Autoencoder with Differential Evolution Approach for Sensor Fault Detection in Robotic Manipulators

Dulce Ibarra-Montes, Francisco Gonzalez Barriga, Alma Y. Alanis, Jesus Hernandez-Barragan, and Gabriel Martinez-Soltero

Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Guadalajara 44430, Mexico
erasmo.martinez@academicos.udg.mx

Abstract. Fault detection is a topic of great relevance due to its importance in the control of various types of systems. In particular, the most common faults usually occur in actuators or sensors. This paper proposes the use of autoencoders for the detection of faults in the joint sensors of a robotic manipulator during trajectory tracking tasks. The proposed methodology addresses the identification of faults caused by disconnections, noise and blockage in the sensors of all the joints of the manipulator. In addition, a differential evolution optimization algorithm was implemented to select the optimal threshold that determines whether or not a fault occurs, which increases the accuracy of the system. The applicability of this approach is demonstrated by detecting faults in the sensors of a 5-degree-of-freedom (DOF) manipulator. The results obtained show that autoencoders are a powerful and versatile tool for fault detection in robotic manipulators, at least in most of their joints.

Keywords: Autoencoder · Robot manipulators · Fault detections. · Differential Evolution

1 Introduction

Robotics has transformed manufacturing, medicine and space exploration, driving the automation of complex, dangerous [19], [9] or repetitive tasks (such as pick and place, painting or welding) [21]. The production lines or tasks where they work often run for long periods of time, which means that both actuators and sensors are prone to failures. Sensor failures often result in noise, loss of information, skew, shunting, etc. [10]. The detection and diagnosis of these faults is crucial to avoid and prevent accidents, guaranteeing the safety of the processes being carried out, as well as of the people around them[18].

Traditional fault detection methods, based on threshold analysis and predefined rules, are often insufficient to detect subtle anomalies, incipient faults or intermittent faults [22], [16]. These methods can generate false positives or negatives, which can lead to unnecessary shutdowns or failure to detect critical faults.

https://doi.org/10.61728/AE20255329

Autoencoders are unsupervised artificial neural networks that learn representations of data by dimensionality reduction, encode and decode input data, capturing latent representations of the information [2], they are also capable of reconstructing missing signals using the remaining signals [8]. Their ability to model the normal distribution of data allows them to detect deviations and anomalies that may indicate the presence of sensor faults, in some cases they can eliminate noise [13], [1]. Some of the applications of autoencoders range from application in heavy mechanical equipment [22] or escalator bearing [12] [5], wind turbines [11], [23], hydraulic machinery [17] and maritime [7], improved Printed Circuit Board (PCB) designs [15] They are even used in nunclear and thermal power plants to detect single failures in multiple subsystems [14, 3] and in same way industrial robots [4].

This work proposes a sensor fault detection method for robotic manipulators using autoencoders. Unlike approaches based on Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN), autoencoders do not require labeled data sets with and without faults, thus avoiding data imbalance problems that hinder training. In addition, the differential evolution algorithm is employed to optimize the threshold that determines the presence of faults, improving the F1-score, a particularly useful metric for unbalanced data sets. The effectiveness of the method is validated by detecting faults in position sensors on a 5-degree-of-freedom robotic manipulator during trajectory tracking tasks.

The article is structured as follows: Section 2 describes the differential evolution algorithm; Section 3 details the autoencoder architecture and threshold selection for fault detection; Section 4 presents the experimental results and Section 5 concludes the work.

2 Differential Evolution

Differential Evolution (DE) is a stochastic, population-based algorithm designed for global optimization. Each individual x_i in the population represents a candidate solution, where $i = 1, 2, \ldots, N$ and N is the population size. The algorithm improves solutions across generations through three key operations: **mutation**, **crossover**, and **selection**.

Mutation for each individual i, a mutant vector v_i is generated using:

$$v_i = x_{r1} + F(x_{r2} - x_{r3}), \tag{1}$$

where x_{r1}, x_{r2}, x_{r3} are randomly selected individuals ($r1, r2, r3 \in [1, N]$) distinct from each other and from i. The parameter $F \in [0, 2]$, known as the *amplification factor*, scales the difference between x_{r2} and x_{r3}.

Crossover a trial vector u_i is created by combining components of the mutant vector v_i and the original individual x_i:

$$u_{ij} = \begin{cases} v_{ij} & \text{if } r_j \leq CR \text{ or } j = r_i, \\ x_{ij} & \text{if } r_j > CR \text{ and } j \neq r_i, \end{cases} \tag{2}$$

where $CR \in [0, 1]$ is the *crossover constant*, $j = 1, 2, \ldots, D$ (with D as the problem dimension), and r_j, r_i are random numbers.

Selection the trial vector u_i is compared to the original vector x_i using an objective function f. If $f(u_i)$ performs better than $f(x_i)$, u_i replaces x_i; otherwise, x_i is retained for the next generation.
For further details on DE and its variants, refer to [6], [20].

3 Methodology

Autoencoders are trained similarly to traditional Neural Networks, where the input and output are identical. The network learns to reconstruct the input as accurately as possible. The input is defined as $X = \{x_1, x_2, x_3, \ldots, x_d\}$, where d is the vector size, and the output is $\hat{X} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \ldots, \hat{x}_d\}$.

The proposed autoencoder architecture, illustrated in Fig. 1, consists of an encoder and a decoder. The encoder comprises four layers, starting with 128 neurons and halving the number in each subsequent layer. The decoder also has four layers, beginning with 16 neurons and doubling the number in each layer, followed by the output layer. All neurons use hyperbolic tangent activation functions, except for the output layer.

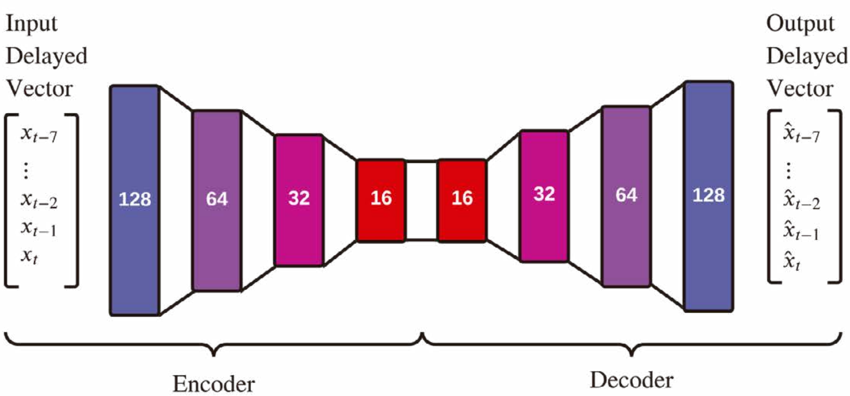


Fig. 1: Arquitecture of the autoencoder with size $d = 8$ for the delayed vector.

For sensor fault detection in a 5-degree-of-freedom (DOF) robotic manipulator, the input vector for the autoencoder is constructed using the current joint measurement at time t and the $(d-1)$ previous joint measurements. The input vector for the i-th joint is defined as:

$$X_i = \{q_i^t, q_i^{t-1}, q_i^{t-2}, \ldots, q_i^{t-(d-1)}\}, \tag{3}$$

where $i = 1, 2, 3, 4, 5$ represents each joint. A separate autoencoder is used for

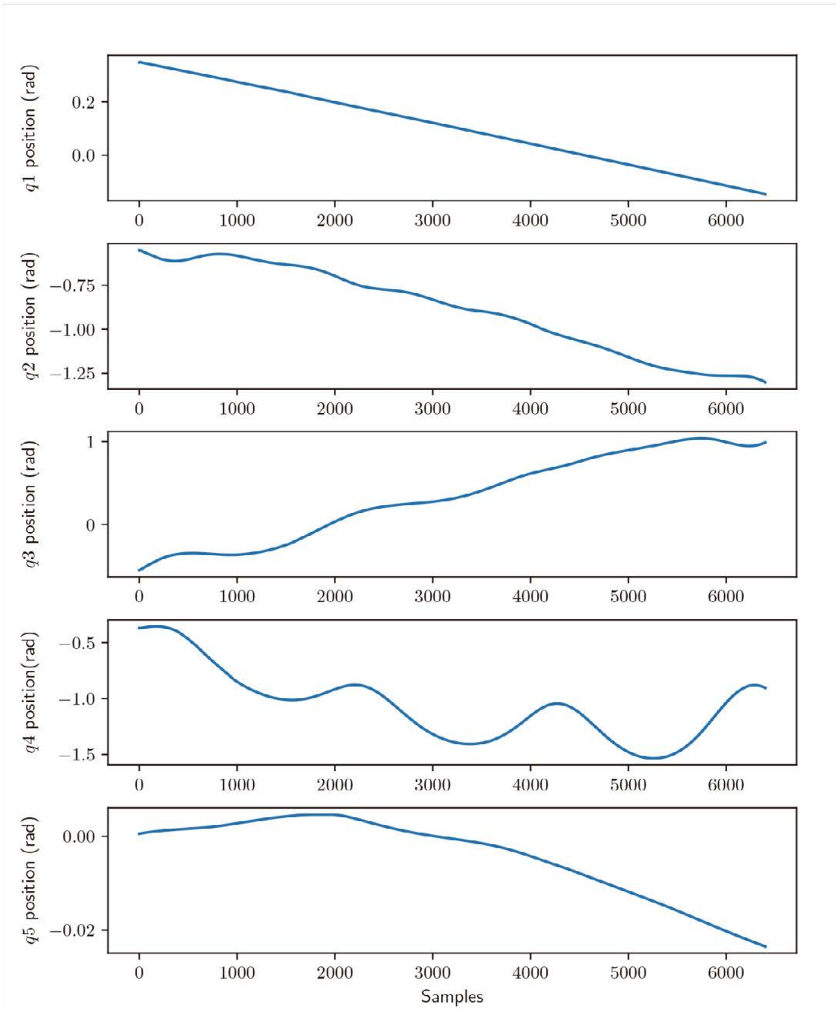


Fig. 2: The sensor measurements for each joint in the 5-degree-of-freedom (DOF) manipulator are represented as q_i, where q_i denotes the position measurement of the i-th joint ($i = 1, 2, 3, 4, 5$).

each joint, all sharing the same architecture as shown in Fig. 1. During a successful trajectory tracking task, a dataset was created containing measurements from all 5 joint position sensors are illustrated in Fig. 2.

The autoencoders were trained using 6276 samples collected over sixty seconds of trajectory tracking, as shown in Fig. 2. The Mean Absolute Error (MAE) was used as the loss function, and the Adam optimizer was employed for training over 50 epochs. Several versions of the autoencoder were trained with varying input sizes ($d = 4, 8, 16, 32$) to determine the optimal configuration. The loss function values during training for each epoch are presented in Fig. 3.

To identify the best input size d, four configurations ($d = 4, 8, 16, 32$) were tested, aiming to determine which size yields the best performance.

Fault detection is performed by calculating the reconstruction error e between the input X and the output $\hat{X}$:

$$e = \sum_{j=1}^{d} |x_j - \hat{x}_j|. \tag{4}$$

A fault is detected when $e > \alpha$, where α is a predefined threshold. This threshold determines whether the reconstruction error e is classified as a fault or not.

To select the optimal threshold value α, the Differential Evolution (DE) algorithm is employed. The objective function f for this optimization is the F1-score, which is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{5}$$

where *precision* (positive predictive value) and *recall* (true positive rate) are calculated as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

Here, TP represents the number of correctly identified fault samples, FP denotes the number of non-fault samples incorrectly classified as faults, and FN corresponds to the number of fault samples incorrectly classified as non-faults.

The F1-score ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates a complete failure in either precision or recall. The optimization problem is formulated as:

$$\arg\max_{\alpha} f(\alpha) = \text{F1-score}. \tag{7}$$

To solve this optimization problem, the DE algorithm is applied with the following parameters: 100 iterations, a population size of $N = 30$, an amplification factor $F = 0.5$, and a crossover constant $CR = 0.3$.

4 Results

To test the autoencoders three different types of faults are introduced via software (disconnection, loss of accuracy, and freeze). For $q1$ a freeze fault from the instant 1000 to 1500, in $q2$ the data goes to zero from instant 1500 to 2000, $q3$ has a freeze from instant 4000 to 4500, $q4$ and $q5$ have noise (loss of accuracy) from instant 5500 to 6000 as is shown in Fig. 3.

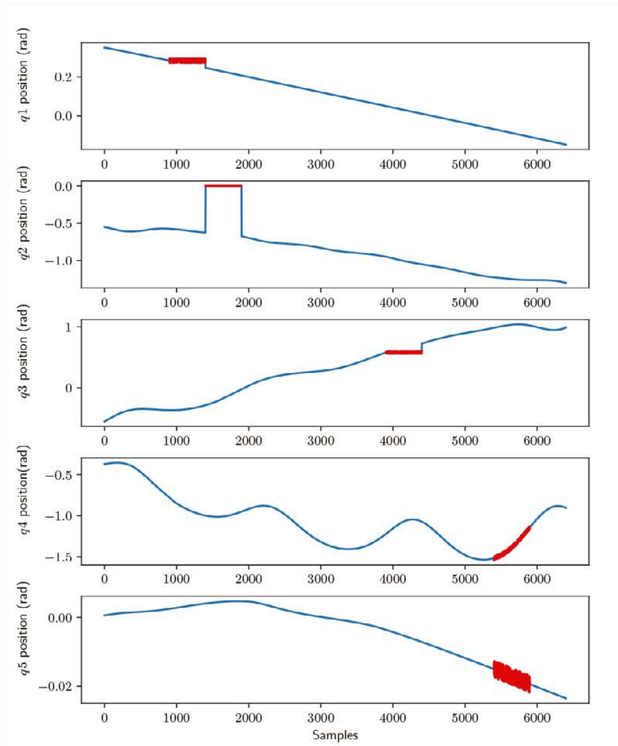
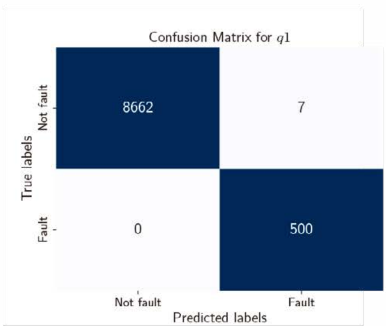


Fig. 3: Signals with fault

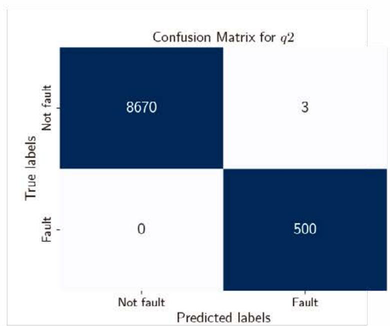
The results of applying DE and varying the size of the delayed vector to select the optimal threshold and size of the input are shown in Table 1 getting that for $q1$ the size selected is eight and the best threshold α is 0.0136 to get a precision, recall and F1-score of 0.9861, 1 and 0.993 respectively for the classification of the fault which can be confirmed with the confusion matrix presented in Fig. 4a. For the joint $q2$ the threshold α value is 0.2229 and the input size is equal to 4, obtaining metrics of precision = 0.994, recall = 1, and F1-score of 0.997 according to Fig. 4b. For these two joints as the recall shows all the faults were correctly detected.

Table 1: Results of the threshold α obtained with DE and their F1-score. Bold letters indicate the best results.

		Delayed Vector Size d			
		4	8	16	32
$q1$	α	0.0234	**0.0136**	0.0918	0.1751
	F1-score	0.570	**0.9930**	0.7725	0.7751
$q2$	α	**0.2229**	0.2229	0.2229	0.2409
	F1-score	**0.9970**	0.9930	0.9852	0.9354
$q3$	α	0.0367	0.0830	**0.1446**	0.3692
	F1-score	0.7339	0.3153	**0.9183**	0.3133
$q4$	α	0.0440	0.0622	**0.1307**	0.2377
	F1-score	0.5601	0.9630	**0.9810**	0.9701
$q5$	α	0.0014	**0.0037**	0.0109	0.0279
	F1-score	0.9744	**0.9929**	0.9890	0.9777



(a)

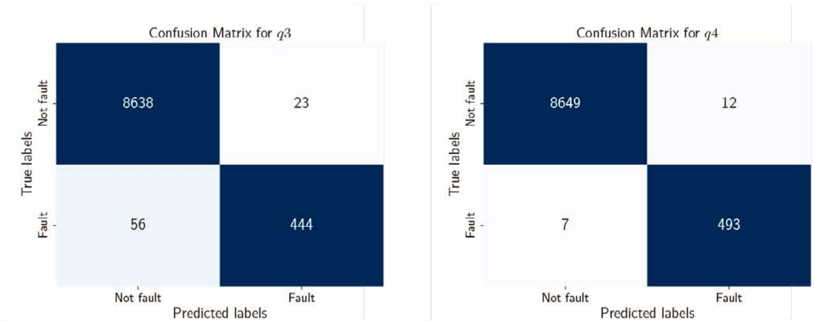


(b)

Fig. 4: Confusion Matrix for $q1$ and $q2$

For the signals of $q3$ and $q4$ the size of the input chosen is 16 having an F1-score of 0.918 and 0.981 with a threshold of 0.1446 and 0.1307 respectively. As is shown in Table 1 $q3$ got the lowest scores among all the joints, even when de F1-score is up to 0.9 the recall is 0.888 according to results shown in the Fig. 5a which could be due to the fact that it is the signal that does not have a noticeable trend or periodicity, and the size of the input could lead to a larger error without actually having a failure. In the case of $q4$ there is no such a problem as with $q3$ just that not all the faults were detected as faults as can be seen in Fig. 5b.

For $q5$ the selected threshold value is 0.0037 being the lowest of all the joints. This threshold gets the metrics of recall an precision of 0.992 and 0.993 respectively according to the confusion matrix in Fig. 6,which represents that less than 1% of the samples with failures were not detected, at the same time less than 1% of the samples without failure were considered failures,



(a) (b)

Fig. 5: Confusion matrix for $q3$ and $q4$

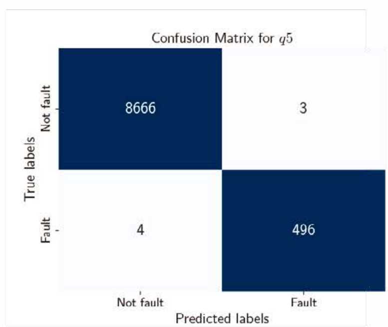


Fig. 6: Confusion matrix for $q5$

5 Conclusion

The fault detection is a significant topic of interest in broad real-world applications. Neural networks prove to be a powerful tool able to obtain characteristics of the observed data to recognize healthy samples and samples with fault. The proposed approach implements a neural network architecture called autoencoder to get a model that can get the characteristics of the signal and with the part of the decoder reconstruct it with minimal error. In this case to reconstruct the signal of five joints rotation of a robot manipulator. It is noticeable all the autoencoders have the same quantity of inner layers with different input and output layers, since some sizes could not have enough values to be representative of the signal. With the addition of the optimal threshold selection made by the Differential Evolution algorithm for all the joints and sizes of the delayed vector lead to achieving the distinction of faults with F1-scores of at least 90% for all the joints of the robot.

References

1. Abboush, M., Knieke, C., Rausch, A.: Machine learning-assisted test records analysis during the real-time validation of automotive software systems based on hil simulation (2025)
2. Aggarwal, C.C., Aggarwal, C.C.: An introduction to outlier analysis. Springer (2017)
3. Ai, X., Liu, Y., Shan, L., Gao, J., Zhang, W.: A robust diagnosis method specifically for similar faults in nuclear power plant multi-systems based on data segmentation and stacked convolutional autoencoders. Annals of Nuclear Energy **213**, 111178 (2025)
4. Chen, T., Liu, X., Xia, B., Wang, W., Lai, Y.: Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. IEEE Access **8**, 47072–47081 (2020)
5. Chen, Z., Li, W.: Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. IEEE Transactions on instrumentation and measurement **66**(7), 1693–1702 (2017)
6. Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., Xu, J.: An improved differential evolution algorithm and its application in optimization problem. Soft Computing **25**, 5277–5298 (2021)
7. Han, P., Ellefsen, A.L., Li, G., Holmeset, F.T., Zhang, H.: Fault detection with lstm-based variational autoencoder for maritime components. IEEE Sensors Journal **21**(19), 21903–21912 (2021)
8. He, Y., Du, L., Chen, W., Chen, F., Xu, Y.: Missing signal reconstruction and aileron fault detection via generative adversarial learning. Neurocomputing **620**, 129205 (2025)
9. Isermann, R.: Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems. Springer Science & Business Media (2011)
10. Jana, D., Patil, J., Herkal, S., Nagarajaiah, S., Duenas-Osorio, L.: Cnn and convolutional autoencoder (cae) based real-time sensor fault detection, localization, and correction. Mechanical Systems and Signal Processing **169**, 108723 (2022)
11. Jiang, G., Xie, P., He, H., Yan, J.: Wind turbine fault detection using a denoising autoencoder with temporal information. IEEE/Asme transactions on mechatronics **23**(1), 89–100 (2017)
12. Jiang, X., Zhang, Z., Yuan, H., He, J., Tong, Y.: Dac-bilstm model for accurate diagnosis of bearing faults in escalator principal drive systems. Processes **13**(1), 202 (2025)
13. Kang, G., Gao, S., Yu, L., Zhang, D.: Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning. IEEE Transactions on Instrumentation and Measurement **68**(8), 2679–2690 (2018)
14. Khalid, S., Azad, M.M., Kim, H.S.: A generalized autonomous power plant fault detection model using deep feature extraction and ensemble machine learning. Mathematics **13**(3), 342 (2025)
15. Kim, J., Ko, J., Choi, H., Kim, H.: Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. Sensors **21**(15), 4968 (2021)
16. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE journal **37**(2), 233–243 (1991)

17. Mallak, A., Fathi, M.: Sensor and component fault detection and diagnosis for hydraulic machinery integrating lstm autoencoder detector and diagnostic classifiers. Sensors **21**(2), 433 (2021)
18. Park, P., Marco, P.D., Shin, H., Bang, J.: Fault detection and diagnosis using combined autoencoder and long short-term memory network. Sensors **19**(21), 4612 (2019)
19. Siciliano, B., Khatib, O., Kröger, T.: Springer handbook of robotics, vol. 200. Springer (2008)
20. Simon, D.: Evolutionary optimization algorithms. John Wiley & Sons (2013)
21. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot modeling and control. John Wiley & Sons (2020)
22. Tsai, D.M., Jen, P.H.: Autoencoder-based anomaly detection for surface defect inspection. Advanced Engineering Informatics **48**, 101272 (2021)
23. Wu, X., Jiang, G., Wang, X., Xie, P., Li, X.: A multi-level-denoising autoencoder approach for wind turbine fault detection. Ieee Access **7**, 59376–59387 (2019)