

UTILIZAÇÃO DE ALGORITMO DE EVOLUÇÃO DIFERENCIAL MULTIOBJETIVO NO PROJETO DE CONTROLADORES

Cidiney J. Silva, Oriane Magela Neto, Flávio Vinícius Cruzeiro Martins

Departamento de Engenharia Elétrica - UFMG

cidiney@cpdee.ufmg.br, oriane@dee.ufmg.br, flavio@cpdee.ufmg.br

Resumo – É de interesse tecnológico e econômico que o controle das plantas presentes nos diversos processos tenha o melhor desempenho possível. A utilização de controladores cujos parâmetros sejam sub ou sobre-estimados pode resultar em oscilações, dinâmica retardada, desperdício de materiais e energia, desgastes dos atuadores, dentre outros. O modo de projetar controladores perpassa desde métodos analíticos, como alocação de pólos, até métodos de tentativa e erro, como o de Ziegler-Nichols. O interesse deste trabalho é demonstrar a utilidade dos Algoritmos Evolucionários, em especial o de Evolução Diferencial, no projeto automático de controladores. Além disso, é demonstrada a utilidade da abordagem de otimização multiobjetivo para o problema de controladores. Com a abordagem multiobjetivo do problema, faz-se necessária a definição de um decisor para a escolha de uma das respostas do conjunto Pareto-Ótimo. Uma possível definição é a escolha do controlador mais barato; neste trabalho escolhe-se o controlador cujo ponto na fronteira de Pareto seja mais próximo da solução utópica do problema. Da forma aqui proposta, basta que sejam definidas as restrições do problema, o modelo da planta a ser controlada e a estrutura do controlador desejado e um decisor para que um controlador ótimo seja definido automaticamente.

Palavras-chave – Controle de Sistemas Lineares, Otimização Multiobjetivo, Evolução Diferencial.

Abstract – It is interesting that technological and economic control of the plants present in the various processes have the best performance possible. The use of controllers whose parameters are under-or over-estimates can result in oscillations, dynamic retarded, waste of materials and energy, wear of the actuators, among others. The method of designing controllers permeates from analytical methods such as pole placement, to methods of trial and error, as the Ziegler-Nichols. The interest of this work is to demonstrate the usefulness of Evolutionary Algorithms, in particular of Differential Evolution, in the automatic design of controllers. Furthermore, it demonstrated the usefulness of multiobjective optimization approach to the problem of drivers. With the approach of multi-objective problem, it is necessary to define a decision-maker to choose one of the responses of all Pareto-Optimal. One possible definition is the choice of controller cheaper, this work picks up the controller which point on the Pareto is closer to the utopian solution of the problem. The way proposed here, just for clarification of the restrictions of the problem, the model of the plant to be controlled and the structure of the desired controller and a decision-maker for a great driver is automatically set.

Keywords – Control of Linear Systems, Multiobjective Optimization, Differential Evolution.

1 Introdução

O projeto de controladores consiste essencialmente na escolha de parâmetros \vec{p} de um subsistema de função de transferência $\mathcal{C}(s, \vec{p})$ – em que s corresponde à frequência complexa $s = \sigma + j\omega$ e \vec{p} é um conjunto de parâmetros – a ser integrado à malha fechada envolvendo uma planta a ser controlada e um elemento sensor cujas funções de transferência são $P(s)$ e $H(s)$, respectivamente (v. figura 1). $\mathcal{C}(s, \vec{p})$ possui uma estrutura pré-concebida e adequada ao tipo de sistema ao qual se vai controlar. Mais freqüentemente, a estrutura adotada para a função de transferência $\mathcal{C}(s, \vec{p})$ é a de um controlador PID (Proporcional, Integral, Diferencial) ou ainda a de compensadores (v. equações 1 e 2).

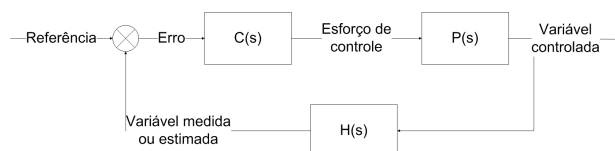


Figura 1: Esquema básico de uma malha de controle fechada

$$C_{PID} = K_P + \frac{K_I}{s} + K_D \cdot s \equiv \frac{K_D \cdot s^2 + K_P \cdot s + K_I}{s} \quad (1)$$

$$C_{Comp} = \frac{a \cdot s + b}{c \cdot s + d} \quad (2)$$

Os parâmetros do controlador são escolhidos de forma a satisfazer alguns critérios de desempenho imediatos do sistema em malha fechada [1] [2]: estabilidade, erro em estado estacionário, tempo de assentamento. Outros critérios podem se fazer necessários, como as margens de ganho e de fase, esforço de controle, índices como ITAE, ISE, rejeição a perturbações, dentre outros. Certamente alguns desses critérios estão diretamente relacionados a outros; por exemplo, a diminuição de erro em estado estacionário também está relacionado à diminuição do índice ITAE. O projeto torna-se desafiador quando há critérios conflitantes que devem ser atendidos; por exemplo, a diminuição ou supressão de sobressinal (*overshoot*) e a diminuição do tempo de assentamento do sistema são critérios conflitantes.

Neste trabalho foram consideradas como funções-objetivo a serem minimizadas os valores RMS de erro e de esforço de controle. Esses objetivos são conflitantes, como pode ser observado na figura 2. A motivação para a escolha dessas funções-objetivo é o fato de que caso haja um sinal de erro persistentemente grande na entrada do controlador, o objetivo da malha fechada não se concretiza, logo, deseja-se um valor médio baixo de erro; caso haja um esforço de controle persistentemente grande, pode-se incorrer no desgaste dos atuadores sobre a planta, portanto, deseja-se utilizar parcimoniosamente os atuadores.

Os pontos mostrados na figura 2 foram coletados tomando-se uma planta conhecida, uma estrutura pré-definida de controlador e variam-se os parâmetros desse controlador com o sistema em malha fechada; para cada conjunto de parâmetros de controlador simula-se a resposta do sistema à entrada em degrau e então tomam-se os valores RMS de erro e esforço (v. figura 1).

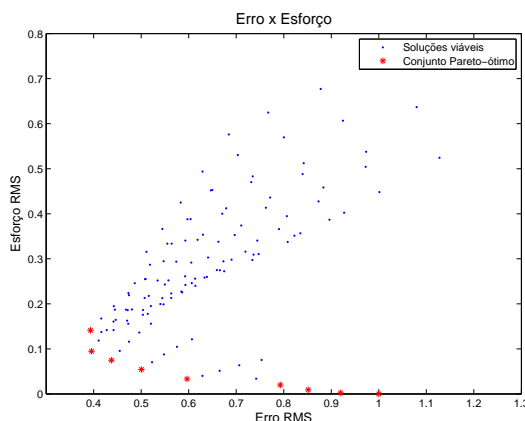


Figura 2: Compromisso entre os valores RMS de erro do sistema e esforço de controle sobre a planta.

Observando-se os possíveis valores de erro e esforço da planta em malha fechada registrados na figura 2, nota-se a formação de uma fronteira a partir da qual não são mais possíveis pares de valores de erro \times esforço. Os pontos dessa fronteira correspondem aos melhores controladores que minimizam ao mesmo tempo erro e esforço de controle para a planta em questão e estão na chamada fronteira de Pareto. A Otimização Multiobjetivo se propõe a encontrar esses pontos dessa fronteira e existem diversos métodos com essa finalidade.

Existem diversos algoritmos voltados à resolução de problemas de otimização do tipo multiobjetivo [3]. Da escola de algoritmos determinísticos, os principais são o Problema Ponderado (P_λ), Programação de Metas (P_μ), e o Problema ϵ -restrito (P_ϵ).

Nos últimos anos, uma ampla coleção de Algoritmos Evolucionários tem sido proposta e estudada para aplicações em otimização multiobjetivo [4]. Em [5] é apresentado o algoritmo NSGA-II. Esse tem sido um algoritmo muito utilizado e estudado devido ao seu bom desempenho em custo computacional e qualidade das fronteiras de Pareto encontradas [6] [7] [8]. Outros exemplos são SPEA-2 [9] [10] e o MOGA [6] [11].

Dentro dessa gama de algoritmos evolutivos, os de evolução diferencial [12] são atrativos por sua simplicidade de implementação, robustez e capacidade de auto-adaptação [13]. Este será o algoritmo de interesse neste texto.

2 Otimização multiobjetivo

A otimização multiobjetivo se ocupa em encontrar não uma única solução apenas, mas um conjunto de soluções que representam a chamada *fronteira de Pareto* em um problema de otimização que envolve mais de uma função objetivo [3]. Considere-se um problema de otimização com m funções objetivo $\vec{F} = [f_1(\vec{p}) \ f_2(\vec{p}) \ \dots \ f_m(\vec{p})]^T$, todas elas variando com o conjunto de parâmetros $\vec{p}_i = [p_1 \ p_2 \ \dots \ p_n]^T$ de dimensão n . Um ponto \vec{p}_j^* é chamado de *solução Pareto-ótima* se minimizar uma das funções $f_j(\vec{p}) \in \vec{F}$ de tal forma que piore simultaneamente todas as demais funções $f_i(\vec{p}) \in \vec{F}$, $i \neq j$. Maiores detalhes em [3], [14], [15].

3 Evolução Diferencial

Cabe ressaltar aqui que as abordagens para evolução que seguem descritas são adotadas por [15] e [13]. A escola de Evolução Diferencial abre precedentes para a geração de muitos algoritmos derivados [16] das idéias básicas dos algoritmos expostos a seguir.

3.1 Algoritmo de Evolução Diferencial Mono-objetivo

O algoritmo de Evolução Diferencial baseia-se principalmente no operador de *mutação diferencial*. Para a aplicação desse algoritmo, os indivíduos de uma população de soluções candidatas de um problema de otimização são vetores de valores reais, e portanto cada indivíduo está contido em um espaço \mathbb{R}^n . Logo, para os indivíduos de uma população estão definidas as operações de soma vetorial e multiplicação por escalar.

O operador de mutação diferencial [13] consiste na geração de uma perturbação que é um *vetor diferencial* entre dois indivíduos aleatórios \vec{x}_2 e \vec{x}_3 de uma população. O resultado da diferença $(\vec{x}_2 - \vec{x}_3)$ é modificada por um escalar $\eta \in]0, 5; 1[$ aleatório e então é somado a um outro indivíduo escolhido aleatoriamente \vec{x}_1 – é importante notar que \vec{x}_1 , \vec{x}_2 e \vec{x}_3 devem ser distintos. Diz-se então que aplicou-se a mutação diferencial sobre o indivíduo \vec{x}_1 . Dessa forma, chega-se à equação fundamental da evolução diferencial:

$$\vec{u}_t = \vec{x}_1 + \eta(\vec{x}_2 - \vec{x}_3) \quad (3)$$

em que \vec{u}_t é chamado de *indivíduo de teste*. Ao se calcular o indivíduo de teste, verifica-se dentre \vec{u}_t e \vec{x}_1 qual tem maior adequação e então escolhe-se para a próxima geração esse indivíduo.

Espera-se que ao longo das execuções do algoritmo, os vetores diferenciais tendam a alinhar-se preferencialmente a alguma direção tal que ao se calcular \vec{u}_t ele seja atraído ao mínimo em uma bacia de atração. Espera-se que esse mínimo seja global porque inicialmente, cada cálculo de $(\vec{x}_2 - \vec{x}_3)$ fornece um vetor com direções e amplitudes aleatórias, e portanto, provavelmente diversas vezes os indivíduos \vec{x}_1 serão modificados de forma a cair na bacia do mínimo global. À medida que os melhores indivíduos vão sendo conservados para a próxima geração, vão ficando provavelmente os indivíduos de uma mesma bacia e então as perturbações tenderão a se alinhar em uma única direção. As amplitudes das perturbações vão diminuindo progressivamente devido à proximidade entre os indivíduos em uma mesma bacia e as sucessivas multiplicações pelo escalar η .

O algoritmo-base da Evolução Diferencial está explicitado no algoritmo 1.

Algorithm 1 Algoritmo de Evolução Diferencial Mono-objetivo

```

t := 1
Inicializa população  $X_t := \{\vec{x}_{t,1} \vec{x}_{t,2} \dots \vec{x}_{t,N}\}$ 
3: enquanto not(Critério de parada) faça
    para i = 1 até i = N faça
        Selecione aleatoriamente  $r_1, r_2, r_3 \in \{1, 2, \dots, N\}$ 
6:     Selecione aleatoriamente  $\delta_i \in \{1, 2, \dots, N\}$ 
        para j = 1 até j = n faça
            se  $\mathcal{U}(0, 1) < C \vee j = \delta_i$  então
9:                  $u_{t,i,j} := x_{t,r_1,j} + \eta(x_{t,r_2,j} - x_{t,r_3,j})$ 
            senão
                 $u_{t,i,j} := x_{t,i,j}$ 
12:     fim se
        fim para
    fim para
15: para i = 1 até i = N faça
    se  $f(\vec{u}_{t,i}) < f(\vec{x}_{t,i})$  então
         $\vec{x}_{t+1,i} := \vec{u}_{t,i}$ 
18: senão
         $\vec{x}_{t+1,i} := \vec{x}_{t,i}$ 
    fim se
21: fim para
    t := t + 1
fim enquanto
    
```

$f(\cdot)$ é o valor de função, δ_i é um índice para componente do indivíduo, n é a dimensão do espaço de busca, N é o tamanho da população.

As vantagens do algoritmo de evolução diferencial são sua simplicidade de implementação, quantidade mínima de cálculos (uma subtração, uma adição e uma multiplicação) e a quantidade mínima de parâmetros (um escalar aleatório). Essa simplicidade implica em um melhor tempo de máquina para resolução de problemas.

3.2 Evolução Diferencial Multiobjetivo

Este algoritmo segue um esquema essencialmente semelhante ao descrito para o caso mono-objetivo (v. algoritmo 1). Entretanto, são dois os objetivos aqui tratados, pelo que não é possível realizar a operação de escolha entre o indivíduo pai $\vec{x}_{t,i}$ e o indivíduo de teste $\vec{u}_{t,i}$ conforme descrito na linha 16 do algoritmo 1. Como procuram-se pontos que componham a fronteira de Pareto do problema, o que é feito é uma operação de união entre o conjunto de pontos da geração presente (pontos $\vec{x}_{t,i}$,

$i \in (1, 2, \dots, N)$) e o conjunto de pontos de teste $\vec{u}_{t,i}, i \in (1, 2, \dots, N)$, conforme descrito no algoritmo 2. Após isso, é realizada uma operação de dominância entre todos os pontos do conjunto $X_{aux} \equiv \{x_{t,i}\}_{i=1}^N \cup \{u_{t,i}\}_{i=1}^N$ e tomam-se os primeiros N pontos do ranque de dominância para a próxima geração.

Algorithm 2 Algoritmo de Evolução Diferencial Multiobjetivo

```

t := 1
Inicializa população  $X_t := \{\vec{x}_{t,1} \vec{x}_{t,2} \dots \vec{x}_{t,N}\}$ 
3: enquanto not(Critério de parada) faça
    para  $i = 1$  até  $i = N$  faça
        Selecione aleatoriamente  $r_1, r_2, r_3 \in \{1, 2, \dots, N\}$ 
6:     Selecione aleatoriamente  $\delta_i \in \{1, 2, \dots, N\}$ 
        para  $j = 1$  até  $j = n$  faça
            se  $\mathcal{U}(0, 1) < C \vee j = \delta_i$  então
9:                  $u_{t,i,j} := x_{t,r_1,j} + \eta(x_{t,r_2,j} - x_{t,r_3,j})$ 
            senão
                 $u_{t,i,j} := x_{t,i,j}$ 
12:        fim se
        fim para
    fim para
15:  $X_{aux} := \{x_{t,i}\}_{i=1}^N \cup \{u_{t,i}\}_{i=1}^N$ 
     $X_{t+1} := \text{dominância}(X_{aux}, N)$ 
     $t := t + 1$ 
18: fim enquanto
    
```

4 Projeto de controladores

A abordagem adotada neste trabalho possibilita encontrar os parâmetros de um controlador ótimo automaticamente. Um modelo de controlador deve ser definido, as funções de restrição da dinâmica desejada para o sistema em malha fechada e o modelo da planta são as entradas básicas para se encontrar uma coleção de parâmetros ótimos – que é o conjunto Pareto-ótimo. Dentre as soluções do conjunto Pareto-ótimo, uma deve ser escolhida como resposta final via função de utilidade do decisor. Neste trabalho o decisor utilizado é o de solução que mais se aproxima da solução utópica. As funções-objetivo são os valores RMS de erro e esforço de controle, pelo que a solução utópica seria um controlador tal que resultasse em erro nulo e esforço nulo sobre a planta; logo, o decisor aqui utilizado consiste na escolha de um ponto no conjunto Pareto que esteja mais próximo da origem.

Alguns casos de teste foram utilizados para validar a eficácia da metodologia.

Os principais parâmetros do algoritmo implementado e do problema de otimização em si seguem listados nas tabelas 1 e 2.

Tabela 1: Parâmetros do algoritmo DE

Parâmetro	Valor
Mopulação	24
Máximo de avaliações da fun. obj.	2400
Probabilidade mutação C	0,4

Os controladores aqui testados eram do tipo PID, pelo que os cromossomos assumem a forma de um vetor $\vec{x}_i = [K_P K_I K_D]^T$, com os limites desses valores de acordo com as restrições da tabela 2. O modelo das plantas é variado para se verificar a eficácia da metodologia para diversos tipos de problema.

5 Resultados obtidos

5.1 Sistema de segunda ordem

Considerando-se um sistema com a seguinte função de transferência

$$G(s) = \frac{s + 1}{s^2 + 1,4s + 1} \quad (4)$$

A partir da execução do algoritmo, a fronteira de Pareto para o problema foi encontrada como ilustrado na figura 3. O controlador encontrado via Evolução Diferencial Multiobjetivo foi

$$C(s) = 1,388s + 42,71 \quad (5)$$

Tabela 2: Parâmetros do problema de otimização

Parâmetro	Valor
Restrição 1	Sobressinal
Restrição 2	T. assentamento
Restrição 3	Estabilidade
Máx. Sobressinal	10%
Máx. T. assentamento	20% de τ_{princ}
Ponderação Rest. 1	100
Ponderação Rest. 2	1
Ponderação Rest. 3	1000
Função objetivo 1	esf. controle
Função objetivo 2	erro
Atraso de tempo	40% de τ_{princ}
Limite máx. ganho P	100
Limite máx. ganho I	100
Limite máx. ganho D	100

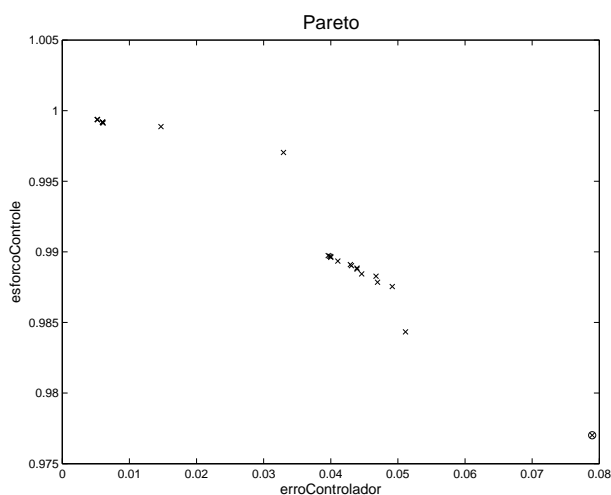


Figura 3: Pareto para a planta cuja função de transferência é dada na equação 4.

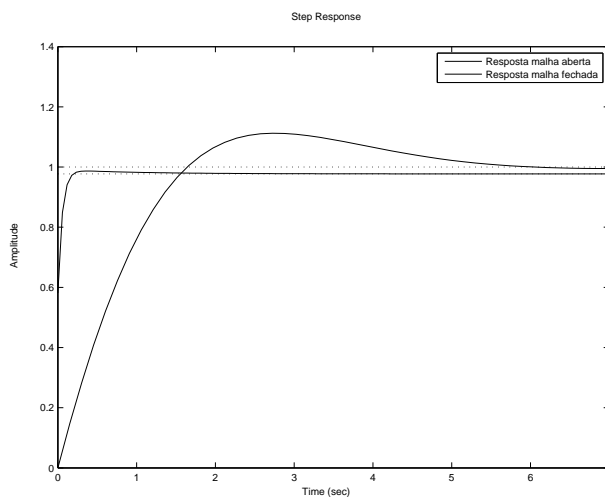


Figura 4: Respostas ao degrau em malha aberta e em malha fechada da planta de função de transferência dada na equação 4.

As respostas ao degrau para o sistema em malha aberta e o sistema em malha fechada estão ilustradas na figura 4.

Os dados de tempo de assentamento e sobressinal em malha aberta e fechada para este sistema estão registrados na tabela 3.

Note-se que os critérios de desempenho dinâmico, postos na forma de restrições ao problema de otimização (v. tabela 2) foram respeitados para esse sistema com exceção do tempo de assentamento, que ficou em 0,61 s em malha fechada e não em 0,22 s como se esperaria (admitindo-se uma aproximação de 5τ para o tempo de assentamento da resposta do sistema).

Tabela 3: Características da planta (v. equação 4) em malha aberta e malha fechada

Parâmetro	Valor M. A.	Valor M. F.
Sobressinal	11,2 %	0%
T. assentamento	1,11 s	0,61 s

Entretanto, de acordo com a tabela 2, há um peso muito maior na restrição de sobressinal, pelo que o algoritmo foi forçado a atender preferencialmente essa restrição à outra.

5.2 Primeira ordem com atraso

Considerando-se um sistema com a seguinte função de transferência

$$G(s) = \frac{10 \cdot e^{-0,04s}}{s + 10} \quad (6)$$

A partir da execução do algoritmo, a fronteira de Pareto para o problema foi encontrada como ilustrado na figura 3.

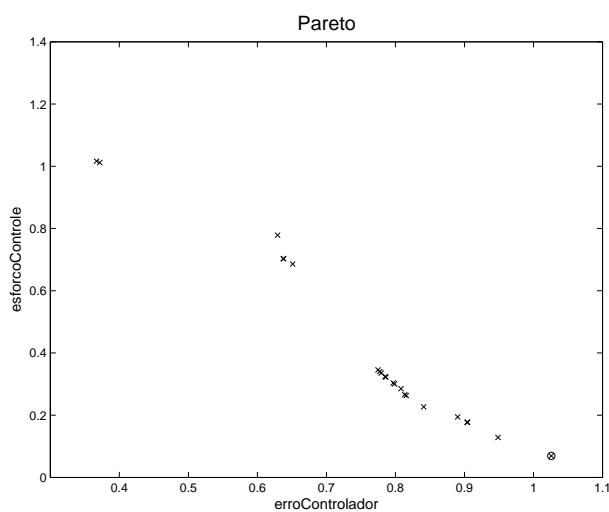


Figura 5: Pareto para a planta cuja função de transferência é dada na equação 6.

O controlador encontrado via Evolução Diferencial Multiobjetivo foi

$$C(s) = 0,024 \quad (7)$$

As respostas ao degrau para o sistema em malha aberta e o sistema em malha fechada estão ilustradas na figura 6.

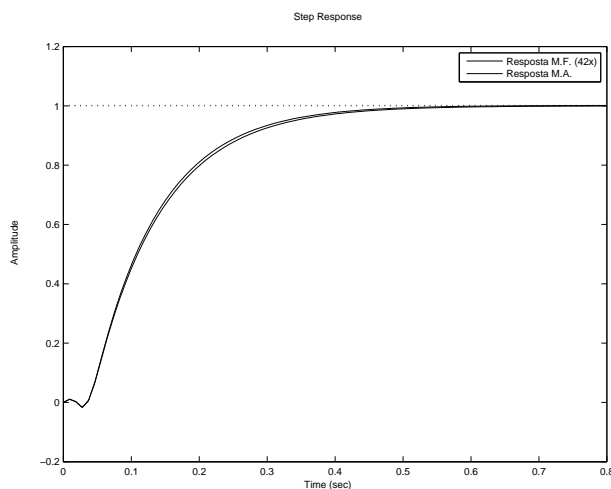


Figura 6: Respostas ao degrau em malha aberta e em malha fechada da planta de função de transferência dada na equação 6.

Os dados de tempo de assentamento e sobressinal em malha aberta e fechada para este sistema estão registrados na tabela 4.

Tabela 4: Características da planta (v. equação 6) em malha aberta e malha fechada

Parâmetro	Valor M. A.	Valor M. F.
Sobressinal	0 %	0%
T. assentamento	0,43 s	0,41 s

Este é um problema de projeto difícil (uma relação θ/τ grande) e portanto o algoritmo encontra uma solução que consiste apenas em uma redução de ganho, em nome da estabilidade, já que não é possível atender aos outros critérios (v. tabela 2).

6 Conclusões

Puderam ser encontradas respostas com bom desempenho no domínio do tempo a partir dos sistemas controlados via controladores calculados pela abordagem proposta. Essa abordagem mostra-se promissora, pois a geração automática de parâmetros ótimos de um controlador é interessante em sistemas com dinâmica variante no tempo. A cada época em que o sistema assume uma função de transferência, um novo controlador ótimo poderia ser obtido automaticamente.

A abordagem de otimização multiobjetivo é vantajosa por permitir que uma família de respostas ótimas possa ser obtida, dado o interesse de otimizar mais de um funcional simultaneamente. Em projeto de controladores, essa atitude deveria ser uma constante dado que sempre ao melhorar um aspecto da malha de controle, outro acaba tendo que “pagar” pela melhoria.

REFERÊNCIAS

- [1] R. C. Dorf and R. H. Bishop. *Sistemas de controle modernos*. LTC, 8th edition, 2001.
- [2] F. G. Shinskey. *Process Control Systems*. McGraw-Hill, third edition, 1988.
- [3] R. H. C. Takahashi. “Otimização Escalar e Vetorial”.
- [4] E. Zitzler. “Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications”. Ph.D. thesis, Swiss Federal Institute of Technology (ETH), 1999.
- [5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, abril 2002.
- [6] J. J. S. Santos, C. Silva, E. F. Wanner, O. M. Neto and R. H. C. Takahashi. “Avaliação do Impacto na Variação de Parâmetros em Algoritmos Genéticos Multiobjetivo”. In *Congresso Brasileiro de Automática*. Sociedade Brasileira de Automática, 2008.
- [7] M. Farina, K. Deb and P. Amato. “Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications”. *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, outubro 2004.
- [8] R. T. N. Cardoso, A. A. Cruz, E. F. Wanner and R. H. C. Takahashi. “A Multiobjective Non-Linear Dynamic Programming Approach for Optimal Biological Control in Soy Farming via NSGA-II”. In *Congress on Evolutionary Computation*, Cingapura, 2007.
- [9] E. Zitzler, M. Laumanns and L. Thiele. “SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization”. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95–100, 2002.
- [10] C. Silva, J. J. S. Santos and E. F. Wanner. “Proposal of a Transductive Least Squares Support Vector Machine”. In *Congresso Brasileiro de Automática*. Sociedade Brasileira de Automática, 2008.
- [11] C. M. Fonseca and P. J. Fleming. “An Overview of Evolutionary Algorithms in Multiobjective Optimization”. *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1995.
- [12] R. M. Storn and K. V. Price. “Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces”. Technical Report TR-95-012, International Computer Science Institute (ICSI), University of California, 1995.
- [13] F. G. Guimarães. “Algoritmos de Evolução Diferencial para Otimização e Aprendizado de Máquina”. In *Congresso Brasileiro de Redes Neurais*, 2009.
- [14] R. de Albuquerque Teixeira. “Treinamento de Redes Neurais Artificiais Através de Otimização Multi-objetivo: Uma Nova Abordagem entre a Polarização e a Variância”. Ph.D. thesis, PPGEE-UFMG, Belo Horizonte, agosto 2001.
- [15] B. V. Babu and M. M. L. Jehan. “Differential Evolution for Multi-Objective Optimization”. In *Congress on Evolutionary Computation*. IEEE, 2003.
- [16] K. Price and R. Storn. <http://www.ICSI.Berkeley.edu/storn/code.html>, 2003.