PROGRAMA COMPUTACIONAL INTERATIVO PARA SIMULAÇÃO E OTIMIZAÇÃO DE CONTROLADORES PID

André Laurindo Maitelli
LECA – Laboratório de Engenharia de Controle e Automação – UFRN/CT/DEE
Campus Universitário – Lagoa Nova – 59072-970 Natal/RN – maitelli@leca.ufrn.br

Fábio Câmara Araújo de Carvalho
LabSAD – Laboratório de Sistemas de Apoio à Decisão – UFSC/CTC/EPS – fbcamara@eps.ufsc.br
R. Lauro Linhares, 689, B.7 ap.406 – Trindade - CEP-88036-002 - Florianópolis/SC

Resumo — A simulação está sendo cada mais utilizada no ensino das engenharias. Ela permite que o aluno realize testes, modifique parâmetros, visualize inúmeras situações, de modo a obter um ganho em termos de aprendizagem, visto que o mesmo pode, em curto espaço de tempo, trabalhar com inúmeros conceitos ao mesmo tempo. O presente trabalho visa apresentar um programa computacional interativo para simulação e otimização de controladores PID a ser utilizado nas disciplinas dos cursos de Engenharia Elétrica, Engenharia de Computação e Engenharia de Controle e Automação. Tais controladores têm o uso difundido nas indústrias, sendo muitas vezes problemático fazer a sintonia dos mesmos apenas por tentativa e erro.

1. Introdução

Os controladores PID são bastante utilizados nas indústrias para controle de nível de líquidos, controle de temperatura, de velocidade de rotação de motores, em máquinas elétricas em geral. O projeto desses controladores pode ser feito baseado na resposta de um modelo de referência de segunda ordem e, a partir do conhecimento da planta que deverá ser controlada, encontrar os parâmetros de ganho proporcional, integral e derivativo de um controlador PID que é utilizado em cascata com esta planta, em malha fechada, para que a mesma acompanhe a resposta do modelo da melhor forma possível.

Os parâmetros do PID podem ser escolhidos com o auxílio de simulações com valores escolhidos aleatoriamente, e podem ser encontrados através das regras de Ziegler & Nichols, que podem não gerar boas respostas e/ou não serem aplicáveis em determinadas plantas. Portanto, faz-se necessário um ajuste fino para encontrar um valor ótimo de parâmetros para proporcionar uma resposta conjunta próxima a do modelo de referência desejado.

Tais ajustes podem ser feitos de várias formas, dentre elas pode-se destacar a otimização que consiste, basicamente, em considerar os três parâmetros como variáveis de uma função tridimensional desconhecida, que pode possuir vários mínimos locais representando valores ótimos, ou sub-ótimos de parâmetros que proporcionam uma resposta próxima ao modelo de referência.

O presente trabalho apresenta um programa computacional desenvolvido a partir de estudos de técnicas de otimização de controladores PID, utilizando algoritmos multidimensionais para funções as quais não se conhece o gradiente. Tal programa pode ser utilizado nas disciplinas de Controle e Automação para encontrar o valor dos parâmetros necessários e simular a resposta ao degrau de uma planta em malha aberta e em malha fechada, com um controlador projetado através dos parâmetros de Ziegler & Nichols e, também com o PID com parâmetros otimizados através do algoritmo dos poliedros flexíveis.

Utilizando a idéia deste trabalho, juntamente com outras ferramentas, foi desenvolvido um outro programa denominado SINCON (Síntese de Controladores), o qual permite o projeto otimizado de outros controladores além do PID, tais como atraso e avanço de fase, [1].

2. Os Controladores PID

Os controladores PID são projetados para inúmeros sistemas. São de baixo custo e de relativa facilidade de projeto.

O projeto desses controladores pode ser feito baseado na resposta de um modelo de referência de segunda ordem e, a partir do conhecimento da planta que deverá ser controlada, encontrar os parâmetros proporcional, integrativo e derivativo – chamados *kp*, *ki* e *kd* – de um controlador PID que é utilizado em série com esta planta, em malha fechada, para tentar "*seguir*" da melhor forma possível a resposta do modelo, conforme o diagrama esquematizado na Figura 1 a seguir:

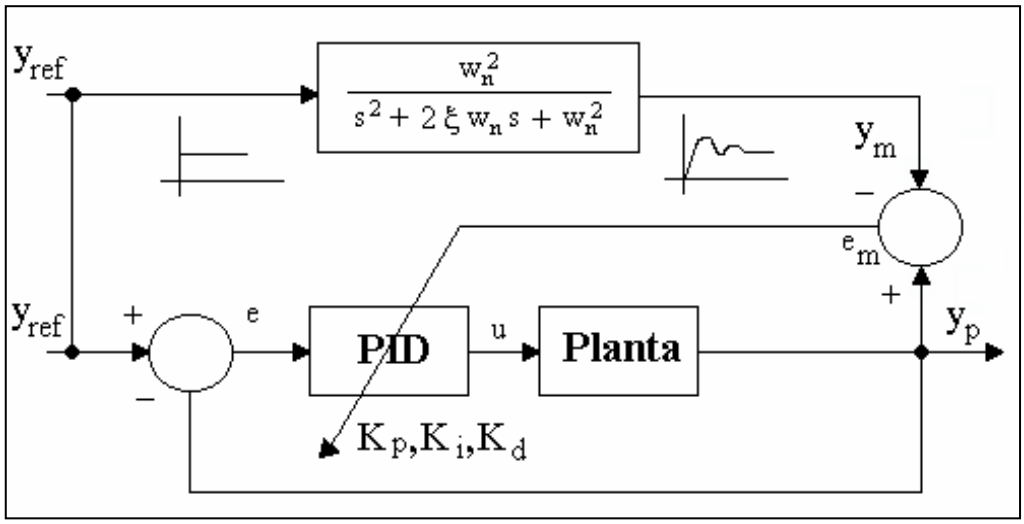

Figura 1: Diagrama Esquemático para o Projeto de um PID

Onde w_n é a freqüência natural, ξ é o coeficiente de amortecimento, y_{ref} é o valor de referência de entrada, y_m é a saída do modelo, y é a saída da planta, e é o erro atuante, u é o sinal de controle e e_m é o erro do modelo.

A Equação (1) foi a equação do PID utilizada para este trabalho:

$$U(s) = Kp + \frac{Ki}{s} + s.Kd \qquad (1)$$

Os três parâmetros devem ser ajustados para proporcionar um desempenho satisfatório da planta. Os mesmos podem ser calculados por simulações com valores escolhidos aleatoriamente, e podem, o que ocorre na maioria das vezes, serem escolhidos através das regras de Ziegler & Nichols, regras utilizadas para a determinação dos valores do ganho proporcional, do tempo integral e do tempo derivativo, baseadas nas características da resposta transitória de uma dada planta.

3. Regras de Ziegler & Nichols

As regras propostas por Ziegler & Nichols são bastante utilizadas em Sistemas de Controle para determinação dos valores do ganho proporcional *Kp*, do tempo integral *Ti* e do tempo derivativo *Td*, baseadas nas características da resposta transitória de uma dada planta, obtidos experimentalmente. Há dois métodos para determinação dos parâmetros, um para plantas que possuem curva de resposta ao degrau em malha aberta na forma de "S" e outro para plantas que se instabilizam, em malha fechada, com o aumento do ganho de um controlador proporcional; ambos os métodos visam a obtenção de 25% de sobre-sinal máximo na resposta ao degrau, [2].

O uso dessas regras para o trabalho de otimização objetiva a obtenção de parâmetros iniciais do controlador. Os três parâmetros trabalhados são: *Kp, Ki* e *Kd*; onde *Ki=kp/Ti* e *kd=kp*Td*. Tais parâmetros podem constituir num controlador PID que, juntamente com a planta, fornecerá uma resposta que, comparada a um modelo de referência desejado, proporcionará um erro relativo pequeno.

Quando a planta não se instabiliza com o aumento do ganho, não é utilizado o método de Zigler & Nichols (Z&N) para estimar os valores iniciais, sendo simulado em 20 iterações um conjunto de "chutes" para os três parâmetros, utilizando uma função randômica, donde se retira os parâmetros que geram um menor valor de erro relativo ao modelo de referência. Já para as plantas que se instabilizam, foi utilizado o Segundo Método de Ziegler & Nichols, onde: *"primeiro estabelecemos Ti = ∞ e Td = 0. Usando a ação de controle proporcional somente, aumentar Kp desde 0 até um valor crítico Kcr onde a saída primeiro exibe oscilações mantidas. (se a saída não exibir oscilações mantidas para qualquer que seja o valor de Kp possa assumir, então este método não se aplica.) Portanto, o ganho crítico Kcr e o período correspondente Pcr são experimentalmente determinados. Ziegler & Nichols sugeriram estabelecer os valores dos parâmetros Kp, Ti e Td de acordo com a fórmula mostrada na Tabela 1."*, [2].

Tipo de Controlador	**Kp**	**Ti**	**Td**
P	0.5 Kcr	∞	0
PI	0.45 Kcr	Pcr/1.2	0
PID	0.6 Kcr	0.5 Pcr	0.125 Pcr

Tabela 1: Valores dos parâmetros Kp, Ti e Td

Os parâmetros do PID que foram calculados pelas regras de Z&N, experimentalmente, possuem aproximadamente 10% a 60% de sobre-sinal máximo em resposta ao degrau, fugindo da média de 25% a qual foi baseada a Tabela 1 acima, [2].

4. Método de Otimização Multidimensional

As curvas geradas pelas plantas, em resposta ao degrau unitário, podem ser representadas por funções a serem otimizadas. Porém, essas funções são desconhecidas.

Há inúmeros métodos de otimização multidimensionais, porém a maioria apenas se aplica a funções que sejam conhecidas, visto que necessitam extrair o gradiente das mesmas. Foi escolhido, para implementação no programa computacional, o método dos poliedros flexíveis, que é um método multidimensional e que não utiliza o gradiente da função, [3].

O Método dos Poliedros Flexíveis, para uma função bi-dimensional, funciona da seguinte forma: são necessários três pontos, 01 (um) além da dimensão da função; esses pontos devem formar um poliedro, que no caso é triangular. Com o valor da função em cada ponto, descobre-se o pior ponto, em seguida é calculado o centróide da face oposta a esse pior ponto e há um rebatimento desse ponto, fazendo uma **reflexão** baseada nesse centróide. Fazendo isso, é gerado um novo triângulo, onde é descoberto um novo pior ponto e repete-se a operação do rebatimento. Esse processo poderá se repetir até um critério de parada que pode ser realizado tirando-se a norma da soma dos vetores de ponto com exceção do pior ponto. O ponto de mínimo é retirado através da média aritmética dos pontos.

Em suma, *"a idéia básica no método dos poliedros flexíveis é deformar, a cada iteração, um poliedro, de modo que este caminhe em uma direção descendente"*, [3].

Outras operações realizadas pelo método, além da **reflexão**, são: a **expansão**, a **redução** e a **contração**, as quais servem para deformar o poliedro em cada iteração. A expansão proporciona um aumento do poliedro para tentar circular o mínimo. A contração e a redução proporcionam uma diminuição do poliedro, sendo que na redução são modificados todos os pontos, menos o melhor; e na contração é modificado apenas o pior ponto. Todas essas quatro operações realizadas pelo método possuem coeficientes - α, γ , δ e β - respectivamente para a reflexão, expansão, redução e contração. Esses coeficientes servem para incremento dos pontos para a iteração seguinte.

A figura 2 mostra um gráfico com as curvas de nível de uma função desconhecida e a aplicação do método, nos moldes da descrição realizada no parágrafo anterior, para uma função bi-dimensional. É uma representação gráfica da idéia do método dos Poliedros Flexíveis, [3].

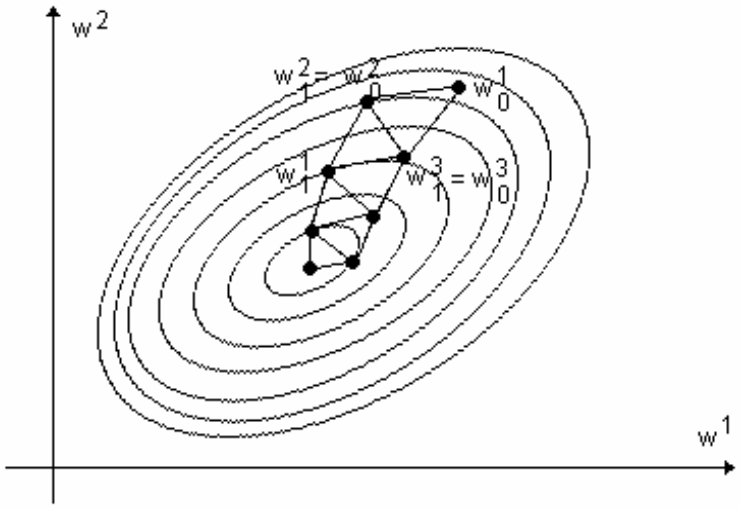

Figura 2: Representação da idéia do Método dos Poliedros Flexíveis.

Na figura acima podemos destacar que o ponto w_0^1 é o pior da série de três pontos iniciais designados para entradas. No segundo passo, aparece um novo ponto w_1^1 e permanece os dois pontos "bons" – w_0^2 e w_0^3 – quando que no terceiro passo, podemos facilmente ver que o pior ponto será o w_1^2, e assim por diante. Essa é a idéia do método o qual será descrito a seguir pelo seu algoritmo, no Quadro 3, [3].

1.	Escolher n+1 pontos $w_0^1,\ldots, w_0^{n+1}$, onde n é a dimensão da função Fazer k=0 Escolher números reais positivos: tol, α,β,γ e δ
2.	Determinar os vértices com pior (w_k^H) e melhor (w_k^L) custo associado J(.): w_k^H tal que $J(w_k^H) = \max \{J(w_k^i)\}$, i=1,…,n+1

w_k^L tal que $J(w_k^L) = \min \{J(w_k^i)\}$, i=1,…,n+1

3. Critério de Parada:

Calcular $P = \sum_{i=1}^{n+1} | w_k^i - w_k^L |$

Se (P < tol), Então ($w_{aprox}^* = 1/(n+1) \sum_{i=1}^{n+1} (w_k^i)$ e fim).

4. Determinar o centróide da face oposta ao vértice w_H:

$c = 1/n \, [\sum_{i=1}^{n+1} w_k^i - w_k^H]$

5. Reflexão:

$w_k^R = c + \alpha(c - w_k^H)$

6. Expansão, se for o caso:

Se ($J(w_k^R) < J(w_k^L)$), Então ($w_k^E = c + \gamma(w_k^R - c)$)

e Fazer $w_{k+1}^i =$ w_k^E, se i corresponde a w_k^H
w_k^i, para os demais vértices

7. Redução, se for o caso:

Se ($J(w_k^R) < J(w_k^H)$)

Então, $w_{k+1}^i = w_k^L + \delta(w_k^i - w_k^L)$, $\forall w_k^i \neq w_k^L$
$w_{k+1}^i = w_k^i$, para $w_k^i = w_k^L$

8. Contração, se for o caso:

Se ($J(w_k^R) > J(w_k^i) \; \forall \; w_k^i \neq w_k^H$),
Então ($w_k^C = c + \beta(w_k^H - c)$)

e Fazer $w_{k+1}^i =$ w_k^C, se i corresponde a w_k^H
w_k^i, para os demais vértices

9. Fazer k=k+1 e Retornar ao Passo 1.

Quadro 1: Algoritmo do Método dos Poliedros Flexíveis.

Com esse algoritmo de otimização, foi designado um algoritmo geral para o trabalho em si, o qual foi realizado computacionalmente em ambiente **MATLAB**, [4], é apresentado no Quadro 2.

1. Definição da Planta;
2. Definição do Modelo de Referência (ξ e w_n);
3. Definição do tempo de simulação;
4. Definição dos parâmetros do método de otimização:
   - tol – tolerância;
   - α - reflexão;
   - β - contração;
   - γ - expansão e
   - δ - redução

Estipular Kp, Ki e Kd – Ziegler & Nichols

5. Entrar no algoritmo de otimização, onde a avaliação de J(.) consiste em: simular o sistema planta+controlador com Kp, Ki e Kd selecionados, obtendo y(t) e,

Calcular $J(Kp, Ki, Kd) = \frac{1}{N} \sum_{i=1}^{N} (y(i) - y_m(i))^2$ $N = t_f - t_i$; ou

$J(Kp, Ki, Kd) = 100x(Mp - Mpo)^2 + (ts - tso)^2$.

Quadro 2: Algoritmo geral realizado.

A implementação seguiu a seqüência mostrada no algoritmo acima. Alguns detalhes merecem ser destacados, o que veremos a seguir.

Os pontos de entrada para o método de otimização deveriam ser 04 (quatro), porque a dimensão é de ordem 03. Cada vetor, portanto, possui 3 coordenadas representadas pelos

parâmetros Kp, Ki e Kd, respectivamente. Foi escolhida a seguinte lógica para os pontos iniciais: o primeiro foi *(Kp Ki Kd)* provenientes de Ziegler & Nichols; o segundo foi *(Kp 0 0)*, ou seja, um controlador proporcional puro; o terceiro, *(0.5Kp 0.5Ki 0)*, ou seja um controlador PI e o quarto, *(0.5Kp 0 0.5Kd)*, ou seja, um controlador PD.

A função de avaliação, ou de custo associado foi realizada de duas formas. A primeira foi bastante eficiente em algumas plantas e consistia em o custo ser o erro quadrático ocorrido entre o sobre-sinal (*Mp*) somado ao erro ocorrido ao tempo de estabilização (*ts*). Essa função, devido ao fato de não ser aplicável em alguns casos simulados durante alguns testes, não foi utilizada como avaliação em decisão final para fechamento do trabalho. A outra função foi a que prevaleceu. Consiste em, mediante simulação do modelo e do conjunto planta + controlador, verificar o somatório ao quadrado do erro a cada instante entre estes dois conjuntos. As duas funções são visualizadas no Quadro 2.

Houve apenas um problema com relação a esse método de avaliação, pois em determinados instantes o método sugeria um valor negativo para um dos parâmetros do controlador, o que tornava o sistema instável em regime permanente. Porém, dependendo do tempo de simulação esse problema poderia não ser perceptível, o que acarretava, em alguns casos, de o custo com um parâmetro negativo ser o melhor de todos anteriores e o sistema chegar ao critério de parada. Para contornar este problema foi feito uma verificação para quando algum parâmetro fosse negativo, o custo era aumentado, o que resolveu esse problema.

Para validação do método, foram realizados vários testes, com vários tipos de plantas diferentes.

Foram utilizadas plantas com inúmeras características particulares: lentas, rápidas, oscilatórias, instáveis, e o resultado foi melhor do que o esperado. O método conseguiu estabilizar até mesmo plantas as quais os parâmetros de Ziegler & Nichols não conseguiam.

5. O Programa Interativo

A partir de um método implementado e funcionando dentro das expectativas foi planejado o desenvolvimento de uma *interface* amigável para entrada e saída dos dados de interesse do usuário que utilizará o programa para calcular parâmetros ótimos ou sub-ótimos do PID.

Como o ambiente de programação utilizado desde o início foi o **MATLAB**, o mesmo foi escolhido para ser também o ambiente de *interface*, [5].

O **MATLAB** possui um conjunto de funções específicas que utilizam os sistemas orientados a objeto do sistema operacional *Windows*, *Macintosh* e *xwindows*. Estas funções são estruturadas para uso nas três *interfaces* citadas, não tendo qualquer alteração grafo-visual em qualquer um dos sistemas, [5].

De posse das funções gráficas e mais algumas informações, juntamente com uma série de simulações e testes, foi criado um *layout* o qual configurasse, em uma só tela, todos os comandos, entradas e saidas do programa.

Foram realizadas inúmeras simulações, desde com plantas simples, até mesmo com instáveis, as quais o método convergiu para valores adequados.

Para ilustrar, foi escolhido um modelo de segunda ordem com freqüência natural de 1 rad/s e coeficiente de amortecimento de 0.5. Para os parâmetros do método, foram escolhidos: 0.001, para a tolerância, 1, para a reflexão, 0.5, para a contração, 2, para a expansão e 0.3 para a redução. A função de transferência da planta utilizada é dada pela Equação 2.

$$G(s) = \frac{1}{s+1} \qquad (2)$$

Na Figura 3 é mostrado a tela principal de *interface* com vários programas, entre eles o principal que otimiza o controlador, motivo principal de investigação deste estudo.

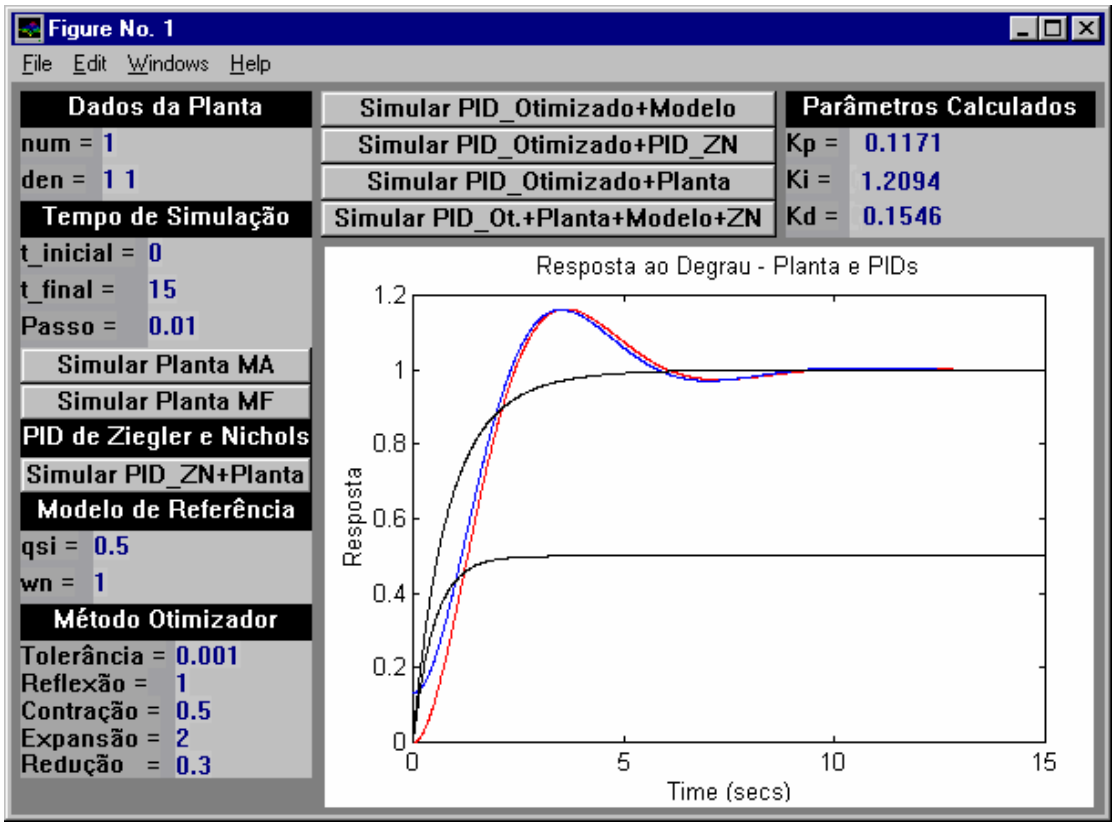

Figura 3: *Layout* da tela gráfica e resultados com controlador

Nota-se que o método sintonizou os parâmetros do PID, tais que a saída da planta se aproximasse ao máximo à do modelo, conforme desejado.

6. Considerações Finais

Pretendeu-se, com este trabalho, contribuir para o aperfeiçoamento das técnicas de projeto de controladores PID, muito utilizado nas indústrias. Os resultados foram além dos esperados, permitindo-se dizer que a utilização de métodos como estes são de grande validade para projetar controladores ótimos ou sub-ótimos.

A grande dificuldade na implementação desse tipo de ferramenta está no processamento do programa computacional. Um dos exemplos de planta utilizada, necessitou de cerca de uma hora para obtenção dos valores finais otimizados e as curvas de resposta ao degrau unitário. Essa carga de processamento impede que esse tipo de aplicação seja empregada em tempo real.

Com relação ao método de otimização, pode-se dizer que o mesmo pode ser aperfeiçoado para variar melhor as entradas dependendo da característica da planta, visto que foi utilizada uma só técnica. Ainda, pode-se utilizar técnicas para se encontrar melhores mínimos, visto que concluímos num dos itens que a maioria desses mínimos são locais. Pode-se, também, melhorar as entradas, reavaliando a rotina de valores iniciais chamada de Ziegler & Nichols.

Com relação as funções de avaliação, que são fundamentais para o método, as mesmas podem funcionar de acordo com um determinado conjunto de plantas com características peculiares. Muito se pode fazer nesse sentido promovendo um melhor resultado de otimização.

A *interface* gráfica, mesmo ainda incipiente, consegue cumprir os objetivos básicos. É uma parte que pode ser mais explorada para melhorar o uso do programa, como uma forma de distribuir e tornar uma ferramenta útil à outros grupos de pesquisa na área.

Enfim, os resultados aqui apresentados fazem com que se abram novos caminhos e se investigue mais métodos de otimização de funções que não utililizem gradiente e tornem iniciativas como essa uma ferramenta para estudos na área de Engenharia de Controle e Automação.

O programa desenvolvido constitui uma ferramenta de simulação que além de fazer com que o aluno visualize inúmeras situações de projeto de controladores, obtenha uma visão global no que se refere ao uso de controladores PID e a função da otimização para melhoria e sintonia dos mesmos.

7. Referências

[1] SILVA, G. A., MAITELLI, A. L., ARAÚJO, A. D. Um ambiente para o projeto de controladores clássicos empregando técnicas de otimização. Anais do XII Congresso Brasileiro de Automática, vol. VI, pp. 1911-1916, Uberlândia-MG, 1998.

[2] OGATA, K. Engenharia de Controle Moderno. Prentice-Hall do Brasil, 1994.

[3] NASCIMENTO Jr, Cairo L. e YONEYAMA, Takashi. Inteligência Artificial em Automação e Controle. São Paulo, 1997.

[4] **MATLAB –** The Student Edition of MATLAB – The Ultimate Computing Environment for Tchnical Education. Version 4.0 – User's Guide. The Math Works Inc., 1995.

[5] **MATLAB** – High-Performance Numeric Computation and Visualization Software. Building a Graphical User Interface. The Math Works Inc., 1993.