

SÍNTESE DE CONTROLADORES ROBUSTOS PELO MÉTODO DE EVOLUÇÃO DIFERENCIAL

DEVAIR DE MOURA MARCOS, SARA JORGE E SILVA, EDUARDO NUNES GONÇALVES*

*Programa de Pós-graduação em Engenharia Elétrica UFSJ/CEFET-MG
Departamento de Engenharia Elétrica - CEFET-MG, Av. Amazonas 7675
Belo Horizonte, MG, Brasil

Email: devauaiso@yahoo.com.br, sarajorge_ufsjs@yahoo.com.br, eduardong@des.cefetmg.br

Abstract— This work presents a new implementation of a $\mathcal{H}_2/\mathcal{H}_\infty$ robust control synthesis procedure applied to uncertain time-invariant linear systems represented by polytopic systems. Synthesis formulations based on linear optimization problems with constraints in the form of linear matrix inequalities are quite popular but can present too conservative results or even not present a result. The robust control problem means to optimize performance ensuring system stability for infinity system combinations, what results in a semi-infinite optimization problem that is difficult to solve. In previous works it was proposed a problem solution based on a two step procedure, synthesis and analysis, implemented by the cone-ellipsoidal algorithm and the Branch-and-bound method, respectively. This paper proposes to apply the differential evolution method for both steps of the synthesis procedure. This algorithm can be more efficient to achieve the optimum solution of non-convex problems, in synthesis step, and may presents less computational cost than the method Branch-and-bound in the analysis step. Thus, the new proposed implementation can be a more efficient alternative to more complex problems.

Keywords— Robust control, polytopic systems, differential evolution algorithm.

Resumo— Este trabalho apresenta uma nova implementação de um procedimento de síntese de controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$ para sistemas lineares invariantes no tempo, incertos, representados por sistemas politópicos. Formulações de síntese baseadas em problemas de otimização linear com restrições na forma de desigualdades matriciais lineares são bastante populares, mas podem apresentar resultados bastante conservadores ou até mesmo não apresentar um resultado. O problema de controle robusto significa otimizar o desempenho garantindo a estabilidade do sistema para infinitas combinações de sistemas, o que resulta em um problema de otimização semi-infinita de difícil solução. Em trabalhos anteriores foi proposto a solução do problema através de um procedimento de dois passos, síntese e análise, implementados pelos algoritmo cone-elipsoidal e pelo método *Branch-and-bound*, respectivamente. Neste trabalho é proposto aplicar o método de evolução diferencial para ambos os passos do procedimento de síntese. Tal algoritmo pode ser mais eficiente para obter a solução ótima de problemas não convexos, no passo de síntese, e pode apresentar menor custo computacional que o método *Branch-and-bound* na etapa de análise. Desse modo, a nova implementação proposta pode ser uma alternativa mais eficiente para problemas mais complexos.

Palavras-chave— Controle robusto, sistemas politópicos, algoritmo de evolução diferencial

1 Introdução

Um problema bastante estudado na literatura é o controle de sistemas lineares invariantes no tempo com parâmetros que não são precisamente conhecidos, mas que variam em um intervalo conhecido. O sistema de controle deve garantir não somente a estabilidade do sistema mas também o seu desempenho ótimo para todos os valores possíveis dos parâmetros incertos, sendo que o desempenho pode ser quantificado em termos de normas $\mathcal{H}_2$ e $\mathcal{H}_\infty$ das matrizes de transferência (Zhou and Doyle, 1998).

Uma forma bastante popular de se modelar sistemas incertos é tratá-los como sistemas lineares invariantes no tempo (SLIT), representados por modelos no espaço de estados, com incerteza politópica. No modelo politópico, cada sistema pertencente ao domínio de incerteza pode ser descrito como a combinação convexa dos sistemas que correspondem aos vértices do politopo. A popularidade de sistemas politópicos se deve em parte às formulações de análise e síntese descritas em termos de problemas de otimização linear com res-

trições na forma de desigualdades matriciais lineares (LMIs, do inglês *linear matrix inequalities*) (Boyd et al., 1994). Através das formulações LMI é possível analisar e projetar sistemas de controle considerando somente os sistemas correspondentes aos vértices do politopo. Problemas de controle robusto representados por formulações LMI podem gerar resultados muito conservadores e, em alguns casos, pode até mesmo não ser obtida uma solução para o problema. Esses dois fatos motivam o desenvolvimento de estratégias de síntese alternativas. Por exemplo, os métodos probabilísticos para projetos de sistemas de controle incertos (Calafiore et al., 2011).

Em uma abordagem alternativa, o problema de controle robusto, na forma original de problema de otimização semi-infinita, foi solucionado através de um procedimento iterativo de dois passos similar ao método proposto por Zakovic and Rustem (2002). Tal procedimento obteve resultados melhores que os obtidos por formulações LMI em diferentes tipos de problemas na área de controle robusto (Gonçalves et al., 2005; Gonçalves et al., 2011). A ideia do método é obter a solução

do problema em dois passos: síntese e análise. No passo de síntese, o domínio infinito é substituído por um conjunto finito de sistemas que pode ser inicialmente os vértices do politopo. O controlador é projetado para minimizar a função objetivo e garantir o atendimento às restrições somente para esse conjunto finito, o que torna o problema mais fácil de ser solucionado. No passo de análise, o controlador obtido no passo de síntese é analisado para todos os infinitos sistemas do politopo. Se for verificado que alguma restrição foi violada ou que o máximo da função objetivo nos passos de síntese e de análise possuem uma diferença significativa, novos sistemas são acrescentados no conjunto finito e os dois passos são executados novamente. O procedimento finaliza quando é verificado que todas as restrições são atendidas para todos os sistemas no domínio infinito e que a diferença entre o mínimo da função objetivo calculada em cada passo está dentro de um limite especificado. Nos trabalhos anteriores, esse procedimento foi implementado considerando o algoritmo cone-elipsoidal (CE) para o passo de síntese e o método *Branch-and-bound* (BB) para o passo de análise. Essa implementação pode apresentar problemas em determinadas situações. Primeiro, o algoritmo cone-elipsoidal é mais adequado para problemas de otimização convexas. Segundo, apesar do método *Branch-and-bound* possuir garantia de convergência para o ótimo global, o custo computacional do método aumenta rapidamente com o aumento da ordem do sistema e do número de vértices do politopo. Desse modo é interessante estudar outras possibilidades de implementação desse procedimento. Nesse trabalho é avaliado a alternativa de se adotar o método de evolução diferencial (DE, do inglês *differential evolution*) (Storn and Price, 1997) em ambos os passos do procedimento. Por ser um método evolucionário baseado em populações, o método DE pode ser aplicado a problemas não diferenciáveis, não convexas e multimodais com maior probabilidade de localização do mínimo global. Apesar de não ter a garantia de convergência, como ocorre com o método *Branch-and-bound*, para problemas mais complexos, o método de evolução diferencial pode requerer custo computacional muito menor, o que torna essa nova alternativa bastante interessante.

Esse trabalho é organizado como descrito a seguir. Na seção 2 é apresentada a formulação do problema de controle robusto em termos de um problema de otimização semi-infinita. Na seção 3 é apresentado o procedimento iterativo de síntese. Na seção 4 é detalhado como o método de evolução diferencial é utilizado para implementar os dois passos do procedimento. Na seção 5 é apresentado um exemplo ilustrativo para a validação da implementação proposta para o procedimento de síntese seguido pelas conclusões.

A notação utilizada nesse artigo é padrão. É

considerada a notação compacta para matrizes de transferência:

$$G(s) = C(sI - A)^{-1}B + D = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right].$$

2 Formulação do problema

Considere o sistema linear invariante no tempo descrito pelo modelo no espaço de estados

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t) \\ z(t) &= C_z x(t) + D_{zu} u(t) + D_{zw} w(t) \\ y(t) &= C_y x(t) + D_{yw} w(t) \end{aligned} \quad (1)$$

sendo $x(t) \in \mathbb{R}^n$ o vetor de estados, $u(t) \in \mathbb{R}^{n_u}$ o vetor de variáveis manipuladas, as saídas do controlador, $w(t) \in \mathbb{R}^{n_w}$ o vetor de entradas exógenas (distúrbios, $d(t)$, e ruídos de medição, $n(t) \in \mathbb{R}^{n_y}$), $z(t) \in \mathbb{R}^{n_z}$ o vetor de saídas controladas (saídas da planta, $c(t)$ e variáveis manipuladas, $u(t)$) e $y(t) \in \mathbb{R}^{n_y}$ o vetor de saídas medidas que são as entradas do controlador dinâmico.

Para simplificar a notação, as matrizes do modelo no espaço de estados, Eq. (1), podem ser reunidas na representação matricial seguinte:

$$\mathcal{S} \triangleq \left[\begin{array}{c|cc} A & B_u & B_w \\ \hline C_z & D_{zu} & D_{zw} \\ C_y & 0 & D_{yw} \end{array} \right]. \quad (2)$$

Considere que a matriz sistema $\mathcal{S}$ pode incluir parâmetros incertos pertencentes a um conjunto convexo compacto, ou politopo, definido pela combinação convexa de seus vértices:

$$\mathcal{P}(\alpha) \triangleq \left\{ \mathcal{S} : \mathcal{S} = \sum_{i=1}^{\eta} \alpha_i \mathcal{S}_i; \quad \alpha \in \Omega \right\}, \quad (3)$$

$$\Omega \triangleq \left\{ \alpha \in \mathbb{R}^{\eta} : \alpha_i \geq 0, \sum_{i=1}^{\eta} \alpha_i = 1 \right\}, \quad (4)$$

sendo $\mathcal{S}_i$, $i = 1, \dots, \eta$, os vértices do politopo e $\alpha = [\alpha_1, \dots, \alpha_{\eta}]^T$ o vetor que parametriza o politopo. A dependência das matrizes do sistema de α será omitida para simplificar a notação.

Neste trabalho, consideramos o controlador dinâmico por realimentação de saída definido pelas seguintes matrizes:

$$K(s) = \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right]. \quad (5)$$

As matrizes do sistema em malha-fechada

$$T_{zw}(s) = \left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & D_f \end{array} \right], \quad (6)$$

são dadas por

$$\begin{aligned} A_f &= \left[\begin{array}{cc} A + B_u D_c C_y & B_u C_c \\ B_c C_y & A_c \end{array} \right] \\ B_f &= \left[\begin{array}{c} B_w + B_u D_c D_{yw} \\ B_c D_{yw} \end{array} \right] \\ C_f &= \left[C_z + D_{zu} D_c C_y \quad D_{zu} C_c \right] \\ D_f &= \left[D_{zw} + D_{zu} D_c D_{yw} \right] \end{aligned} \quad (7)$$

O objetivo do sistema de controle   manter o sistema no ponto de opera  o desejado minimizando o efeito de dist rbios e de ru dos de medi  o. Desse modo, o problema de s ntese de controlador robusto multiobjetivo pode ser estabelecido como sendo: dado um sistema linear invariante no tempo com modelo de incerteza polit pico, $\mathcal{P}(\alpha)$, $\alpha \in \Omega$, encontre o controlador din mico por realimenta  o de sa da, $K(s)$, que minimiza a norma $\mathcal{H}_\infty$ m xima da matriz de transfer ncia em malha-fechada que relaciona o vetor de dist rbios com o vetor de sa das da planta, $T_{cd}(s)$, e a norma $\mathcal{H}_2$ m xima da matriz de transfer ncia em malha-fechada que relaciona o vetor de ru dos de medi  o com o vetor de vari veis manipuladas, $T_{un}(s)$, garantindo a estabilidade do sistema em malha-fechada para todo o dom nio polit pico de incerteza:

$$K^*(s) = \arg \min_{K(s)} \begin{bmatrix} \max_{\alpha \in \Omega} \|T_{cd}(s, \alpha, K)\|_\infty \\ \max_{\alpha \in \Omega} \|T_{un}(s, \alpha, K)\|_2 \end{bmatrix} \quad (8)$$

sujeito a: $K \in \mathcal{F}$,

sendo $\mathcal{F}$ o conjunto de controladores que resultam em sistemas a malha-fechada robustamente est veis, com estruturas e restri  es especificadas. Os objetivos de controle s o conflitantes, sendo o segundo objetivo diretamente relacionado   minimiza  o do esfor o de controle. Se necess rio,   poss vel incluir restri  es adicionais para garantir especifica  es sobre as caracter sticas da resposta transit ria. O problema (8)   um problema de otimiza  o semi-infinita de dif cil solu  o.

3 Procedimento de S ntese Iterativo

Para facilitar a solu  o do problema (8), pode ser considerado um procedimento baseado em dois passos:

Passo 1 = S ntese. O dom nio com infinitos sistemas, Ω , do problema (8)   substituído por um conjunto finito de pontos, $\tilde{\Omega}$. Al m disso, para aplicar t cnicas de otimiza  o escalar, o problema de otimiza  o multiobjetivo (8)   transformado em um problema de otimiza  o escalar tratando a segunda fun  o objetivo como restri  o:

$$K^* = \arg \min_K \max_{\alpha \in \tilde{\Omega}} \|T_{cd}(s, \alpha, K)\|_\infty$$

sujeito a: $\begin{cases} \max_{\alpha \in \tilde{\Omega}} \|T_{un}(s, \alpha, K)\|_2 \leq \epsilon_2 \\ K \in \mathcal{F} \end{cases} \quad (9)$

sendo ϵ_2 o valor da restri  o que pode ser variado para gerar diferentes solu   es candidatas a Pareto- timo do problema multi-objetivo.

Passo 2 = An lise. O controlador calculado no passo 1, $K^*(s)$, tem que ser validado para todo o dom nio infinito de incerteza, Ω .

O problema de an lise de estabilidade robusta do sistema polit pico pode ser formulado como um problema de otimiza  o em que   desejado determinar a maior parte real dos n autovalores das infinitas matrizes $A_f(\alpha)$:

$$\alpha_u = \arg \max_{\alpha \in \Omega} f_u(\alpha), \quad f_u(\alpha) \triangleq \max_i \Re(\lambda_i(A_f(\alpha))) \quad (10)$$

sendo $\Re(\lambda)$ a parte real de $\lambda \in \mathbb{C}$ e $\lambda_i(A)$ o conjunto dos autovalores de $A \in \mathbb{R}^{n \times n}$. Se $f(\alpha_u) \geq 0$, isso significa que existe polo no semi-plano direito do plano- S e o sistema em malha-fechada com $K^*(s)$ n o   robustamente est vel. Caso contr rio, existe uma grande probabilidade do sistema ser robustamente est vel, mas n o pode-se afirmar com certeza uma vez que trata-se de um problema de otimiza  o n o convexo com m ximos em diferentes regi  es de Ω .

Se o sistema   robustamente est vel, deve-se verificar os valores das fun   es objetivos. Os problemas de c lculo dos custos garantidos $\mathcal{H}_2$ e $\mathcal{H}_\infty$ de um sistema polit pico podem ser formulados como problemas de otimiza  o:

$$\alpha_\infty = \arg \max_{\alpha \in \Omega} f_\infty(\alpha), \quad f_\infty(\alpha) \triangleq \|T_{cd}(s, \alpha, K^*)\|_\infty, \quad (11)$$

$$\alpha_2 = \arg \max_{\alpha \in \Omega} f_2(\alpha), \quad f_2(\alpha) \triangleq \|T_{un}(s, \alpha, K^*)\|_2. \quad (12)$$

Defina $\tilde{f}_\infty \triangleq \max_{\alpha \in \tilde{\Omega}} \|T_{cd}(s, \alpha, K)^*\|_\infty$, o valor  timo obtido na etapa de s ntese para o conjunto finito, novos pontos no conjunto finito $\tilde{\Omega}$ s o acrescentados nas seguintes situa   es:

1. Se $f_u(\alpha_u) \geq 0$, ent o $\tilde{\Omega} \leftarrow \tilde{\Omega} \cup \{\alpha_u\}$.
2. Se $|f_\infty(\alpha_u) - \tilde{f}_\infty| > \epsilon$, ent o $\tilde{\Omega} \leftarrow \tilde{\Omega} \cup \{\alpha_\infty\}$.
3. Se $f_2(\alpha_2) > \epsilon_2$, ent o $\tilde{\Omega} \leftarrow \tilde{\Omega} \cup \{\alpha_2\}$.

O conjunto $\tilde{\Omega}$   inicializado com um conjunto finito de pontos sendo adotado os η v rtices do politopo (α 's iguais as colunas da matriz identidade de ordem η). Se no passo 2 s o acrescentados pontos no conjunto $\tilde{\Omega}$, α_u ou α_∞ e/ou α_2 , os passos 1 e 2 devem ser executados novamente, caso contr rio o procedimento   finalizado. Na pr xima se  o   descrito como os passos do procedimento de s ntese podem ser implementados pelo m todo de evolu  o diferencial.

4 Algoritmo de Evolu  o Diferencial

O algoritmo de Evolu  o Diferencial   um algoritmo de otimiza  o evolucion rio para solu  o de problemas com fun   es com dom nio real (Storn and Price, 1997; Das and Suganthan, 2011). O algoritmo DE inclui operadores similares aos empregados por algoritmos evolucion rios padr es: muta  o, cruzamento ou recombina  o e sele  o.

Seja $\mathcal{U}_{(a,b)}$ um n mero pseudo-aleat rio com distribui  o uniforme no intervalo (a, b) ; $\mathcal{I}_{(m)}$ um n mero pseudo-aleat rio com distribui  o uniforme no intervalo $[1, m]$; $\chi \in \mathbb{R}^n$ o vetor de vari veis de otimiza  o; e N o n mero de indiv duos (solu   es candidatas) da popula  o. Defina a popula  o na k - sima itera  o, $X_k = \{\chi_{k,i}; i = 1, \dots, N\}$, sendo a i - sima solu  o:

$$\chi_{k,i} = \begin{bmatrix} \chi_{k,i,1} \\ \vdots \\ \chi_{k,i,m} \end{bmatrix}. \quad (13)$$

Os operadores do DE s o descritos a seguir.

4.1 Popula  o Inicial

Na solu  o do problema (9) na etapa de s ntese, as vari veis de otimiza  o s o os par metros do controlador, $\chi_{k,i} = [\theta_1, \dots, \theta_p]^T$, sendo $m = p$ o n mero de

par metros. A popula  o inicial   distribu da uniformemente no intervalo entre $-L$ e L , sendo o valor de L especificado pelo projetista, isto  , $\chi_{1,i,j} = \mathcal{U}_{(-L,L)}$, $i = 1, \dots, N$ e $j = 1, \dots, p$. O tamanho da popula  o foi fixado em $N = 3p$.

Nas solu  es dos problemas (10), (11) e (12) na etapa de an lise, as vari veis de otimiza  o s o as coordenadas do politopo, $\chi_{k,i} = [\alpha_1, \dots, \alpha_\eta]^T$, $m = \eta$. Neste trabalho, foi adotado que a popula  o inicial   composta pelos v rtices, tr s pontos determin sticos sobre cada aresta e η solu  es distribu das de forma aleat ria tamb m sobre as arestas do politopo Ω . Seja $I_i \in \mathbb{R}^\eta$ a i - sima coluna da matriz identidade, a popula  o inicial   criada pelo seguinte algoritmo:

```

para  $i \leftarrow 1$  at   $\eta$ 
   $\chi_{1,i} \leftarrow I_i$ 
fim para
 $i \leftarrow \eta$ 
para  $p \leftarrow 1$  at   $\eta - 1$ 
  para  $q \leftarrow p + 1$  at   $\eta$ 
     $\chi_{1,i+1} \leftarrow 0,25\chi_{1,p} + 0,75\chi_{1,q}$ 
     $\chi_{1,i+2} \leftarrow 0,5\chi_{1,p} + 0,5\chi_{1,q}$ 
     $\chi_{1,i+3} \leftarrow 0,75\chi_{1,p} + 0,25\chi_{1,q}$ 
     $i \leftarrow i + 3$ 
  fim para
fim para
para  $j \leftarrow 1$  at   $\eta$ 
   $r_1 \leftarrow \mathcal{I}_{(\eta)}$ 
  repita
     $r_2 \leftarrow \mathcal{I}_{(\eta)}$ 
  at   $r_1 \neq r_2$ 
     $\alpha \leftarrow \mathcal{U}_{(0,1)}$ 
     $i \leftarrow i + 1$ 
     $\chi_{1,i} \leftarrow \alpha\chi_{1,r_1} + (1 - \alpha)\chi_{1,r_2}$ 
  fim para

```

O tamanho total da popula  o   dado por $N = \eta + 3\eta(\eta - 1)/2 + \eta$.

4.2 Muta  o diferencial

Considere os  ndices $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ dados por $r_j = \mathcal{I}_{(N)}$, $j = 1, \dots, 5$. No passo de s ntese e na verifica  o de estabilidade no passo de an lise, adotamos que a i - sima solu  o mutante   obtida como sendo:

$$\mathbf{v}_{k,i} = \chi_{k,r_1} + F_i(\chi_{k,r_2} - \chi_{k,r_3}) \quad (14)$$

$i = 1, \dots, N$. Neste caso, adotamos o fator de escala aleat rio para cada solu  o, sendo $F_i = \mathcal{U}_{(0,5,1)}$

No c lculo do m ximo de norma no passo de an lise, a i - sima solu  o mutante   obtida como sendo:

$$\mathbf{v}_{k,i} = \chi_{k,r_1} + F_{1,i}(\chi_{k,r_2} - \chi_{k,r_3}) + F_{2,i}(\chi_{k,r_4} - \chi_{k,r_5}), \quad (15)$$

$i = 1, \dots, N$, sendo $F_{j,i} = \mathcal{U}_{(0,1,1)}$, $j = 1, 2$.

4.3 Cruzamento

O cruzamento entre as i - simas solu  es da k - sima popula  o, X_k , e da popula  o mutante, V_k , gera a popula  o tentativa, U_k :

$$u_{k,i,j} = \begin{cases} v_{k,i,j}, & \text{se } \mathcal{U}_{(0,1)} \leq C_r \text{ ou } j = \delta_i \\ \chi_{k,i,j}, & \text{caso contr rio} \end{cases}, \quad (16)$$

para $j = 1, \dots, m$, $i = 1, \dots, N$, sendo $C_r \in [0, 1]$ a taxa de cruzamento. Adotamos $C_r = 0,5$. O  ndice $\delta_i = \mathcal{I}_{(\eta)}$ garante que $\mathbf{u}_{k,i} \neq \chi_{k,i}$.

4.4 Tratamento das restri  es

Na etapa de s ntese, n o   realizada nenhuma restri  o adicional sobre os par metros do controlador, admitindo que os valores de θ_i , $i = 1, \dots, p$, assumam valores fora da faixa $|\theta_i| \leq L$. Com rela  o   restri  o sobre a norma $\mathcal{H}_2$ e a restri  o do sistema a malha-fechada ser est vel para $\alpha \in \bar{\Omega}$, ver problema (9),   utilizado o m todo de penalidades. Para a solu  o que corresponde a um sistema inst vel   atribuído um valor maior para a fun  o objetivo do que a solu  o que viola a restri  o sobre a norma $\mathcal{H}_2$, que por sua vez possui um valor de fun  o objetivo maior que a solu  o que n o viola nenhuma das duas restri  es. Quando comparadas duas solu  es que violam a mesma restri  o, a penalidade   maior para a solu  o mais distante da regi o fact vel.

No etapa de an lise, uma solu  o   fact vel se $\chi_{k,i} \in \Omega$, isto  , $\chi_{k,i,j} \geq 0$, $\forall j$, e $\sum_{j=1}^\eta \chi_{k,i,j} = 1$. Neste trabalho, optamos por for ar que toda solu  o atenda essas restri  es para evitar o uso de tratamento de restri  o pelo m todo de penalidades. Para isso, aplicamos as seguintes opera  es sobre as solu  es tentativa obtidas pela opera  o de muta  o e cruzamento:

$$\begin{aligned} u_{k,i,j} &= |u_{k,i,j}|, & j &= 1, \dots, \eta; \\ \mathbf{u}_{k,i} &= \mathbf{u}_{k,i} / \|\mathbf{u}_{k,i}\|_1, & i &= 1, \dots, N. \end{aligned} \quad (17)$$

A primeira opera  o garante que $u_{k,i,j} > 0$ e a segunda que $\sum_{j=1}^\eta u_{k,i,j} = 1$.

4.5 Sele  o

A opera  o de sele  o determina qual solu  o, se o alvo, $\chi_{k,i}$, ou a tentativa, $\mathbf{u}_{k,i}$, sobrevive para pr xima gera  o:

$$\chi_{k+1,i} = \begin{cases} \mathbf{u}_{k,i}, & \text{se } f(\mathbf{u}_{k,i}) \leq f(\chi_{k,i}) \\ \chi_{k,i}, & \text{caso contr rio} \end{cases}, \quad (18)$$

$i = 1, \dots, N$. Como o m todo DE   apresentado nesse trabalho para resolver problemas de minimiza  o, no passo de an lise   suficiente trocar o sinal das fun  es $f_u(\alpha)$, $f_\infty(\alpha)$ e $f_2(\alpha)$.

4.6 Crit rio de parada

Adotamos como crit rios de parada um n mero m ximo de gera  es, N_g , ou a converg ncia da popula  o comparando os valores m ximos e m nimos de $f(\chi)$ da k - sima popula  o, $f_{\max}(\chi) - f_{\min}(\chi) \leq \epsilon$, $\chi \in \mathbf{X}_k$, sendo ϵ um n mero pequeno. No passo de s ntese, adotamos $N_g = 8.000$ e $\epsilon = 5 \times 10^{-4}$, e no passo de an lise, na verifica  o da estabilidade adotamos $N_g = 3.000$ e $\epsilon = 10^{-4}$, e no c lculo da m ximo da norma adotamos $N_g = 200\eta$ e $\epsilon = 10^{-8}$. No c lculo do m ximo das normas, o algoritmo tamb m   interrompido caso seja verificado um sistema inst vel.

5 Exemplo Ilustrativo

N s consideramos um sistema idealizado que requer v rias itera  es para converg ncia do procedimento

analisado. Considere o sistema de 6^a ordem com $w = [d \ n_1 \ n_2]^T$, d o dist rbio do sistema e n_1 e n_2 ru dos de medi  o, $z = [c \ u]^T$, c a sa da controlada e u a vari vel manipulada, e $y = [x_5 + n_1 \ x_6 + n_2]^T$. As matrizes do modelo no espa o de estados s o:

$$A = \begin{bmatrix} -0,05 & -0,01 - \gamma & 0 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & -0,01 + \gamma & -0,1 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ -0,01 - \beta & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ -0,01 + \beta & -0,05 & -1 & \dots \\ 0 & 1 & 0 & \dots \end{bmatrix}, \quad (19)$$

$$B_u = \begin{bmatrix} 0,1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad B_w = \begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}, \quad (20)$$

$$C_z = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0,1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (21)$$

$$D_{zu} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad D_{zw} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (22)$$

$$C_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (23)$$

$$D_{yu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad D_{yw} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (24)$$

sendo que os par metros incertos γ e β pertencem ao politopo no espa o $\mathbb{R}^2$ com 5 v rtices e 5 arestas definidas por: $0 \leq \gamma \leq 1$, $0 \leq \beta \leq 1$ e $\gamma + \beta \leq 1,5$. O objetivo de projeto   obter o controlador que minimiza a norma $\mathcal{H}_\infty$ da fun  o de transfer ncia que relaciona a perturba  o e a sa da controlada, $\|T_{cd}(s, \alpha)\|_\infty$, e a norma $\mathcal{H}_2$ da fun  o de transfer ncia que relaciona os ru dos de medi  o com a vari vel manipulada, $\|T_{un}(s, \alpha)\|_2$, garantindo que o sistema seja robustamente est vel para todo $\alpha \in \Omega \subset \mathbb{R}^5$.   adotado um controlador de 2^a ordem com a seguinte estrutura:

$$\begin{aligned} \frac{dx_c}{dt} &= \begin{bmatrix} \theta_1 & 1 \\ \theta_2 & 0 \end{bmatrix} x_c + \begin{bmatrix} \theta_3 & \theta_4 \\ \theta_5 & \theta_6 \end{bmatrix} y, \\ u(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_c + \begin{bmatrix} 0 & 0 \end{bmatrix} y, \end{aligned} \quad (25)$$

sendo $\theta_1, \dots, \theta_6$ os par metros de otimiza  o.

Para gerar uma curva candidata a Pareto- tima,   adotado os seguintes valores para a restri  o: $\epsilon_2 \in \{0,01; 0,05; 0,10; 0,25; 0,50; 0,75; 1,00; 1,25; 1,50; 1,75; 2,00\}$. Em todos os casos foi necess rio acrescentar pontos em $\tilde{\Omega}$ para obter um sistema em malha-fechada robustamente est vel, para garantir a restri  o ϵ_2 da norma $\mathcal{H}_2$ e para garantir a minimiza  o da norma $\mathcal{H}_\infty$. Foram necess rias de 5 a 10 itera  es do procedimento de s ntese proposto. Para $\epsilon_2 = 0,01$ e $\epsilon_2 = 0,05$, adotamos distribuir a popula  o inicial na etapa de s ntese no intervalo $-0,01 \leq \theta_i \leq 0,01$, $\forall i$. Nos demais casos, escolhemos $-5 \leq \theta_i \leq 5$, $\forall i$. A Fig. 1 apresenta as fronteiras candidatas a Pareto- tima obtida pelo procedimento implementado pelo m todo DE e pelos m todos CE/BB onde pode ser observado o melhor resultado do m todo DE. Para esse

exemplo espec fico, do ponto de vista de custo computacional, uma implementa  o n o supera a outra em todos os casos de ϵ_2 . O m todo BB teve dificuldade de converg ncia no passo de an lise para alguns casos.

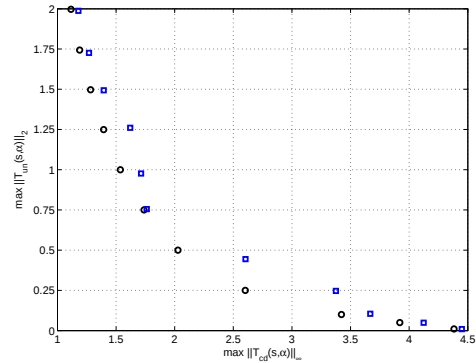


Figura 1: Fronteira candidata a Pareto- tima obtida pelos m todos DE, ' ', e CE/BB, ' '.

Para ilustrar o funcionamento do m todo, para $\epsilon_2 = 0,01$, foram necess rios dez itera  es para converg ncia do procedimento de s ntese proposto. Somente na 2^a itera  o, ap s acrescentar um novo ponto no conjunto $\tilde{\Omega}$, obteve-se um controlador que estabilizava o sistema para todo $\chi \in \Omega$. A Fig. 2 mostra a nuvem de polos para o sistema em malha-fechada obtido na 1^a itera  o onde pode ser observado a presen a de polos no semi-plano direito. Cabe ressaltar que a an lise de estabilidade n o   realizada com base na nuvem de polos e sim pela solu  o do problema (10). Na 2^a itera  o, na etapa de otimiza  o, foi obtido $\max_{\alpha \in \tilde{\Omega}} \|T_{cd}(s, \alpha)\|_\infty = 3,9657$ e $\max_{\alpha \in \tilde{\Omega}} \|T_{un}(s, \alpha)\|_2 = 0,0099$ ao passo que, na etapa de an lise, foi verificado que, para todo o conjunto Ω , os valores s o na verdade $\max_{\alpha \in \Omega} \|T_{cd}(s, \alpha)\|_\infty = 16,0054$ e $\max_{\alpha \in \Omega} \|T_{un}(s, \alpha)\|_2 = 0,0135$. A Fig. 3a apresenta a superf cie de $\|T_{cd}(s, \alpha)\|_\infty$ na 2^a itera  o. Al m da diferen a alta no valor da norma $\mathcal{H}_\infty$ entre as etapas de s ntese e an lise, ainda   observado que a restri  o sobre a norma $\mathcal{H}_2$ n o   atendida. Ap s 10 itera  es, com adi  o de 11 pontos em $\tilde{\Omega}$, o m todo converge para $\max_{\alpha \in \tilde{\Omega}} \|T_{cd}(s, \alpha)\|_\infty = 4,3822$ e $\max_{\alpha \in \tilde{\Omega}} \|T_{un}(s, \alpha)\|_2 = 0,0100$, que corresponde ao menor valor na Fig. 1. A Fig. 3b apresenta a superf cie de $\|T_{cd}(s, \alpha)\|_\infty$ na 10^a itera  o, ao passo que a Fig. 4 apresenta as curvas de n vel correspondente e a localiza  o de todos os pontos adicionados no espa o $[\gamma \ \beta]^T \in \mathbb{R}^2$. Pode ser observado pelas curvas de n vel que o problema de determina  o do valor m ximo de norma   n o-diferenci vel, n o-convexo e multimodal. As matrizes do controlador final s o dadas por:

$$K = \left[\begin{array}{cc|cc} -0,1049 & 1 & -0,0025 & -0,0001 \\ -0,6576 & 0 & 0,0001 & -0,0028 \\ \hline 1 & 0 & 0 & 0 \end{array} \right]. \quad (26)$$

6 Conclus es

Foi proposta uma nova implementa  o para um procedimento de s ntese de sistemas de controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$ baseado em dois passos interativos. No passo

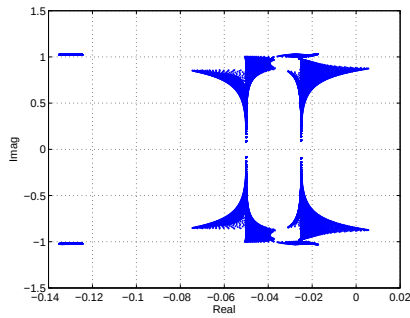


Figura 2: Nuvem de polos de $\|T_{zw}(s, \alpha)\|_\infty$ na 1^a itera  o para $\varepsilon_2 = 0,01$.

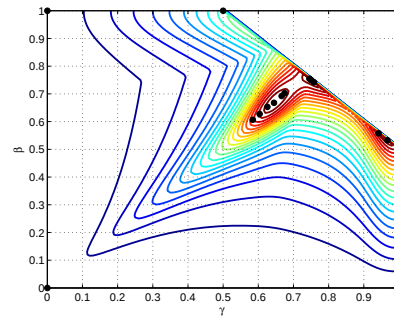
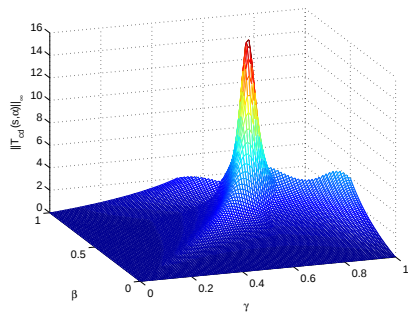
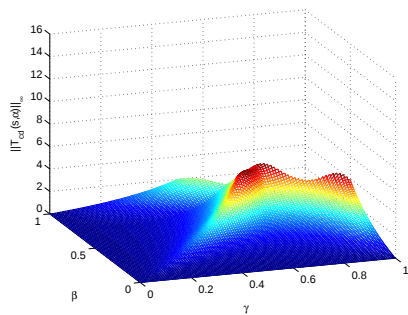


Figura 4: Curvas de n vel de $\|T_{cd}(s, \alpha)\|_\infty$ com os pontos de $\hat{\Omega}$, $\bullet$, na 10^a itera  o para $\varepsilon_2 = 0,01$.



(a) 2^a itera  o.



(b) 10^a itera  o.

Figura 3: Superf cie de $\|T_{cd}(s, \alpha)\|_\infty$.

de s ntese   determinado o controlador que minimiza a fun  o objetivo e atende  s restri  es considerando um n mero finito de pontos do dom nio polit pico de incerteza. No passo de an lise, o controlador obtido   avaliado para todo o dom nio infinito de incerteza e   verificada a necessidade de se acrescentar novos pontos no conjunto finito de pontos considerado no passo de s ntese. Neste trabalho, foi estudada a implementa  o de ambos os passos pelo m todo de evolu  o diferencial. No passo de s ntese, o m todo de evolu  o diferencial possui maior probabilidade de obter uma solu  o melhor do que o algoritmo cone-elipsoidal, utilizado em implementa  es anteriores. No passo de an lise, o m todo *Branch-and-bound*   capaz de determinar exatamente se um sistema   robustamente est vel ou n o e tamb m capaz de calcular o m ximo da norma no dom nio infinito com a precis o especificada, mas pode apresentar custo computacional proibitivo. Apesar do m todo de evolu  o diferencial n o ter a garantia de solu  o  tima global, ele pode ser

efetivo com menor custo computacional. Foi apresentado um exemplo ilustrativo especialmente formulado para gerar itera  es do procedimento. A implementa  o do procedimento de s ntese proposta nesse trabalho funcionou de forma satisfat ria tendo gerado uma fronteira candidata a Pareto- tima melhor do que a implementa  o anterior.

Agradecimentos

Os autores agradecem os apoios das ag ncias CAPES, CNPq, [306005/2012-3] - Produtividade em Pesquisa - PQ - 2012, e FAPEMIG, [APQ-02943-15] - Edital 01/2015 - Demanda Universal.

Refer ncias

- Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*, SIAM, Philadelphia, PA.
- Calafiore, G. C., Dabbeneb, F. and Tempo, R. (2011). Research on probabilistic methods for control system design, *Automatica* **47**(7): 1279–1293.
- Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation* **15**(1): 4–31.
- Gon alves, E. N., Bachur, W. E. G., Palhares, R. M. and Takahashi, R. H. C. (2011). Robust $\mathcal{H}_2/\mathcal{H}_\infty$ /reference model dynamic output-feedback control synthesis, *International Journal of Control* **84**(12): 2067–2080.
- Gon alves, E. N., Palhares, R. M. and Takahashi, R. H. C. (2005). Improved optimisation approach to robust $\mathcal{H}_2/\mathcal{H}_\infty$ control problem for linear systems, *IEE Proceedings Control Theory & Applications* **152**(2): 171–176.
- Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* **11**(4): 341–359.
- Zakovic, S. and Rustem, B. (2002). Semi-infinite programming and applications to minimax problems, *Annals of Operations Research* **124**: 81–110.
- Zhou, K. and Doyle, J. C. (1998). *Essentials of Robust Control*, Prentice-Hall, Inc., New Jersey.