

# OTIMIZAÇÃO DE CONTROLADOR PID PARA O SELF BALANCING SEGWAY UTILIZANDO BAT ALGORITHMS

## **Mateus Pereira Sátiro Reyes**

Graduando em Engenharia da Computação. Universidade Potiguar.  
E-mail: mateus\_sat@hotmail.com

## **Iugo Baggio Albano Bezerra**

Graduando em Engenharia da Computação. Universidade Potiguar.  
E-mail: iugobaggio@hotmail.com

## **Nielsen Castelo Damasceno**

Mestre em Engenharia Elétrica e de Computação. Universidade Potiguar.  
E-mail: nielsen.castelo@unp.br

**ENVIO EM:** Setembro de 2015

**ACEITE EM:** Maio de 2016

**RESUMO:** Este presente artigo apresenta um estudo inicial do controlador PID sintonizado com o Bat Algorithms. Apresentamos a fundamentação teórica necessária para o aprofundamento do tema proposto neste trabalho. Inicialmente, realizou-se simulações computacionais utilizando Matlab ao controle do sistema dinâmico do SegWay. As simulações computacionais apresentaram resultados promissores. Isso nos motiva, em um outro momento, continuar o estudo com a implementação do SegWay e aplicar a lógica de controle sintonizado pelo Bat Algorithm.

**Palavras-chave:** PID. *Bat-Algorithm*. SegWay, Otimização.

### **OPTIMIZATION OF A PI CONTROLLER FOR OF SELF BALANCING SEGWAY USING THE BAT ALGORITHM METAHEURISTIC**

**ABSTRACT:** This present article discusses an initial study of the tuned PID controller with Bat Algorithms. Here is the theoretical foundation necessary for the issue of deepening proposed in this paper. Initially held computer simulations using Matlab to control the dynamic of the SegWay system. Computer simulations have shown promising results. This motivates us, at another time, continue the study with the implementation of the Segway and implement control logic tuned by the Bat Algorithm

**Keywords:** PID. Bat Algorithm. SegWay. Optimization.



# 1 INTRODUÇÃO

Muito se tem discutido e estudado na engenharia a obtenção de novos algoritmos inteligentes para que se tornem mais eficientes, rápidos, e ao mesmo tempo de simples manuseio e fácil entendimento para os pesquisadores. O PID é um dos controladores de realimentação mais usados em sistemas dinâmicos (OGATA, 2003). Uma importante vantagem desse tipo de sistema de controle é que não se faz necessário o conhecimento preciso da modelagem analítica da planta que está sendo controlada.

Em 2010, desenvolvido por Xin-Shen Yang, o *Bat-Algorithm* é um método que utiliza o comportamento de eco localização dos morcegos. O que pretendemos apresentar neste artigo é o estudo do *Bat-Algorithm* para sintonizar dos parâmetros do PID no controle de um SegWay. Em seguida, verificar a eficiência comparando com o trabalho proposto por (JOHNNY et al., 2009) e (WITZAND, 2009). O SegWay é basicamente um dicitelo (meio de transporte de duas rodas lado a lado) inventado por Dean Kamen e revelado em dezembro de 2001, que funciona a partir do equilíbrio do indivíduo que o utiliza, inicialmente criado para pessoas com deficiências visuais (PEREIRA, 2010).

As próximas seções apresentam as fundamentações teóricas e os estudos iniciais para realização do trabalho proposto neste artigo. Na seção 2, faremos uma breve revisão do PID. Na seção 3, descreve-se o modelo do sistema utilizado para os experimentos. Na seção 4, apresenta-se a função custo. Na seção 5, descreve-se sobre o algoritmo bio-inspirado. Na seção 6, discorre-se sobre os resultados e discussões. E finalmente, as considerações finais e trabalhos futuros.

## 2 PID

Proporcional-Integral-Derivativo (PID) é um dos controladores mais usados na indústria e tem sido utilizado em todo o mundo para sistemas de controle industrial (OGATA, 2003). O controlador possui três parâmetros principais: o Proporcional, o Integral e o Derivativo. A fórmula do PID se resume em:  $u(t) = Kp \cdot e(t) + Ki \cdot \int_{t=0}^t e(t) dt + Kd \frac{de(t)}{dt}$ , onde,  $u(t)$  é a saída em relação ao tempo;  $e(t)$  é a entrada menos o erro em relação ao tempo;  $kp$  é a constante proporcional;  $ki$  é a constante integral e  $kd$  é a constante derivativa.

A função proporcional no controlador produz um valor na saída proporcional ao erro obtido na realimentação. A resposta proporcional pode ser ajustada a partir da constante de ganho  $kp$ . Quanto maior  $kp$ , maior será o ganho do erro e mais instável será o sistema. Mas se a constante  $kp$  for muito pequeno, menor será o seu tempo para a resposta. A função integrativa soma todos os erros instantâneos e a somatória é multiplicada pela constante  $ki$ . A função integral do controlador acelera o movimento do processo até o ponto desejado e elimina o erro que ocorre na função anterior (NOGUEIRA, 2009). Como a função soma dados instantâneos, o resultado do processo pode ultrapassar o ponto desejado, também conhecido como *overshoot*. A função derivativa retarda a taxa de variação de saída do controlador. Essa função diminui o *overshoot* da função anterior e melhora a estabilidade do controlador. Por outro lado, a função derivativa causa um retardo na resposta e é muito suscetível à ruídos. Isso acontece porque essa função amplifica o ruído e caso o ruído e o ganho  $kd$  forem muito grandes, podem causar instabilidade no controlador (OGATA, 2003).

### 3 MODELO DA PLANTA

Para realizar os experimentos utilizando a metaheurística para sintonia do PID, utilizou-se o modelo matemático do SegWay proposto por (JOHNNY et al., 2009) e GELway (WITZAND, 2009). Esses dois projetos servirão de comparativo apresentado neste artigo. Antes de apresentar a equação de espaço de estados do sistema, são apresentados na Tabela 1 todos os parâmetros do modelo do robô e as suas respectivas unidades.

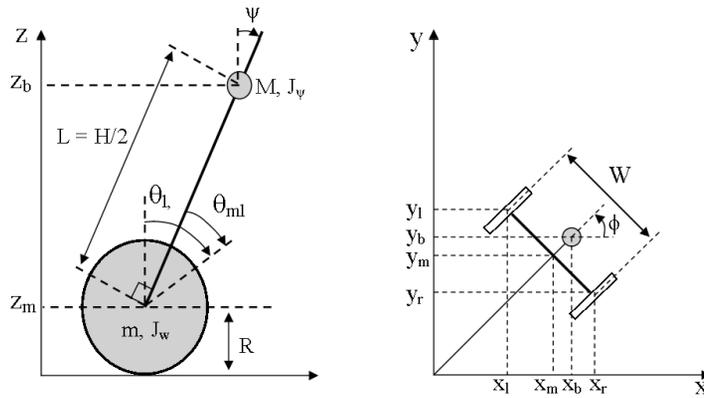
**Tabela 1: Parâmetros do robô SegWay.**

Parâmetros	Unidade	Descrição
$\theta$	[deg]	Ângulo médio das rodas
$\psi$	[deg]	Ângulo de inclinação do corpo
$\Phi$	[deg]	Ângulo de guinada corpo
$\theta_m$	[deg]	Ângulo do motor DC
$g$	[m/sec <sup>2</sup> ]	Gravidade da aceleração
$m$	[kg]	Peso das Rodas
$R$	[m]	Radianos das Rodas
$J$	[kgm <sup>2</sup> ]	Momento de inércia de rodas
$M$	[kg]	Peso do Corpo
$W$	[m]	Largura do corpo
$D$	[m]	Profundidade do corpo
$H$	[m]	Altura do Corpo
$L$	[m]	Comprimento do Corpo
$J$	[kgm <sup>2</sup> ]	Momento de inércia
$J_\psi$	[kgm <sup>2</sup> ]	Arremesso do Corpo no momento da inércia
$J_\phi$	[kgm <sup>2</sup> ]	Corpo no momento de guinada inercial
$J_m$	[kgm <sup>2</sup> ]	DC do motor no momento de inércia
$n$		Proporção da engrenagem
$K_t$	[Nm/A]	Momento de torção constante do DC do motor
$K_b$	[V sec/rad]	Constante da volta EMF do motor DC
$f_m$		Coefficiente de atrito entre o corpo e motor DC
$f_\omega$		Coefficiente de atrito entre a roda e o chão

Fonte: (WITZAND, 2009).

As coordenadas do sistema utilizado para derivar as equações do movimento são apresentado no Figura 1. Os parâmetros que estamos mais preocupados são ângulo médio das rodas ( $\theta$ ) e o ângulo de inclinação do corpo ( $\psi$ ). Com estes são utilizados para calcular o sinal de erro para o balanceamento do controlador. As equações de movimento linearizadas para uma das rodas são apresentado a seguir.

Figura 1: Sistema de coordenadas para a equação do SegWay



Fonte: (STEVEN, 2009).

$$((2m + M)R^2 + 2J_\omega + 2n^2J_m)\ddot{\theta} + (MLR - 2n^2J_m)\ddot{\psi} = F_\theta \quad (1)$$

$$(MLR - 2n^2J_m)\ddot{\theta} + (ML^2 + J_\psi + 2n^2J_m)\ddot{\psi} - MgL\psi = F_\psi \quad (2)$$

$$\left(\frac{1}{2}mW^2 + J_\theta + \frac{W^2}{2R^2}(J_\psi + n^2J_m)\right)\ddot{\phi} = F_\phi \quad (3)$$

A equação (1) e (2) podem ser agrupados, pois ambos constituídos por ângulos  $\theta$  e  $\psi$ . A equação (3) tem somente o desvio do ângulo  $\theta$ .

As primeiras duas equações podem agora ser expressos como:

$$S \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + T \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} + U \begin{bmatrix} \theta \\ \psi \end{bmatrix} = V \begin{bmatrix} v_t \\ v_r \end{bmatrix} \quad (4)$$

$$\text{Onde, } S = \begin{bmatrix} (2m + M)R^2 + 2J_\omega + 2n^2J_m & MLR - 2n^2J_m \\ MLR - 2n^2J_m & ML^2 + J_\psi + 2n^2J_m \end{bmatrix}, T = 2 \begin{bmatrix} \beta + f_\omega & -\beta \\ -\beta & \beta \end{bmatrix},$$

$$U = 2 \begin{bmatrix} 0 & 0 \\ 0 & -MgL \end{bmatrix}, V = 2 \begin{bmatrix} \alpha & \alpha \\ -\alpha & -\alpha \end{bmatrix}, \alpha = \frac{nK_t}{R_m} \text{ e } \beta = \frac{nK_t K_b}{R_m} + f_m.$$

Além disso, a equação (3) pode ser escrita na forma:

$$I\ddot{\phi} + J\dot{\phi} = K(v_r - v_t) \quad (5)$$

$$\text{Onde, } I = \frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}(J_\omega + n^2J_m), J = \frac{W^2}{2R^2}(\beta + f_\omega) \text{ e } K = \frac{W}{2R}\alpha.$$

O próximo passo é escrever as equações de espaço de estado. Usando as seguintes variáveis, , conforme as variáveis de estado para equação (4) e (5), respectivamente, e é a variável de entrada:

$$x_1 = \begin{bmatrix} \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad x_2 = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad u = \begin{bmatrix} v_t \\ v_r \end{bmatrix} \quad (6)$$

Podemos reescrever a equação (4) e (5), no estado de espaço como se segue:

$$\dot{x}_1 = A_1 x_1 + B_1 u \quad (7)$$

$$\dot{x}_2 = A_2 x_2 + B_2 u \quad (8)$$

As matrizes de espaço de estado, são mostrados na sua forma completa no apêndice A1 no trabalho de Witzand (2009). A matriz A1 apresenta instabilidade (o terceiro valor é 7.4367). Portanto, faz necessário utilizar uma estratégia de estabilização do controlador utilizando o método regular quadrático linear (RQL). O RQL foi usado para otimizar as matrizes de espaço de estado do controlador para o motor e parâmetros ângulo de inclinação do corpo. O método RQL pode ser usado para reduzir erros de estado estacionário, fornecendo ponderação de feedback para os parâmetros de entrada que estabilizam o sistema. A função LQR em Matlab foi então utilizada para determinar os pesos específicos para cada um dos feedbacks do controlador. Considere a dupla entrada, sistema de saída dupla para o Segway, notando que existe agora um estado adicional devido a um integrador a ser incluído no loop de realimentação:

$$\dot{x}_1 = A_1 x_1 + B_1 u. \quad (9)$$

Onde,

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & A_1(3,2) & A_1(3,3) & A_1(3,4) & 0 \\ 0 & A_1(4,2) & A_1(4,3) & A_1(4,4) & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_1(3) & B_1(3) \\ B_1(4) & B_1(4) \\ 0 & 0 \end{bmatrix}$$

e o feedback

$$u = -Kx = - \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \\ k_1 & k_2 & k_3 & k_4 & k_5 \end{bmatrix} x.$$

A razão pela qual existem dois sinais de controle de feedback idênticas é que a ação de controle é dividida entre dois atuadores e, devido à simetria, os mesmos ganhos são utilizados para o controlador de esquerda e direita. As equações completas para as matrizes A1 e B1 pode ser encontrada no apêndice A1 do trabalho de Witzand (2009).

O método RQL usa a seguinte equação para determinar a melhor solução para o sistema, observando que a matriz Q determina os fatores de ponderação para o feedback e R é o fator de ponderação escalar (RICHARD; BISHOP, 2005):

$$A_1^T P + P A_1 - P B_1 R^{-1} B_1^T P + Q = 0 \quad (10)$$

Resolver esta equação para irá determinar os parâmetros de feedback otimizados para o nosso sistema de controle. Usando as documentações NXTway-GS, foram utilizados os seguintes matrizes Q e R:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 * 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 * 10^2 \end{bmatrix}, \quad R = \begin{bmatrix} 1 * 10^3 & 0 \\ 0 & 1 * 10^3 \end{bmatrix} \quad (11)$$

Finalmente, a função LQR em Matlab foi então utilizada para determinar os pesos especificado para cada um dos feedbacks do controlador. Os coeficientes de correção do controlador são mostrado na Tabela 2.

**Tabela 2: Parâmetros utilizados para linearizar o sistema.**

Variável	Ganho	Valor
Ângulo do Motor, $\theta$	$k_1$	-0.8351
Ângulo do Giro, $\psi$	$k_2$	-34.1896
Velocidade do Motor, $\dot{\psi}$	$k_3$	-1.0995
Velocidade do Giro, $\dot{\theta}$	$k_4$	-2.8141

Fonte: (STEVEN, 2009).

## 4 ÍNDICE DE DESEMPENHO

Além da estabilidade e da convergência ao referencial, existem fatores importantes que devem ser analisados na sintonia dos PID. A análise da resposta do sistema baseada nos índices de desempenho fornece aos profissionais da área de controle evidências sobre o grau de otimização do controlador. A equação  $ISTSE = \int_0^T t^2 e^2(t) dt$  é um dos principais índices de desempenho (SHINNERS, 1998). Esse índice (Integral of Squared Time Multiplied by the Squared Error) é calculado no intervalo de tempo  $0 \leq t \leq T$ , onde  $T$  é o tempo necessário para que a resposta da planta estabilize num valor final. Em sistemas de segunda ordem é uma prática comum escolher  $T = T_s$ . Os controladores sintonizados pelos índices ponderados pelo tempo (ISTSE) tendem a melhorar a performance do erro nos estágios intermediário e final da resposta da planta, quando uma parte significativa do tempo de simulação já é decorrida.

## 5 BAT ALGORITHM

Foi criado um algoritmo baseado na ecolocalização dos morcegos, o Algoritmo Morcego ou Bat Algorithm (BA), proposto por (YANG, 2010). BA é uma metaheurística inspirada no método do morcego se localizar em voos noturnos. Na forma computacional é gerado um conjunto de morcegos na forma de vetor, onde, cada morcego representa uma solução a ser obtida. O objetivo principal é alcançar a sua presa em uma rota mais eficiente em um curto período de tempo, ou seja, uma solução ótima que minimiza a função de custo (DAMASCENO et al., 2015). Basicamente, os passos do algoritmo são descritos na forma de um pseudocódigo conforme mostra a imagem a seguir.



Figura 3: Pseudocódigo do Bat Algorithm.

```

1: Parâmetros de entrada:  $n, \alpha, \lambda$ 
2: Inicializa morcegos  $\vec{x}_i$ 
3: Avalia  $f(\vec{x}_i)$  para todos os morcegos
4: Atualiza melhor morcego  $\vec{x}_*$ 
5: enquanto critério de parada não atingido faça
6:   para  $i = 1$  até  $n$  faça
7:      $f_i = f_{min} + (f_{max} - f_{min})\beta, \beta \in [0,1]$ 
8:      $\vec{v}_i^{t+1} = \vec{v}_i^t + (\vec{x}_i^t - \vec{x}_*^t)f_i$ 
9:      $\vec{x}_{temp} = \vec{x}_i^t + \vec{v}_i^{t+1}$ 
10:    se  $rand < r_i, rand \in [0,1]$  to  $n$  então {Faz busca local}
11:     $\vec{x}_{temp} = \vec{x}_* + \varepsilon A_m, \varepsilon \in [-1,1]$ 
12:    fim se
13:    Realiza perturbação em uma dimensão de  $\vec{x}_{temp}$ 
14:    se  $rand < A_i$  or  $f(\vec{x}_{temp}) \leq f(\vec{x}_i), rand \in [0,1]$  então {Aceita solução temporária}
15:     $\vec{x}_i = \vec{x}_{temp}$ 
16:     $r_i^{t+1} = 1 - \exp(-\lambda t)$ 
17:     $A_i^{t+1} = \alpha A_i^t$ 
18:    fim se
19:    Atualiza melhor morcego  $\vec{x}_*$ 
20:   fim para
21: fim enquanto
22: Pós-processamento

```

Fonte: Autores.

Na linha 7 a frequência do morcego  $f_i \in [f_{min}, f_{max}]$  é atualizada e ponderada pela variável  $\beta \in [0,1]$  que representa um número aleatório gerado a partir de uma distribuição uniforme. A nova frequência gerada  $f_i$  é utilizada para determinar a nova velocidade do morcego  $\vec{v}_i^{t+1}$  (linha 8) tomando como referência o melhor morcego global no instante de tempo  $t$ .

Com a velocidade  $\vec{v}_i^{t+1}$  calculada é possível determinar a nova posição temporária do morcego dada por,  $\vec{x}_{temp}$  (linha 9). Nas linhas 10 e 11 é feita uma busca local aleatória chamada de (Random Walk) para explorar as soluções candidatas do problema, aonde  $\varepsilon \in [-1,1]$  é um número aleatório tirado de uma distribuição uniforme e  $A_m$  é a média da amplitude de todos os morcegos no instante  $t$ , e o valor da posição temporária,  $\vec{x}_{temp}$ , é atualizado.

Na linha 13, é realizado uma perturbação em uma dimensão de  $\vec{x}_{temp}$ , o vetor resultante é então avaliado por  $f(\vec{x}_i)$  na (linha 14), caso minimize a função custo ou tenha a amplitude  $A_i$  menor que a variável de distribuição uniforme aleatória,  $rand \in [0,1]$ , a nova posição temporária  $\vec{x}_{temp}$ , é aceita como permanente para morcego (linha 15), o que irá aumentar sua taxa de pulso  $r_i$  e reduzirá sua amplitude  $A_i$  (linhas 16 e 17) sendo que quando  $t \rightarrow \infty, r_i \rightarrow 0$ .

No fim da execução das linhas 17 e 18, a nova solução encontrada é comparada com a solução global  $\vec{x}_*$  (o melhor morcego), caso aquela seja melhor, o valor de  $\vec{x}_{temp}$  passa a sobrescrever o valor de  $\vec{x}_*$ , tornando-se assim o melhor morcego global (linha 19).

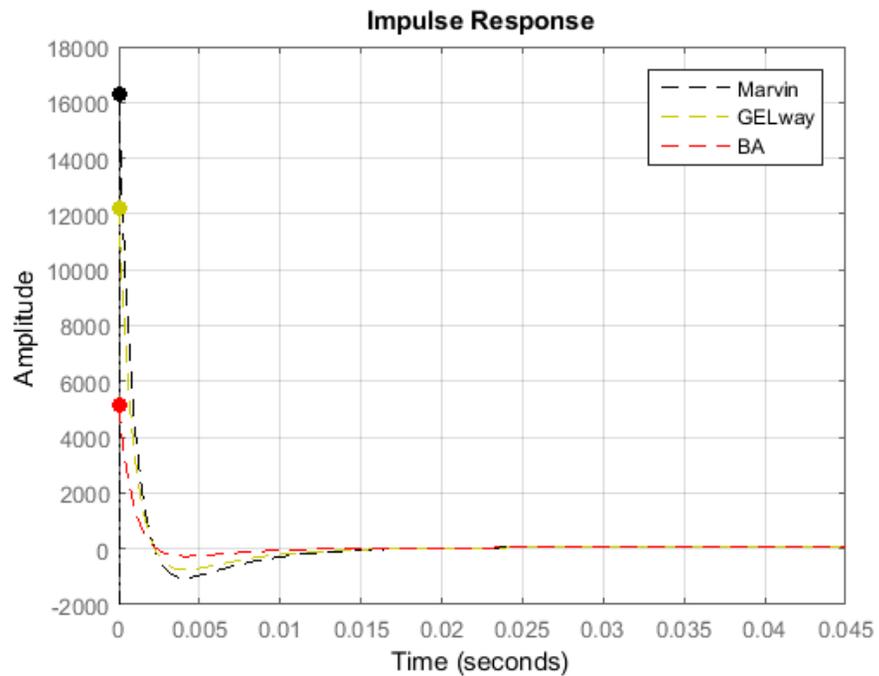
## 6 RESULTADOS E DISCUSSÃO

A população do BA são parâmetros do PID, ou seja, cada morcego corresponde ao  $Kp, Ki$  e  $Kd$ . Para esses estudos iniciais utilizou-se uma população de 40 morcegos. O critério de parada empiricamente foi 100 iterações. A função custo utilizado foi ISTSE. Os parâmetros  $A_i$  e  $r_i$  foram 0.5 e a frequência máxima e mínima foram 0 e 2, respectivamente.

O primeiro teste foi analisar o comportamento da resposta do sistema a uma função impulso. Os parâmetros do PID empiricamente escolhido foi  $k_p=1.5048$ ,  $k_i=2.3383$  e  $k_d=0.0619$ . A figura 4 apresenta esse resultado. Observe que nessa figura a *overshoot* foi  $5.13e+03$  (em vermelho) e é menor que os outros métodos.

A Tabela 3, mostra um comparativo numérico entre os três métodos utilizado para estimar o PID. Neste resultado houve uma melhora significativa do *overshoot*.

Figura 4: Resposta do sistema ao impulso com PID sintonizado pelo BA



Fonte: Autores.

Tabela 3: Comparativo com os métodos e BA.

Método	$K_p$	$K_i$	$K_d$
Marvin	1.00	0.20	0.20
GELway	1.15	0.10	0.15
BA	1.50	2.33	0.06

Fonte: Autores.

## 7 CONSIDERAÇÕES FINAIS

Os resultados mostram que o BA é superior aos dois métodos comparados. Os próximos passos neste trabalho é aplicar os resultados obtidos neste estudo utilizando EV3 da *Mindstorms*. Pretendemos realizar uma análise mais significativa em um projeto futuro e demonstrar que o BA pode ser um ótimo método para sintonizar o PID de uma forma rápida e precisa. Isso nos motiva a realizar outros experimentos, como por exemplo, comparar o BA com o algoritmo da Polinização das Flores (FPA) e Otimização por Enxame de Partícula (PSO) e averiguar na prática se realmente o BA tem uma resposta mais satisfatória.

## REFERÊNCIAS

DAMASCENO, Nielsen Castelo; PAULA, Paulo Cesar Oliveira de; RÊGO Joilson B. A. Otimização de controlador PI para um trocador de calor utilizando a metaheurística Bat Algorithm. **Anais SBAI**. 2015.

JOHNNY, Rieper; BENT, Bisballe N.; KASPER, Sohn. **Marvin - The Balancing Robot**. Aarhus University, 2009.

NOGUEIRA, João C. Basilio; Victor N. **Ajuste de controle PI e PID para plantas estáveis e instáveis**. p. 7, COPPE - Programa de Engenharia Elétrica Cidade Universitária - Ilha do Fundão, Universidade Federal do Rio de Janeiro, 2009.

OGATA, katsuhiko. **Engenharia de controle moderno**, 4ed. São Paulo: Prentice Hall Ppearson), p. 788, 2003.

PEREIRA, Guilherme Machado. **Projeto e controle de um transportador pessoal robótico auto equilibrante**. 11 p. Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro, 2010.

RICHARD, C. Dorf; BISHOP, Robert H. **Modern Control Systems**, 10th Edition. Pearson Education, Inc., 2005.

SHINNERS, Stanley M. **Modern control system theory and design**. John Wiley & Sons, Cap. 5. p. 290-292, 1998.

WITZAND, Steven J. **Coordinated Lego Segways**. Monografia Engenharia Elétrica, Universidade de New South Wale, Australia, 2009.

YANG, X.-s. A New Metaheuristic Bat-Inspired Algorithm. In: YANG, X.-s. **Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)** (Eds. J. R. Gonzalez et al.): Studies in Computational Intelligence. Berlin: Springer, p. 65-74, 2010.

