

# TOOLBOX MATLAB PARA SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO DINÂMICA

L. S. SANTOS, A. R. SECCHI e E. C. BISCAIA Jr.

Universidade Federal do Rio de Janeiro, Departamento de Engenharia Química - PEQ – COPPE.  
E-mail para contato: lizandrossantos@gmail.com

**RESUMO** – O objetivo deste trabalho é apresentar uma nova *toolbox* desenvolvida em ambiente MATLAB para solução de problemas de otimização dinâmica com diferentes características: tempo final livre, restrições de trajetória, restrições finais e múltiplas variáveis de controle. A *toolbox* disponibiliza de um ambiente intuitivo para modelagem, com inserção dos modelos algébrico-diferenciais, parâmetros, função objetivo e restrições. Adicionalmente foram desenvolvidos algoritmos que permitem o cômputo automático das equações de sensibilidade (via computação simbólica e diferenciação automática), paralelização do código e integração com simuladores externos como o SIMULINK, EMSO, ASPEN e HYSYS além de diversas ferramentas de avaliação gráfica e relatórios em texto e Excel. O algoritmo proposto utiliza um procedimento de discretização adaptativa, baseado em bases wavelets. Tal procedimento possibilita ajustar a discretização recursivamente durante a etapa de otimização, gerando uma malha com número reduzido de elementos e aumentando o desempenho computacional.

## 1. INTRODUÇÃO

A literatura recente divide os métodos para solução dos problemas de otimização restritos por sistemas de equações algébrico-diferenciais (DAOPs) em duas classes: **(i)** métodos diretos (numéricos) e **(ii)** indiretos (analíticos). Os métodos indiretos propõem a solução analítica do problema, aplicando o Princípio de Máximo de Pontryagin (PMP), porém apresenta sérias limitações para solucionar problemas mais complexos, que envolvam grande quantidade de equações, variáveis e restrições. Outra alternativa é a abordagem baseada em métodos numéricos, os quais se aplicam diretamente as condições necessárias de otimalidade ao sistema de equações, gerando um problema de Programação Não Linear (NLP). Tais métodos são classificados como **(a)** simultâneos (discretização total das variáveis de estado e de controle, **(b)** sequenciais ou *single shooting* (discretização das variáveis de controle e integração do modelo algébrico-diferencial) e **(c)** *multiple shooting* (partição do caso b em subdomínios). Uma das principais vantagens dos métodos sequenciais são a relativa facilidade de implementação, em se tratando da possibilidade de discretização apenas da variável de controle, e subsequente integração do sistema algébrico-diferencial por *solvers* de integração robustos e já consolidados (como as famílias de métodos de Runge-Kutta e de BDF).

Os métodos simultâneos se caracterizam pela realização da discretização total do sistema de

equações, aproximando-se as variáveis por polinômios, gerando um sistema algébrico que é otimizado e integrado (por métodos de quadratura numérica) em uma etapa só. Dessa maneira, as equações são efetivamente satisfeitas somente ao fim da otimização, caracterizando por isso essa estratégia como de caminho inviável (do inglês *infeasible path*). Tais métodos são geralmente menos dispendiosos que os métodos sequenciais, porém isso não é uma regra geral e não há um consenso estabelecido de qual é a melhor estratégia. Já na estratégia *multiple shooting* cada subintervalo do perfil de controle é integrado independentemente, impondo novas restrições ao algoritmo de otimização, que interligam os valores das variáveis de estado entre o fim de um intervalo e início do intervalo vizinho. A estratégia *multiple shooting* em geral é mais estável que a estratégia *single shooting* e pode se tornar bastante atrativa se utilizados algoritmos paralelos para integração dos períodos em diferentes processadores.

Há diversos códigos e softwares estruturados para solucionar DAOPs. Dentre elas, destacam-se DYNOPT (Cizniar *et al.*, 2005), ACADO (Logist *et al.*, 2011), TOMLAB PROPT (Rutquist e Edvall, 2010), além de ferramentas comerciais como ASPEN DYNAMICS e gPROMS. No Brasil, Pfeifer (2007) desenvolveu um algoritmo, OPCOL, que envolve diversos códigos para solução de DAOPs que envolvem redução de índice diferencial e solução via métodos indiretos.

Neste trabalho é apresentada uma ferramenta para solução de DAOPs, desenvolvida em ambiente Matlab. Basicamente diferentes estratégias de *wavelet-threshoding* (Santos *et al.*, 2012) são utilizadas para comprimir a malha temporal e indicar regiões prospectivas para discretização ao longo do cálculo de otimização. Além disso, disponibiliza um ambiente intuitivo para inserção dos modelos algébrico-diferenciais e também restrições e funcional objetivo, possibilitando também o cômputo automático das equações de sensibilidade, paralelização do código e integração com simuladores externos como o EMSO e ASPEN DYNAMICS além de diversas ferramentas de avaliação gráfica e relatórios em texto e Excel. Acredita-se que a considerável difusão e conhecimento do MATLAB possam facilitar o avanço e desenvolvimento da ferramenta para estudos e aplicações de DAOPs. Para ilustração da aplicabilidade da ferramenta é usado um problema de controle ótimo e sua configuração na *toolbox*.

O trabalho se estrutura na seguinte maneira: no Tópico 2 apresenta-se a estrutura geral de um DAOP. No Tópico 3 descreve-se a estrutura da *toolbox*. Já no Tópico 4 apresenta-se um problema exemplo de controle ótimo com a estratégia *wavelet-threshoding*. Finalmente, no Tópico 5, conclui-se as principais funcionalidades da ferramenta e implementações futuras.

## 2. ESTRUTURA GERAL DE DAOPs e MÉTODOS IMPLEMENTADOS

### 2.1 Estrutura Geral DAOP

Primeiramente, define-se a estrutura geral de um DAOP:

$$\min_{\mathbf{u}(t), t_f, \mathbf{p}} J = \min_{\mathbf{u}(t), t_f, \mathbf{p}} \left[ \varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f), \mathbf{p}) \right] \quad (1)$$

$$\begin{aligned}
 \text{sujeito a } & \mathbf{h}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}) = 0 \\
 & \mathbf{g}(t, \mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}) \leq 0 \\
 & \mathbf{e}(t_F, \mathbf{x}(t_F)) \leq 0 \\
 & \mathbf{u}^{mín} \leq \mathbf{u}(t) \leq \mathbf{u}^{máx} \\
 & \mathbf{x}_0 = \mathbf{x}_0(t_0), t_0 < t < t_f
 \end{aligned} \tag{2}$$

em que:  $J \in \mathbb{R}$  é a função objetivo,  $\mathbf{h} \in \mathbb{R}^{n_h}$  é o sistema DAE,  $\mathbf{g} \in \mathbb{R}^{n_g}$  são as restrições de trajetória,  $\mathbf{e} \in \mathbb{R}^{n_e}$  são as restrições finais,  $\mathbf{u} \in \mathbb{R}^{n_u}$  é a variável de controle,  $\mathbf{x} \in \mathbb{R}^{n_x}$  é a variável de estado diferenciável,  $\mathbf{y} \in \mathbb{R}^{n_y}$  é a variável algébrica,  $\mathbf{p} \in \mathbb{R}^{n_p}$  são possíveis parâmetros independentes do tempo e  $t$  é a variável tempo.

### 3. ESTRUTURA BÁSICA DA *TOOLBOX* MATLAB

O algoritmo tem como objetivo propor melhorias quanto a critérios de adaptação de malhas, a fim de desonerar a etapa de otimização. Em outras palavras, pretende-se construir um procedimento no qual a discretização se adapte automaticamente à própria solução do problema. Tal mecanismo não só reduz a dimensão no sistema NLP como também possibilita a identificação *a priori* dos pontos de junção dos perfis de controle. O resultado final é maximizar o desempenho computacional garantindo a convergência para o ótimo teórico (ou ótimo de referência).

A estrutura central do algoritmo (Figura 1) se baseia em ciclos de otimização sucessivos, com malhas sequencialmente ajustadas por análise wavelets. Os métodos *multiple shooting* e *single shooting* foram implementados na *toolbox*. A inovação se baseia na inserção de critérios *wavelet-thresholding*, apresentados em Santos *et al.* (2012).

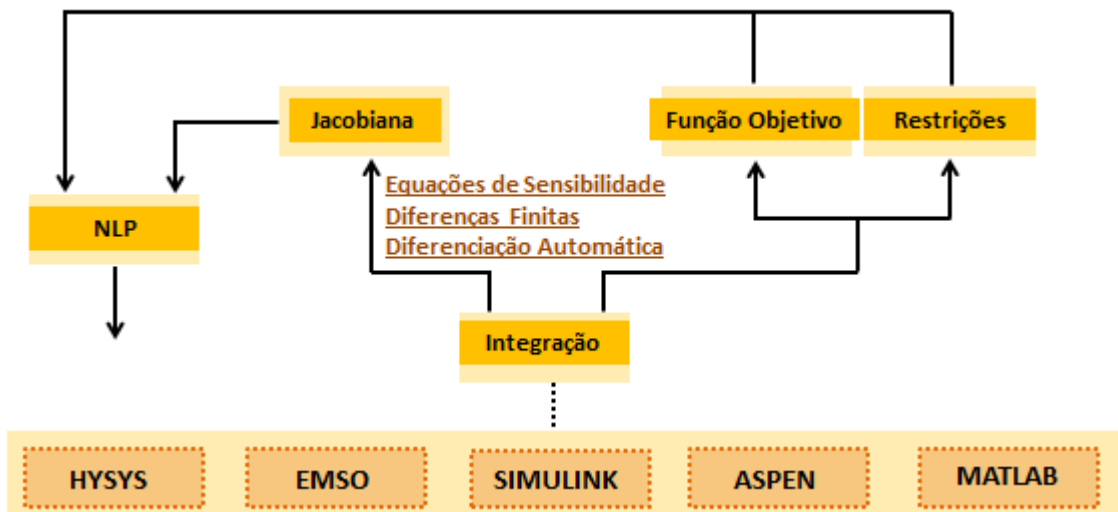


Figura 1. Estrutura básica da toolbox Matlab.

Conforme ilustrado na Figura 1, na *toolbox*, há possibilidade de inserção de modelos utilizando o SIMULINK, EMSO e o próprio MATLAB, além de disponibilizar uma estrutura pronta para conexão com rotinas externas como o ASPEN DYNAMICS e HYSYS. No MATLAB, além dos rotinas internas para integração, possibilita-se inserção de códigos DASSLC e SUNDIALS.

O exemplo seguinte ilustra a configuração básica de um problema de controle ótimo na interface da *toolbox*.

## 4. EXEMPLO ILUSTRATIVO

### 4.1. Reator Semi-batelada

Considere o seguinte modelo de reator semi-batelada (QUINTO, 2010), solucionado pelo método *single shooting*:



$$\frac{dx_a}{dt} = -k_1 x_a x_b - \frac{F}{V} x_a \quad (4)$$

$$\frac{dx_b}{dt} = -k_1 x_a x_b - 2k_2 x_b^2 + \frac{F}{V} (x_{b,in} - x_b) \quad (5)$$

$$\frac{dV}{dt} = F_b \quad (6)$$

$$x_c = \frac{1}{V} (x_{a,o} V_o - x_a V) \quad (7)$$

$$x_d = \frac{1}{2V} \left[ (x_a + x_{b,in} - x_b) V - (x_{a,o} + x_{b,in} - x_{b,o}) V_o \right] \quad (8)$$

em que a Equação 4 descreve a reação do componente *A*, com as seguintes condições iniciais:

$$x_a(0) = x_{a_0}; \quad x_b(0) = x_{b_0}; \quad V(0) = V_0 \quad (9)$$

O objetivo do problema é maximizar a seguinte função:

$$\max_u J = x_c(t_f) V(t_f) \quad (10)$$

sujeita às seguintes restrições:

$$\begin{cases} u_{\min} \leq u \leq u_{\max} \\ x_b(t_f) \leq x_{b_f, \max} \\ x_d(t_f) \leq x_{d_f, \max} \end{cases} \quad (11)$$

A interface de configuração do problema pode ser visualizada na Figura 2. As equações de sensibilidade são computadas automaticamente e incorporadas ao modelo dinâmico, bastando ao usuário fornecer apenas o sistema algébrico diferencial. A Figura 3 ilustra a trajetória típica de controle ótimo, destacando-se as regiões de refinamento da malha.

```
function [ dx y par e_var a_var c_var J cr_eq cr_deq input] = input_model

[ dx y par e_var a_var c_var J cr_eq cr_deq] = dae_default;
input.t0 = 0; % Instante inicial
input.tf = 250; % Instante final
input.x0 = [ 0.72 0.05 1]; % Condições Iniciais
input.u_lb = [0]; % Limite inferior da variável de controle
input.u_ub = [0.001]; % Limite superior da variável de controle
input.u0 = [0.0005]; % Estimativa inicial da variável de controle
input.nU = 1; % Número de variáveis de controle
input.is0 = 8; % Número mínimo de elementos
input.isF = 256; % Número máximo de elementos % declaração de parâmetros
par(1)={'k1 = 0.053'}; par(2)={'k2 = 0.128'}; par(3)={'cb_in = 5'}; par(4)={'Vo = 1'}; par(5)={'lamb=1^8'};
par(6)={'cao=0.72'}; par(7)={'cbo=0.05'}; par(8)={'cd_mais = 0.15'}; par(9)={'cb_mais = 0.025'};
% Modelo dinâmico
dx(1) = {'-k1*x(1)*x(2)-( u(1)/x(3) )*x(1)'};
dx(2) = {'-k1*x(1)*x(2) - 2*k2*x(2)^2 + ( (u(1)/x(3))*(cb_in - x(2)))'};
dx(3) = {'u(1)'};
% Função objetivo
J = {'-cao*Vo + x(end,1)*x(end,3) '}; % calculo da Fobj no tempo final.
% Restrições de Igualdade
cr_eq(1) = {'[]'};
% Restrições de Desigualdade
cr_deq(1) = {'((0.5/x(end,3))*((x(end,1)+cb_in-x(end,2))*x(end,3) - (cao+cb_in-cbo)*Vo)) - cd_mais'};
cr_deq(2) = {'x(end,2) - cb_mais'};

end
```

Figura 2 – Interface de configuração (inserção do modelo).

## 4.2. Controle Singular

O exemplo seguinte ilustra a aplicação dos métodos *single shooting* para solucionar um problema de controle ótimo singular. Segundo Biegler (2010), os métodos diretos apresentam resultados satisfatórios para problemas com arcos singulares. Porém, dificuldades aparecem quando um perfil de controle requer níveis de discretização mais refinados. No caso do método simultâneo, a matriz Hessiana se torna mal condicionada à medida que o nível de discretização tende a zero.

$$\min_u x_3\left(\frac{\pi}{2}\right) \quad (12)$$

$$\dot{x}_1 = x_2(t) \quad x_1(0) = 0 \quad (13)$$

$$\dot{x}_2 = u(t) \quad x_2(0) = 1 \quad (14)$$

$$\dot{x}_2 = x_2(t)^2/2 - x_1(t)^2/2 \quad x_3(0) = 0 \quad (15)$$

sujeita às seguintes restrições:

$$-1 \leq u(t) \leq 1 \quad (16)$$

Note que, ao aplicar o PMP sobre o problema, é possível obter a expressão para o perfil de controle. Define-se a função Hamiltoniano:

$$\hbar(t) = \lambda_1(t)x_2(t) + \lambda_2(t)u(t) + \lambda_3(t)\left(x_2(t)^2/2 - x_1(t)^2/2\right) \quad (17)$$

$$\dot{\lambda}_1 = -\frac{\partial \hbar(t)}{\partial x_1(t)} = x_1(t)\lambda_3(t) \quad (18)$$

$$\dot{\lambda}_2 = -\frac{\partial \hbar(t)}{\partial x_2(t)} = -\lambda_1(t) - \lambda_3(t)x_2(t) \quad (19)$$

$$\dot{\lambda}_3(t) = -\frac{\partial \hbar(t)}{\partial x_3(t)} = 0, \quad (20)$$

$$\frac{\partial \hbar(t)}{\partial u(t)} = \lambda_2(t) = 0 \quad (21)$$

Note que o Hamiltoniano tem dependência linear em relação à variável de controle. Nesse caso, ao diferenciar em relação ao tempo até explicitar a variável de controle:

$$\frac{\partial}{\partial t} \frac{\partial \hbar}{\partial u} = \dot{\lambda}_2 = -\lambda_1(t) - \lambda_3(t)x_2(t) \quad (22)$$

$$\frac{d^2}{dt^2} \frac{\partial \hbar}{\partial u} = \ddot{\lambda}_2 = -\dot{\lambda}_1 - \underbrace{\dot{\lambda}_3 x_2(t)}_0 - \dot{x}_2 \lambda_3(t) = -x_1(t)\lambda_3(t) - u(t)\lambda_3(t) = 0 \quad (23)$$

Logo, depois de duas diferenciações:

$$u(t) = -x_1(t) \quad (24)$$

As condições finais são obtidas de:

$$\lambda_1(\pi/2) = \frac{\partial \varphi}{\partial x_1} = 0 \quad (25)$$

$$\lambda_2(\pi/2) = \frac{\partial \varphi}{\partial x_2} = 0 \quad (26)$$

$$\lambda_3(\pi/2) = \frac{\partial \varphi}{\partial x_3} = 1 \quad (27)$$

Sabendo-se que  $\varphi = x_3(\pi/2)$ .

A Figura 4 ilustra a trajetória de controle obtido por meio do método *single shooting*, aplicado no algoritmo, com 32 estágios de discretização e a trajetória da primeira variável de estado: Observando a Figura 4 nota-se que a solução numérica converge para a solução analítica obtida por PMP.

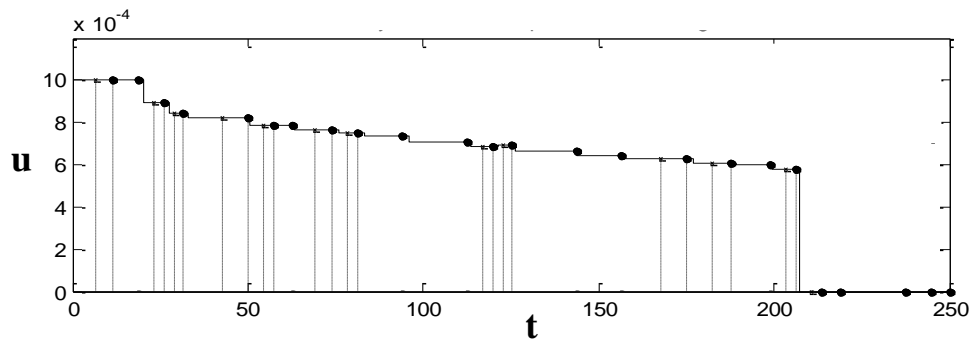


Figura 3 – Trajetória de controle ótimo do problema 4.1.

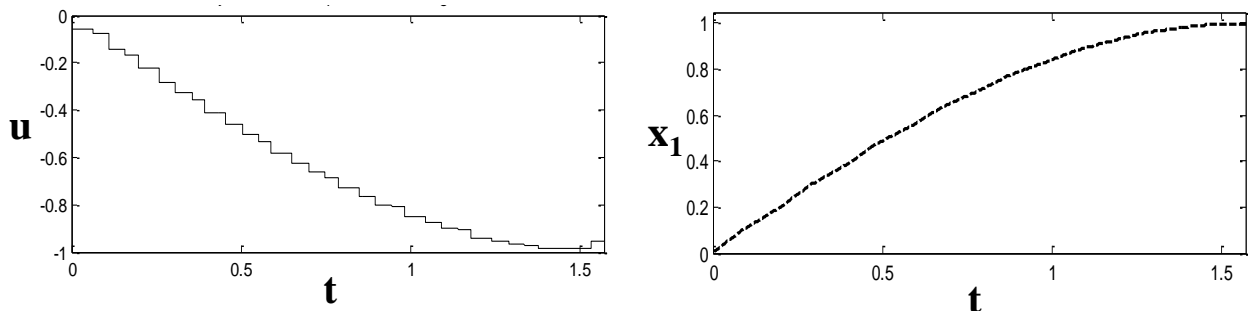


Figura 4 – Trajetória de controle: método *Single Shooting*.

## 5. CONCLUSÕES

Uma *toolbox* em MATLAB foi desenvolvida para solucionar problemas de otimização

dinâmica. A estratégia *wavelets-thresholding* foi incorporada a algoritmos numéricos como o *single shooting* e *multiple shooting*. O acervo de exemplos já resolvidos pela ferramenta evidencia uma significativa redução do custo computacional. O primeiro exemplo mostrado ilustrou a configuração básica do problema da interface MATLAB da *toolbox*. Já o segundo exemplo caracterizou a robustez da ferramenta em solucionar um problema de arco singular, geralmente reportados na literatura como problemáticos, principalmente quando aplicados métodos de discretização total.

## 6. REFERÊNCIAS

- BIEGLER, L.T. Nonlinear Programming Concepts, Algorithms, and Applications to Chemical Processes. Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, Society for Industrial and Applied Mathematics. MOS-SIAM Series on Optimization, 2010.
- CIZNIAR, M., SALHI, D., FIKAR, M., LATIFI, M.A. A Matlab Package for Orthogonal Collocations on Finite Elements in Dynamic Optimisation. *In.: 15th Int. Conference Process Control*, Strbské Pleso, Slovakia, 2005.
- LOGIST, F., VALLERIO, M., HOUSKA, B., DIEHL, M., VAN INPE, J. Multi-objective optimal control of chemical processes using ACADO toolkit. *Computers and Chemical Engineering*, v. 37, n. 10, p. 191-199, 2012.
- PFEIFER, A. A. Controle Ótimo de Sistemas Algébrico-Diferenciais com Flutuação do Índice Diferencial. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2007.
- QUINTO, T. Abordagem Algébrico-Diferencial da Otimização Dinâmica de Processos com Índice Flutuante. Dissertação de Mestrado. Programa de Engenharia Química-COPPE-UFRJ, 2010.
- RUTQUIST, P.E., EDVALL, M.M. TOMLAB OPTIMIZATION. PROPT - Matlab Optimal Control Software, 2010.
- SANTOS, L., S., SECCHI, A., R., BISCAIA JR., E.C. Wavelet-Threshold Influence in Optimal Control Problems. *Computer Aided Chemical Engineering*, v. 30, p. 1222-1226, 2012.