

## BAT Algorithm aplicado à localização de robôs móveis<sup>1</sup>

### *BAT Algorithm applied to mobile robots localization*

Afonso da Fonseca Braga<sup>2</sup>

João Pedro Carvalho de Souza<sup>3</sup>

Fabício de Oliveira Coelho<sup>4</sup>

André Luís Marques Marcato<sup>5</sup>

DOI: <https://doi.org/10.34019/2179-3700.2019.v19.29911>

#### Resumo

A robótica assistiva está presente em diversas áreas de pesquisa do mundo atual. Trabalhos voltados para o aumento da produtividade e para o auxílio de pessoas com deficiência física são alguns exemplos de como a robótica pode facilitar e melhorar a qualidade de vida do ser humano. Com o desenvolvimento de aplicações remotas é possível controlar diferentes dispositivos sem a necessidade de estar presente no local de atuação. Este artigo tem como objetivo controlar um robô humanoide remotamente através do reconhecimento de sinais de eletromiografia, bem como localizá-lo em seu ambiente.

**Palavras-chave:** Robótica assistiva; Localização; Humanóide.

#### Abstract

The assistive robotics is present in many areas of research in the modern world. Projects aiming productivity and helping people with disabilities are examples how robotics can improve and make our lives better. With remote application development is possible to control a variety of gadgets without the need to be next to it. This paper aims to localize and control a humanoid robot remotely with help of arm muscles movements.

**Keywords:** Assistive robotics; Localization; Humanoid.

## 1 INTRODUÇÃO

### 1.1 NAO

O NAO (Figura 1) foi o primeiro robô humanoide desenvolvido pela SoftBank

<sup>1</sup> Trabalho premiado no Seminário de Iniciação Científica da UFJF em 2017.

<sup>2</sup> Graduando em Engenharia Elétrica – habilitação em Robótica e Automação. Bolsista do Programa Institucional de Bolsas de Iniciação Científica(PIBIC/CNPq) da Universidade Federal de Juiz de Fora (UFJF).

<sup>3</sup> Mestre em Engenharia Elétrica pela Universidade Federal de Juiz de Fora (UFJF).

<sup>4</sup> Mestre em Engenharia Elétrica pela Universidade Federal de Juiz de Fora (UFJF).

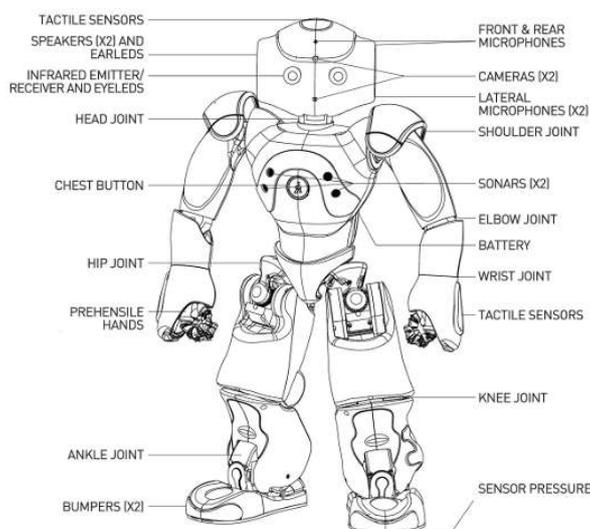
<sup>5</sup> Doutor em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO). Endereço profissional: Universidade Federal de Juiz de Fora, Faculdade de Engenharia, Departamento de Energia Elétrica. Av. Presidente Costa e Silva, S/N - Galpão 3 - Fac. Engenharia - PPEE - Sala 206 – CEP: 36.036-330 - Juiz de Fora, MG – Brasil



Robotics. Criado em 2006, com a finalidade de ser um robô de companhia e para fins educativos, o humanoide já ultrapassou 10 mil unidades vendidas no mundo inteiro.

Em sua quinta versão, o NAO conta com 25 graus de liberdade, wireless, duas câmeras, quatro microfones, sensores do tipo sonar, inércia, infravermelho, sensores táteis e sensores de pressão. Para comunicação, possui um sintetizador de voz, LEDs e dois alto-falantes. Com seu processador ATOM e uma memória de armazenamento interno de 8GB, são infinitas as possibilidades para a programação do humanoide.

Figura 1 - Robô NAO



Fonte: SoftBank Robotics. Disponível em: <[http://doc.aldebaran.com/2-4/family/nao\\_h25/index\\_h25.html#nao-h25](http://doc.aldebaran.com/2-4/family/nao_h25/index_h25.html#nao-h25)>. Acesso em: 25 abr. 2018.

## 2 METODOLOGIA

O presente trabalho foi dividido em três etapas: a primeira etapa é responsável por catalogar, processar e identificar sinais de eletromiografia que serão traduzidos para movimentos específicos no humanoide. Na segunda etapa foi desenvolvido um algoritmo capaz de controlar o NAO, fazendo com que ele se locomova e segure objetos de acordo com a força informada pelas leituras processadas na primeira etapa. A terceira etapa constituiu o desenvolvimento do BAT Algorithm para a localização do robô em um ambiente mapeado.

### 2.1 Etapa de reconhecimento dos movimentos da mão

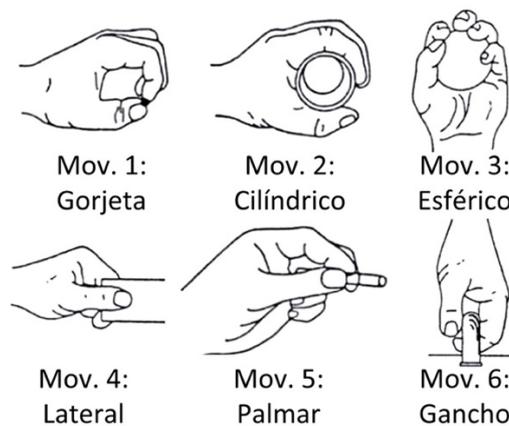
O desenvolvimento de classificadores é feito em partes como o sensoriamento e a

extração de características, a seleção de parâmetros, a classificação e a validação (THEODORIDIS; KOUTROUMBAS, 2008). Os dados que são submetidos a testes neste trabalho estão presentes em um banco de dados público disponibilizado em Sapsanis et al. (2013).

### 2.1.1 Sensoriamento e extração das características

O banco de dados é constituído com sinais referentes a seis tipos distintos de movimentos da mão humana. Esses movimentos, junto com a nomenclatura utilizada são apresentados na Figura 2.

Figura 2 - Movimentos captados.



Fonte: Adaptado de Sapsanis et al. (2013).

Cada um dos movimentos foi realizado por um voluntário durante seis segundos. Foram 30 repetições para cada um dos movimentos em um total de cinco voluntários.

De acordo com Sapsanis et al. (2013), a extração dessas características é realizada por eletrodos de EMG superficiais, os quais são fixados ao antebraço do voluntário, sobre os músculos flexor ulnar do carpo e extensor radial curto do carpo.

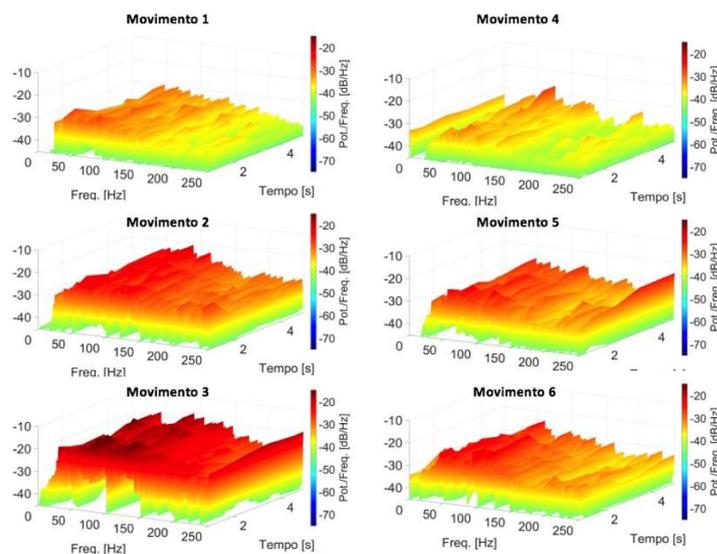
O pré-processamento dos dados realizado no presente artigo é executado sobre o software Matlab, com uma taxa de amostragem de 500 Hz. Dessa maneira, cada um dos movimentos apresenta 3 mil amostras. Os sinais são selecionados por um filtro que rejeita uma faixa com frequência central de 50 Hz e seus múltiplos inteiros até a frequência de 250 Hz. O objetivo dessa filtragem é que frequências relacionadas à rede elétrica local não sejam processadas. É retirado também a componente DC dos sinais. Os filtros

utilizados são do tipo Butterworth. Esse filtro não apresenta *ripple* na banda passante, logo não interfere nas características dos sinais a serem classificados.

### 2.1.2 Seleção de parâmetros

Para a seleção de parâmetros do trabalho proposto é feita a análise de características no domínio da frequência. Contudo, os sinais não são estacionários, tornando-se um empecilho para a análise simplificada da transformada de Fourier. Assim, visando contornar esse problema, a técnica de análise da densidade espectral de potência (DEP) aplicada em janelas ao longo do sinal é uma opção válida. Cada janela tem uma sobreposição, reduzindo o erro inerente à análise da transformada rápida de Fourier (fast Fourier transform, FFT) aplicada sobre um sinal não estacionário. A Figura 3 ilustra exemplos de espectrogramas aplicados a cada sinal referente a um evento de um movimento realizado por um voluntário. São utilizadas janelas de mil amostras com sobreposição de 500 amostras.

Figura 3 -Espectrogramas de um evento para cada movimento.



Com base nos espectrogramas é possível inferir propriedades visuais presentes em cada movimento, diferenciando-os. Essas características se repetem ao longo de movimentos de amostras distintas. Elas são relativas a faixas de potência ao longo do tempo, dada uma frequência de interesse.

### 2.1.3 Classificação

Como mencionado na seção 2.1.2, a análise da DEP em janelas permitiu a evidência de determinados padrões possíveis de distinguir. Como já dito, esses padrões são extraídos acumulando-se ao longo do tempo toda potência da frequência de interesse. Logo as frequências de interesse tornam-se as dimensões do problema. Com os parâmetros extraídos, aborda-se o desenvolvimento do classificador. A técnica de classificação por K vizinhos mais próximos com KD-Tree foi a adotada neste artigo.

A busca por K vizinhos mais próximos, ou Knn, é um método de classificação com base nas seguintes regras:

- Determina-se o número K de vizinhos. Preferencialmente ímpar.
- Calcula-se a distância do ponto aos K vizinhos do conjunto de treinamento.
- Retira-se os pontos com as K menores distâncias.
- O ponto de teste é classificado de acordo com o grupo de maior incidência sobre os K selecionados.

Devido à característica do algoritmo, a viabilidade do método é comprometida em razão de seu custo computacional elevado. Esse custo é associado com o tamanho dos conjuntos de treinamento e dimensionalidade do problema. Para contornar essa situação, diversas técnicas são empregadas, como as discutidas em Alippi et al. (2016). Uma técnica alternativa é associada com o Knn, denominada KD-Tree, explicitada em Bergmanne Tartakovski (2009).

## 2.2 Controlador para o Humanoide NAO

Para realizar a comunicação entre o robô e o computador, foi utilizado o ROS junto com os drivers disponíveis no próprio site do framework. Com esses drivers é possível fazer com que o robô se locomova informando coordenadas cartesianas ou cilíndricas. Infelizmente, para o controle da força na mão do NAO não havia nenhum driver específico que realizasse essa tarefa.

Uma solução encontrada para o problema foi criar um driver capaz de realizar essa tarefa. Desenvolvido na linguagem Python, em sua primeira versão, o chamado *NAOstiffnesstem* a função de alterar a quantidade de corrente fornecida para a junta da mão do robô. Com a variação de corrente é possível controlar a intensidade da força exercida pela mão ao se fechar.

Os valores da corrente fornecida estão entre 0 e 1, os quais significam 0 e 100%, respectivamente. Esses valores são enviados através de uma mensagem do tipo *float*. Após o envio, qualquer outro comando para movimentação da junta será executado com o valor de corrente fornecida.

Observando o artigo como um todo, percebe-se que a maior parte dos códigos foi desenvolvida em Matlab. Para uma melhor integração entre todos os códigos gerados, foi criada também uma função em Matlab. Nessa função, utilizando a capacidade do Matlab de se comunicar com o ROS, é possível enviar comandos diretos ao robô, utilizando os drivers anteriormente desenvolvidos.

### **2.3 Localização através do Algoritmo de Morcegos**

Um problema importante que deverá ser considerado no decorrer do trabalho é a localização do sistema robótico. De acordo com Thrun et al. (2005), para todas as tarefas complexas que o robô realize de maneira autônoma é necessário um método que encontre a posição dele no ambiente, assim a localização é tratada como um problema básico na robótica. Neste trabalho, o robô NAO poderá se deslocar pelo ambiente, fazendo com que o sistema proposto tenha a necessidade de se localizar.

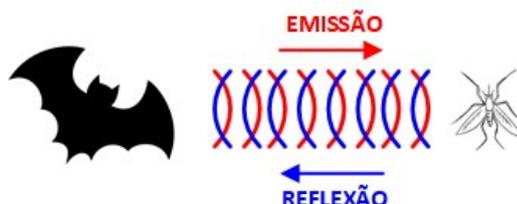
Entre os métodos que existem na literatura, destacam-se aqueles com base no problema de Monte Carlo ou Filtro de Partículas (FOX, 1999), que é bastante utilizado na robótica. Um dos fatores que explicam o sucesso do método é a abordagem probabilística, a qual torna possível a aplicação em um problema com grandes dimensões. Entretanto, como pontos negativos estão o alto custo computacional agregado ao método e a dificuldade de localizar o robô em ambientes com características ambíguas (FURTADO et al., 2015). Localizações a um menor custo computacional podem ser observadas a partir das técnicas inteligentes denominadas bioinspiradas, ou seja, em estudos envolvendo o comportamento de agentes da natureza será possível traçar técnicas e aplicá-las na robótica com propósitos de localização.

Técnicas como a Colônia de Formigas (DORIGO; DI CARO, 1999), a qual aborda o comportamento desses seres vivos na busca por alimentos; os Sistemas Imunológicos Artificiais (SIA) (THRUN et al., 2005), que consiste no estudo do sistema de defesa do organismo humano contra agentes causadores de doenças; e o Algoritmo de Morcegos (REKABY, 2013; YANG, 2010), que basicamente estuda o princípio de ecolocalização que os morcegos utilizam para se posicionar em relação às suas presas, são métodos

totalmente inspirados na natureza e que poderão ser aplicados na área da robótica.

O Algoritmo de Morcegos tem o objetivo de localizar o morcego em relação à sua presa a partir da emissão de ondas sonoras (BASTOS, 2016). Basicamente, o morcego é sensível à fração de tempo que a onda sonora leva para ir e voltar do morcego até a presa (Figura 4).

Figura 4 - Princípio da ecolocalização dos morcegos.



De acordo com Furtado et al. (2015), a técnica do Algoritmo de Morcegos é satisfatória na resolução de problemas denominados como programação inteira ou binária, ou seja, aqueles com regiões que apresentam soluções não convexas (mínimos locais). A solução dos problemas convexas é fortemente dependente das condições iniciais. Em contrapartida, os algoritmos evolucionários, como o de Morcegos, solucionam problemas sem a dependência dessas condições iniciais.

A técnica é baseada em um conjunto de  $\eta$  morcegos com velocidades ( $v_i$ ), taxa de pulso ( $r_i$ ), amplitude ( $A_i$ ), frequência ( $f_i$ ) e posições aleatórias ( $x_i$ ) pelo ambiente. O índice  $i$  é a assinatura de determinado morcego dentro do algoritmo. Quando o morcego detectar a presa, o pulso emitido é acelerado junto com a amplitude para que o local onde a presa se encontra não seja perdido (BASTOS, 2016).

O Algoritmo de Morcegos inicia a execução avaliando a solução encontrada para todos os  $\eta$  morcegos presentes, desse modo, é possível extrair desse processo qual é o melhor indivíduo naquela população.

Depois, os parâmetros dos morcegos são ajustados proporcionalmente à efetividade de terem encontrado presas no instante anterior. A velocidade também é ajustada para que os outros morcegos se desloquem para o local onde há o melhor morcego, ou seja, a posição do morcego que está diante da melhor presa. Ao final dessa etapa, é inicializado um processo chamado *busca local*, ou seja, alguns morcegos irão receber uma componente aleatória que atualizará sua posição, independentemente da velocidade. Em Yang (2010), foi proposto que essa aleatoriedade pudesse ocorrer através

de mutações não uniformes, apresentando bons resultados. Essa característica é extremamente satisfatória, pois irá gerar algumas soluções aleatórias que, por sua vez, ajudarão na eliminação de soluções locais.

Por fim, há o aumento da taxa de pulso que assume valores de 0 a 1 quando o tempo  $t$  do algoritmo tende ao infinito. Essa característica é favorável para que a busca local se intensifique com o decorrer do tempo. A amplitude também é atualizada ao longo do algoritmo decrescendo proporcionalmente a um parâmetro  $\alpha$ . Com uma amplitude alta, o Algoritmo de Morcegos é mais suscetível a aceitar novas soluções; caso contrário, para amplitudes baixas, o aceite de novas soluções ruins ficará mais restrito. Ressalta-se que é prudente aceitar algumas soluções ruins, pois é possível que elas possam melhorar no futuro e auxiliar a busca por respostas ótimas.

### 3 RESULTADOS

#### 3.1 Resultados da classificação dos movimentos

Nesta seção, será apresentada a etapa de validação. São abordadas análises utilizando o classificador K vizinho mais próximo com  $K = 1, 3$  e  $5$ . Nas simulações foram utilizadas, de maneira recursiva, 18 e 12 dados de cada movimento para treinamento e testes, respectivamente. Assim, os conjuntos de treinamento e testes resultaram um total de 108 e 72 amostras, cada um.

Os resultados das Tabelas 1, 2 e 3 correspondem às médias de acerto para cada um dos indivíduos submetidos ao teste. Ao todo são realizados 3 mil testes, variando aleatoriamente o conjunto de treinamento e teste.

Tabela 1 - Resultados obtidos para  $K = 1$

Indivíduo	M.1 (%)	M.2 (%)	M.3 (%)	M.4 (%)	M.5 (%)	M.6 (%)
1	100	99	97	82	86	98
2	100	79	93	98	69	70
3	98	88	91	92	87	87
4	100	89	87	99	85	82
5	91	96	97	81	89	98

Tabela 2 - Resultados obtidos para  $K = 3$

Indivíduo	M.1 (%)	M.2 (%)	M.3 (%)	M.4 (%)	M.5 (%)	M.6 (%)
1	70	95	94	70	63	96
2	84	82	94	90	53	42
3	87	81	88	90	73	80
4	52	82	78	99	55	61
5	89	94	95	74	81	95

Tabela 3 - Resultados obtidos para  $K = 5$

Indivíduo	M.1 (%)	M.2 (%)	M.3 (%)	M.4 (%)	M.5 (%)	M.6 (%)
1	64	92	94	75	60	94
2	84	77	93	89	58	33
3	87	77	85	90	68	75

4	41	83	76	99	53	53
5	90	92	94	72	78	91

Os resultados médios finais para cada um dos voluntários estão presentes na Tabela 4. É possível observar que os melhores resultados foram obtidos para  $K = 1$ , isto é, quando apenas um vizinho era levado em consideração nas comparações. A densidade de pontos do conjunto de treinamento é pequena devido ao número reduzido de amostras do banco de dados, e assim, o aumento do parâmetro  $K$  não implica melhoria nos resultados obtidos.

Tabela 4 - Índices de acerto médio

K =	1	3	5
Indivíduo 1	93%	81%	80%
Indivíduo 2	85%	74%	72%
Indivíduo 3	91%	83%	80%
Indivíduo 4	90%	71%	68%
Indivíduo 5	92%	89%	86%

### 3.2 Controlador para o humanoide NAO

Por fim, conseguimos programar o NAO através do Matlab e variar a força exercida pela sua mão ao se fechar. Foram enviados comandos para que o NAO estendesse a mão para segurar um objeto. Após a inserção do objeto em sua mão, ele a fecharia. As Figuras 5 a 8 ilustram os movimentos com um balão de ar. Os valores de corrente fornecidos para a mão do NAO foram 0; 30; 60 e 100%.

Figura 5 - Balão sem ser pressionado



Figura 6 - Balão sendo pressionado com 30% da força total



Figura 7 - Balão sendo pressionado com 60% da força total



Figura 8 - Balão sendo pressionado com 100% da força



#### 4 DISCUSSÃO E CONCLUSÃO

Com a implementação da metodologia descrita neste artigo, o robô será capaz de se locomover sabendo sua posição no ambiente, sendo controlado por sinais de eletromiografia, reproduzindo movimentos com os braços e alterando a força exercida pela mão.

Em trabalhos futuros, pode-se finalizar a integração entre as três etapas descritas ao longo deste artigo. A partir dos resultados alcançados, é possível o desenvolvimento de novas aplicações remotas integradas às descritas anteriormente.

#### 5 AGRADECIMENTOS

Os autores agradecem à Universidade Federal de Juiz de Fora (UFJF), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Faculdade de Engenharia, ao GRIn (Grupo de Robótica Inteligente da UFJF) e ao Litel (Laboratório de Instrumentação e Telemetria da UFJF) pelas excelentes oportunidades concedidas.

#### REFERÊNCIAS

ALIPPI, C. et al. **Computational intelligence techniques to detect toxic gas presence.** In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR MEASUREMENT SYSTEMS, 2006, IEEE, p. 40-44. Disponível em: <https://ieeexplore.ieee.org/document/4016821/>. Acesso em: 25 abr. 2018.

BASTOS, L. F. **Otimização bioinspirada aplicada na localização de robôs móveis.** 2016, 81. Dissertação – Universidade Federal de Juiz de Fora, Juiz de Fora, 2016. Disponível em: <https://repositorio.ufjf.br/jspui/handle/ufjf/4174>. Acesso em: 25 abr. 2018.

BERGMANN, R.; TARTAKOVSKI, A. **Improving KD-Tree based retrieval for attribute dependent generalized cases.** In: FLAIRS CONFERENCE, 22., 2009, FLAIRS, p. 319-324. Disponível em: <http://aaai.org/ocs/index.php/FLAIRS/2009/paper/download/23/312>. Acesso em: 25 abr. 2018.

BERSINI, H.; VARELA, F. J. Hints for adaptive problem solving gleaned from immune networks. In: INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE. 1990, **Anais...** Berlim, Heidelberg, Springer, 1990, p. 343-354. Disponível em: <https://link.springer.com/chapter/10.1007%2FBFb0029775>. Acesso em: 25 abr. 2018.

DORIGO, M.; DI CARO, G. Ant colony optimization: a new meta-heuristic. In: CONGRESS ON EVOLUTIONARY COMPUTATION, 1999, **Anais...** Washington, IEEE, 1999. Disponível em: <https://ieeexplore.ieee.org/abstract/document/782657/>. Acesso em: 25 abr. 2018.

FOX, D. et al. Monte Carlo localization: Efficient position estimation for mobile robots. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 16., 1999, **Anais...** Orlando, AAAI, 1999, p. 343-349. Disponível em: <http://aaai.org/Papers/AAAI/1999/AAAI99-050.pdf>. Acesso em: 25 abr. 2018.

FURTADO, L. et al. Bat Search Algorithm aplicado na localização de robôs móveis. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 12., 2015. **Anais...** Natal, 2015. Disponível em: [http://www.ufjf.br/gohb/files/2015/07/SBAI\\_2015\\_submission\\_326.pdf](http://www.ufjf.br/gohb/files/2015/07/SBAI_2015_submission_326.pdf). Acesso em: 25 abr. 2018.

REKABY, A. Directed Artificial Bat Algorithm (DABA) – A new bio-inspired algorithm. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS, 2013, **Anais...** Mysore, IEEE, 2013. Disponível em: <https://ieeexplore.ieee.org/document/6637355/>. Acesso em: 25 abr. 2018.

SAPSANIS, C. et al. **EMG based classification of basic hand movements based on time-frequency features**. In: MEDITERRANEAN CONFERENCE ON CONTROL & AUTOMATION, 21., 2013, IEEE, p. 716-722. Disponível em: <https://ieeexplore.ieee.org/document/6608802/>. Acesso em: 25 abr. 2018.

THEODORIDIS, S. et al. **Pattern recognition**. 4. ed. Academic Press, 2008.

THRUN, S. et al. **Probabilistic robotics**. MIT Press, 2005.

YANG, X.S. A new metaheuristic bat-inspired algorithm. In: González J.R. et al. **Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)**. **Studies in Computational Intelligence**. Berlim, Heidelberg: Springer, 2010. p. 65-74.