

「魅せるモデリング」のTIPS

モデリングに MATLAB/Simulink を利用する場合、抽象的なシミュレーション結果をいかに分かりやすく見せるか、という点が一つの課題になります。せっかく適切にモデリングできていても、その解析結果が何を意味しているのかが分かりにくければ、効果半減です。

2次元グラフを書いてみよう

2次元グラフを描画することは、MATLAB の操作の中ではよく行われる作業です。ここでは連続系と離散系の2次元グラフを例にとり、グラフを加工する方法についてサンプルを用いて説明します。グラフの編集を行う方法には、以下の2通りがあります。

- 編集コマンドの利用
- プロパティ・エディタの利用

ここでは主に編集コマンドを利用する方法について見ていきます。

グラフィックスを描画すると、MATLAB は各グラフィックス・オブジェクトに固有の ID 番号(数値データ)を割り当てま

す。この ID のことをハンドルと呼び、内部的にはこのハンドルを用いて各オブジェクトを管理します。編集コマンドで編集したいオブジェクトを指定する場合、そのオブジェクトに割り当てられたハンドルを指定します。このことから、編集コマンドを使用する際には、目的のグラフィックス・オブジェクトのハンドル番号を取得する必要があります。一般に、編集コマンドによるグラフィックス作成の手順は以下のようになります。

- 1) 編集したいオブジェクトのハンドルを取得する
- 2) 編集コマンドを実行してオブジェクトのプロパティを変更する⁽¹⁾

連続系のグラフを描くポイント

図1(a)のサンプル・コードを例に、アナログ波形などの連続系の2次元グラフを描くポイントを説明します。実行結果は図1(b)のようになります。

図1(a)のポイント1では、 x 軸のデータを t 、 y 軸のデータを $\sin(t)$ として、関数 `subplot` を用いてグラフの1行目にプロットします。`subplot` の引き数が「211」の場合、2行1列の1行目となります。プロット・オブジェクトのハンドルを $h1$ として取得し、 $h1$ のプロパティを変更しています。ここでは、 $h1$ のプロパティとして、`LineWidth`(線幅)、`Color`(色)、および `Marker`(マーカー)を変更します。

さて、グラフィックス・オブジェクトには親子関係があります。オブジェクトの親子関係を図2に示します。プロット・オブジェクトの親は `Axes` オブジェクトになります。ポイント2では、 $h1$ の親のハンドルを $H1$ として取得し、 $H1$ のプロパティの `FontSize`(フォント・サイズ)、`XGrid`(x 軸のグリッド)、`XLim`(x 軸の範囲)、および `YLim`(y 軸の範囲)を変更しています。

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 連続系のグラフ・サンプル
% データ作成
t=0:pi/10:2*pi
y1=sin(t)
y2=cos(t)

subplot(211);% グラフ 2 行 1 列目
h1=plot(t,y1);% ハンドル取得

set(h1,'LineWidth',2)% 線幅変更(2)
set(h1,'Color','r')% 色変更(赤)
set(h1,'Marker','*')% マーカー変更(*)

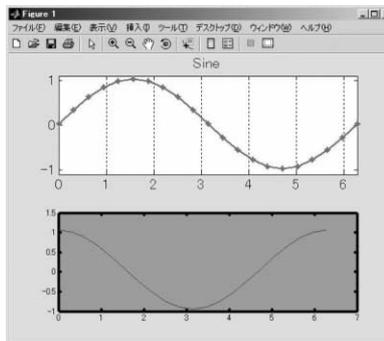
H1=get(h1,'Parent')% h1の親ハンドル取得(グラフ1行目)
set(H1,'FontSize',[16])% フォント・サイズ変更(16)
set(H1,'XGrid','On')% x軸のグリッド線描画
set(H1,'XLim',[0 2*pi])% x軸の範囲変更([0 2*pi])
set(H1,'YLim',[-1.1 1.1])% y軸の範囲変更([-1.1 1.1])

h_t=get(H1,'Title')% グラフ1行目のタイトルのハンドル取得
set(h_t,'String','Sine')% タイトルの文字列変更(Sine)
set(h_t,'FontSize',[16])% タイトルのフォント・サイズ変更(16)

subplot(212);% グラフ 2 行 2 列目
plot(t,y2);% ハンドル取得

set(gca,'Color',[0 1 0])% gcaの色変更(緑)
set(gca,'LineWidth',3)% gcaの線幅変更(3)
set(gcf,'Color',[1,1,0])% gcfの背景色(黄)
    
```

(a) サンプル・コード



(b) 実行結果

図1 連続系のグラフを描画する

四つのポイントに配慮して、連続系の2次元グラフを描いた。

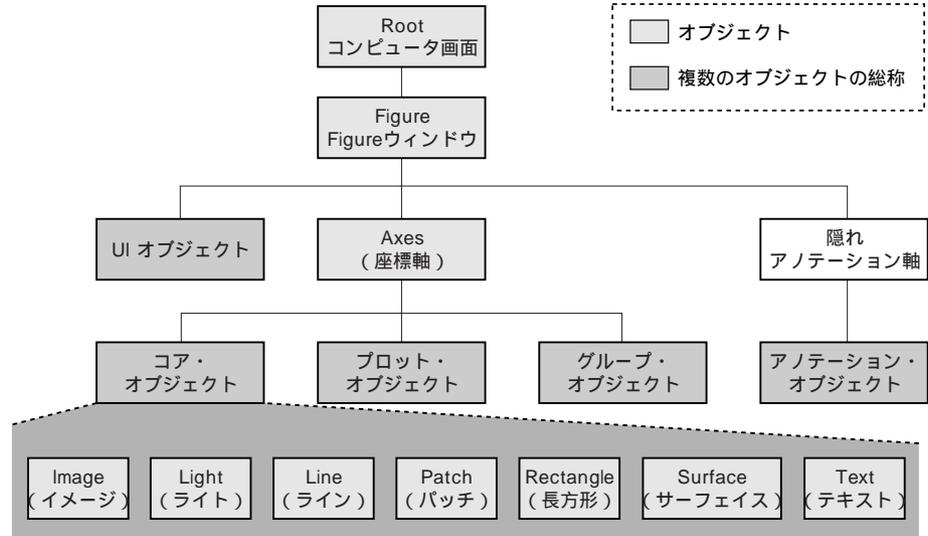


図2 オブジェクトの親子関係²⁾

グラフィックス・オブジェクトには親子関係がある。例えばプロット・オブジェクトの親はAxesオブジェクトになる。

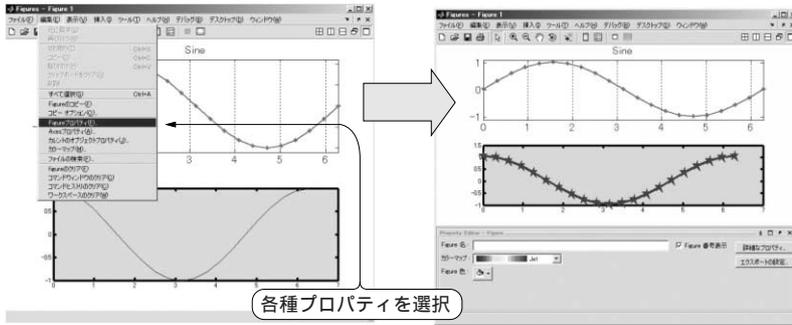


図3 プロパティ・エディタ

プロパティ・エディタはグラフィックス編集を行うGUIツール。マウス操作を主体にして、グラフの加工を行える。

ポイント3では、H1のグラフ・タイトルのハンドルをh_tとして取得し、h_tのプロパティのString(文字列)、FontSize(フォント・サイズ)を変更しています。

ポイント4では、ハンドルをコマンドによって取得しています。gcaはget current axesの意味で、カレントAxesのハンドルを取得します。gcfはget current figureの意味で、カレントFigureのハンドルを取得します。ここでは、plotコマンドで書いた現オブジェクトに対し、setコマンドを用いてgcaのColor(色)およびLineWidth(線幅)を変更し、gcfのColor(色)を変更しています。

ポイント1～ポイント4の実行結果が図1(b)に反映されている点を確認してください。また、変更内容の詳細については、図1(a)のコメント文を参照してください。

プロパティ・エディタは、Figureウィンドウのメニューの「編集」を選択するか、あるいはpropertyeditorコマンドで起動します(図3)。プロパティ・エディタはグラフィックス編集を行うGUIツールで、マウス操作を主体としたグラフの加工

```

% 離散系のグラフ・サンプル
alpha=0.01;
beta=0.5;
t=0:0.5:10*x軸データ

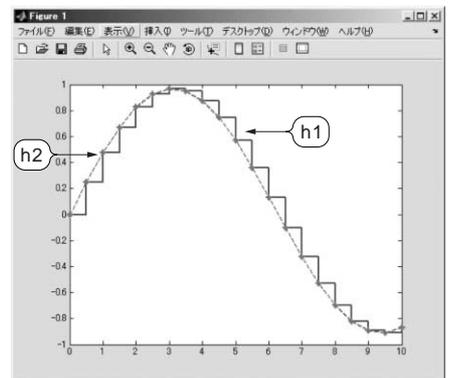
f=exp(-alpha*t).*sin(beta*t);%y軸データ

h1=stairs(t,f)% 離散系のプロット

hold on
h2=plot(t,f,'--*')
hold off

set(h1,'LineWidth',2)
set(h2,'LineWidth',2)
set(h2,'Color','r')
    
```

(a) サンプル・コード



(b) 実行結果

図4 離散系のグラフを描画する

三つのポイントに配慮して、離散系のゼロ次ホールドの時系列データをプロットした。

を行えます。

離散系のグラフを描くポイント

デジタル(離散系)のゼロ次ホールドの時系列データをプロットするサンプル・コードと実行結果を図4に示します。図4(a)