

第2週目：Simulinkの基本的な使用法

第1週目では主に数値計算とグラフ書きのツールとしてのMATLABを学んだ。また、数ブロックから成る簡単な構造を持つシステムの応答をシミュレートすることも出来ることを見た。しかし、突風外乱やセンサーノイズなどの外生信号、飽和などの非線形要素、モデル化されていないプラントダイナミクスが存在する場合など、もっと現実に近いシステムのシミュレーションやモデリングはMATLABでは扱いにくい。このような状況ではSimulinkが便利であるので、今回はその基本的な操作について学ぶ¹。

1. Simulinkの基本操作

MATLABと違ってSimulinkはシステムをグラフィックに表現し、「ブロック線図」を作成する要領でモデリングして行くことができる。例えば積分器であれば、積分器というブロック要素が用意されており、そのブロックを作図用のモデルウィンドウにコピーし、他のブロックなどと結線していく。各種のブロック要素を揃えたライブラリを「ブロックライブラリ」と言う。例えば、1次遅れ系の伝達関数を含む簡単な動的システムをブロック線図とSimulinkモデルで表すと、以下のようになるが、その類似点が明らかである。



(a) ブロック線図

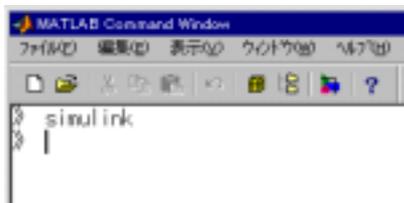
(b) Simulink モデル

Figure1.1 ブロック線図と Simulink モデル

1.1 基本操作方法

Simulinkの起動方法

1. MATLABのコマンドプロンプト上で"Simulink"と入力
2. ツールバーのアイコンから起動 (Windows)



(a) "Simulink"と入力



(b) ツールバーのアイコンをクリック

Figure1.2 Simulinkの起動

Simulinkを起動すると、ライブラリ・ブラウザが立ち上がる。ここからブロックを選択し、モデルウィンドウにブロックを置いてモデルを作成する。

¹今週の内容の多くはサイバネットシステム株式会社の承諾を得て Simulink 入門セミナー(1999年5月版)のテキストを編集したものである。

ライブラリブラウザの"Simulink"と書いてある部分をダブルクリックするか、その左側の"+"マークをクリックすると、Simulinkのブロックライブラリがツリー構造で現れる。

各ライブラリに用意されているブロックはつぎの通り。

| | |
|----------------------|------------------------------------|
| Continuous : | 連続システムに関するブロック |
| Discrete : | 離散システムに関するブロック |
| Functions & Tables : | 関数コール、テーブルに関する ブロック |
| Math : | 四則演算、論理演算、複素数の計算、 代数ループに関するブロック |
| Nonlinear : | 非線形要素に関するブロック |
| Signals & Systems : | ブロックの接続などに関する ブロック |
| Sinks : | 信号を観測するためのブロック |
| Sources : | 信号を生成するためのブロック |

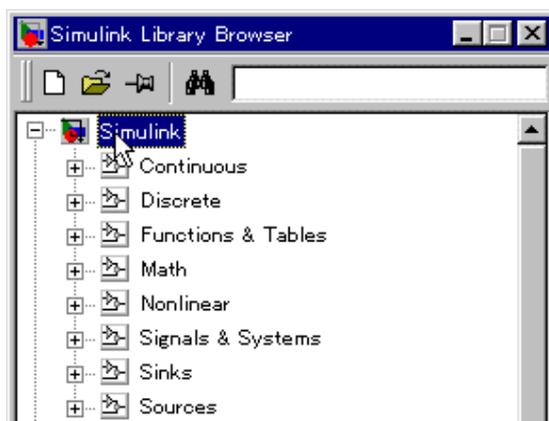


Figure1.3 ブロックライブラリ

このうち本実験項目で頻繁に用いるのは Continuous、Math、Sinks、Sources、である。

モデルウィンドウの起動

新規のモデルウィンドウを起動する方法は2つある。

1. ライブラリブラウザから起動の下図のボタンをクリックする。
2. MATLAB コマンドプロンプトのメニュー"ファイル - 新規作成 - Model"を選択する。



(a) ツールバーのアイコンをクリック



(b) メニューを選択

Figure1.4 モデルウィンドウの起動

ブロックのコピー、リサイズ、回転

モデルウィンドウに"Sources - Sine Wave"ブロックを置いてみる。ブロックを選択する方法はつぎの2種類です。

1. ライブラリブラウザで"Simulink - Sources"の下にある"Sine Wave"を直接探す。
2. ライブラリブラウザの検索機能を使う。

"Sine Wave"を選択後、モデルウィンドウまで、マウスの左ボタンでドラッグします。

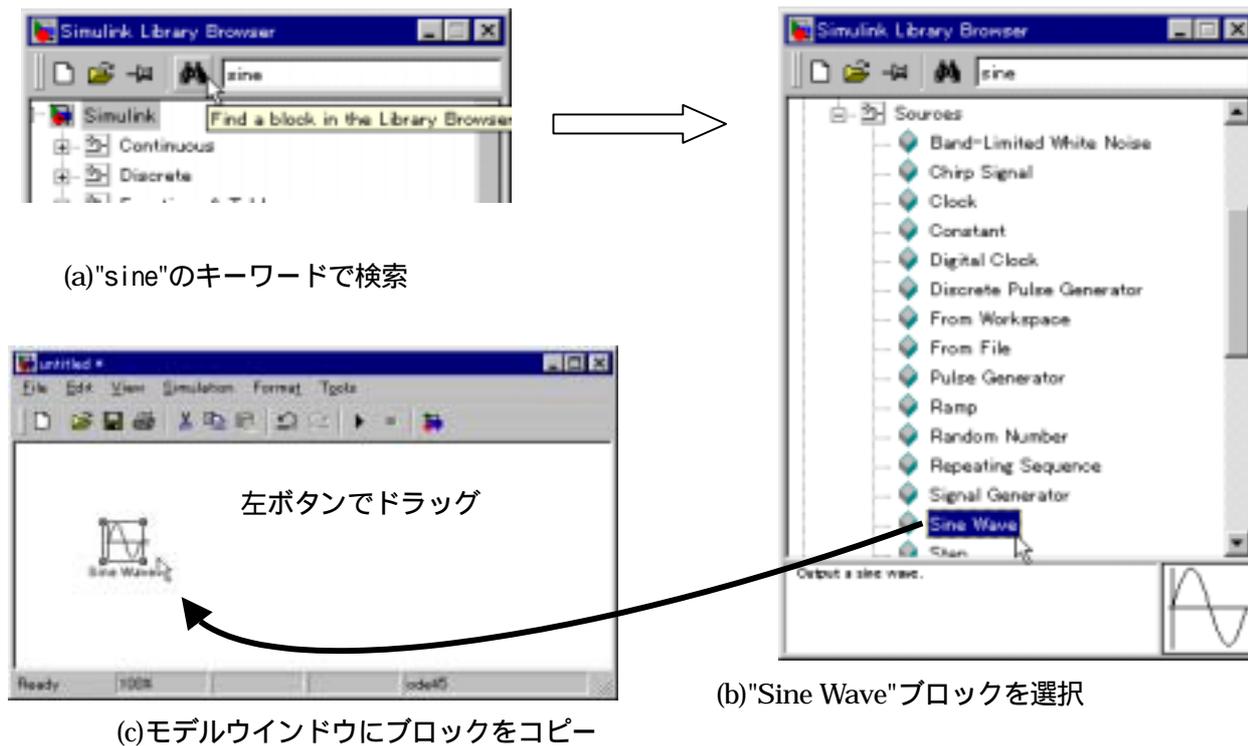


Figure1.5 ブロックのコピー

ブロックの大きさを変えるには(リサイズ)、ブロックの四隅のうちどれかにマウスカーソルを近づけ、左マウスボタンでドラッグする。

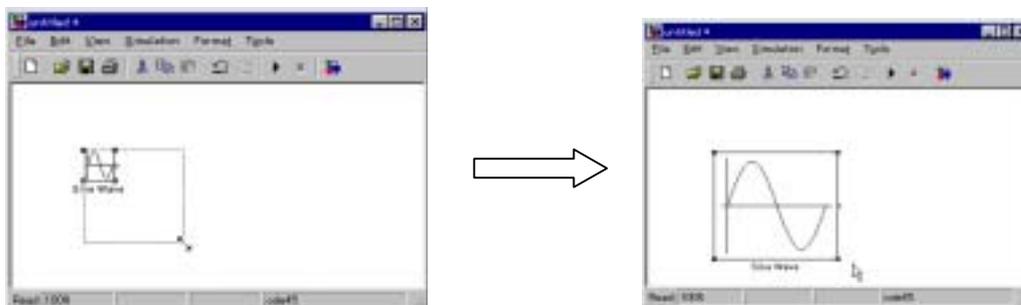
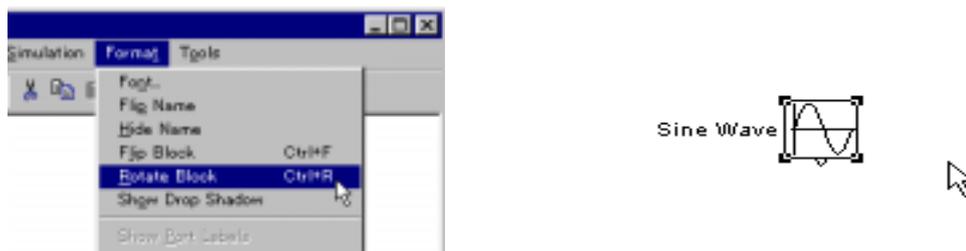


Figure1.6 ブロックのリサイズ

フィードバック部にあるブロックを結線する場合など、ブロックを回転したいことがある。例えば 90 度回転させるには、メニューの"Format - Rotate Block"を選択するか、*Ctrl+R* キーを押す。また 180 度回転のときは"Flip Block"を選択するか、*Ctrl+F*キーを押す。



(a)"Rotate Block"メニュー (b)90度回転したブロック

Figure1.7 ブロックの回転

同じブロックをモデル内にコピーするには、右マウスボタンでドラッグするとよい。



Figure1.8 ブロックのモデル内コピー

尚、ブロックの回転やコピーなど、良く使うメニューコマンドは、右マウスボタンのコンテキストメニューに用意されている。

結線の方法

同様に"Sink - Scope"ブロックをコピーした後、結線をする。結線をしたいブロックの片方のポートを選択し、もう片方のブロックのポートまでマウスでドラッグする。

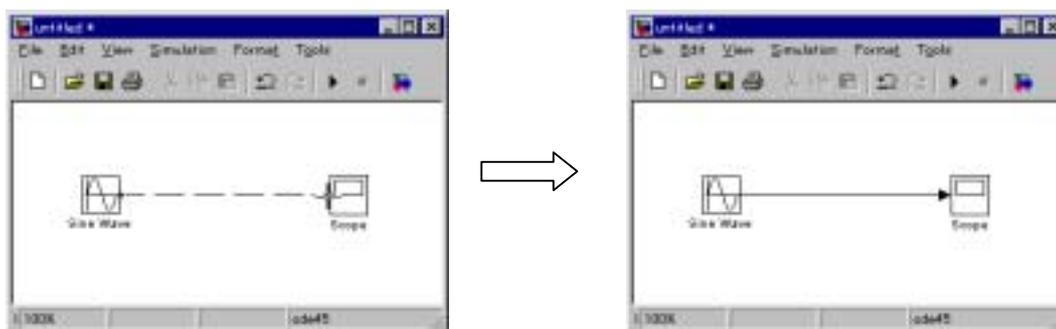
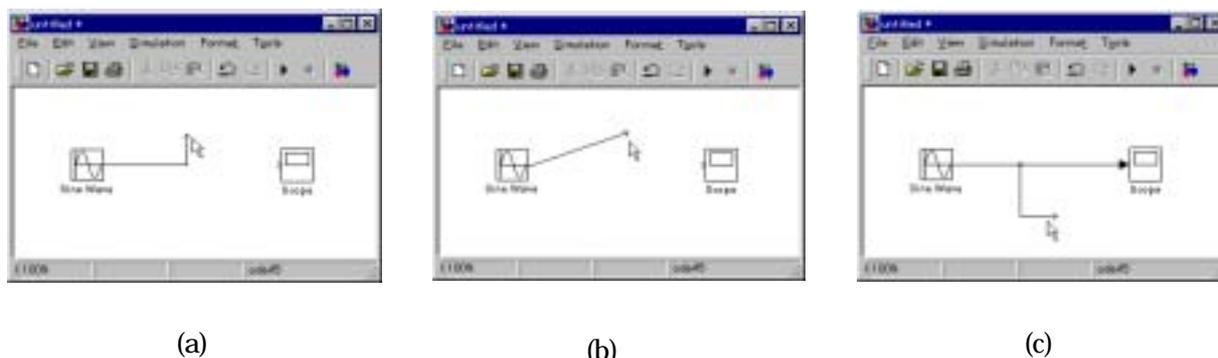


Figure1.9 ブロックの結線

いろいろな結線の操作をつぎに示す。

表 1.1 結線の操作

| | マウスの操作 | 図番 |
|--------------|------------------------|---------------|
| 信号線を直角に曲げる | 左マウスボタンでドラッグする | Figure1.10(a) |
| 任意の方向に引き出す | Shift + 左マウスボタンでドラッグする | Figure1.10(b) |
| 信号線の途中から引き出す | 右マウスボタンでドラッグする | Figure1.10(c) |



(a)

(b)

(c)

Figure1.10 いろいろな結線の操作

信号線にラベルをつけるには、信号線上でダブルクリックし、文字を入力すればよい。

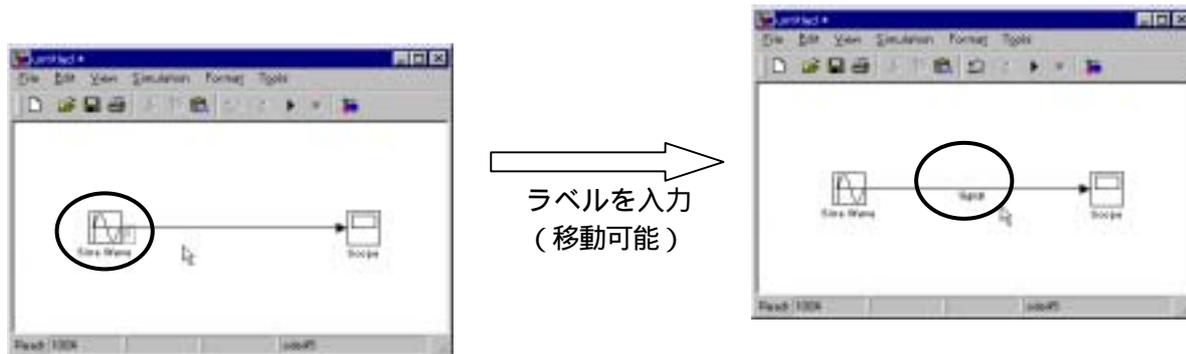


Figure1.11 信号線のラベル付け

ブロックパラメータの設定

パラメータを変更したいブロックをダブルクリックする。ここでは" Sine Wave"ブロックのパラメータを変更する方法を示す。

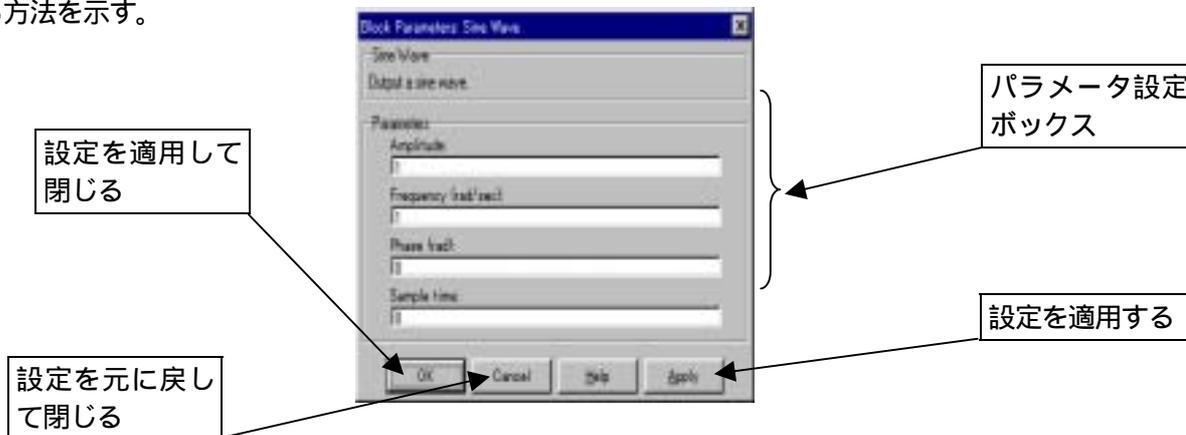


Figure1.12 ブロックパラメータの設定

例えば、正弦波の振幅を変更するには" *Amplitude*" の値を変更する。

ブロックの追加、削除

" *Continuous - Integrator*" を、モデルウィンドウにコピーし、既に結線されたブロック間に挿入する。ブロックの入出力ポートを線に近づけるように移動すると、自動的に結線される(1入力1出力のブロックのみ)。

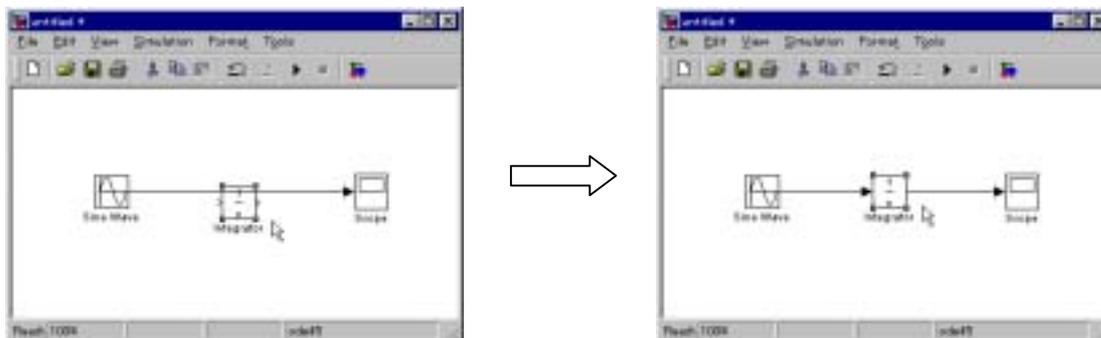
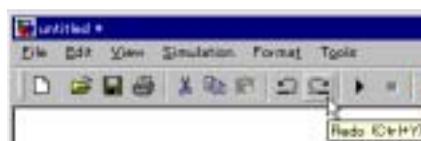
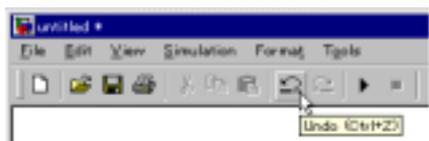


Figure1.13 ブロックの追加

削除したいときは、ブロックを選択した状態で *Delete* キーを押す。

Undo、Redo 機能について

ブロックを削除した後、削除する前の状態に戻りたいときはメニュー "*Edit - Undo Delete*" を選択するか、ツールバーのアイコンを選択する (Figure1.14(a))。Redo 機能も同様に使える (Figure1.14(b))。



(a)Undo 機能

(b)Redo 機能

Figure1.14 Undo、Redo 機能

ブロックの移動

ブロックを移動するには、2つの方法がある。

1. ブロックを選択した状態で、左マウスボタンを押したままドラッグする。
3. ブロックを選択した状態で、キーボードの矢印キーを押す。

ブロックを結線からはずしたいときは、Shift + 左マウスボタンでブロックを移動する。

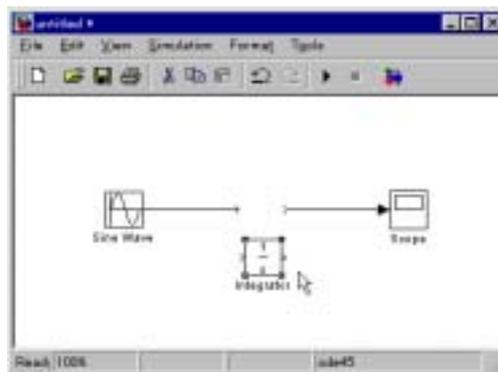


Figure1.15 ブロックを結線からはずす。

複数ブロックの選択

隣り合っているブロックを選択するには、Figure1.16のように、左マウスボタンで領域を囲むようにドラッグする。離れているブロックを選択するには、Shift + 左マウスボタンで選択する。

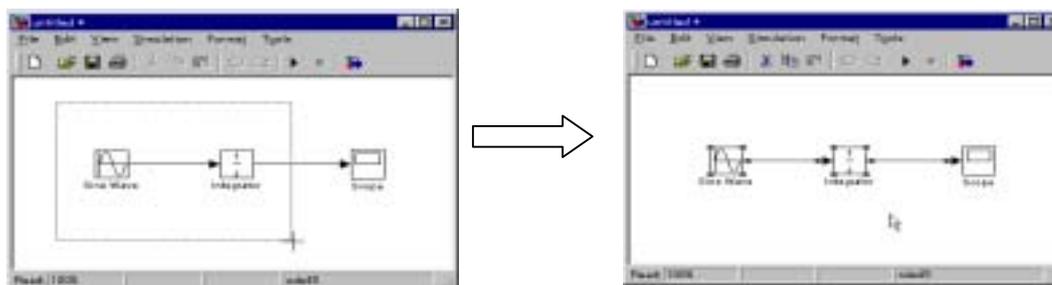
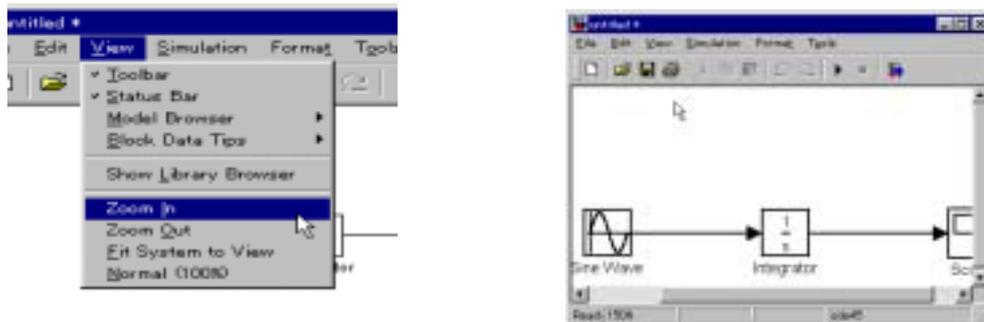


Figure1.16 複数ブロックの選択

モデルのズームイン・アウト機能

メニューの"View - Zoom In (Zoom Out)"を使って、モデルのズームイン（アウト）を行うことができる。



(a)Zoom In を選択

(b)ズームインしたモデル

Figure1.17 ズームイン機能

モデルの保存と起動

作成したモデルを保存するには、メニューの"File - Save"を選択するか、ツールバーのアイコンをクリックする。



Figure1.18 モデルの保存

ここでは"sinesys.mdl"という名前で保存している。Simulink のモデルの拡張子は.mdl である。 Figure1.18 の"ファイル名"に拡張子をつけなかった場合は、.mdl が自動的につく。

モデルを再起動するには、MATLAB コマンドプロンプト上で

》 sinesys

と入力すればよい。

シミュレーションの実行

シミュレーションを実行するには、メニューの"Simulation - Start"を選択するか、ツールバーのアイコンをクリックする (Figure1.19)。このとき、Scope ブロックをダブルクリックして、ウインドウを開いておくと、シミュレーションの経過をみることができる (シミュレーション実行後に Scope ウインドウを開いても、もちろん結果は表示される)。

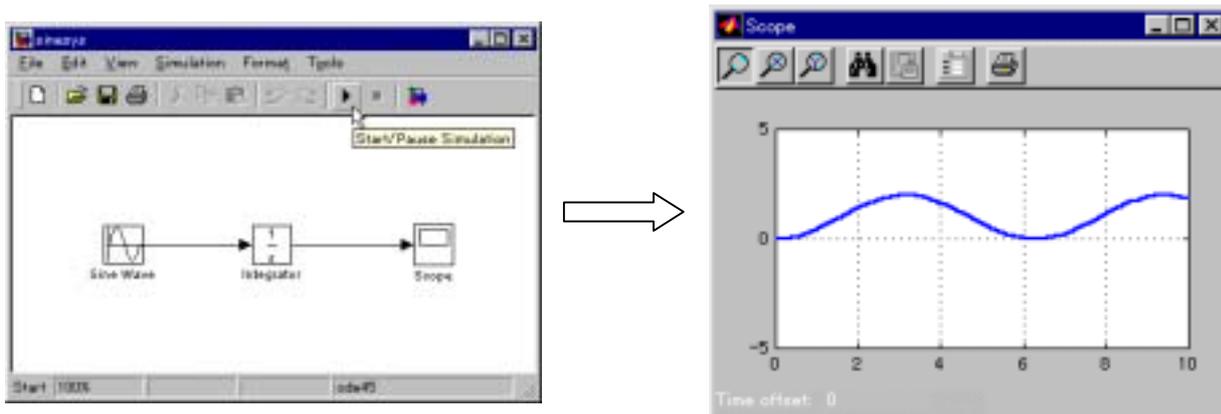


Figure1.19 シミュレーションの実行 ($\int_0^t \sin \tau d\tau = -\cos t + 1$)

以上が基本的な操作方法である。つぎに、これらの機能を使って、実際にモデリングをして行く。

2. モデリング

以下に示す 1 自由度振動系をモデル化 (モデリング) しよう。

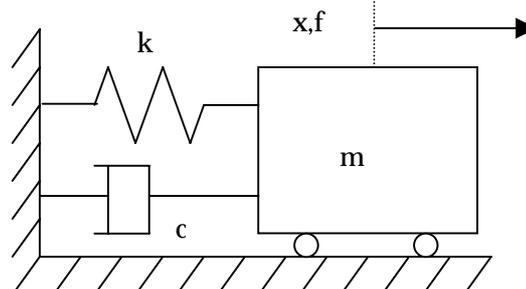


Figure2.1 1 自由度振動系

ここで、物体の質量を m [Kg]、ダンパの減衰係数を c [Ns/m]、バネ定数 k [N/m]とし、簡単のため摩擦項は無視する。また、外力 (入力) を f [N]、平衡状態からの変位 (出力) を x [m]とすると、このシステムの運動方程式はつぎのようになる。

$$m \ddot{x} + c \dot{x} + kx = f \quad (2.1)$$

モデリングのための基本的な考え方は、つぎの 2 点である。

1. 最高次の微係数について式を整理する

(2.1)式は 2 階の微分方程式であり、最高次の微係数について式を整理すると、つぎのようになる。

$$\ddot{x} = \frac{1}{m} (f - c \dot{x} - kx) \quad (2.2)$$

この式を元にモデリングをして行く。

2. 積分器を中心にブロックを置いていく

たとえば、つぎの図のように、積分器の出力が x であれば、その積分器の入力は x の 1 階微分のはずである。よって(2.2)式の 2 階微分方程式を実現するには積分器が 2 個必要である。

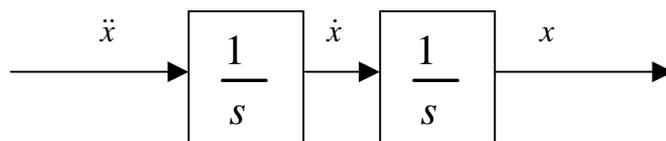


Figure2.2 積分器の入出力関係

また、1. で求めたように、 x の2階微分が(2.2)式の右辺になるように生成する必要があるので、

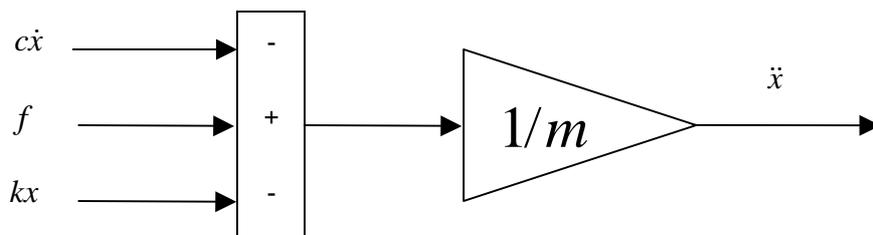


Figure2.3 (2.2)式の実現 1

とする。ここで入力 f はステップ信号としよう。また $c\dot{x}$ と kx の項は積分器の出力を利用して次のように生成できる。

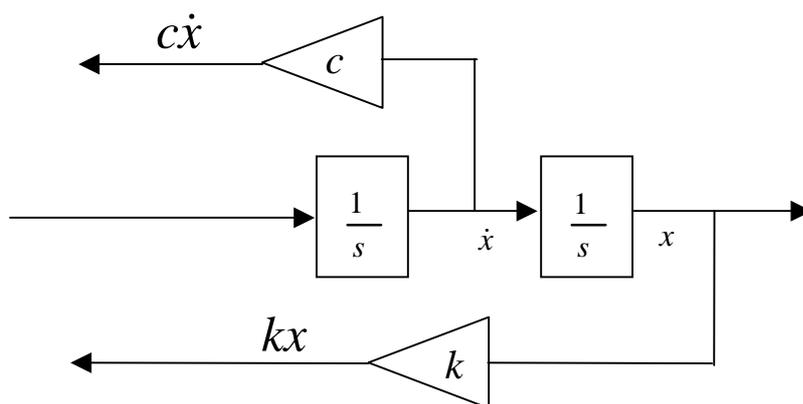


Figure2.4 (2.2)式の実現 2

以上を参考に、Simulink 上でモデリングができ、Figure2.5 のようになる。(モデル名 *onedof.mdl*)

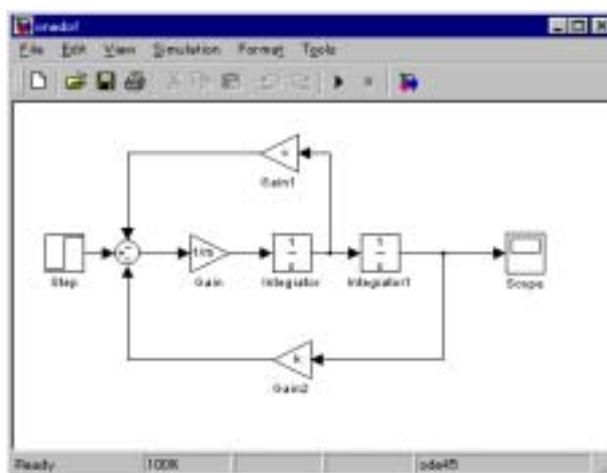


Figure2.5 onedof.mdl

表 2.1

| ブロックライブラリ | ブロック要素 |
|------------|------------|
| Continuous | Integrator |
| Math | Gain |
| | Sum |
| Sinks | Scope |
| Sources | Step |

各ブロックのパラメータの設定はつぎのようにする。

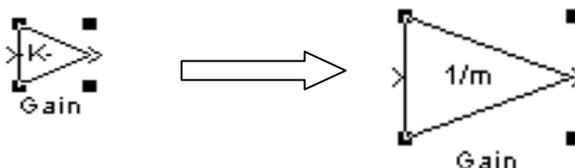
表 2.2

| ブロック (ラベル) | パラメータ名 | パラメータ値 |
|------------|---------------|--------|
| Gain | Gain | 1/m |
| Gain1 | Gain | c |
| Gain2 | Gain | k |
| Sum | List of signs | - + - |

さらに、パラメータ m 、 c 、 k は、MATLAB ワークスペース上で定義する。

- 》 $m=1$;
 - 》 $c=1$;
 - 》 $k=10$;
- これらのパラメータは M ファイルで指定する方がよいであろう。

尚、Gain ブロックは、パラメータの "Gain" を変更すると表示が変わるので、もし下図に示すような表示になったら、ブロックをリサイズすれば適当な大きさになり正しく表示する。



3 シミュレーション (時刻歴応答解析)

3.1 シミュレーションパラメータの設定

シミュレーションを行う前にパラメータを設定するため、メニューの "Simulation - Parameters" を選択する。

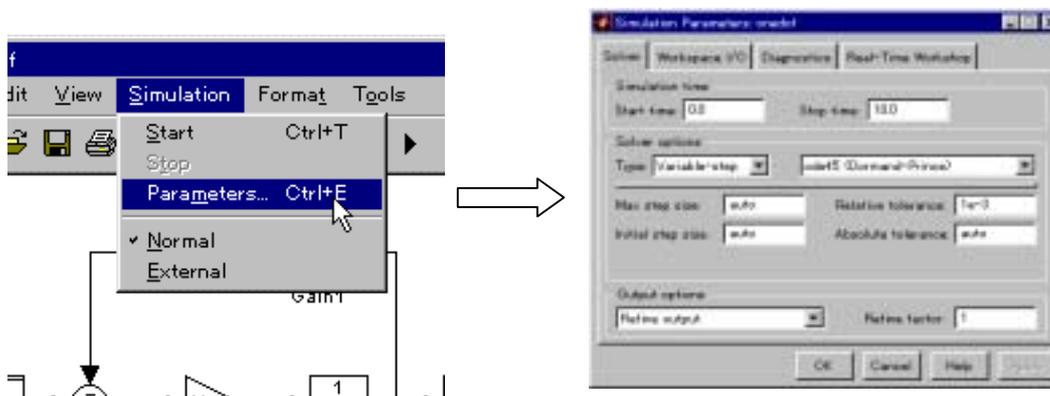
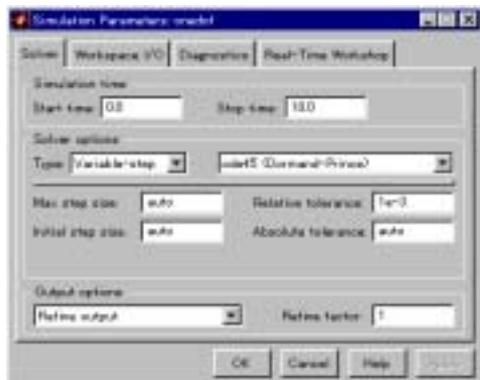


Figure3.1 シミュレーションパラメータの設定

Solver 画面



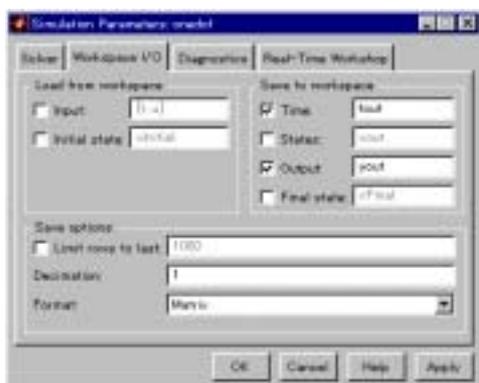
Simulation time :
← シミュレーション開始・終了時間

Solver options :
← 積分手法の選択
ステップサイズ、誤差の設定

Output options :
← 出力データの補間

Figure3.2 Solver 画面

Workspace I/O 画面



Load from workspace :
Save to workspace :
← ワークスペースのデータ入出力の設定 ()

Save options :
← 保存データ点数の制限(Limit rows to last)
保存データの間引き点数設定(Decimation)
保存データ形式(Format)

Figure3.3 Workspace I/O 画面

Input, Output は、それぞれ入力、出力ポートが指定されている時のみに有効。ここでは、出力データをワークスペースに保存する必要がないので、*Workspace I/O*ページの"Save to Workspace"のチェックを外しておく。(チェックしたままだと、必要のない動作のためにメモリを圧迫することになる。)

3.2 シミュレーション結果の出力

出力信号の観測 (Scope ウィンドウの使い方)

Scope ウィンドウを開いてシミュレーションを開始する。

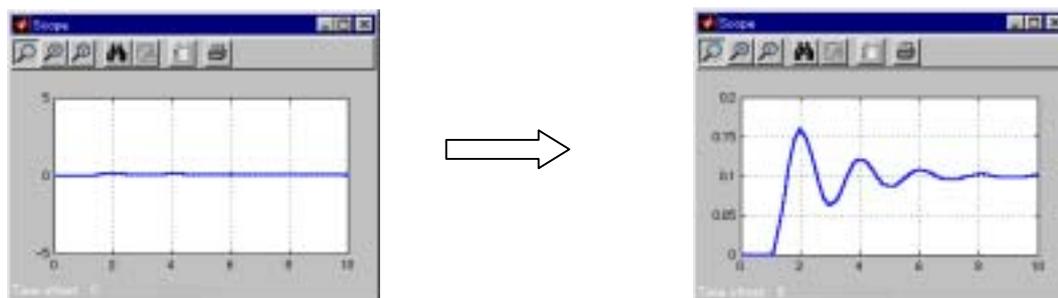


Figure3.4 Scope ウィンドウ

Figure3.4 の右側のような、適切なスケールに変更するには、つぎの方法が便利である。

1. ツールバーのアイコンをクリック (Figure3.5(a))
2. 軸上で右クリックし、"Autoscale"を選択 (Figure3.5(b))

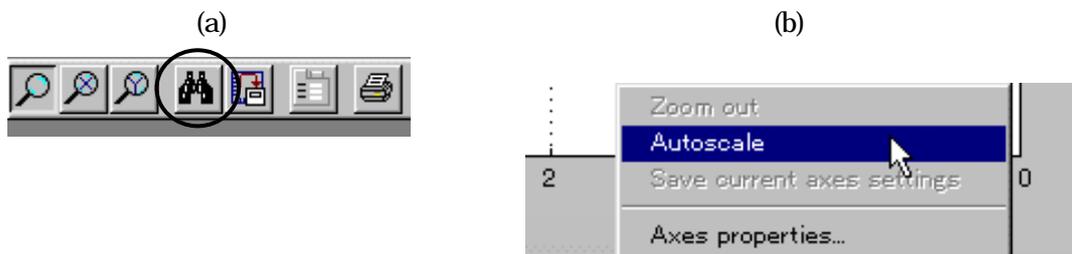


Figure3.5 Scope スケール変更

ツールバーのアイコンによるスケール変更

Scope ウィンドウのツールバーには、スケールに関するアイコンがいくつか用意されている (Figure3.6 の囲ってある部分)。



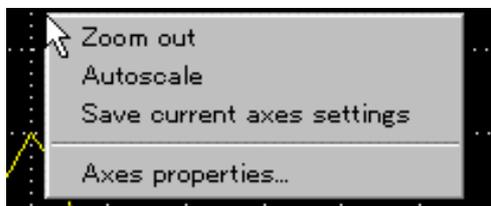
- : 表示範囲をマウスで選択
- : 横軸の表示範囲をマウスで選択
- : 縦軸の表示範囲をマウスで選択
- : 適切な表示範囲に自動スケール
- : 現在のスケールを保存

Figure3.6 Scope ウィンドウのツールバーと役割

をクリックすると、'Scope' properties ウィンドウが立ち上がる。その中の "Time range" で横軸の表示範囲の値を直接入力することができる。

右クリックメニューによるスケール変更

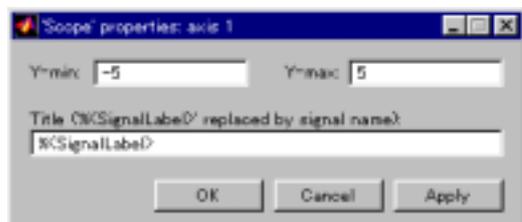
マウскарソルを軸の中に置き、右マウスボタンをクリックすると、スケールに関するメニューが開かれ、設定を行うことができる。



- Zoom out : ズームインする一つ前のスケールに戻す
- Autoscale : 適切な表示範囲に自動スケール
- Save current axes settings : 現在のスケールを保存

Axes properties : (b)図のようなメニューが立ち上がり、縦軸の範囲の値を直接入力することができる

(a)Scope ウィンドウ右クリックメニュー



← 縦軸の範囲を設定

Figure3.7 Scope ウィンドウ右クリックメニューによるスケール変更

(b)Axes properties ウィンドウ

Scope ウィンドウの軸にタイトルをつけることができる。Scope ブロックに結線されている信号線のラベルをつければ、その後シミュレーションを行うと、軸のタイトルはラベルの文字列を継承する形になる。

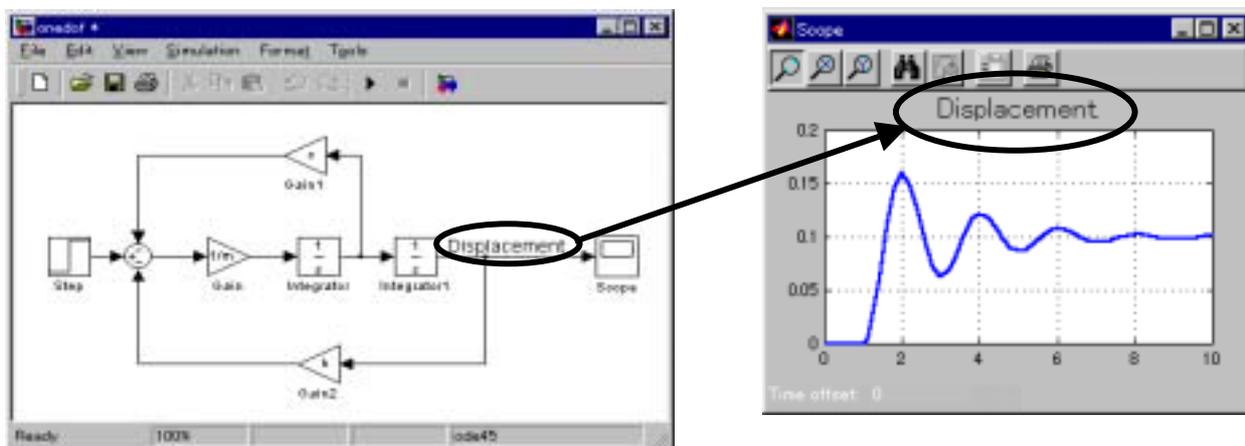


Figure3.8 信号線のラベルと Scope ウィンドウ

複数信号の観測 (Scope ウィンドウの使い方 2)

これまでは変位の応答を観測してきたが、つぎに速度の応答をあわせて観測する方法を考える。Scope ウィンドウに 2 つの応答を一度に表示する方法はつぎの 2 通りがある。

- 1 . 1 つの軸に 2 つの応答を表示する
- 2 . 軸を 2 つにしてそれぞれの軸にそれぞれの応答を表示する

方法 1 . 1 つの軸に 2 つの応答を表示する

モデルをつぎのように変更し、シミュレーションを行う。

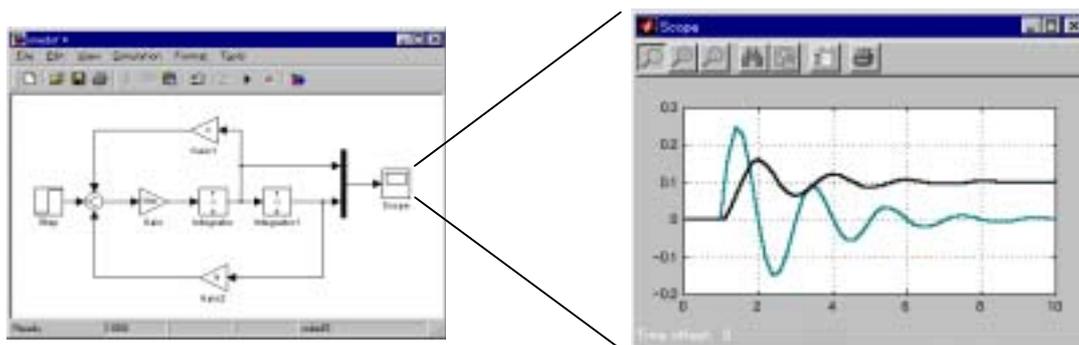


Figure3.9 変位と速度の観測 1

ここで追加したブロックは以下の通り。

表 2.2

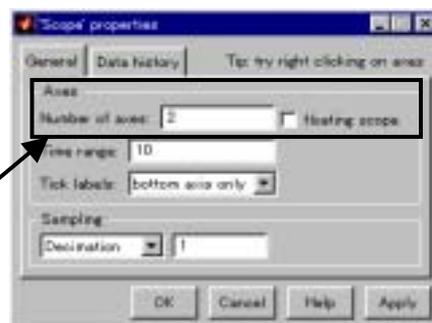
| ブロックライブラリ | ブロック要素 |
|-------------------|--------|
| Signals & Systems | Mux |

Mux ブロックの入力は、一番目 (上) が速度、二番目 (下) が変位、の順になっているので、出力は [速度 変位] のベクトルになっている。これはそのまま Scope ブロックの入力となっているので、応答の表示も同じ順序で示される。表示の色は、一番目の応答が黄色、二番目の応答がマゼンタである。

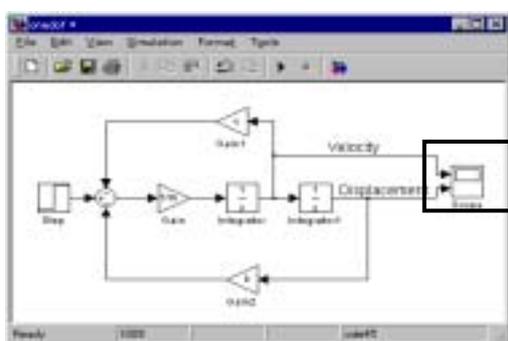
方法2 . 軸を2つにしてそれぞれの軸にそれぞれの応答を表示する

Scope ウィンドウのツールバーのアイコンをクリックし、'Scope' properties ウィンドウを開く。
"General - Axes - Number of axes"を2に設定し、"OK"ボタンをクリックすると、Scope ウィンドウとブロックの外観が変わる。
Mux ブロックをはずし結線し直して、シミュレーションをすると Figure3.10 のようになる。

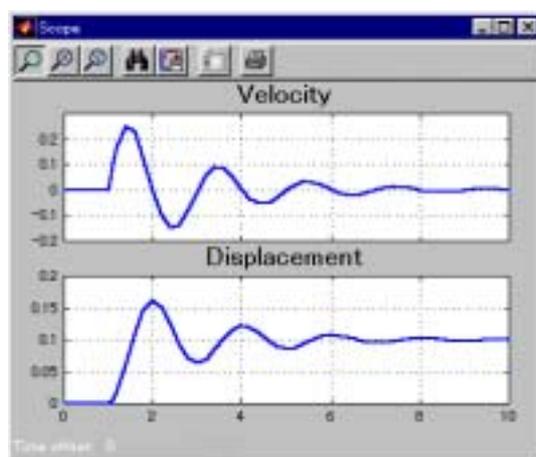
軸数を設定



(a)Scope の軸数を設定



(b)ブロックの外観の変化



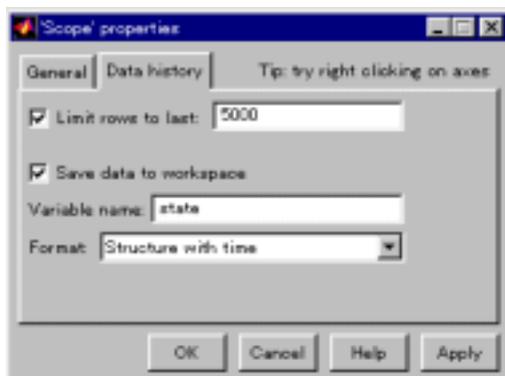
(c)シミュレーション結果

Figure3.10 変位と速度の観測 2

MATLAB グラフィックス機能の利用

最後に、Simulink でのシミュレーション結果を MATLAB ワークスペースに保存する。ここでは、Figure3.10 のモデルを使う。

Scope ウィンドウのツールバーのアイコンをクリックし、'Scope' properties ウィンドウを開く。"Data history - Save data to workspace"のチェックボックスを on にし、Variable nameを"state"に変更する。



ワークスペース保存の設定

Figure3.11 ワークスペースへの保存

この状態でもう一度シミュレーションし、MATLAB ワークスペースに存在する変数を確認する。

》 whos

```
Name      Size      Bytes Class
c          1x1        8 double array
k          1x1        8 double array
m          1x1        8 double array
state     1x1      2664 struct array
```

Grand total is 236 elements using 2688 bytes

変数 state も存在していることがわかる。関数 simplot を用いると、

》 simplot(state)

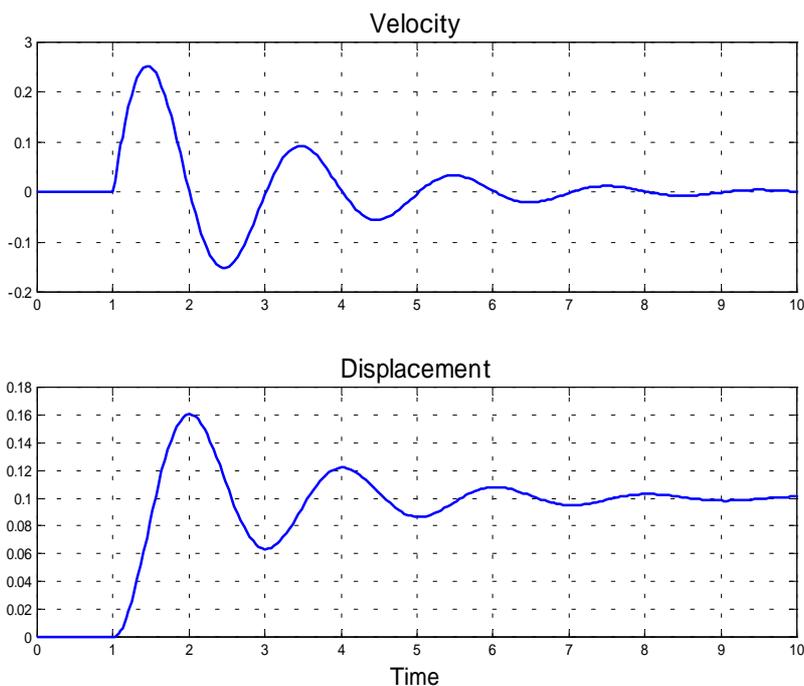


Figure3.12 simplot を用いた表示

simplot に加えて MATLAB のグラフィック機能を使うことにより、軸ラベルや凡例、ラインスタイルの変更などを行い、よりきれいで見やすいグラフを表示することができる。Figure3.9 の Mux ブロックを用いたモデルで、出力結果を MATLAB ワークスペースに出力し、simplot と MATLAB グラフィック関数を用いた例をつぎに示す。

```
》 simplot(state)
》 h = get(gca, 'child');           %これはここでは無視してよい。(ハンドル番号を取得する)
》 set(h(1), 'linestyle', ':');     %これも無視してよい。(速度のプロットを点線にする)
》 legend('Position', 'Velocity');
》 title('One degree of freedom spring-mass system');
》 xlabel('time [s]');
》 ylabel('displacement [m] & velocity [m/sec]');
```

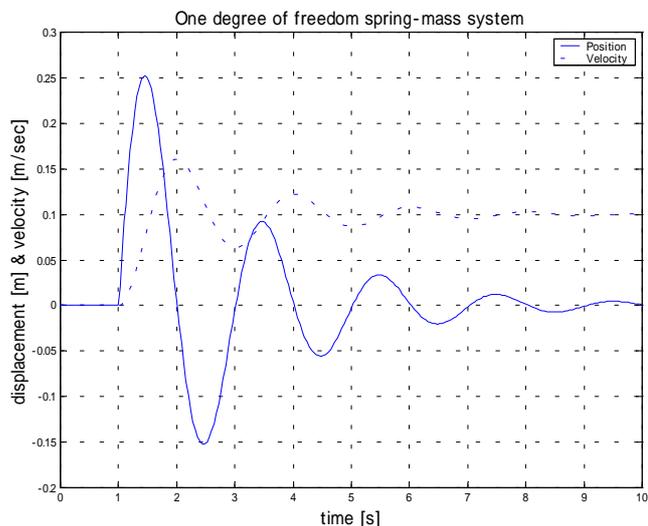


Figure3.13 simplot、MATLAB

4. Simulink ブロックのカスタマイズ

Simulink の特徴の一つに、ブロックを容易にカスタマイズできることが挙げられる。Simulink (Blockset) で作成したモデルのブロック群を、一つのブロックにまとめてカスタマイズし、自分用のライブラリとして登録しておくこともできる。イメージとしては、つぎのような形である。

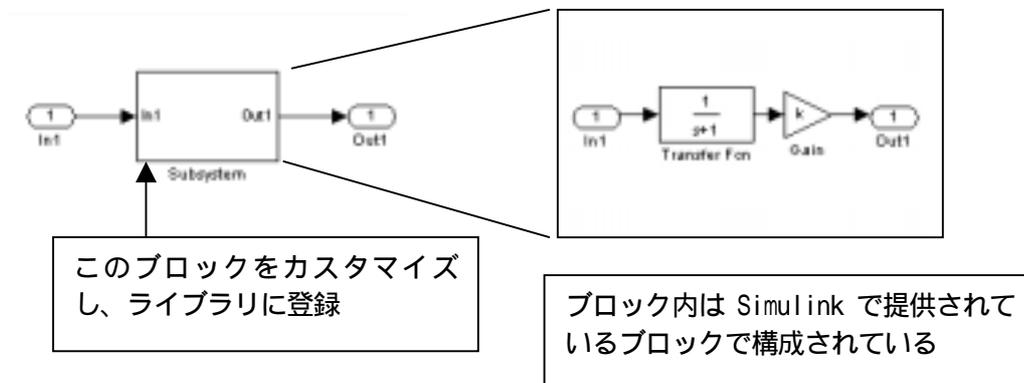


Figure4.1 自分用のブロックを作成

作業の流れは、

サブシステムの作成 (ブロック群の階層化、グループ化)

マスク機能の適用 (カスタマイズ)

ライブラリ化

である。以下では、サブシステムの作成についてのみ簡単に紹介する。

4.1 サブシステムの作成方法

モデルが大きくなると、ブロックの数も増え、煩雑になってくる。Simulink では、既存のブロックを機能ごとにまとめて一つのブロックとして扱うことができ、そのブロックを「サブシステム」と言う。サブシステムの作成方法は、つぎの2つがある。

1. モデル内のブロックを役割ごとにグループ化する
2. ブロックライブラリの "Signals & Systems / Subsystem" ブロックをモデルに置き、内部を作成する

ここでは、グループ化による作成方法を紹介する。

グループ化による方法

Figure2.5 の Simulink モデルを考える。このモデル内の "step"、"scope" ブロック以外は、一自由度振動系の機能を持つ部分であるから、この部分をグループ化する。下記のようにグループ化する範囲を選択し、メニューの "Edit - Create Subsystem" を選択する。

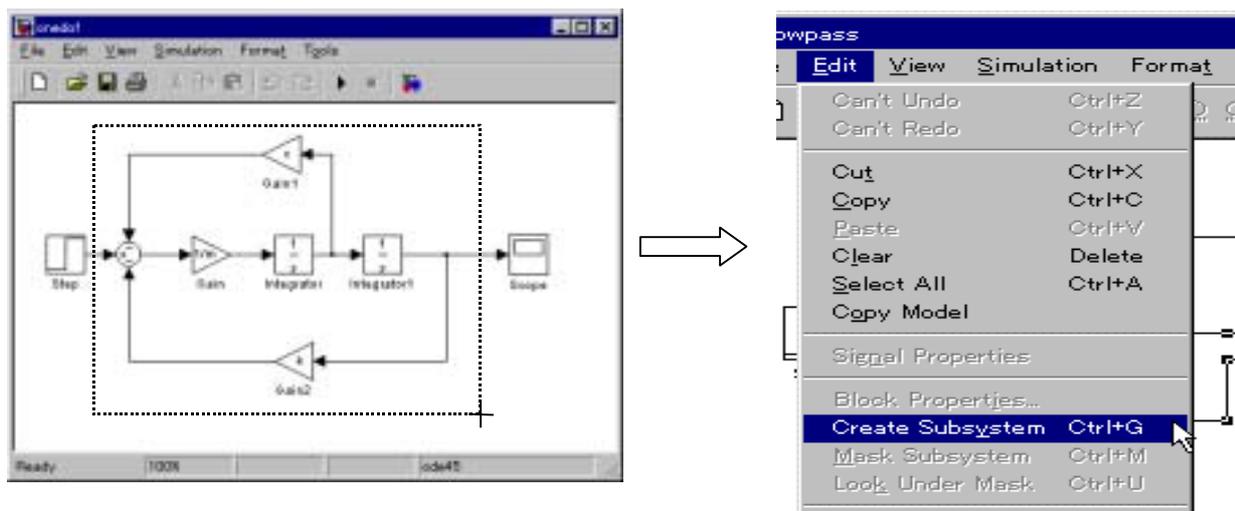


Figure4.2 サブシステムの作成方法（グループ化）

これでサブシステムが生成でき、図には一つのブロックで表示される。このブロックをダブルクリックするグループ化した部分が現れる。

ブロック名の変更

Subsystem ブロックの名前はつぎの手順で変更できる。

1. ブロック下部の名称部分にマウカーソルを当て、左クリックする。
 2. テキストが入力できる状態になるので、名称を入力する
- この結果が Figure4.3 である。

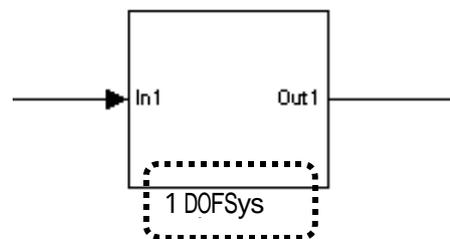


Figure4.3 名称の変更

5. VTOL 機の Simulink モデル作成

第一週目で見たと (1) 式で表される VTOL 機を 2 節で述べた要領 1 で変形すると

$$\begin{aligned} \dot{v}(t) &= -g\theta(t) + X_v v(t) + X_\delta \delta(t) \\ \ddot{\theta}(t) &= M_q \dot{\theta}(t) + M_v v(t) + M_\delta \delta(t) \end{aligned} \quad (5.1)$$

となり、これを元に要領 2 で Simulink モデルを作成できる。まずブロック線図を紙に下書きし、それが (5.1) 式を満たしていることを確認すること。その後、実際に Simulink モデル用の mdl ファイル (VTOL_MODEL.MDL) を作成せよ。実験終了時にはこのモデルをスクリーン上に提示し、教官か TA に説明をしながらシミュレーションするので注意して行うこと。

積分器は Simulink-Continuous-Integrator にブロックがあり、モデルウィンドウに左ボタンでドラッグすればよい。積分器は (5.1) 上式に一つ、下式には二つの合計三つ必要であるが、一つあればそれから右ボタンでドラッグすればコピーが出来る。同様に Simulink-Math-Gain で信号と安定微係数などの掛け算を、Simulink-Math-Sum でこれら各項の加減算を行える。入力は昇降舵角であるが、ここでは 5 度のステップ信号としよう。これは Simulink-Sources-Step から取ればよい。ダブルクリックして時刻 0 に 5 度のステップが入るようにパラメータを設定せよ。

Simulink では、時間応答をシミュレーションする際の積分手法を選択できる。

- ode45 (Runge-Kutta 法) はもっとも汎用的な手法であり、通常はこの手法で充分である。シミュレーションを試すときはまず ode45 からやってみるとよい。
- ode15s (NDFs 法) のように、最後に s のついている手法は Stiff なシステム (低周波成分と高周波成分が混在しているシステムなど) の場合に有効な積分手法である。

ode45 で、シミュレーションが極端に遅い場合は、つぎに ode15s を試してみるとよい。この VTOL 機の場合にはどの方法でも同じ結果が得られる。

積分手法と絶対許容誤差値の設定は以下のようにして出来る。

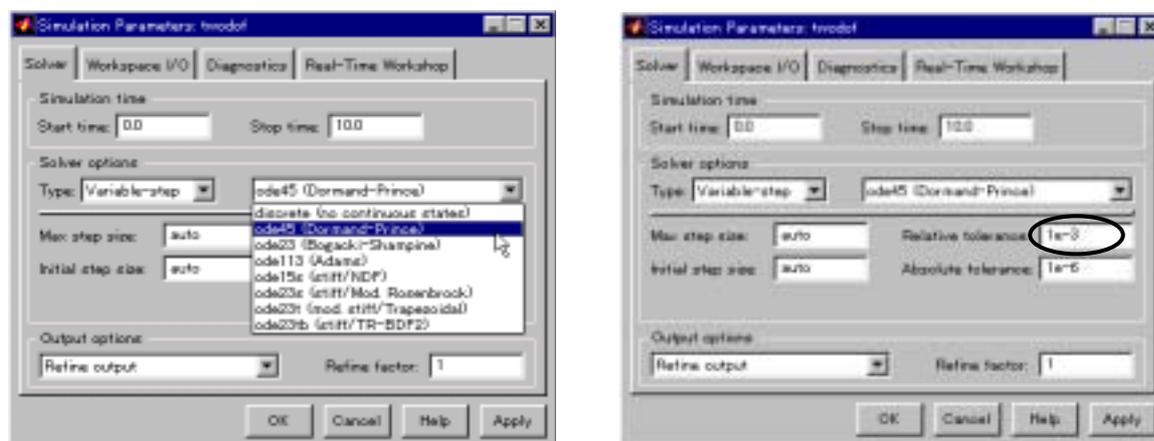


Figure5.1 積分手法の選択

ピッチ角のステップ応答は Simulink-Sinks-Scope のスコープで見ることが出来る (ToWorkspace を利用して MATLAB コマンドでプロットしてもよい)。シミュレーションのパラメータを 0 から 15 秒までに設定して色々とステップ応答を調べてみよう。

次に昇降舵からピッチ角までの全てのブロックを選択して sublock 化し、適当にリサイズしよう。以降は必要に応じてこのブロックをダブルクリックすれば中身がどうなっているかを見ることが出来る。(Viewer + Model Browser + Show Model Browser でも見れる。)

この積分器ブロックベースのモデルではピッチ角のみならず、ピッチレートや前後速度の応答も見ることが出来るので大変便利であるが、ピッチ角のみであれば伝達関数や状態方程式を使ってもよい。第一週目ではこれらをすでに求めているので、Simulink-Continuous-TransferFcn と Simulink-Continuous-StateSpace によりこれらもモデルファイルに含めてみよう。またこれら三つの応答を比較し、すべてが同じであることを確認すること。

6. 【その他。この項は単に参考情報である。

コマンドラインシミュレーション: Simulink では、MATLAB コマンドラインから時間応答解析を行うこともできる。例えばシミュレーションパラメータを設定したり、シミュレーション結果 (時間、状態、出力) を得たりすることができる。この機能を使うと、つぎのような利点がある。

- ・パラメータ設定、シミュレーションの流れを全てコマンドラインから行うことができる。
また、結果を MATLAB、Toolbox の関数で利用することができる。
- ・上記の流れをスクリプト M-File にし、バッチ処理的に実行することができる。

・ファンクション M-File 中でシミュレーションを行い、結果を関数の中で利用することができる。

コマンドラインシミュレーションの流れは以下のようになる。

パラメータ (変数、シミュレーションパラメータ) の設定

シミュレーション (時間応答解析) の実行

結果のグラフ化 (MATLAB グラフィック機能)

線形化・平衡点算出 : Simulink では、時間応答解析だけではなく、線形化、平衡点算出という解析も行うこともできる。必要であればモデルのメニュー "*Tools - Linear Analysis*", LTI Viewer 等を利用してボード線図も書ける。】

7. 終了

一通り項目を終了したら教官あるいは TA にその旨を告げ、**口頭試問**を受ける。これにパスしたらメダタク終了となる。

各自が作成したファイルは Z ドライブあるいはディスクなどに**保管**してあることを確かめる。特に、3L204 で実験を行った場合は、各グループで作成したファイルをフロッピーなどにコピーして保管し、MATLABR11/WORK/のサブディレクトリーから消去すること。ファイルをセーブした後に正規の手続きによりからネットワークからログアウトあるいはパソコンをシャットダウンする。

来週は VTOL 機のピッチ角制御システムを根軌跡法と呼ばれる方法により設計する。時間があればどのような方法であるか下調べをしておくといよい。(この方法は制御の入門書にはほぼ必ず載っている。)