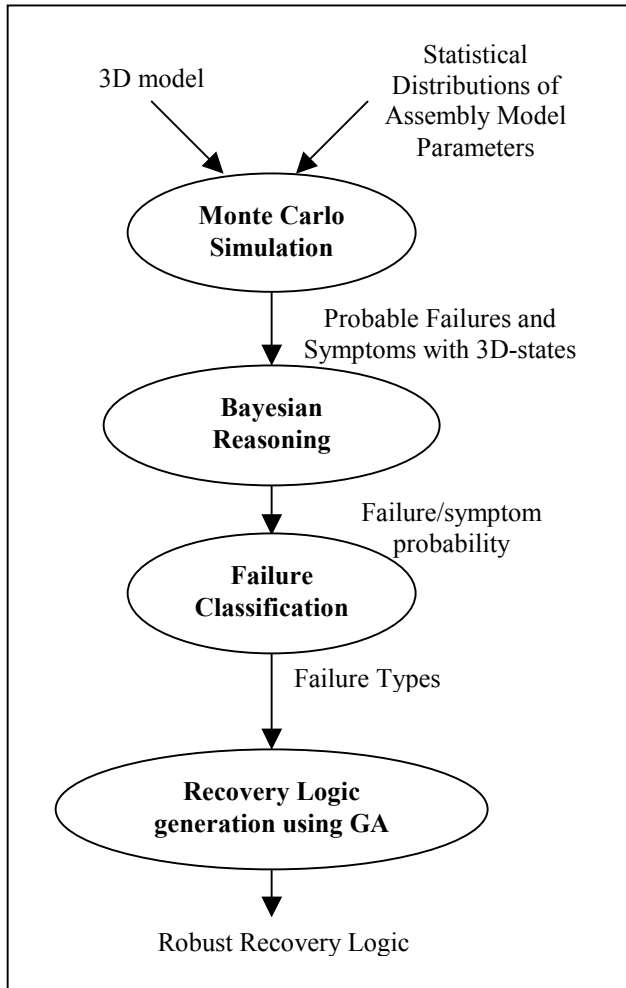


Figure.2: Working mechanism of the proposed approach

3.1 Prediction of Error Scenarios:

A widely used technique for simulating the errors is applying statistical methods to tolerance analysis of mechanical assemblies. At this step, Monte Carlo simulation is used to predict the possible errors. Process parameters are sampled from the appropriate distributions and simulations are performed. The main drawback of this method is that, to get accurate estimates, it is necessary to generate very large samples, which is computationally expensive. However, with the availability of high-speed computers this was overcome. For each simulated error with relatively high likelihood of occurrence, error diagnosis logic will be synthesized for effective error recovery from the error. Fundamental difference from the diagnosis in on-line cases is that a complete sequence of the events, which caused the

detected error, is readily available in terms of the sampled parameters.

3.2 Error Diagnosis:

Since only providing error recovery logic is not adequate for the complete recovery process, a diagnosis system is necessary to identify the correct source(s) of error. Error diagnosis is implemented in the following way. First, from the simulation results conditional probabilities of each error situation is obtained for the sampled parameters. Since Monte Carlo simulation is being used, complete sequence of the events that caused an error and their likelihood are readily available. A reasoning engine is developed based on the symptoms (outputs from the sensory values) and the probable error conditions as it is suggested in [13]. This engine will process each possibility of failure and come up with most probable one (or multiple) of the five error classifications. The belief value of each type of failure is calculated by using Bayesian reasoning using the following formula:

$$Bel(F_k) = \frac{P(Y_o \setminus F_k) * P(F_k)}{\sum_{\forall l} P(Y_o \setminus F_l) * P(F_l)}$$

In the above formula, Y_o indicates the given symptoms from the sensor array. The sensor values can only get 0 or 1 depending on their activeness. F_k is the type of failure from the following failure array given in the table below. Depending on the number of sensors or the assembly process each failure type can take any numerical value.

Table.1: Failure Array

Failure Array = {d, e, f, g, h}
d= Grasping Error
e= Collision Error
f= Sensor Failure
g= Misplacement Error
h= Flawed Parts

3.3 Error Recovery:

The proposed approach provides the generation of the error recovery logic using a method called Genetic Programming (GP). The term “Genetic Programming” was first introduced by Koza [12] and it enables a computer to do useful things by automatic programming. It uses the working principles of Genetic algorithms (GAs). In Genetic Programming, each member in the population is a computer program for the solution of the problem. Using an error situation obtained with the sampled parameters, a fitness function based on the allowed recovery criteria can be defined. After the

definition of this fitness function, genetic programming can be used to explore an efficient recovery algorithm.

The performance of the error recovery logic can be tested in a *generate and test* fashion [2] such that, several recovery logic algorithms are generated with the genetic programming engine and tested with the commercial software package, Workspace. The results of this evaluation will then be inputted to the GP engine and improved recovery logic is generated based on the obtained results [3,4].

4. EXPERIMENTAL RESULTS

A sample assembly process, which was mainly focused on inserting a peg in a hole, is experimented. The sampled parameters are statistical variations in the *dimensions of peg and hole, robot repeatability* (both translational ability and wrist repeatability effect), *grasping ability* and sensor reliability for the *grasping sensor* in the gripper and the *position sensor* located above the peg. Figure.3 shows the studied system.

The assembly process is as follows: First, a peg is grasped from the table. During this process, a camera is examining the position of peg, detecting whether it was grasped correctly or not. A sensor in the gripper is also detecting whether the peg is in the gripper or not. After peg is grasped, it is inserted into the hole. During the insertion, process torque/force sensors indicate whether a collision is occurred or not. In addition, during the releasing step of the peg, gripper sensor detects the whether the peg is released correctly or not.

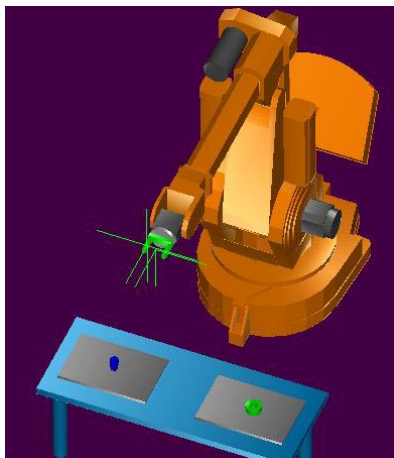


Figure.3: Monte Carlo Simulation of a Peg-in-Hole Problem

During the Monte Carlo simulation, normal distribution is assumed for the variations of peg, hole and robot repeatability, as it is suggested in [7]. Robot repeatability is a measure of positional deviation from the average position achieved by repetitive motion commands

from a start position to target position. The parameters and tolerances are as follows:

Table.2: Tolerances of the Sampled Parameters

<u>Sampled Parameter</u>	<u>Value</u>
Hole diameter	50.3 ± 0.0508 mm
Peg diameter	49.92 ± 0.03175 mm
Robot Repeatability	0.2 mm
Wrist Angle Repeatability	0.006 deg

For calculating the mean and standard deviation of the normal distribution, it is assumed that the tolerances are taken in 3σ-range as suggested in [7]. Therefore, the distribution values of the sampled parameters are given in the table below:

Table.3: Distribution values of sampled parameters

<u>Sampled Parameter</u>	<u>Distribution type</u>
Peg	Normal (0, 0.0106mm)
Hole	Normal (0, 0.169 mm)
Robot Repeatability	Normal (0, 0.67 mm)
Wrist Angle Repeatability	Normal (0,0.002 deg)
Grasping ability	Uniform (0.9 probability)
Position sensor	Uniform (0.95 probability)
Gripper sensor	Uniform (0.95 probability)

The number of simulations is taken as 1000 and the process capability is investigated. Out of 1000 simulations 317 errors are occurred. The observed results for each error type are shown in the table below:

Table.4: Failure Types and percentage

<u>Failure Type:</u>	<u>Percentage:</u>
Collision Error	36.6 %
Grasping Error (Picking)	27.44 %
Grasping Error (Releasing)	27.1 %
Misplacement Error	0
Sensor Failure	0
Flawed Parts	0
Collision + Sensor Failure	5.397 %
Gripper + Sensor Failure	3.45 %

It is realized that none of the pure sensor failures is observable. This does not mean that no sensor failures occurred, the reason is sensor failures cannot be detected at this stage and may be propagated with the later steps of the assembly process to produce detectable errors. Also there is coupled collision and sensor failures that are detected as pure collision errors from the detection system, however Monte Carlo simulation reveals that the percentage of this coupled error type is 5.397 %.

A failure diagnosis system is developed based on a probabilistic Bayesian reasoning discussed. The system uses the symptoms obtained from the assembly system and calculates the likelihood of each possible failure based on these symptoms, coming up with most probable failure type(s). The sensor and failure arrays are given in Table.5.

The sensor values can only get 0 or 1 depending on the activeness. The values for the failure array are given in Table.6 below:

Table.5: Sensor and Failure Arrays

<u>Sensor Array</u> <u>{a, b, c}</u>	<u>Failure Array</u> <u>{d, e, f, g, h}</u>
a = Gripper Sensor	d= Grasping Error
b = Torque/Force Sensor	e= Collision Error
c = Camera	f= Sensor Failure
	g= Misplacement Error
	h= Flawed Parts

Table.6: Failure Types and Associated Values

<u>Failure Type:</u>	<u>Values:</u>
Grasping Error	0-none, 1-picking, 2-releasing
Collision Error	0-none, 1-collision
Sensor Failure	0-none 1-grripper sens., 2-camera, 3-both
Misplacement Error	0-none, 1-misplacement
Flawed Parts	0-none, 1-flawed part

For each failure type, a belief value is calculated using the equation discussed before. The following results are obtained with the given input symptoms as shown in Table.7:

Table.7: Results of Diagnosis Process

<u>Symptom</u>	<u>Failure Type/Probability</u>
(1,0,0)	(2,0,0,0,0) / 0.996
(0,1,0)	(0,1,0,0,0) / 0.989
(0,0,1)	(1,0,0,1,0) / 1
(1,0,1)	(1,0,0,0,0) / 1

It is realized that having a symptom from torque/force sensor does not mean that the error is due to pure collision. There is a possibility of having this error coupled with one/all sensor failures but the most probable action is using the error recovery strategy for collision.

A virtual recovery system is developed based on the obtained errors and their states in 3D Workspace. The system uses Genetic Algorithms to provide error recovery logic and based on a system developed before and discussed in [3,4]. A robust recovery code for collision error is generated in RAPID language. The following

lines are belong to the generated code for the collision recovery only:

```

MODULE collision recovery
CONST robtargt NewGP005 := [[1553.09,
530.71, 770.341], [0.707107, -8.65927E-017,
0.707107, 8.65927E-017], [0, 0,-1, 0],
[9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
CONST robtargt NewGP006 := [[1553.09,
530.89, 770.341], [0.707107, -8.65927E-017,
0.707107, 8.65927E-017], [0, 0,-1, 0],
[9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
CONST robtargt NewGP007 := [[1553.09,
530.89, 722.85], [0.707107, -8.65927E-017,
0.707107, 8.65927E-017], [0, 0,-1, 0],
[9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
PERS tooldata t_Gripper := [ TRUE, [[0,
2.27777E-014, 93],[1, 0, 0,
0]], [0.01, [0.01, 0.01, 0.01], [1, 0, 0,
0], 0, 0, 0]];
PERS tooldata t_Nil := [ TRUE, [[0, 0,
0], [1, 0, 0, 0], [0.01, [0.01, 0.01,
0.01], [1, 0, 0, 0], 0, 0, 0]];
PERS wobjdata w_Nil := [ FALSE, TRUE,
"", [[0, 0, 0], [1, 0, 0, 0]], [[0, 0, 0], [1,
0, 0, 0]]];

PROC Path1()
MoveJ NewGP005, v1000, z1, t_Gripper;
MoveJ NewGP006, v1000, z1, t_Gripper;
MoveL NewGP007, v1000, z1, t_Gripper;
!- ThisDocument.RunBehavior "Gripper",
"OpenGripper", ""

!- ThisDocument.RunBehavior "Gripper",
"UnGrasp", ""
ENDPROC

PROC main()
Path1;
ENDPROC
ENDMODULE

```

This case study demonstrated the validity of the proposed approach. The sensory information is mapped on the failure domain efficiently, to predict the probable failures and their 3D state. Another advantage is the generation of the recovery code using the virtual assembly system saves the time on finding a robust recovery logic algorithm for the system.

5. CONCLUSIONS

A new approach on the investigation of error diagnosis and recovery is discussed in this paper. Current systems use intelligent reasoning on the diagnosis and recovery for the automated assembly lines. However, they are deficient in anticipating all errors and they leave the 3D state of the possible errors out of consideration, which makes the generated recovery codes non-robust. Because of these facts, the need is identified as to predict all possible error conditions as well as their likelihood of occurrence and the associated 3D-state to provide efficient and robust error recovery means.

The proposed system uses a commercial software package for robotic simulation for the prediction, diagnosis and recovery of the possible failures during an assembly process in four steps:

- Modeling of the assembly system using a commercial off-line robotic software package.
- Monte Carlo simulation of assembly processes to predict the possible error conditions and their likelihood of occurrence.
- Logic synthesis for error diagnosis and recovery from the predicted error scenarios based on the three-dimensional model using Bayesian reasoning and Genetic Algorithms.
- Downloading the developed recovery codes to the robotic controller to patch the assembly process against the unexpected errors.

A case study was conducted by developing a virtual assembly system, which is responsible from a peg-in-hole assembly process. The obtained results showed that, the system is capable of identifying the possible failures and their likelihood as well as their 3D geometrical state. The system is also capable of generating robust recovery codes as in discussed in our previous works [2,3,4].

The proposed method aims to cover the gap between the configuration space approach and the abstract-level knowledge-based approach. Future studies will be performed on more complicated assembly process composed of multiple assembly stations. It is expected that with this new approach, errors can be diagnosed and recovered accurately and costly downtime of robotic assembly systems will be reduced.

References

[1] Abu-Hamdan M. G., El-Gizawy A. S., "Computer Aided Monitoring System for Flexible Assembly Operations", Computers in Industry, Vol. 34, pp. 1-10, 1997.

[2] Baydar C., Saitou K., "Off-line error recovery logic synthesis in automated assembly lines by using genetic programming", Proceedings of the 2000 Japan-USA Symposium on Flexible Automation, 2000.

[3] Baydar C., Saitou K., "A Genetic Programming Framework for Error Recovery in Robotic Assembly Systems", Genetic Evolution and Computation Conference 2000, 2000.

[4] Baydar C., Saitou K., "Generation of robust recovery logic in assembly systems using multi-level optimization and genetic programming", Proceedings of the ASME-DETC2000/CIE Conference, 2000.

[5] Brnyjolfsson S., Arnstrom A., "Error Detection and Recovery in Flexible Assembly Systems", International Journal of Advanced Manufacturing Technology, Vol.5, pp. 112-125, 1990.

[6] Cao T.C., Sanderson A.C., "Sensor-based error recovery for robotic task sequences using fuzzy petri-nets", Proceedings of

the 1992 IEEE International Conference on Robotics and Automation, Vol.2, pp.1063-1069, 1992.

[7] ElMaraghy H. A., ElMaraghy W. H., Knoll L., "Design Specification of Parts Dimensional Tolerance for Robotic Assembly", Computers in Industry, v10, pp: 47-59, 1988.

[8] Evans E.Z., Lee S.G., "Automatic Generation of Error Recovery Knowledge Through Learned Activity", Proceedings of the 1994 IEEE International Conference on Robotics and Automation, Vol. 4., pp. 2915-2920, 1994.

[9] Jennings J., Donald B., Campbell D., "Towards Experimental Verification of an Automated Compliant Motion Planner Based on a Geometric Theory of Error Detection and Recovery", IEEE International Conference on Robotics and Automation, pp. 632-637, 1989.

[10] Jing Q., Xisen W., Zhihua P., Youngcheng X., "A Research on Fault Diagnostic Expert System Based on Fuzzy Petri Nets for FMS Machining Cell", Proceeding of IEEE International Conference on Industrial Technology, pp. 122-125, 1996.

[11] Kang L., Wenhan Q., "Fuzzy Expert System in Robotic Assembly Workcell", Proceedings of IEEE TENCON, pp. 738-741, 1993.

[12] Koza J. R., "Genetic Programming: On the Programming of Computers by Natural Selection", MIT Press, Cambridge, MA, 1992.

[13] Lopes L.S., Camarinho-Matos L.M., "Towards Intelligent Execution Supervision for Flexible Assembly Systems", IEEE International Conference on Robotics and Automation, pp.1225-1230, 1996.

[14] Lunze J., Schiller F., "An Example of Fault Diagnosis by means of Probabilistic Logic Reasoning", Control Engineering Practice, Vol.7, pp. 271-278, 1999.

[15] Luxhoj J.T., Riis J. O., Thorsteinsson U., "Trends and Perspectives in Industrial Maintenance Management", Journal of Manufacturing Systems, Vol.16, No.6, 1997.

[16] Lam R. K., Pollard N. S., Desai R. S., "Studies in Knowledge-Based Diagnosis of Failures in Robotic Assembly", IEEE Conference on Robotics and Automation, pp. 60-65, 1990.

[17] Sampath M., Sengupta R., Lafortune S., Sinnamohideen K., Teneketzis C., "Failure Diagnosis using Discrete-Event Models", IEEE Transactions on Control Systems Technology, v4, n2, pp. 105-124, 1996.

[18] Srinivas S., "Error Recovery in Robot Systems", Ph.D Thesis, California Institute of Technology, 1977.

[19] Tzafestas S. G., Stamou G. B., "Concerning Automated Assembly: Knowledge-Based Issues and a Fuzzy System for Assembly under Uncertainty", Computer Integrated Manufacturing Systems, Vol. 10, No.3, pp. 183-192, 1997.

[20] Visinsky M.L., Cavallaro J.R., Walker I. D., "Expert System Framework for Fault Detection and Fault Tolerance in Robotics", Computers in Electrical Engineering Vol. 20, No.5, pp 421-435, 1994.

[21] Zhou M. C., DiCesare F., "Adaptive design of petri-net controllers for error recovery in automated manufacturing systems", IEEE Transactions on Systems, Man and Cybernetics, Vol.19, No.5, pp.963-973, 1989.

[22] Workspace 5 User-Manual, 2000.