Abstract—A multi-resolution wavelet analysis coupled with a neural network based approach is applied in the problem of fault diagnostics of industrial robots. The multi-resolution analysis implements discrete wavelet transforms with filters and decomposes the signal in various levels. The approximate and detailed coefficients of the decomposed signals are then used for training a feedforward neural network whose output determines the state (faulty or normal) of the robot. The neural network classifier was then implemented and monitored in a Matlab-Simulink environment using a state-flow model. Validation of the method was performed offline using experimental data obtained from an industrial robot manipulator used in the semi-conductor industry.

1. INTRODUCTION

One of the most critical aspects of industrial robotic systems for manufacturing applications is reliability. A failure of the robot results in production line down-time and, in some cases, damage to the product that is being processed. Typically, it takes several days to bring the production line back to operation because the entire tool and workstation needs to be cleaned once the robot has been repaired or replaced. If product damage occurs, it may represent a loss in the order of several tens of thousands of dollars, depending on how far in the production cycle the damage takes place. The risk of unexpected failures can be significantly reduced by monitoring the performance of the robotic system and predicting failures. This information can be used to prevent failures by servicing the robot as a part of the regular maintenance schedule of the tool. Those techniques are called health maintenance or condition (based) maintenance of the industrial robot.

During the last decades, a large number of approaches have been developed towards machine condition monitoring, aiming to fault detection [1]. The common aspect of all is the intention to detect changes in characteristic aspects of the “normal” or “nominal” machine behavior that would possibly imply a fault, wear or abnormal operating condition. Due to the large quantities of information present in the dynamic response of machines, a number of signal processing methods have been especially adapted and modified, in order to expose characteristic aspects of the “signature” of the machine related to specific faults. Quite common for example, is the case of rotating machines under constant operating conditions, where traditional spectral analysis based on Fourier transform is used, in order to associate specific peaks (or peak families) of the vibration spectrum to specific types of machine faults (e.g. unbalance/misalignment). However, due to the inefficiency of the Fourier transform to cope with situations where rapid changes occur during the measurement horizon, several signal processing methods recently developed, have been applied. These methods include: Wavelets [2]; High Order Statistics and Cyclostationary Analysis [3]; Blind Source Separation [4]; Non-linear/Chaotic Analysis [5]; ARMA models [6].

Although a large number of diagnosis methods and relevant applications for industrial equipment already exist, the current research in the area of fault diagnosis of industrial robotic systems is rather poor, being restricted practically to a limited number of faults, like joint backlash [7]. A possible explanation is the fact that certain additional restrictions exist in this field, like the large variability of faults, the unsteady and non-uniform operating conditions, the very small number of sensors that can be used and the rather limited time records of the equipment.

In this paper we present a novel approach to perform fault diagnostics of industrial robotic systems using Neural Networks and Wavelet Multi-Resolution Analysis (WMRA). WMRA analysis is used for data reduction and for capturing the required features that will be used in the training of a feedforward neural network. The neural network classifier is then implemented and monitored in a Matlab Simulink environment using state-flow modeling. Validation of the method, with excellent results, was performed offline using experimental data obtained from an industrial robot manipulator used in the semi-conductor industry.

II. METHODS

A. Discrete Wavelet Transforms

The wavelet transform (WT) is a relatively new type of transform. One of the main strengths that characterizes this transform is its ability to provide information about the time-frequency representation of the signal. A detailed tutorial on fundamentals of WT is available in [8]. The signals in the concerned problem are non-stationary in nature; hence a
time-frequency representation gives more information about the occurrence of fault than simply a frequency domain representation. For most practical applications there are two kinds of wavelets available which are the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT). During CWT analysis the wavelet is shifted smoothly over the full domain of the analyzed signal and thus calculating the wavelet coefficient at every scale generating a huge amount of data. Due to the huge amount of data generated through CWT, training classifiers based on its coefficients at different scales can often become cumbersome. Instead, for practical implementation DWT is preferred, which is the sampled version of continuous wavelet transforms and can be implemented through multi-resolution analysis.

In this paper we perform wavelet multi-resolution analysis (WMRA) on an industrial robot’s joint motor signals by decomposing the signal into three resolution levels. Figure 1 shows the implementation of a 3-level wavelet multi-resolution analysis. At each level, the high pass filter produces detailed information, $d[n]$, and the low pass filter produces coarse approximation, $a[n]$. With this approach, the time resolution becomes good at high frequencies, while the frequency resolution becomes good at low frequencies. The approximate coefficient at the $j^{th}$ resolution can be calculated as [9]:

$$a_j(n) = 2^{(-j/2)} \sum_{k} x(n) \phi(2^{-j}n - k)$$  

The detailed coefficient at the $j^{th}$ level are computed as

$$d_j(n) = 2^{(-j/2)} \sum_{k} x(n) \psi(2^{-j}n - k)$$

where $\phi(n)$ is the scaling function, $\psi(n)$ is the wavelet basis function, $j$ represents the scaling (dilation) index, $k$ represents the translation (shifting) index and $x(n)$ represents the original signal. Thus the approximation signal $A_j(n)$ and the detail signal $D_j(n)$ at the $j$th resolution can be represented as

$$A_j(n) = \sum_{k} a_j \phi \left( 2^j n - k \right)$$

$$D_j(n) = \sum_{k} d_j \psi \left( 2^j n - k \right)$$

Fig. 1: Three Level Signal Decomposition Using Wavelet Multi-Resolution Analysis

In DWT, the most prominent information in the signal appears in high amplitudes and the less prominent information appears in very low amplitudes. The high frequency band outputs are taken as the detail coefficient ($d_1$, $d_2$, $d_3$) and the low frequency band outputs are taken as the approximate coefficients ($a_1$, $a_2$, $a_3$). As shown in Figure 1 there is a down-sampling process carried out at each stage by ignoring the second sample of every sampling pair thus improving the computation time and data storage capabilities. In our method we select the $a_3$ and $d_3$ coefficients as the primary inputs to the feed-forward neural network based classifier.

B. Neural Network For Fault Classification

Figure 2 shows the structure of the neural network that is used in this work to perform the fault classification. The complexity of a neural network depends on its architecture characterized by a number of hidden layers and a number of neurons present in each layer. With a smaller size network the training time is comparatively lesser and thus more suitable for online deployment. With this view in mind we design our classifier with only one hidden layer and then test the performance of the network with 10, 20 and 40 neurons in the hidden layer, as shown in Figure 2. For this network we used the hyperbolic tangent function as the activation function of the hidden layer and a saturated linear function for the output layer. The neural network was trained based on Marquardt-Levenberg algorithm. The network has two inputs that are the outputs of the 3-level WMRA ($a_3$, $d_3$) and six outputs (see Table 1 and Figure 2) representing each one of the five faults considered in this project and the normal case. Based on the fault type, the desired output neurons should be either 0 or 1 as shown in Table 1. Table 1 shows the desired output sequence used for training the neural network for different normal and faulty cases.

![Fig. 2: Feedforward Neural Network Structure.](image)

<table>
<thead>
<tr>
<th>Fault Type</th>
<th>N</th>
<th>HBD</th>
<th>MBD</th>
<th>HC</th>
<th>SC</th>
<th>PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBD</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MBD</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

III. EXPERIMENTAL PROCEDURES

To validate experimentally the efficiency of the proposed WMRA-neural network based method for fault diagnostics of industrial manipulators, a single end-effector, scara type, four degree of freedom arm from the semi-conductor industry was used for data collection and inducing faults in the system. For simplicity of the experimental procedures we limited the motion of the robot into one axis, that of displacement along the z-direction. The robot shown in Figure 3 was commanded to move up and down along the z-direction. Data sets were collected for the robot operating under normal conditions and different fault scenarios at dif-
ferent speed settings during the z-downward motion of the robot. The overall methodology followed in this paper is shown in Figure 3b.

The z-motion of the robot is produced through a horizontal rail with a carriage driven by a motor through a ball-screw arrangement. The duty cycle can be viewed as a normalized measure of the overall voltage applied to the motor and hence of the joint torque produced. A 24-mm motion of the carriage was used for testing purposes. The motion is performed in both directions (up and down) since some of the modes of failure exhibit direction-dependent symptoms. However data shown in Figures 4-6 represent only downward motion, when the robot homes back to the zero position. During the tests, numerous variables were monitored and recorded which including position, velocity, acceleration, motor torque in duty cycle etc. Out of these variables, the duty cycle of the motor showed distinctive symptoms under different modes of failure. Duty cycle can be defined as a fixed repetitive load pattern over a given period of time which is expressed as the ratio of on-time to cycle period.

Brake drag (HBD & MBD): The motor is equipped with an electromagnetic brake, which opens when voltage is applied and engages when voltage is removed. The brake may cause undesirable drag when voltage is lost, e.g., due to faulty cabling, or if it is mechanically misadjusted. As shown in Figure 3a, in our experiments, the brake drag was induced by introducing various thin metal shims of different thicknesses between the brakes to induce different level of brake drags. As shown in Figures 5a and 5b, this type of brake drag fault is accompanied by an increase of the motor duty cycle in both directions of motion with a periodic behavior that corresponds to one rotation of the ball-screw arrangement. The data in Figure 5a are obtained by introducing a metal film with greater thickness compared to the data shown in Figure 5b.

Collision with external obstruction (HC & SC): Figures 6a and 6b illustrate data from an experiment where a collision with a hard object, such as an obstacle in the workspace of the system, occurred. In our test, we opened the robot chassis and introduced two different kinds of foam of different hardness along its way while the robot was homing. Both the foams were pressed down at different rates thus yielding different graphs for hard and soft collision. We observe a steeper rise in the motor duty cycle for the hard foam and a gradual rise for the soft foam since depending on their resistance to motion. The motor duty cycle is limited by ±65%, and the motor is disabled by a safety mechanism to prevent damage to the test system due to excessive current and heat buildup.

To better illustrate the collected data, Figure 7 shows the distribution of the data in two-dimensional space. **Sammon Mapping**, used primarily in the area of multidimensional scaling, was used to explore the data and to find any
possible inherent clusters or underlying distributions in the collected experimental data.

![High Brake Drag](image1)

![Medium Brake Drag](image2)

Fig. 5: a) (Top) High Brake Drag (HBD); b) (Bottom) Medium Brake Drag (MBD).

![Hard Collision](image3)

![Soft Collision](image4)

Fig. 6: a) (Top) Hard Collision (HC); b) (Bottom) Soft Collision (SC).

The axes in Figure 7 represent the first Sammon projection along the x-axis and the second Sammon projection along the y-axis. We can see that the brake drag data are very well separated from the other conditions. There is a lot of overlap and thus there exists a higher correlation between the phase, collision and normal data sets.

![Original Signal Used for Analysis](image5)

Fig. 8: Original Signal Used for Analysis

IV. RESULTS

A. WAVELET PREPROCESSING RESULTS

Since the signals are non-stationary in nature we employed WMRA to extract features from them. Here wavelets act as the data pre-processor and extract the approximate and the detail coefficient at each resolution. The extracted features after the third level downsampling were fed to the artificial neural network based classifier for training. Figure 8 shows the data that was collected experimentally and used for the analysis. From Figure 8 we can see that the brake drag can be easily distinguished from the rest by visual inspection. However, it is hard to distinguish the rest of the faults from the raw data and preprocessing with WMRA will be required. To provide a more intuitive understanding of the data, we performed a spectrographic analysis on the data using CWT as described in [10]. The corresponding energy allocation of the signal is shown in Figure 9. The different color regions on the surface plot show different energy levels associated with different operating conditions (faulty and normal).

In order to extract the important features from the signal we performed WMRA on the signal x(n) shown in Figure 8 to extract the approximate coefficient $a_3$ and the detail coefficient $d_3$. The original signal had 9500 sampling points and after down-sampling both $a_3$ and $d_3$ have 1200 sampling points. Therefore, a significant amount of data reduction was performed. Figure 10 shows the approximate coefficient signal and the detail coefficient signal with 1200 sampling points. There is a distinct pattern for different faults represented in the detail and the approximate signal which if combined can form the input to the neural network based classifier.

The wavelet transformation at high frequencies shows good time resolution and poor frequency resolution while at low frequency it gives good frequency resolution but poor time resolution. When the system is running under normal conditions the operation can be described as a low frequency in which case we get good frequency resolution but bad time resolution. However, when a fault condition occurs, which is typically reflected in the signal as a high frequency disturbance the wavelet transformation gives us good time resolution which is critical to detecting the fault signature.

![Scatter Plot of the Different Z-Motor Torque (Duty Cycle) Projected in 2d Space](image6)

Fig. 7: Scatter Plot of the Different Z-Motor Torque (Duty Cycle) Projected in 2d Space, x-axis represents the first Sammon projection and y-axis the second Sammon projection.
B. NEURAL NETWORK TRAINING & EVALUATION

A convergence precision, mean square error (MSE) is introduced to assist in the real time calibration and monitoring of the network prediction accuracy:

\[
MSE = \frac{1}{N} \sum_{i=1}^{N} [t(i) - o(i)]^2
\]  

(5)

where \(t(i)\) and \(o(i)\) represents the targeted value and the output value of the network.

During the neural network training the data was randomly partitioned into three parts, one used for the network training, one for validation and the last for testing of the network. Figure 11 shows the performance curve of the neural network as derived using Equation (5). The training seizes if no improvement in the network’s performance is measured.

The residual errors of various neural networks are shown in Table 2 for three different sets of hidden neurons (10, 20, and 40). The goal was to have the simplest network with the less number of neurons in the hidden layer without losing the network’s overall prediction accuracy, measured through the mean squared error (MSE). From Table 2 we can see that there is a significant decrease in the MSE between 10 neurons and 20 neurons, however there is not much difference between 20 and 40 neurons except for the fact that the computation time is longer with 40 neurons.

In order to evaluate the prediction accuracy of the neural classifier we use a confusion matrix as shown in Table 3. A confusion matrix [11] contains information about actual and predicted classifications done by a fault classification system. Every row in Table 3 represents an actual mode of the robot (normal or faulty). Each column represents the classification made by the neural network based fault diagnostics algorithm. For example, row 1 of Table 3 demonstrates that for all tests performed while the robot was in a normal mode, the neural network fault classifier identified the correct mode of operation in 90% of the cases while in 3.7% of the cases misdiagnosed a hard collision, 2.7% misdiagnosed a soft collision and 3.6% of the cases misdiagnosed an incorrect commutation angle. From the confusion matrix we observe that all normal/faults were detected successfully at a very high percentage except in the cases of hard and soft collisions where the classifier had problems distinguishing them. If, the soft and hard collisions are considered as one fault, then the classifier shows a high success rate in this case too.

C. REAL TIME MONITORING USING STATE FLOW

A finite state machines (FSM) model consists of a set of states, a set of transitions between the states and a set of actions associated with the states. Thus, the state in FSM represents the current state of the system depending on the input to the system. In this section we demonstrate the development of a real time monitoring system for diagnosis and fault detection in a Matlab Simulink environment via a state flow model.

The neural network classifier was trained using the neural fitting toolbox in Matlab and was generated into Simulink using the \texttt{gensim} command. To represent different normal/faulty data sets, source blocks from the Simulink window were utilized that called upon data from the Matlab workspace. A \texttt{mux} block from the signal routing toolbox and six on-off switches were used to determine the kind of data flowing. For continuous update a state flow model update block was provided to the state flow block outline in green.

Even though there were six different types of data available thus representing six different states (i.e. 5 faults and normal)
one normal case), we used only four states for our representation. From the confusion matrix we observed that the classifier performed poorly in distinguishing hard collisions from soft collisions. To overcome this problem we grouped hard and soft collision as one state and also the high brake drag and the medium brake drag as another state. Thus we represented the data with only four states corresponding to the normal, brake drag, collision and incorrect commutation angle (phase). The state flow box was tuned so that corresponding to the sequence of input shown in Table 1 it will have its own response. In this example, the state flow response to normal data was denoted by 1, phase by 2, brake drag by 3 and collision by 4 and all are shown in Figure 12. When the system goes online, Figure 13, all the modules inside the state flow block get highlighted. Each one of the four modules has two states corresponding to the off and on transition state. In our representation we used the Roboton module, Figure 13, to denote the normal working condition. Thus, if the industrial robot works perfectly with no faults observed then the On state block in the Roboton module will be highlighted, signifying the normal working state of the robot. To create a fault scenario and observe the effectiveness of the stateflow model, we turned on the on switch for the soft collision, such that data related to soft collision is being sent to the neural network. Once the output of the neural network is sent into the stateflow window, then the corresponding On block shown inside module 4 gets highlighted, however all other boxes in modules 1, 2 and 3 highlight the Off position. Thus, by looking at the stateflow model at any given time we can infer the state of the system since each event can be efficiently represented graphically in a transition block.

Fig. 12: Real Time Simulation in Simulink of the Neural Classifier

Fig. 13: Real Time Monitoring Provided By a State Flow Model, Indicating the Current State of the System

V. CONCLUSIONS

In this paper we applied wavelet multi-resolution analysis in the preprocessing of data used in the fault diagnostics of industrial robots. The output of WMRA was then used as the input of a feedforward neural network classifier whose output predicts the state of the industrial robot. As states we considered the normal operation and 5 faulty conditions which are: brake drag (high & medium), collision (hard & soft) with external obstruction and incorrect motor commutation (phase angle). Verification of the proposed algorithm was performed off-line using experimental data obtained from an industrial robot used in the semiconductor manufacturing industry. The WMRA was excellent for data reduction and capturing the required features of the signal needed for the neural network training. The experimental results showed that the neural network classifier had a very high fault detection success rate (around or above 90%) for all faults. However, it showed difficulty distinguishing between the hard and soft collisions. Combining these two relevant faulty conditions into one fault that of just “collision” the neural network based classifier achieves a high rate of success for that fault too. A Matlab Simulink state flow model was developed for providing real time monitoring at any given time. The proposed approach demonstrates high fault detection accuracy with low computational burden that can allow real time implementation of the method. Also the performance of the classifier is dependent on the knowledge base of the neural network thus posing a serious limitation when exposed to new data.

ACKNOWLEDGMENTS

The authors would like to express their appreciation to Brooks Automation of Chelmsford, MA for the support of this work.

REFERENCES