

## APLICAÇÃO DO CLASSIFICADOR NAIVE BAYES PARA IDENTIFICAÇÃO DE FALHAS DE UM MANIPULADOR ROBÓTICO

**Bruna dos Santos Lazéra Wanke, brunawanke@poli.ufrj.br<sup>1</sup>**  
**Vivian Oliveira Costa, vivian.oliveira993@poli.ufrj.br<sup>1</sup>**  
**Aloísio Carlos de Pina, aloisiopina@dcc.ufrj.br<sup>2</sup>**  
**Armando Carlos de Pina Filho, armando@poli.ufrj.br<sup>3</sup>**

<sup>1</sup>Universidade Federal do Rio de Janeiro, Escola Politécnica.

<sup>2</sup>Universidade Federal do Rio de Janeiro, Departamento de Ciência da Computação.

<sup>3</sup>Universidade Federal do Rio de Janeiro, Escola Politécnica, Programa de Engenharia Urbana.

**Resumo:** Ao executar tarefas de manipulação, um robô pode apresentar falhas, principalmente em sua movimentação. Se o sistema de controle detecta uma falha no funcionamento, uma função de diagnóstico é utilizada para confirmar e caracterizar a falha. Baseado na descrição da falha, um plano de recuperação pode ser implementado. Esse diagnóstico corresponde a um processo de decisão que requer um modelo sofisticado da tarefa, do sistema e do ambiente. Construir tal modelo sofisticado não é um problema simples. Até mesmo especialistas no domínio têm dificuldade em especificar o mapeamento necessário entre os sensores disponíveis e as classificações de falhas. Sendo assim, a aplicação de métodos de inteligência computacional pode representar uma solução para identificação dessas falhas, visto que tais métodos têm sido utilizados com sucesso em diversos problemas onde as relações entre suas variáveis não podem ser facilmente definidas pelo ser humano. Logo, o objetivo deste trabalho é programar um método de inteligência computacional e aplicá-lo no diagnóstico de falhas na movimentação de um manipulador robótico, dadas as forças e torques medidos após a detecção das falhas. Para isso, foi usado um conjunto de dados fornecido por Luis S. Lopes e Luis M. Camarinha-Matos, da Universidade Nova de Lisboa, em Portugal. Esse conjunto de dados foi pré-processado para identificar e eliminar as variáveis menos relevantes para o processo de aprendizado, o que reduziu o número de variáveis em mais de 50%. O classificador Naive Bayes, algoritmo escolhido, foi implementado usando a linguagem de programação Python, e em seguida, foram realizados os experimentos usando o método da validação cruzada, a fim de dar suporte estatístico para avaliação correta dos resultados, permitindo assim a determinação do modelo mais adequado para aplicação proposta.

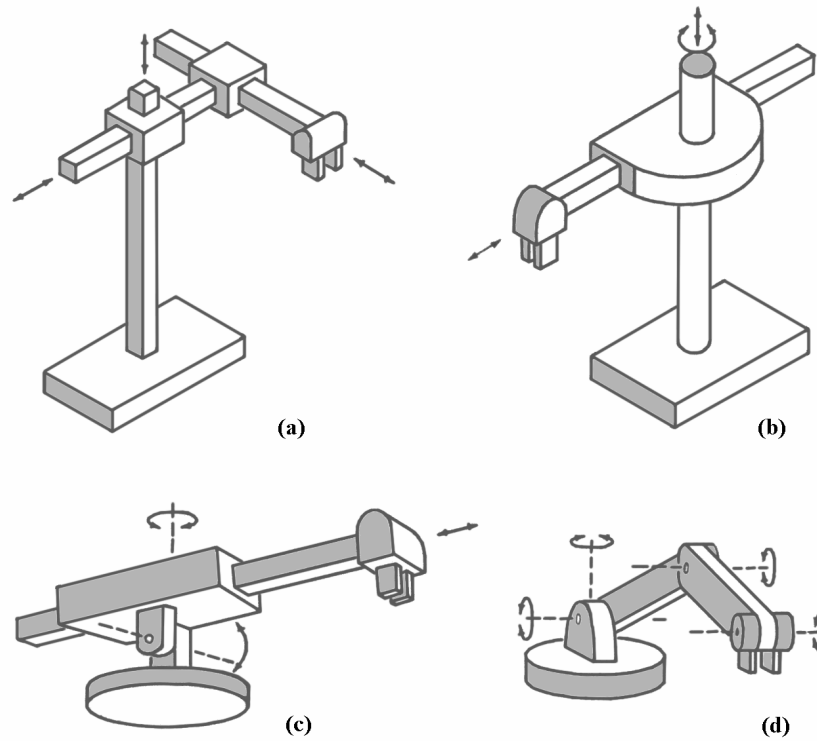
**Palavras-chave:** inteligência computacional, robótica, identificação de falhas.

### 1. INTRODUÇÃO - MANIPULADORES ROBÓTICOS

Manipuladores robóticos são mecanismos multifuncionais e programáveis projetados para mover materiais, partes ou ferramentas através de movimentos variáveis, possibilitando assim uma variedade de tarefas. Um sistema robótico industrial inclui não somente robôs, mas também quaisquer mecanismos e/ou sensores para que o robô possa realizar suas tarefas, bem como para monitoração e controle das mesmas (OSHA Technical Manual, <http://www.osha.gov>). Um manipulador robótico é constituído de dois elementos básicos: juntas e elos (*links*). Em geral, as juntas podem ser: rotacionais, quando executam rotações em torno de um eixo; prismáticas, quando executam deslizamentos num eixo; e esféricas, quando executam rotações em torno de todos os eixos.

Cada junta representa um grau de liberdade (com exceção das juntas esféricas). As juntas podem envolver movimento linear ou movimento rotacional entre os elos adjacentes. Os elos são considerados como estruturas rígidas ligadas por juntas. Manipuladores robóticos estão disponíveis comercialmente em uma diversidade de tamanhos, formas e configurações. A maioria dos robôs possui uma das quatro configurações básicas (Fig. 1):

- a) configuração de coordenadas cartesianas;
- b) configuração cilíndrica;
- c) configuração polar ou esférica;
- d) configuração articulada.



**Figura 1. Configurações básicas dos manipuladores robóticos (Groover, 1989; Pina *et al.*, 2004).**

O manipulador robótico do tipo articulado é o mais utilizado, principalmente para realização de tarefas industriais, recebendo muitas vezes a denominação de robôs ou manipuladores industriais. A Figura 2 apresenta um exemplo de manipulador robótico articulado, para o qual os métodos aplicados nesse trabalho foram implementados.



**Figura 2. Robô manipulador HP50-80 da Motoman (<http://www.motoman.com>).**

Cada tipo de robô possui um volume de trabalho característico, que é o conjunto de todos os pontos que o efetuador consegue alcançar. Logicamente, esse volume varia de acordo com o tamanho dos elos e do alcance das juntas, as quais são acionadas por atuadores que podem ser hidráulicos, pneumáticos ou elétricos.

Os primeiros são pistões ou motores hidráulicos que utilizam como fluido um óleo bastante viscoso e fornecem grande precisão aos robôs, assim como grande capacidade de carga. Os atuadores pneumáticos são similares aos hidráulicos, porém utilizam como fluido o ar que por ser bastante compressível a altas pressões dão aos robôs uma baixa precisão. Os atuadores elétricos utilizam motores elétricos, fornecendo aos robôs uma boa precisão e com uma

vantagem em relação aos hidráulicos: eles não correm o risco de contaminar a área de trabalho. Porém têm a desvantagem de ter menor capacidade de carga e não poderem ser usados em ambientes com risco de incêndio.

Há três tipos básicos de controle utilizados nos robôs manipuladores. O primeiro é chamado *bang-bang*, que tem como objetivo levar o efetuador de um ponto a outro, sendo que esses pontos são dados pelas características mecânicas dos atuadores. Esse tipo de controle opera em malha aberta e não dá ao robô muita precisão. O segundo tipo é o de trajetória ponto a ponto, que também tem como objetivo levar o efetuador de um ponto inicial até um final, podendo ou não passar por pontos intermediários, não importando a trajetória que ele fará. Esse tipo de controle opera em malha fechada com um sensor de posição. O terceiro e último tipo de controle é o de trajetória contínua e faz com que o robô vá de um ponto a outro seguindo uma trajetória pré-programada. Para isso, ele opera em malha fechada com sensores de posição e velocidade.

Mais detalhes sobre robôs manipuladores e robótica em geral podem ser vistos em Rivin (1988), Groover *et al.* (1989), Pina Filho e Dutra (2004), e Campos *et al.* (2010).

A partir da escolha dos atuadores e do controle, para que o manipulador possa realizar seus movimentos serão fornecidos torques e forças às juntas e elos, e as medidas de torques e forças após a detecção de uma falha é que serão utilizadas para diagnóstico da mesma.

Dentre as falhas a serem analisadas e que poderão ser detectadas pelo método proposto nesse trabalho têm-se:

Lp1 - falhas na aproximação da posição de manipulação, para pegar uma peça/objeto;

Lp2 - falhas na transferência de uma peça/objeto;

Lp3 - falhas na posição da peça/objeto após a transferência;

Lp4 - falhas na aproximação da posição de manipulação, para soltar a peça/objeto;

Lp5 - falhas na movimentação do manipulador com a peça/objeto.

O diagnóstico e identificação de cada uma dessas falhas corresponde a um processo de decisão bastante complexo, e a aplicação de métodos de inteligência computacional pode representar uma solução adequada para o problema, visto que tais métodos têm sido utilizados com sucesso em diversos problemas onde as relações entre suas variáveis não podem ser facilmente definidas pelo ser humano. Alguns trabalhos interessantes para embasamento teórico incluem: Mitchel (1997), Mayers (2004) e Charette (2007). Logo, o objetivo deste trabalho é programar um método de inteligência computacional, mais especificamente o Classificador Naive Bayes, e aplicá-lo no diagnóstico de falhas na movimentação de um manipulador robótico, dadas as forças e torques medidos após a detecção das falhas, como será visto nas próximas seções.

## 2. CLASSIFICADOR NAIVE BAYES

Um programa de aprendizado de máquinas (Mitchell, 1997) é um programa capaz de aprender com a experiência, a partir de exemplos. Um exemplo é uma combinação de valores das variáveis do problema.

Em um problema de classificação, a classe representa a informação que se deseja aprender. As demais variáveis do problema, chamadas atributos, são variáveis auxiliares cujos valores podem ser determinados e podem ajudar a prever a classe. Portanto, cada exemplo é uma combinação de valores dos atributos para os quais se sabe a classe associada. Um classificador recebe como entrada um conjunto de exemplos que é usado no aprendizado e então torna-se capaz de responder a classe para qualquer combinação de valores de atributos.

Redes Bayesianas são algoritmos de Aprendizado de Máquinas capazes de fornecer predições associadas à valores de probabilidades. Dentre outras vantagens, elas permitem o tratamento de fenômenos associados ao tempo, considerando a dependência temporal dos dados.

O estudo de Redes Bayesianas é complexo e exige um conhecimento razoável sobre probabilidade (mais informações em Pearl, 2000). Entretanto, o classificador Naive Bayes é uma simplificação conceitualmente fácil de ser entendida e de ser aplicada.

O Classificador Naive Bayes (Fig. 3) aprende a partir dos dados de treinamento a probabilidade condicional de cada atributo dado o valor da classe.

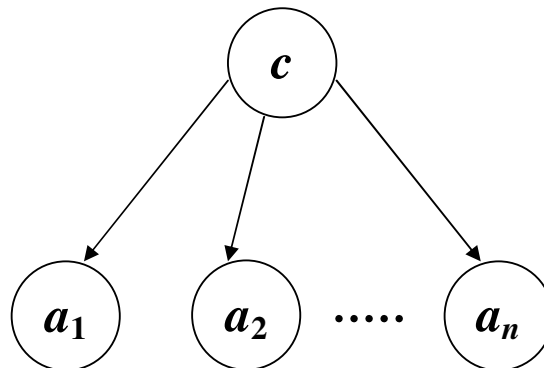


Figura 3. Classificador Naive Bayes (criada pelos Autores).

Para entender o método, são necessárias algumas noções básicas sobre probabilidades.

Probabilidade condicional, é a probabilidade que um evento aconteça, dado que outro aconteceu. A probabilidade que o evento A aconteça, dado que o evento B aconteceu é representada por uma barra (veja Eq. (1)).

$$P(A | B) \tag{1}$$

Quando dois eventos são independentes dado um terceiro, a probabilidade de acontecerem ao mesmo tempo é igual à multiplicação de suas probabilidades, dado o terceiro evento, como mostra a Eq. (2).

$$P(A, B | C) = P(A | C) \cdot P(B | C) \tag{2}$$

A Regra de Bayes (Bayes e Price, 1763) diz que:

$$P(A | B) = \frac{P(A, B)}{P(B)} \tag{3}$$

Da mesma forma:

$$P(B | A) = \frac{P(A, B)}{P(A)} \tag{4}$$

Considerando as Eq. (3) e (4), tem-se que:

$$P(A | B) \cdot P(B) = P(B | A) \cdot P(A) \tag{5}$$

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)} \tag{6}$$

A classificação é feita aplicando-se a regra de Bayes para calcular a probabilidade de cada classe, dados os valores dos atributos no caso-teste, e escolhendo a que resulta em maior probabilidade:

$$c_{casoteste} = \arg \max_c P(c | a_1, a_2, \dots, a_n) \tag{7}$$

Usando a Regra de Bayes:

$$c_{casoteste} = \arg \max_c \frac{P(a_1, a_2, \dots, a_n | c) \cdot P(c)}{P(a_1, a_2, \dots, a_n)} \tag{8}$$

Como o denominador é independente de  $c$ :

$$c_{casoteste} = \arg \max_c P(a_1, a_2, \dots, a_n | c) \cdot P(c) \tag{9}$$

Esse cálculo só é possível devido à forte suposição de que todos os atributos são independentes dada a classe, ou seja, dada uma classe, a probabilidade de um atributo não depende da probabilidade de nenhum outro.

Embora essa suposição raramente seja verdade, o classificador Naive Bayes alcança desempenho surpreendente.

Portanto, assumindo a independência dos atributos dada a classe:

$$c_{casoteste} = \arg \max_c P(a_1, a_2, \dots, a_n | c) \cdot P(c) \tag{10}$$

$$c_{casoteste} = \arg \max_c P(c) \cdot P(a_1 | c) \cdot P(a_2 | c) \cdot \dots \cdot P(a_n | c) \tag{11}$$

$$c_{casoteste} = \arg \max_c P(c) \cdot \prod_i P(a_i | c) \tag{12}$$

O cálculo das probabilidades é feito pela simples contagem de exemplos no conjunto de treinamento:

$$P(a_i | c) = \frac{P(a_i, c)}{P(c)} \tag{13}$$

$$P(c) = \frac{|S_c|}{|S|} \tag{14}$$

$$P(a_i, c) = \frac{|S_{a_i, c}|}{|S|} \tag{15}$$

$$P(a_i | c) = \frac{P(a_i, c)}{P(c)} = \frac{|S_{a_i, c}|}{|S|} \cdot \frac{|S|}{|S_c|} = \frac{|S_{a_i, c}|}{|S_c|} \tag{16}$$

### 3. APLICAÇÃO DO ALGORITMO E RESULTADOS

Os dados usados nos experimentos foram fornecidos por Luis S. Lopes e Luis M. Camarinha-Matos, da Universidade Nova de Lisboa, em Portugal (Camarinha-Matos *et al.*, 1996).

Os conjuntos de dados (Tab. 1) contêm medidas de força e torque em um manipulador robótico após a detecção de falha. Cada falha é caracterizada por 15 amostras de 3 forças e 3 torques coletadas em intervalos de tempo regulares, começando imediatamente após a detecção da falha. A janela de observação para cada instância de falha foi de 315ms.

**Tabela 1. Conjuntos de dados.**

Conjunto de Dados	Número de Exemplos	Classes de Falhas	Exemplos/Classe
Lp1	88	collision	17
		fr_collision	16
		normal	21
		obstruction	34
Lp2	47	back_col	7
		front_col	6
		left_col	9
		normal	20
		right_col	5
Lp3	47	lost	3
		moved	15
		ok	20
		slightly_moved	9
Lp4	117	collision	72
		normal	24
		obstruction	21
Lp5	164	bottom_collision	26
		bottom_obstruction	21
		collision_in_part	47
		collision_in_tool	26
		normal	44

Antes da avaliação da precisão do classificador Naive Bayes aplicado ao problema, foi feito o pré-processamento dos dados a fim de adequá-los ao algoritmo e para determinar a quantidade de amostras necessárias.

Os intervalos de valores de forças e torques foram discretizados para que os atributos pudessem ser tratados como simbólicos, em vez de numéricos, permitindo assim a aplicação direta do algoritmo, que calcula probabilidades através da contagem de valores.

Através da aplicação do algoritmo de seleção de atributos Forward Search (Guetlein *et al.*, 2009) verificou-se que é possível reduzir o número de amostras de forças e torques usadas para apenas 7 sem degradação significativa dos resultados. Basicamente, o algoritmo Forward Search funciona incluindo iterativamente no conjunto de atributos necessários aquele que proporciona o maior aumento na precisão dos resultados, parando assim que nenhum dos atributos restantes satisfaça essa condição.

Dessa forma, cada exemplo do conjunto de dados é formado por 42 atributos (7 amostras de 6 forças/torques) e a correspondente classe de falha.

Cada conjunto de dados é dividido em conjunto de treinamento, com o qual são calculados todos os valores de probabilidades necessários, e conjunto de teste, usado para avaliar a precisão do algoritmo (porcentagem de exemplos do conjunto de teste classificados corretamente pelo algoritmo).

Usar apenas um par de conjuntos de treinamento e de teste não fornece uma estimativa da precisão estatisticamente significativa. Usando validação cruzada (Dietterich, 1997), para construir cada par de conjuntos de treinamento e de teste, o conjunto de dados  $D$  é dividido aleatoriamente em  $n$  conjuntos disjuntos de igual tamanho (chamados partições),  $T_1, \dots, T_n$ . Então são conduzidas  $n$  rodadas. Em cada rodada, o conjunto de teste é  $T_i$  e o conjunto de treinamento é a união de todos os outros  $T_j, j \neq i$ . A precisão do algoritmo então é dada pela média das precisões das  $n$  rodadas. Nesta pesquisa foi usada validação cruzada com 10 partições.

Todos os códigos fonte foram escritos na linguagem de programação Python. Os experimentos foram realizados em um computador com CPU Intel Core i7, 8GB RAM, sob Microsoft Windows 7 Professional 64Bit.

**Tabela 2. Resultados experimentais.**

Conjunto de Dados	Precisão	Classes de Falhas	Precisão/Classe
Lp1	0,907	collision	0,708
		fr_collision	0,882
		normal	1
		obstruction	0,962
Lp2	0,806	back_col	1
		front_col	0,6
		left_col	0,625
		normal	0,833
		right_col	1
Lp3	0,843	lost	1
		moved	0,778
		ok	0,909
		slightly_moved	0,75
Lp4	0,945	collision	0,985
		normal	0,923
		obstruction	0,833
Lp5	0,725	bottom_collision	0,808
		bottom_obstruction	0,81
		collision_in_part	0,634
		collision_in_tool	0,379
		normal	0,936

A Tabela 2 apresenta a precisão alcançada pelo algoritmo aplicado a cada um dos 5 conjuntos de dados. A última coluna da tabela mostra a precisão obtida na identificação de cada uma das classes de falhas. Pode-se notar que para algumas delas houve 100% de acerto. Para apenas uma classe de falha a porcentagem de acerto foi inferior a 60%: collision\_in\_tool, com precisão de cerca de 38%.

Considerando todos os dados, foi possível alcançar uma precisão média de cerca de 83% de acerto na identificação das falhas, dadas as medidas de força e de torque. O método da validação cruzada comprovou que os resultados apresentados pela implementação do algoritmo são significativos do ponto de vista estatístico.

#### 4. CONCLUSÃO

O objetivo deste trabalho foi programar um método de inteligência computacional e aplicá-lo no diagnóstico de falhas na movimentação de um manipulador robótico, dadas as forças e torques medidos após a detecção das falhas. Para isso, foi usado um conjunto de dados fornecido por Luis S. Lopes e Luis M. Camarinha-Matos, da Universidade Nova de Lisboa, em Portugal. Esse conjunto de dados foi pré-processado para identificar e eliminar as variáveis menos relevantes para o processo de aprendizado, o que reduziu o número de variáveis em mais de 50%. Devido a sua simplicidade e bom desempenho, o algoritmo Naive Bayes foi escolhido como algoritmo a ser implementado, usando a linguagem de programação Python, e em seguida, foram realizados os experimentos usando o método da validação cruzada, alcançando-se uma precisão de cerca de 83% de acerto na identificação das falhas, resultado que ainda pode ser melhorado modificando-se o programa para que os valores de forças e torques possam ser tratados como atributos numéricos, substituindo a discretização e contagem de valores pelo cálculo de densidades de probabilidades.

#### 5. AGRADECIMENTOS

Esta pesquisa foi parcialmente financiada pela CAPES e pela FAPERJ.

#### 6. REFERÊNCIAS

- Bayes, T. e Price, R., 1763, "An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S.", *Philosophical Transactions of the Royal Society of London* 53 (0): 370-418. doi:10.1098/rstl.1763.0053.
- Camarinha-Matos, L.M., Seabra Lopes, L. e Barata, J., 1996, "Integration and Learning in Supervision of Flexible Assembly Systems", *IEEE Transactions on Robotics and Automation*, 12 (2), pp. 202-219.
- Campos, F.O., Pina Filho, A.C.de e Pina, A.C.de, 2010, "Estudo e Modelagem Computacional de um Robô Aplicado em Processos de Usinagem", *Anais do IX Simpósio de Mecânica Computacional (SIMMEC)*, São João Del-Rei - MG, Brasil.
- Charette, R.N., 2007, "Learning from software failure", <http://www.spectrum.ieee.org>.
- Dietterich, T.G., 1997, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", *Neural Computation*, 10, pp. 1895-1924.
- Groover, M.P., Weiss, M., Nagel, R.N. e Odrey, N.G., 1989, "Robótica: Tecnologia e Programação", Trad.: D.M. Savatovsky, Rev.: R. Camacho, McGraw-Hill, São Paulo, Brasil.
- Guetein, M., Frank, E., Hall, M. e Karwath, A., 2009, "Large Scale Attribute Selection Using Wrappers", In: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 332-339.
- Mayers, G.J., 2004, "The art of software testing", New Hersey: John Wiley & Sons.
- Mitchel, T.M., 1997, "Machine Learning", Pittsburgh: McGraw Hill.
- Motoman, disponível em: <http://www.motoman.com>, acessado em fevereiro de 2014.
- OSHA Technical Manual, disponível em: <http://www.osha.gov>, "Industrial Robots and Robot System Safety", Seção IV, Capítulo 4, U.S. Government Printing Office, Washington, Estados Unidos.
- Pearl, J., 2000, "Causality: Models, Reasoning, and Inference", Cambridge University Press.
- Pina Filho, A.C.de e Dutra, M.S., 2004, "Performance Data for Evaluation of Industrial Robots", *Anais do III Congresso Nacional em Engenharia Mecânica (CONEM)*, Belém - PA, Brasil.
- Pina, A.A.de, Pina Filho, A.C.de e Pina, A.C.de, 2004, "Análise Cinemática e Simulação Computacional de um Manipulador Robótico", *Anais do III Congresso Nacional em Engenharia Mecânica (CONEM)*, Belém - PA, Brasil.
- Rivin, E.I., 1988, "Mechanical Design of Robots", McGraw-Hill Book Company.

#### 7. RESPONSABILIDADE AUTORMAL

Os autores são os únicos responsáveis pelo conteúdo deste trabalho.

## APPLICATION OF NAIVE BAYES CLASSIFIER FOR FAILURE IDENTIFICATION OF A MANIPULATOR ROBOT

**Bruna dos Santos Lazéra Wanke, brunawanke@poli.ufrj.br<sup>1</sup>**  
**Vivian Oliveira Costa, vivian.oliveira993@poli.ufrj.br<sup>1</sup>**  
**Aloísio Carlos de Pina, aloisiopina@dcc.ufrj.br<sup>2</sup>**  
**Armando Carlos de Pina Filho, armando@poli.ufrj.br<sup>3</sup>**

<sup>1</sup>Universidade Federal do Rio de Janeiro, Escola Politécnica.

<sup>2</sup>Universidade Federal do Rio de Janeiro, Departamento de Ciência da Computação.

<sup>3</sup>Universidade Federal do Rio de Janeiro, Escola Politécnica, Programa de Engenharia Urbana.

**Abstract.** *When performing manipulation tasks, a robot may fail, especially in its handling. If the control system detects a malfunction, a diagnostic function is used to confirm and characterize the failure. Based on the description of the failure, a recovery plan can be implemented. This diagnostic corresponds to a decision process that requires a sophisticated model of the task, the system and the environment. Build such a sophisticated model is not a simple problem. Even experts in the field have difficulty specifying the necessary mapping between the available sensors and ratings failures. Thus, the application of artificial intelligence methods may represent a solution for the identification of these flaws, since such methods have been used successfully in many problems where the relationship between its variables can not be easily defined by humans. Therefore, the aim of this work is to program a computational intelligence method and applying it in the diagnosis of faults in handling a robotic manipulator, given the forces and torques measured after detection of faults. For this, we used a dataset provided by Luis S. Lopes and Luis M. Camarinha-Matos, from New University of Lisbon in Portugal. This dataset was preprocessed to identify and eliminate less relevant variables for the learning process, which reduced the number of variables in more than 50%. The Naive Bayes classifier, selected algorithm, was implemented using the Python programming language, and next, experiments were performed using cross-validation method in order to provide statistical support for correct evaluation of the results, thereby allowing determination of the most appropriate model for the proposed application.*

**Keywords:** *computational intelligence, robotics, fault identification.*