Otimização utilizando Matlab

Agosto de 2018

Margarida Giestas Lima

Índice

Lista de Figuras

Lista de Tabelas

1-Introdução

O objetivo deste texto é apresentar três conceitos. O primeiro, matemático, trata o problema da otimização de funções, o segundo, informático, a utilização da linguagem MATLAB para o seu tratamento e o terceiro, mais utilitário, estabelece como executar esse tratamento através da biblioteca toolbox da qual o Simulink é uma interface de visualização.

2-Otimização

Apenas os problemas mais simples recorrendo a funções objetivo simples podem ser resolvidos por via analítica. Os restantes, que são a maior parte, caiem num conjunto onde é necessário recorrer à resolução numérica. Esta resolução assume diversas formas de acordo com a situação em causa.

2.1-Notação

A notação mais genérica para os problemas de otimização consiste em:

Minimizar a função $f(\underline{X})$

s.a. $\underline{g}(\underline{X})$

$\underline{h}(\underline{X})$

onde X é o vetor de parâmetros a otimizar, f a função objetivo, g o conjunto de igualdades e h o conjunto de restrições (g e h são vetores).

2.2. Minimização sem restrições

A otimização consiste em encontrar uma ou mais soluções ótimas para uma determinada função ou funções. Por solução ótima entende-se aquela que satisfaz o problema que se deseja otimizar.

Os métodos para a solução de problemas de otimização dividem-se em determinísticos e aleatórios.

Considere-se a definição dada em 2.1 de uma forma mais detalhada.

Minimizar f(X) sujeita a $X \in S$,

Em que $f: \mathfrak{R}n \rightarrow \mathfrak{R}_e$ $S \subset \mathfrak{R}^n$ onde S é chamado *conjunto factível*.
O vetor $X = [x_1, x_2, ..., x_n]$ é composto pelas variáveis do problema em causa

Definição 2.1: "Um ponto $X^* \in S$ é um minimizador local de f em S se e somente se existe $\varepsilon > 0$ tal que $f(X) \geq f(X^*)$ para todo $X \in S$ tal que $\| X - X^* \| < \varepsilon$. Se $f(X) > f(X^*)$ para todo $X \in S$ tal que $X \neq X^*$ e $\| X - X^* \| < \varepsilon$, considera-se que se trata de um minimizador local estrito em S."

Definição 2.2: "Um ponto $X^* \in S$ é um minimizador global de f em S se e somente $f(X) \geq f(X^*)$ para todo $X \in S$. Se $f(X) > f(X^*)$ para todo $X \in S$ tal que $X \neq X^*$, considera-se que se trata de um minimizador global estrito em S."

Definem-se de forma análoga maximizadores locais e globais. Observe-se que "maximizar *f*" é equivalente a "minimizar –*f*", razão pela qual se pode, sem perda de generalidade, falar apenas de "minimização".
Um resultado fundamental, relacionado com o problema de otimização, é dado pelo Teorema de Bolzano-Weierstrass:

"Uma função real contínua *f*, definida em um conjunto fechado e limitado $S \subset \mathfrak{R}^n$, admite um minimizador global em S."

Condições necessárias de primeira ordem:

"Seja $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$, $f \in C1$. Se X^* é um minimizador local de f em $\mathfrak{R}^n$, então $\nabla f(X^*) = 0$."

Condições necessárias de segunda ordem:

"Seja $f: \mathfrak{R}n \rightarrow \mathfrak{R}$, $f \in C2$. Se X^* é um minimizador local de f em $\mathfrak{R}^n$ então:

(i) $\nabla f(X^*) = 0$

(ii) $H(X^*)$ é semi-definida positiva."

Condições suficientes de segunda ordem:

"Seja $f: \Re^n \rightarrow \Re$, $f \in C2$. Se $X^* \in \Re^n$, $\nabla f(X^*) = 0$, e $H(X^*) > 0$, então X^* é um minimizador local estrito de f em $\Re^n$."

Modelo Geral de Otimização

É conveniente pensar em algoritmos de otimização como uma aplicação iterativa. A maioria dos algoritmos de otimização requer um conjunto inicial de variáveis de projeto X_0. A partir daí, o projeto é atualizado iterativamente:

$$X^{q+1} = X^q + \alpha^* S^q$$

Em que q representa o número da iteração, X é o vetor das variáveis de projeto, S^q o vetor direção de busca no espaço de projeto, α^* é o escalar multiplicador que define o passo que se deseja dar na direção de S.

2.3- Funções de várias variáveis sem restrições

Nem sempre é possível minimizar funções de várias variáveis por meio de uma solução analítica. Surgem, então, métodos numéricos, baseados em modelos com algoritmos de buscas direcionais.

Dentro dos Métodos de primeira ordem consideram-se: O Método De Direção de Descida, O Método da Descida Máxima, o Método Quasi-Newton.
Passando para os Métodos de Segunda Ordem o mais conhecido é quando f(X) é expandida em série de Taylor até à 2ªordem.

2.4 Minimização com restrições

O problema mais geral abrange a existência de restrições de igualdade, desigualdade e laterais.

Minimizar $f(X), X = [x_1, x_2, \ldots, x_n]^T \quad , X \in \Re^n$

s.a

$g_j(X) \geq 0, j=1,2,...,J$
$h_k(X) = 0, k=1,2 \ ,..., K$

$Xi^{(L)} \leq \ x \ \leq xi^{(U)} \ i= 1, 2 ,..., n$

Método do Multiplicador de Lagrange Aumentado (MMLA)

Neste método reduz-se a dependência do algoritmo em relação à escolha dos coeficientes de penalidade e a maneira pela qual são utilizados.

A Função Lagrangeana clássica associada ao problema de otimização é dada por

$$L(X,\lambda) = f(X) + \sum_{j=1}^{m} \lambda_j g_j(X) + \sum_{k=1}^{l} \lambda_{k+1} h_k(X)$$

Sendo λ_i os multiplicadores de Lagrange.

2.5 Métodos heurísticos de otimização.

Os métodos determinísticos baseiam-se no cálculo de derivadas ou aproximações destas produzindo bons resultados quando as funções são contínuas, convexas e unimodais.

Em geral os problemas que surgem especialmente em engenharia são complexos, não lineares. A sua representação não é possível através de funções diferenciáveis sendo necessário recorrer-se a métodos numéricos.

Este tipo de problema utiliza regras de probabilidade. Requerem um número elevado de avaliações do problema. Trata-se de procedimentos iterativos que tentam simular os processos usados na natureza para resolver problemas difíceis.

Métodos mais importantes a considerar: Algoritmos Genéticos, Otimização multi- objetivo, Ponderação dos objetos.

Algoritmos Genéticos

Neste método a população de indivíduos é um conjunto de pontos do domínio da função que se pretende otimizar. Trata-se de algoritmos iterativos em que em cada iteração a população é modificada, usando as melhores características dos elementos da geração anterior. Comporta três tipos de operações:

i) Reprodução: processo no qual cadeia é copiada. Calcula-se de seguida o valor da função de adaptação.

ii) Cruzamento: a combinação em partes de cada um de dois cromossomas gera um novo descendente.

iii) Mutação: modificação aleatória ocasional do valor de um alelo da cadeia.

Todos os indivíduos da população original são modificados e submetidos aos operadores genéticos; reprodução, cruzamento e mutação.

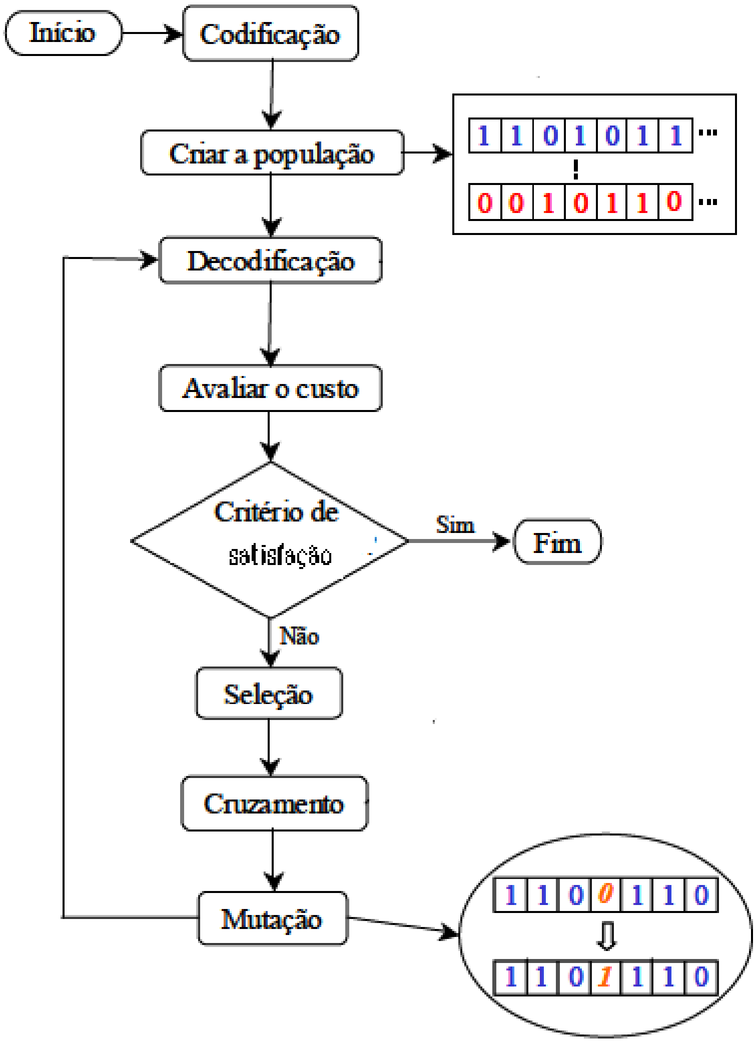

Fig. 1. Esquema do processo que ocorre com a aplicação do Método AG

Optimização Multi-objectivo

Este tipo de otimização é utilizado quando várias funções objetivo simultaneamente contêm uma função em conflito com outra.

Problema de Otimização dos objetivos

Trata-se de um método para a solução de problemas multiobjectivo que se baseia na substituição destes problemas por um de otimização escalar. Tal é conseguido criando uma função de forma.

3-Matlab

O Matlab é um programa de computador da MathWorks. A sua versatilidade centra-se, como ambiente de trabalho orientado para: o cálculo numérico, o desenvolvimento de modelos analíticos, a simulação, controlo, ensaio e monotorização de modelos e finalmente a manipulação simbólica.

O Matlab existe em várias plataformas operativas: Windows, MacOs, Linux e variantes de Unix (workstations).

É constituído pelo núcleo do programa (o Kernel) responsável pelo ambiente gráfico.

Neste kernel encontram-se as funções base do programa. A funcionalidade do Matlab é aumentada através de bibliotecas de funções externas, designadas por toolboxes, da qual o Simulink faz parte.

Existem milhares de algoritmos que já estão no MATLAB e que devem ser introduzidos diretamente no Simulink. Para tal basta adicionar código MATLAB

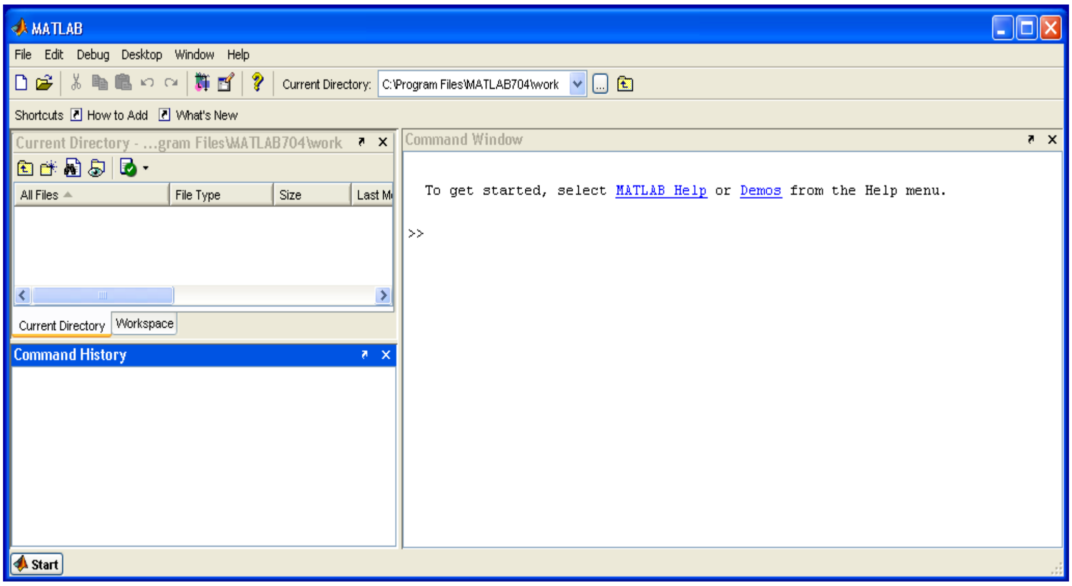

Fig. 2 Janela de funcionamento que surge na primeira vez que o programa é executado

As toolboxes que devem ser instaladas de início são: Signal processing Toolbox; Data Aquisition Toolbox; Instrument Control Toolbox e Excel Link. Este último é importante para transferir estruturas de dados entre o Matlab e o Excel.

A Janela apresentada na Figura 2 corresponde ao ambiente de trabalho do Matlab. A comunicação com as várias bibliotecas é feita por intermédio da janela do canto da direita "Command Window".

Existem dois tipos de comandos:

a) Internos (funções aritméticas e algébricas elementares, comandos de representação gráfica e de interface com os periféricos),
b) Funções externas (são ficheiros editáveis, com extensão ".m" que executam uma operação, com uma estrutura similar às funções das linguagens de programação, como o Pascal ou o C, e com espaço de memória próprio.

Aceitam argumentos e devolvem argumentos – por valor e por referência.

3.1-Estrutura de dados

Internamente o Matlab encara os operandos como matrizes e todas as operações a serem efetuadas sobre matrizes. Assim não é necessária uma reserva de memória antecipada de espaço de memória para a criação de uma variável como é o habitual em linguagens estruturadas como o Pascal ou o C. As variáveis são dinâmicas sendo possível criar, apagar, redimensionar ou modificar a sua natureza (data "type") em qualquer momento. Esta facilidade tem um preço em termos de desempenho

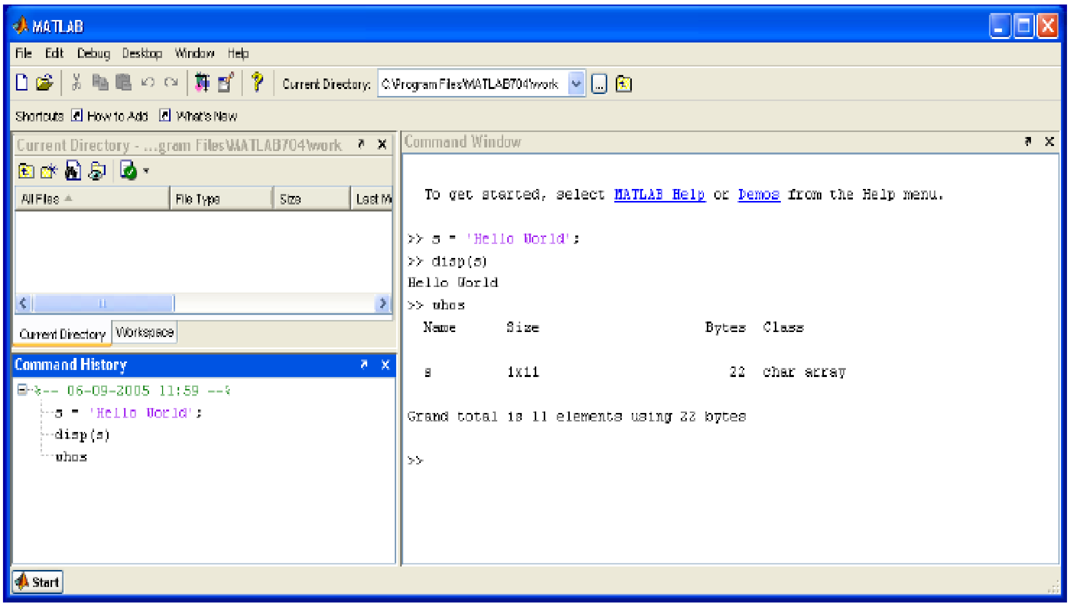

Fig 3 Janela de comando com alguns comandos internos do Matlab.

3.2- Comandos e funções externas

Também é possível a execução automática de um conjunto de comandos e funções a partir de um ficheiro com a extensão ".m". Para chamar e executar o conteúdo do ficheiro ".m" é suficiente escrever o nome do mesmo a partir da linha de comandos do Matlab, desde que o ficheiro esteja na lista de diretorias onde o Matlab irá procurar pelo ficheiro.

A Figura 4 mostra a localização da opção de menu onde é possível alterar a lista de diretorias que o Matlab utiliza para procurar os comandos e funções externas. Esta opção abre uma janela idêntica à que se encontra na Figura 5. Nesta janela existem duas opções para adição de diretorias. Numa delas, *add folder*, cada diretoria é adicionada individualmente enquanto na outra, *add with subfolders*, a diretoria é adicionada conjuntamente com todas as subdiretorias que esta possa conter.

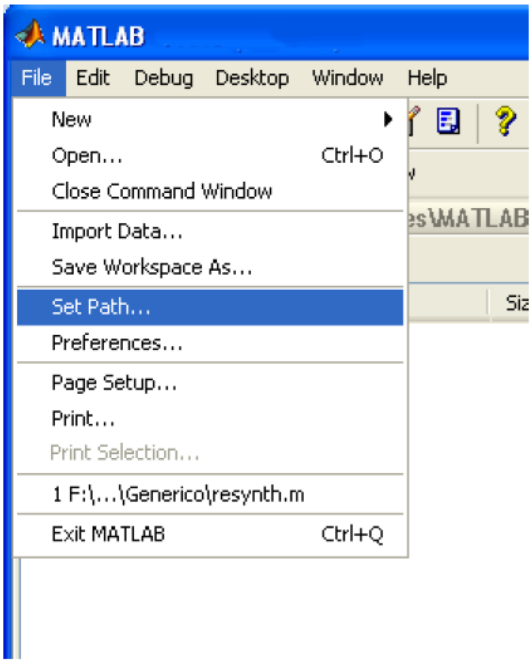

Fig 4. Alteração da lista de directorias onde o Matlab procura os comandos e funções externas.

É importante após adicionadas as directorias, gravar a configuração utilizando o botão "*save".*

Os ficheiros ".m" ,sendo ficheiros de texto, podem ser criados através de quaquer editor de texto ou do próprio Matlab. O editor do Matlab é bastante útil pois possui um modo *debug* e a identificação das estruturas de programação com cores e indentação automática.

Existem duas formas possíveis para criar um ficheiro ".m", a partir do Matlab:

i) através do menu – *File/New/M-File*;

ii) utilizando o botão de *new document* que aparece por defeito à esquerda na barra de ferramentas (*toollbar*) por baixo do menu

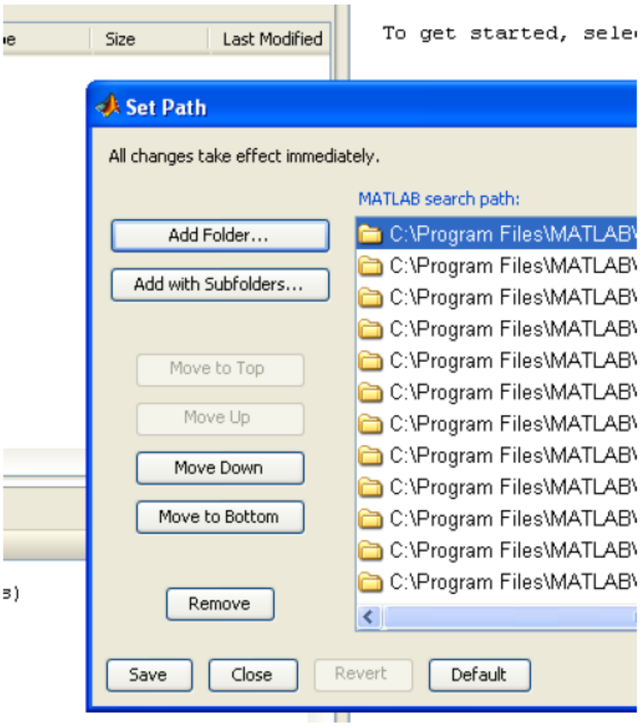

Fig.5 Janela utilizada para a adição manual de diretorias onde o Matlab procura as rotinas externas

4-Simulink

O MATLAB e o Simulink trabalham juntos.

Ao utilizar o MATLAB® e o Simulink® juntos, o programador combina a programação textual e gráfica para projetar o seu problema num ambiente de simulação.

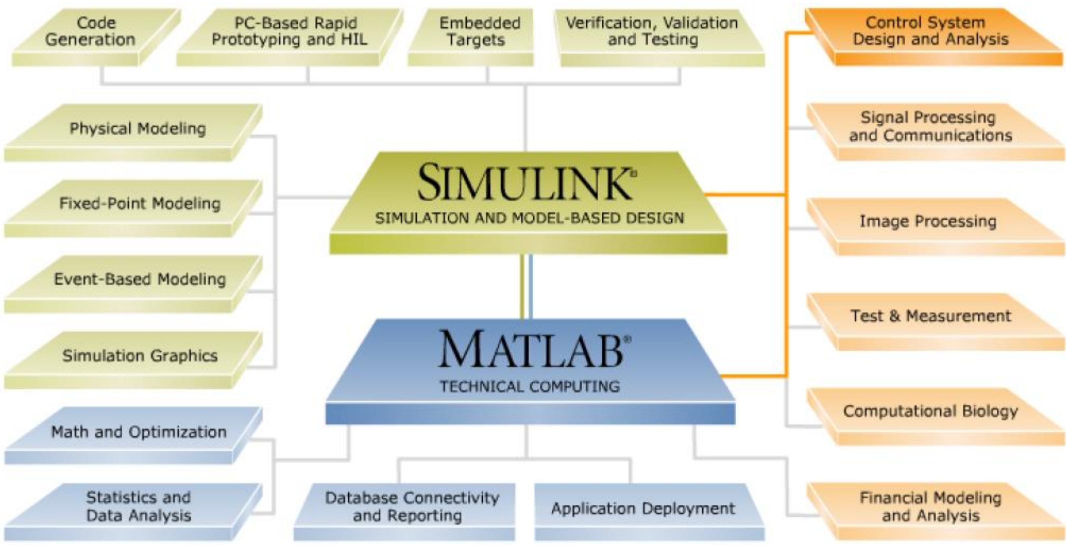

Fig 6 - Estrutura funcional do MATLAB e do Simulink. Imagem retirada do Mathworks

5-Toolbox de otimização

O Optimization Toolbox ™ fornece funções para encontrar parâmetros que minimizam ou maximizam os objetivos enquanto satisfazem as restrições.

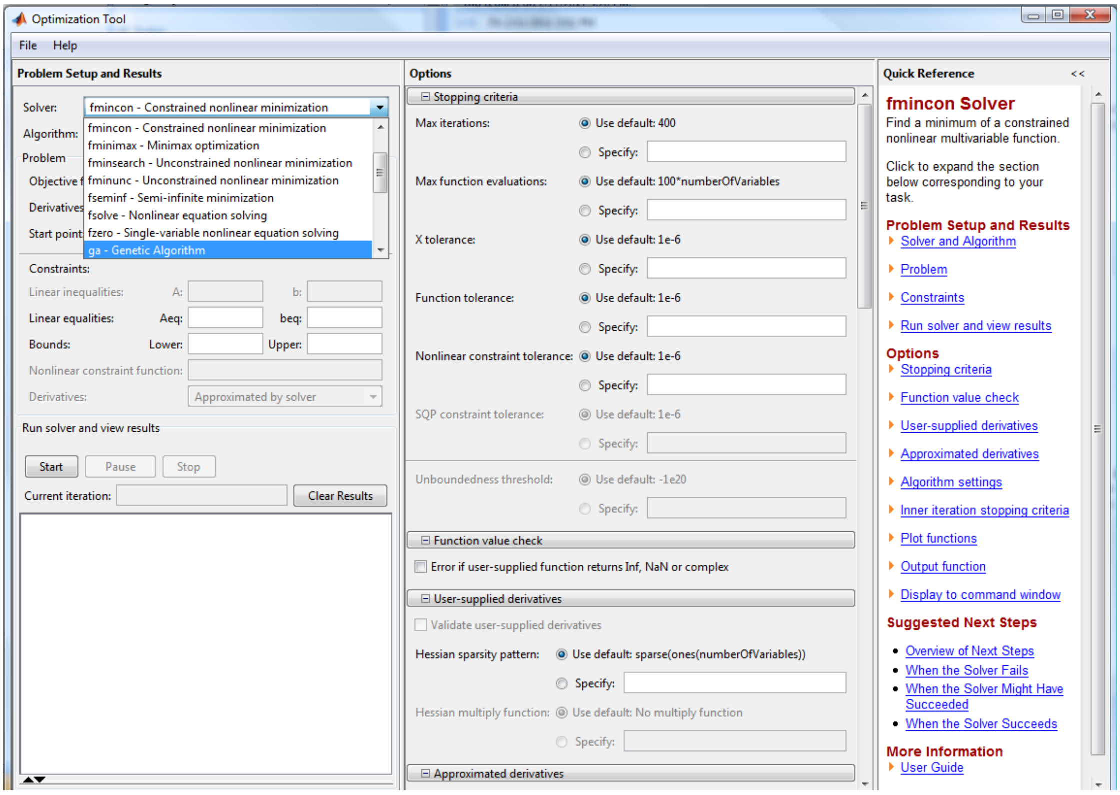

Fig 7. Visualização da toolbox de optimização

A optimization toolbox é uma coleção de funções que estendem a capacidade de MATLAB. Incluem

- •Otimização sem restrições
- •Otimização com restrições lineares e não-lineares.
- • Programação Quadrática e programação linear
- • Nonlinear least squares and curve fitting
- • Nonlinear systems of equations solving
- • Constrained linear least squares

• Algoritmos para large scale problems

Tabela 1-Algoritmos de minimização

Type	Notation	Function
Scalar Minimization	$\min_{a} f(a)$ such that $a_1 < a < a_2$	fminbnd
Unconstrained Minimization	$\min_{x} f(x)$	fminunc, fminsearch
Linear Programming	$\min_{x} f^T x$ such that $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	linprog
Quadratic Programming	$\min_{x} \frac{1}{2}x^T Hx + f^T x$ such that $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	quadprog

Type	Notation	Function
Constrained Minimization	$\min_{x} f(x)$ such that $c(x) \leq 0,\ \ ceq(x) = 0$ $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	fmincon
Goal Attainment	$\min_{x,\gamma} \gamma$ such that $F(x) - w\gamma \leq$ goal $c(x) \leq 0,\ \ ceq(x) = 0$ $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	fgoalattain
Minimax	$\min_{x} \max_{\{F_i\}} \{F_i(x)\}$ such that $c(x) \leq 0,\ \ ceq(x) = 0$ $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	fminimax
Semi-Infinite Minimization	$\min_{x} f(x)$ such that $K(x, w) \leq 0$ for all w $c(x) \leq 0,\ \ ceq(x) = 0$ $A \cdot x \leq b,\ \ Aeq \cdot x = beq,\ \ l \leq x \leq u$	fseminf

Tabela2- Algoritmos para resolver equações

Type	Notation	Function
Linear Equations	$C \cdot x = d$, n equations, n variables	\ (slash)
Nonlinear Equation of One Variable	$f(a) = 0$	fzero
Nonlinear Equations	$F(x) = 0$, n equations, n variables	fsolve

Tabela3- Algoritmos de Mínimos quadrados

Type	Notation	Function
Linear Least-Squares	$\min_x \|C \cdot x - d\|_2^2$, m equations, n variables	`\ (slash)`
Nonnegative Linear-Least-Squares	$\min_x \|C \cdot x - d\|_2^2$ such that $x \geq 0$	`lsqnonneg`
Constrained Linear-Least-Squares	$\min_x \|C \cdot x - d\|_2^2$ such that $A \cdot x \leq b, \quad Aeq \cdot x = beq, \quad l \leq x \leq u$	`lsqlin`
Nonlinear Least-Squares	$\min_x \frac{1}{2}\|F(x)\|_2^2 = \frac{1}{2}\sum_i F_i(x)^2$ such that $l \leq x \leq u$	`lsqnonlin`
Nonlinear Curve Fitting	$\min_x \frac{1}{2}\|F(x, xdata) - ydata\|_2^2$ such that $l \leq x \leq u$	`lsqcurvefit`

A maioria destas rotinas de otimização exigem a definição de um M- arquivo que contém a função *f* a ser minimizada. Opções de otimização são passadas para os algoritmos do Opt. Toolbox.

Os parâmetros *default* da otimização podem ser mudados numa estrutura própria

Conclusão

Pode-se concluir deste pequeno relatório que a utilização do Matlab não é complicada tendo o fim em causa, a otimização, podendo ser esta feita ou através de linha de comando ou utilizando uma das toolboxes existentes. A sua representação gráfica no Simulink é desejável uma vez que dá uma ideia precisa do objetivo pretendido.

Referências

ESCHENAUER, H.; KOSKI, J.;OSYCZKA, A. Multicriteria Design Optimization. Berlin, Springer-Verlag, 1990

FRIEDLANDER, A. Elementos de programação não-linear. Campinas, SP: Ed.da UNICAMP, 1994. 123 p.

GRACE, A. Optimization Toolbox- Foruse with Matlab. The Math Works Inc.,Natick, 1992.

MATLAB Optimization Toolbox Iury Steiner de Oliveira Bezerra
Introdução ao MATLAB Donizetti. Novembro 2002

EQE709 Prof. Marcus Vinicius Duarte-Controle e Simulação de Processos