

図2 MATLABエディタ

する際は、グローバル・ワークスペースを介して受け渡しを行います。特筆なき限り、単にワークスペースと言った場合は、グローバル・ワークスペースを指すものとします。

MATLABのプログラム作法

- コマンドの集まりがスクリプト、拡張子は .m
MATLABではコマンド・ウィンドウにてコマンドを実行して各種計算を実行します。コマンド数が多い場合は、それらを1つのテキスト・ファイルにまとめて書いておきます。それがMATLABのスクリプト(プログラム)です。MATLABスクリプトは拡張子 .m を持ち、「.mファイル」と呼ばれます。
例として、第3部で紹介するドローンの数学モデルのシミュレーション計算を実行するために用意した.mファイルである runsim_sysid.mを開いてみます。

runsim_sysid.mが含まれる drone_flight_control.zip は本誌ウェブ・ページからダウンロード・データとして提供します。

<https://www.cqpub.co.jp/interface/download/contents.htm>

これを MATLAB Online に ZIP データとしてアップロードし、unzip コマンドで展開します。

drone_flight_control フォルダ下の 1_sysid フォルダに移動します。パスが表示されている部分のすぐ左隣のアイコン(緑色の曲がった下向きの矢印が描かれたアイコン)をクリックしてフォルダを選択するか、コマンド・ウィンドウにて「cd <移動先のフォルダ名>」を入力して [Enter] キーを押すと、作業用フォルダを変更できます。すると、図1のように「現在のフォルダ」ウィンドウにいくつかのファイルが表示されます。この中にある runsim_sysid.m をマウスの左ボタンでダブルクリックすると、図2のように MATLAB エディタ(テキスト・エ

イントロ
第1部

A / 画像処理基礎

第2部

テスト駆動開発

第3部

ドローン飛行制御

第4部

フィードバック制御

第5部

姿勢推定

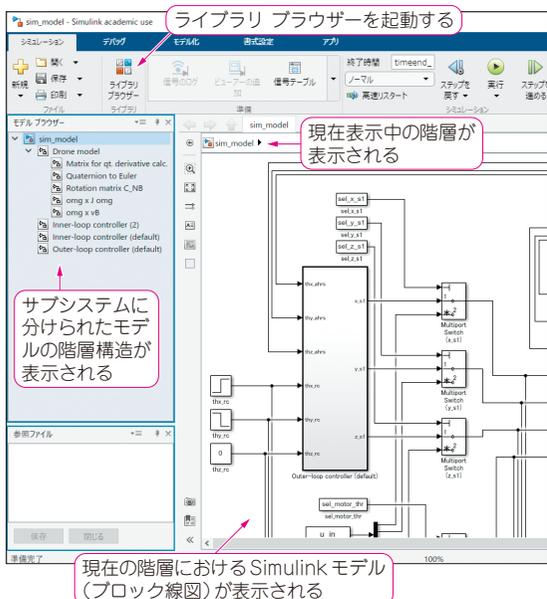


図3 Simulink エディタ

ディタ)が起動し、`runsim_sysid.m`の内容が表示されます。%文字から行末まではコメントになり、これ以外の部分がコマンドになります。mファイルを実行する際は、コマンド・ウィンドウにそのmファイルのファイル名を入力して[Enter]キーを押します。ただし、拡張子(.m)は入力しません。mファイルから別のmファイルを呼び出す際も、拡張子は入力しません。例えば、`runsim_sysid.m`の場合、11行目の`sim_param`と書いてある行は、別のmファイルである`sim_param.m`を実行する命令となります。

● コマンドの終わりはセミコロン

コマンドの終わりは、C言語と同じようにセミコロン(;)を置きます。ただし、これは必須ではなく、改行で終わらせても構いません。セミコロンがあると、実行結果をコマンド・ウィンドウに表示しません。全てのコマンドの実行結果をいちいち表示するとコマンド・ウィンドウに多くの文字が表示され、見づらくなるので、通常はセミコロンを入れて表示させないようにします。あえて実行結果を表示させたい場合はセミコロンを入れなくておきます。例えば、変数の値を表示させる場合は、変数名を書いて、セミコロンを置かずに改行しておく、その行を実行した際にコマンド・ウィンドウにその時点での変数の値が表示されます。

● 変数を作る

ワークスペースに変数を作る場合は、等号(=)を用

いた代入文を記述します。例えば、図2に表示されているスクリプトの21行目は、`sel_expdat`という名前の変数を作り、値として2を代入しています。C言語とは異なり、事前に変数の宣言はしなくてよく、デフォルトでは倍精度実数になります。ワークスペースにない変数名であれば変数が新規作成され、既にワークスペースにある変数名であれば、既存の値を書き換えます。

この他、C言語に類するfor文やswitch文などの構文や、MATLABの特徴である、行列を操作するための特殊な構文もあります。ヘルプに詳しく書かれているので、そちらも参照してください。

Simulinkの使い方

● 拡張子は.slxまたは.mdl

もう1つの主役であるSimulinkを見ていきます。Simulinkは、MATLABのメイン・ウィンドウのリボンにある「Simulink」と書かれたアイコンをクリックして起動できます。既にSimulinkでブロック線図(Simulinkモデル)を作成し、ファイルに保存してある場合は、現在のフォルダー・ウィンドウにて、そのファイルをダブルクリックすれば、Simulinkが起動して開けます。Simulinkモデルのファイルの拡張子は.slxまたは.mdlです。

● ドローンの数学モデルや姿勢制御器を含む Simulinkモデルsim_model.slxを開いてみる

特集 第3部のドローンの数学モデルや姿勢制御器を含むSimulinkモデル`sim_model.slx`を開いてみます。現在のフォルダにある`sim_model.slx`の上でダブルクリックします。図3のようなSimulinkエディタの画面が開きます。タイトル・バーの下のリボンは、MATLABのメイン・ウィンドウ(図1)と同様、この画面でできる主な作業メニューがタブに整理されて表示されます。

● モデルの階層構造

モデル・ブラウザ・ウィンドウは、Simulinkモデルの階層構造がツリーで表示されます。これが表示されない場合、画面左端に折りたたまれていれば、その「モデル ブラウザー」と書かれた部分をクリックすると表示できます。あるいは、リボンの「モデル化」タブを選択して[環境]-[モデル ブラウザー]で表示できます。

サブシステムというブロックを使うと、ブロック線図の一部をひとまとめにして、外見上1つのブロックに整理できます。ツリーの各要素は、そのサブシステ

ムに付けられた名前を示しています。サブシステムの中にサブシステムを作れば、階層が深くなっていきます。

モデルブラウザーの右側には、現在の階層と、その階層におけるブロック線図が表示されます。図3の例では、最上位階層を表示しています。現時点では、一部のブロックの枠の周囲が赤く表示されているかもしれません。これは、そのブロック内でエラーが検出されたことを表しています。このモデルを動かすのに必要な変数をワークスペースにまだ作っていないので、エラーになりますが、後で作るので、今は気にしないでください。

● ブロック線図の作り方

ブロック線図を作るには、Simulinkエディタ上に、ブロック線図の部品となるブロックを配置します。基本的なブロックは、Simulinkに用意されています。それを開くには、「シミュレーション」タブをクリックしてリボンにある[ライブラリ ブラウザー]をマウスで左クリックします。すると、図4のウィンドウが表示されます。左半分はブロックの分類を階層構造で表し、選択した分類に属するブロックが右側に表示されます。ブロックは、Simulinkエディタへドラッグ&ドロップすれば配置できます。制御系設計においてよく使う分類を以下に挙げます。いずれも「Simulink」の下にあり、Simulinkの標準機能として提供されるものです。筆者の経験上、ブロック線図の作成だけなら、Simulink以外のツールボックスが提供するブロックが必要になる場面はあまり多くありません。各ブロックの機能について解説します。

- Continuous…連続時間系を表現する
- Discontinuities…不連続や滑らかでない関数を表現する
- Discrete…離散時間系を表現する
- Logic and Bit Operations…論理演算を表現する
- Math Operations…基本的な数学演算を表現する
- Ports & Subsystems…サブシステムを作る
- Signal Routing…複数の信号をひとまとめにしたり、信号を取り出したりする際に使用する
- Sinks…信号の終端に置く。信号の値のグラフへの表示や、グローバル・ワークスペースへの値の書き込み、使わない信号の終端処理など
- Sources…信号発生器やグローバル・ワークスペースの変数を読み込むなど、信号の起点となる
- User-Defined Functions…ユーザ定義の関数を作る。MATLABスクリプトでSimulinkブロックの動作を記述できるMATLAB Functionなど

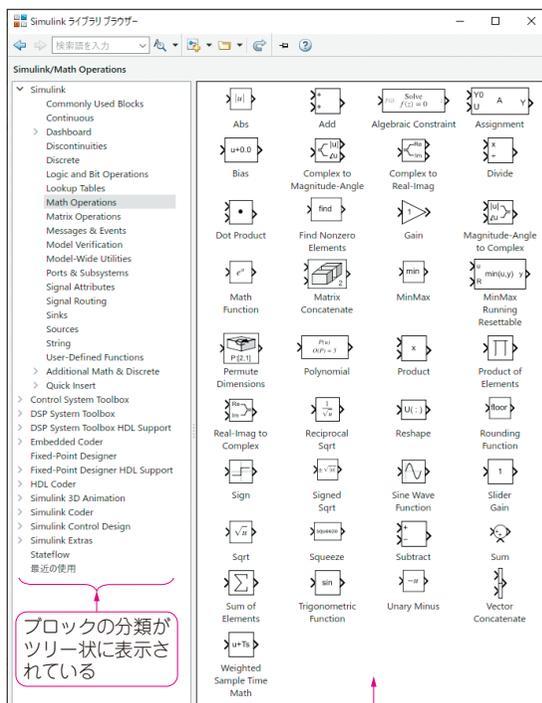


図4 Simulink ライブラリ ブラウザー

● ブロックの入出力は信号で結ばれる

ブロックの入出力は信号です。各ブロックは関数と同じで、数値が入力され、計算結果を出力します。あるブロックの出力を別のブロックの入力に接続することで、計算結果を渡して、次々に計算を実行します。

ブロックをつなぐには、ブロックの出力を表す外向き矢印(>の形)の部分で左クリックし、ドラッグして、入力先ブロックの内向き矢印(>の形)の部分でマウス・ボタンを放します。これによって作られた線が信号です。通常、信号は、1本で1つ(スカラ)または同一データ型の複数(ベクトル)の信号が含まれることがあります。他に、異なるデータ型の信号をひとまとめにできるバス信号もあります。

ふじわら・だいご

イントロ
第1部

AI / 画像処理基礎

第2部

テスト駆動開発

第3部

ドローン飛行制御

第4部

フィードバック制御

第5部

姿勢推定