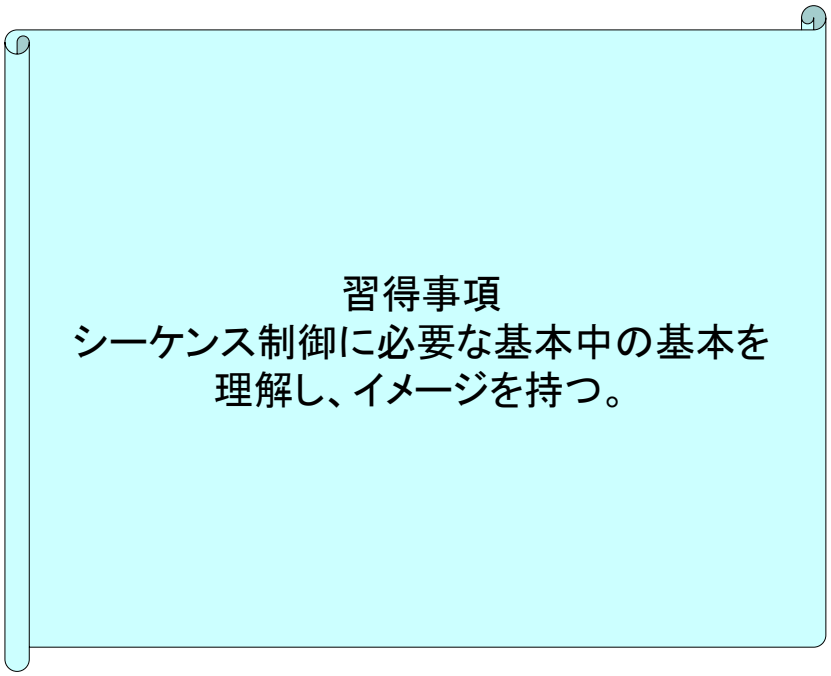
内容

注記：本資料は三田 宇洋著、「MATLAB/Simulink によるモデルベースデザイン入門」、オーム社,2013 より抜粋

2021/4/24　roboOne MATLAB セミナー資料

Stateflow とシーケンス制御

6.1　導入



本章では、Simulink®のオプションプロダクトである Stateflow®を用いて、シーケンシャルなアルゴリズムを記述する方法について、簡単なサンプルを元に説明します。

Stateflow の表記方法では大別して、フローチャートとステートチャート(状態遷移図)があります。

Simulink は、ブロック線図環境でシステムのモデリングを行います。ブロック線図は、データがあるブロックの入力に入り、ブロックで演算された結果が出力されます。この結果を別のブロックに入力し、計算ブロックで演算された結果が出力されます。この機作を繰り返すことで、システム全体がモデリングされます。ブロック線図はデータフローのシステムをモデリングするのに適します。

シーケンス制御は、分岐、判断および繰り返しを伴うロジックが頻出します。分岐、判断および繰り返しを伴うロジックは、フローチャートとステートチャートで高い可読性をもって表現できます。

分岐、判断および繰り返しでは、一般に次のような構文があります。Simulink はこれらの構文のモデリングも可能ですが、Stateflow を使うことで、システムはより可読性の高い明確な表現となります。

if then else

do while

switch case

for
try catch

この章では、Stateflow に親しんでもらうため、基本的な事項に絞りこみ説明します。詳細に関しては文献[1]をご覧ください。以降では、シーケンス制御に関連する事項で必ず押さえたいポイントを簡単に説明します。

6.1.1 補足　シーケンス制御の簡単な説明

実産業で使われる制御は大別して、シーケンス(デジタル)制御とフィードバック(アナログ)制御に分かれます。

シーケンス制御は”逐次制御”ともいいます。外部からの指令に対応して処理が順次進行していく制御です。いろいろな”モノ”がシーケンス制御で制御されています。

例えば、学校や大型建築物の空調(冷暖房)、照明等の管理はシーケンス制御が多くなりました。大型のシステムでは、”制御盤”と呼ばれる筐体に制御装置が収められることが多いです。これらの制御盤は、街中でも見受けられます。例えば、工場の大型機器、貯水槽、街灯、ポンプ場、太陽光/風力発電の付近には、それらの制御のために制御盤が置かれます。

家庭にもシーケンス制御は多く使われています。クーラー、給湯システム、冷蔵庫、テレビ、電子レンジ、洗濯機等ほとんどの複雑な機械にはシーケンス制御が組み込まれているといっても過言ではありません。

例えば、洗濯機を取り上げます。洗濯機のスイッチを入れると水が充填され、洗濯、脱水、乾燥といった処理が自動的に進行します。洗濯機の中にマイコンがあり、このような処理をするようプログラムされています(図 6-1 参照)。

シーケンス制御は、人間に例えると、頭脳の役割はマイコンのような計算機が司ります。五感に相当するセンサーや腕時計に相当するタイマーの情報を元に、頭脳が判断し、手足に相当するアクチュエータを動かす仕組みです。シーケンス制御が扱う情報はデジタルの ON/OFF(1/0)やアナログ値に対する判断、時間のカウントが主です。このことからシーケンス制御はデジタル制御とも呼ばれます[2]。

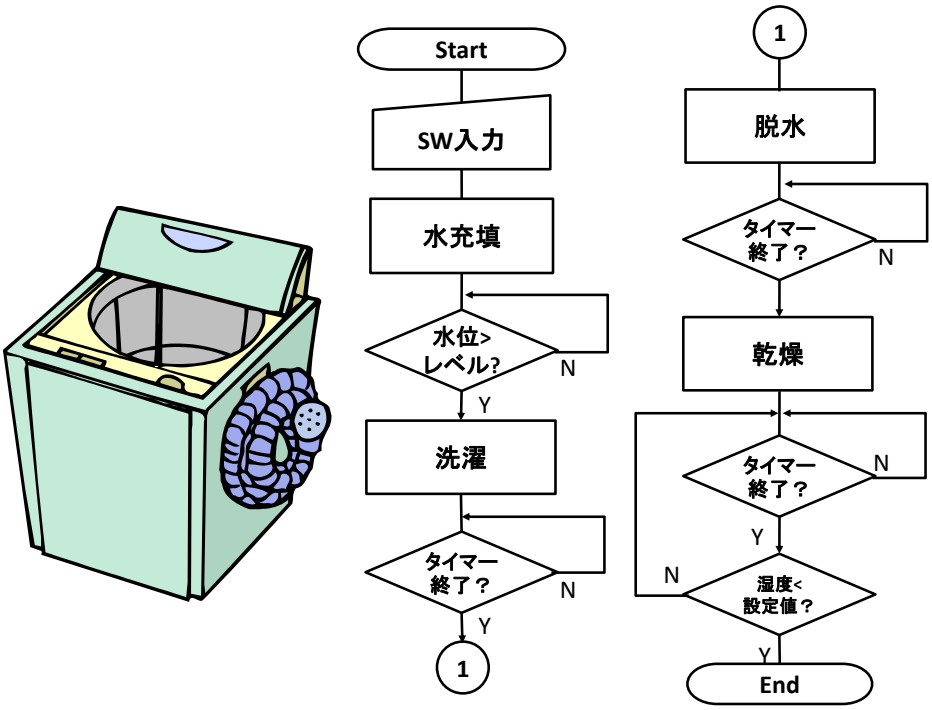


図 6-1　洗濯機のシーケンス　イメージ

6.1.2　補足　フィードバック制御の簡単な説明

フィードバック制御は、センサーで”モノ”の情報をとり、制御器は、センサー値を元にどのように制御するか計算して”モノ”を動かす制御です。典型的なフィードバック制御は、与えられた設定値に対し、設定値に忠実に”モノ”を動かそうとする機構を持ちます。この機構を“サーボ”と呼びます(図 6-2)。サーボは外部から与えられた設定値とセンサーの情報の差分を取り、差分から”モノ”をどう動かすか計算します。差分が大きければ、より大きく動き、差分が小さければ小さく動くよう、制御器の出力(操作量)を変えます。ただし、この操作量はデジタルの計算値であり、この値には直接”モノ”を動かすパワーはありません。

”モノ”を動かすためには、操作量を物理的に変換する必要があります。モータ制御では、ドライバがその役割を担います。操作量が D/A 変換されアナログ値(電圧や電流)に変えられます。そのアナログ値がアクチュエータを動かし、その結果”モノ”が動きます。センサーはモノの情報を測定します。測定する値は電圧や電流です。電圧や電流が A/D 変換でデジタル値に変えられ、制御器に戻されます。この動作が制御器の制御周期毎に繰り返されます[2]。

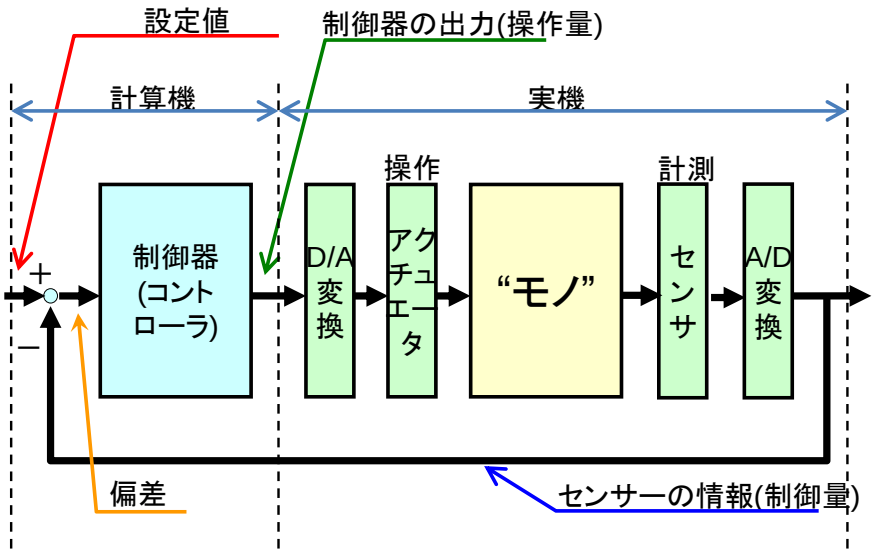


図 6-2　サーボ機構

6.1.3　補足　シーケンス制御とフィードバック制御の違い

シーケンス制御は、逐次的に工程の処理が進行します。シーケンス制御器はセンサーの情報を見て、工程のステップを進行させるか否かを判断します。しかし、シーケンス制御は、センサーのアナログ情報を見て、サーボとして”モノ"を設定値通りに動かすことはしません。あくまで、工程の進行を進める判断条件としてアナログ情報を見ています。

フィードバック制御は、工程の進行は管理しません。それはシーケンス制御に任せ、サーボとして、センサーのアナログ情報を所望の値に制御するようアクチュエータに指令します。シーケンス制御とフィードバック制御はお互いに協力し、期待される”モノ”の動きを実現します。一般に、制御システムの内部はシーケンス制御が 8 割以上占めることが多いようです[2]。

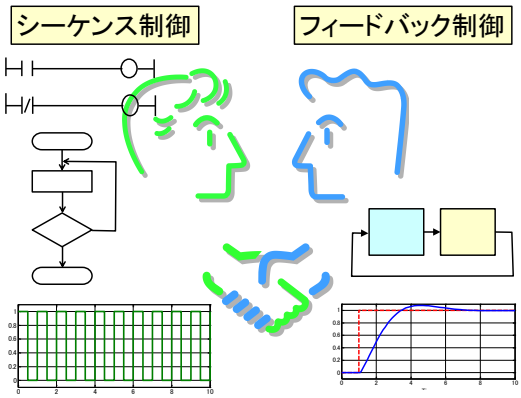


図 6-3　シーケンスとフィードバックの協力

6.1.4　補足　ドライバ回路の簡単な説明

マイコンは小さな電子回路の集合体です。通常マイコンは直接大電力を扱うことはできません。マイコンが流せる電流はせいぜい 20[mmA]といった微小な電流です。マイコン単体では、遊具用の小さなモータですら動かすことはできません。しかし、アクチュエータは、油圧機器、電気機械等大電力を必要とするものが多くなります。

マイコンはこれら大型のアクチュエータをどのように動かすのでしょうか？

それにはドライバと呼ばれる回路が必要になります。ドライバは半導体を含む電気回路です。ドライバはアクチュエータに大電力を流す仕掛けを持ちます。マイコンからの微小な信号が半導体の駆動信号になります。駆動信号は ON/OFF(1/0)で半導体を導通/導通停止させることができます。駆動信号の ON/OFF(1/0)に伴い、アクチェータ用電源から流れる電流が ON/OFF します。電流 ON 時にアクチュエータが動作します。アクチェータ用電源はアクチュエータを動かすために必要な電力を供給できます(図 6-4 参照)。

この仕組みを応用して、小電力のマイコンが大電力のアクチュエータを動かすことができます[2]。

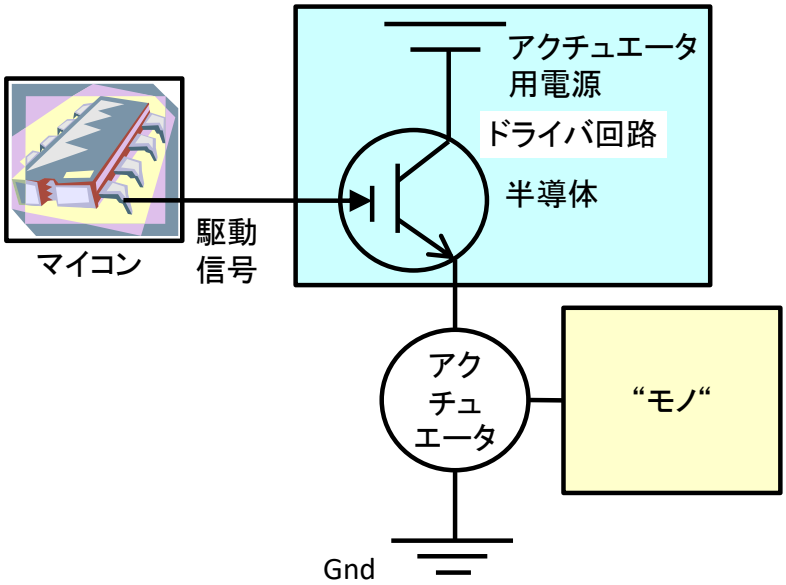


図 6-4　ドライバ回路のイメージ

6.1.5　補足　I/O の簡単な説明

制御器に入る信号と制御器から出す信号を I/O と呼びます。マイコンのような計算機と実センサーやドライバ回路をつなぐために端子台を経由して結線します。実例を図 6-5 に示します。図 6-6 は図 6-5 のモータを制御する仕組みです。ポテンショメータが角度のセンサー、モータドライバがアクチュエータの役割を持ちます。計算機が制御器の役割を持ちます。

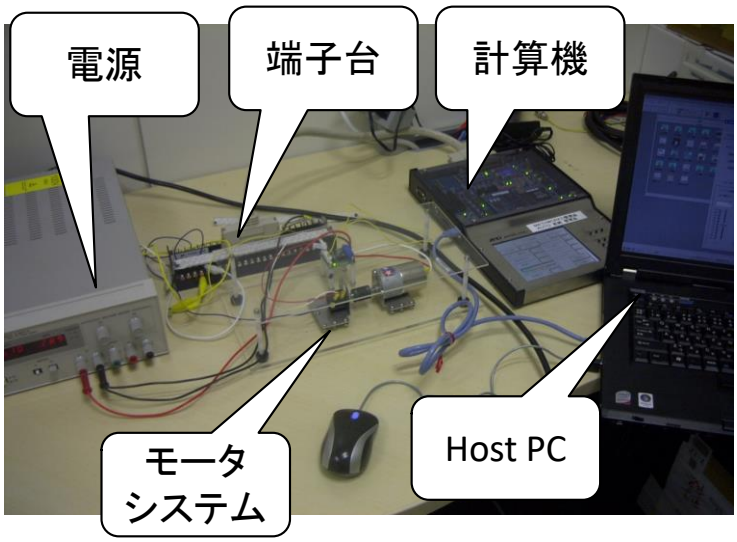


図 6-5　制御システムの例[3]

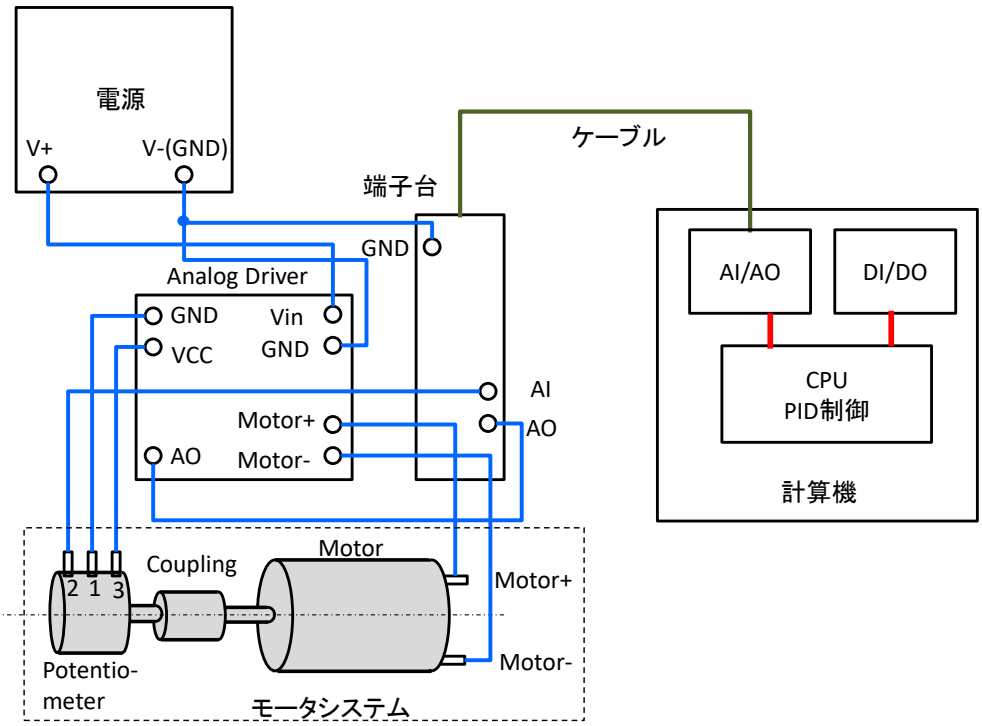


図 6-6　図 6-5/制御システムの模式図[3]

代表的な I/O には次の種類があります。

入力(Input)

(1) DI: Digital Input

DO: Digital　Output

信号の ON/OFF(1/0) を例えば 5/0[V]で表現します。デジタル値とみなします。 DI で 5[V]が入力されれば ON(1)です。DO で 5[V]を出力すれば ON(1)です。

(2) AI: Analog Input

AO:Analog Output

信号を例えば 4-20[mmA]のレンジとします。4-20[mmA]を正規化し 0-100[%]とします。

AI ではセンサーから取り入れる信号の値を 0-100[%]の範囲の値とします。

例えば入力値が 12[mmA]なら 50[%]です。

AO では、アクチュエータに与える信号をを 0-100[%]の範囲の値とします。

(3) PI: Pulse Input

PO:Pulse Output

PI は DI の高機能版です。高速なパルス列の計数機能を持ちます。エンコーダやパルスカ

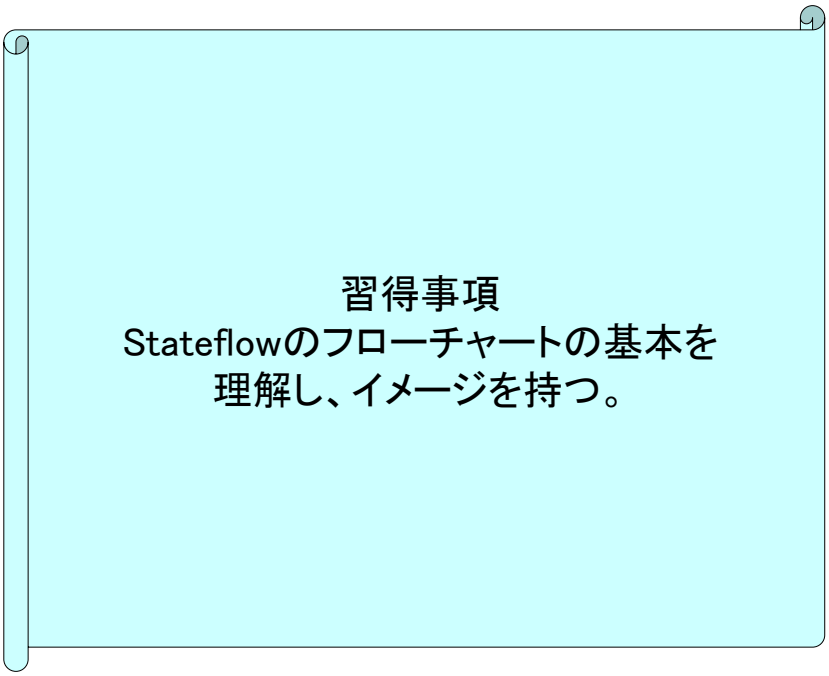
ウントと呼ばれます。
PO は DO の高機能版です。高速なパルス列の出力機能を持ちます。カウンターやパルストレインとも呼ばれます[2]。

6.2 Stateflow について

Stateflow は、ステートマシンとフローチャートに基づいて組み合わせとシーケンシャルの判定ロジックをモデル化およびシミュレーションするための環境です。Stateflow では、状態遷移図、フロー チャート、状態遷移表、真理値表などのグラフィカルな表現と表形式の表現を組み合わせて、イベント、時間ベースの条件、外部入力信号に対するシステムの応答をモデル化することができます。

Stateflow は Simulink と一緒に用いることで、ロジック的な挙動（故障管理やモード切替えなど）とアルゴリズム的な挙動（フィードバック制御や信号の条件付け等）を組み合わせた組込みソフトウェアのモデルを作成することができます。また、Simulink にてシステムとその環境のモデルを作成し、アルゴリズムとロジックとの相互作用を検討するために、ハイブリッドシミュレーションを行うこともできます。
また、Stateflow のモデルはコードの生成に対応します。
Simulink® Coder™（オプション製品）を用いて、Stateflow チャートをさまざまなアプリケーションに展開することができます[4]。

6.3 Stateflow とフローチャート



簡単な例で、Steteflow におけるフローチャートを説明します。

データを昇順(小から大へ)、または降順(大から小へ)に処理したいことがあります。このような場合、データを順に並べるためにソート・アルゴリズムが使われます。バブルソート(Bubble Sort)は、単純で最も普及しているソート・アルゴリズムの一つです。繰り返しの度に、値が泡(bubble)のように配列の下から上に上るので、この名で呼ばれます[5]。バブルソートのフローチャートを図 6-7 に示します。

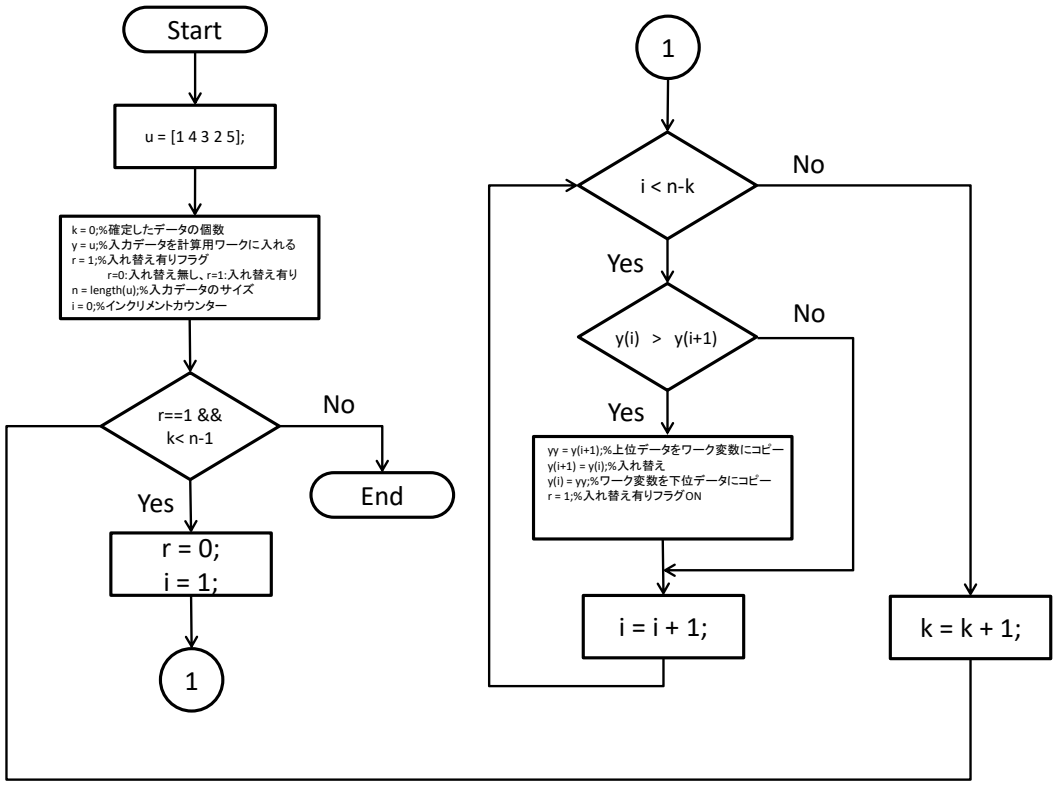


図 6-7　バブルソートのアルゴリズム

図 6-7 のフローチャートを MATLAB プログラムで書いたバブルソートの例を図 6-8 に示します。プログラムの実行結果を図 6-9 に示します。処理の進行につれ、小さい数値が左、大きい数値が右に移動するのがわかります。

```
u = [4 3 2 1 5];%入力データ

k = 0;%確定したデータの個数
y = u;%入力データを計算用ワークに入れる
r = 1;%入れ替え有りフラグ r=0:入れ替え無し、r=1:入れ替え有り
n = length(u);%入力データのサイズ
i = 0;%インクリメントカウンター

while((r == 1) && (k < n-1))
    r = 0;%入れ替え有りフラグOFF
        for i = 1:n-k-1
            if(y(i) > y(i+1))
                yy = y(i+1);%上位データをワーク変数にコピー
                y(i+1) = y(i);%入れ替え
                y(i) = yy;%ワーク変数を下位データにコピー
                r = 1;%入れ替え有りフラグON
            end
        end
        k = k+1;%確定したデータの個数を１増やす
k,y
end
```

(sort_logic.m)

図 6-8　MATLAB で書いたバブルソートの例

```
k =   1
y =   3   2   1   4   5

k =   2
y =   2   1   3   4   5

k =   3
y =   1   2   3   4   5

k =   4
y =   1   2   3   4   5
```

図 6-9　実行結果

同様のバブルソートのアルゴリズムを Stateflow に実装した例とその計算結果を図 6-10 に示します。MATLAB の場合と同様、計算結果は昇順に並べ替えられていることがわかります。

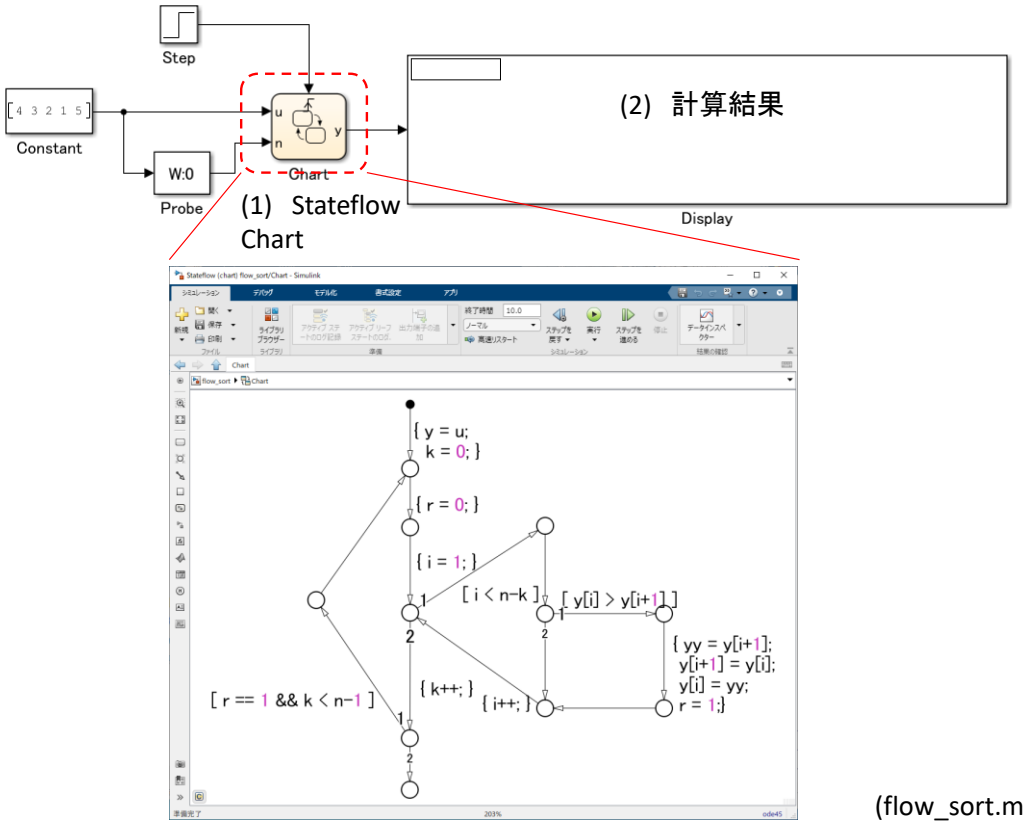


図 6-10　Stateflow によるバブルソートのモデルと実行結果

Stateflow のフローチャートは、基本的には、図 6-11 の 3 つのオブジェクト：(1)既定遷移、(2)遷移及び(3)コネクティブジャンクションで 表されます。(1)既定遷移は、Stateflow Chart がアクティブになった時の処理の開始点を示します。フローチャートの処理は(2)遷移の矢印の方向に進行します。(3)コネクティブジャンクションは、処理の区切りや分岐判断を示すのに用います。

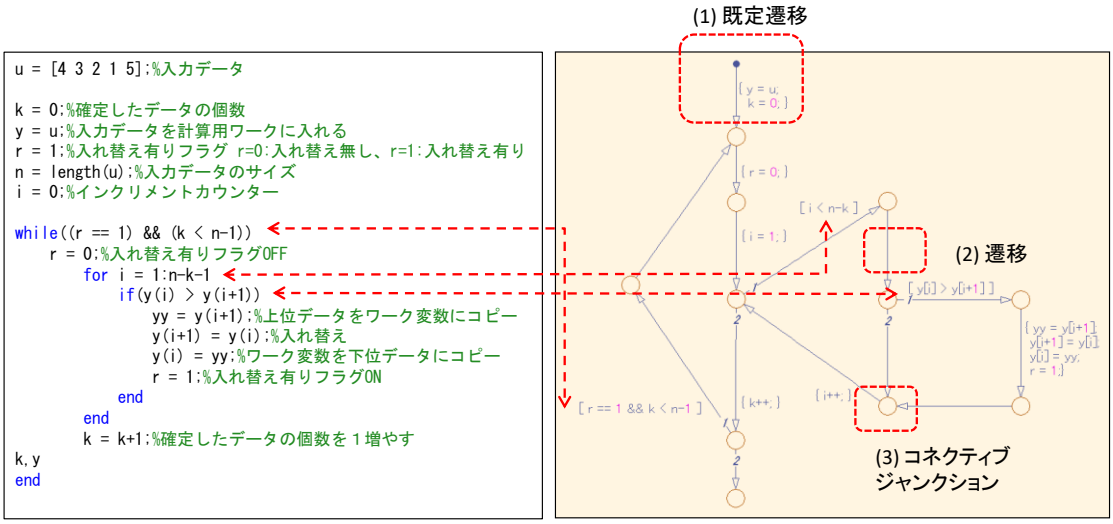
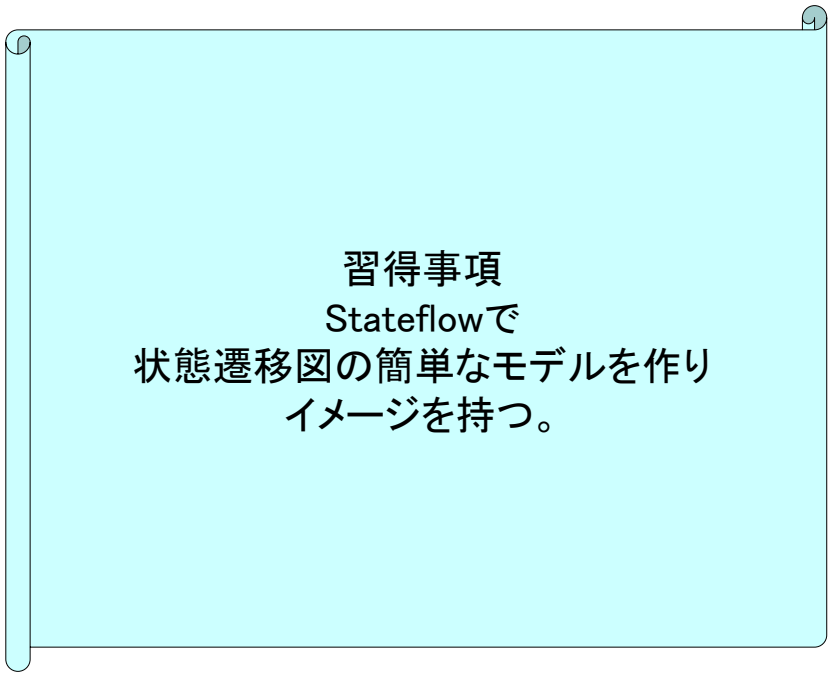


図 6-11　バブルソート MATLAB と Stateflow の対応

6.4　Stateflow とステートチャート　演習



6.4.1　課題

本節では、次の課題が与えられたとします。この課題を演習方式で実施します。

【課題】

図 6-12 は図 6-5 のシステムの一部拡大である。図 6-12 の DC モータを正逆転、停止制御するロジックを Stateflow で作成する。

ドライバ回路の入力信号は 5[v]でモータ正転、2.5[v]でモータ停止、0[v]でモータ逆転する。DC モータとドライバ回路及び Stateflow の関係を図 6-13 に示す。ドライバ回路と Stateflow chart の間には、(2)数値→電圧変換器があり、Stateflow の出力信号を等しい電圧に変換してドライバ回路に入力できる。Stateflow は出力信号を次の値とすればよい。

モータ正転:5[-]、モータ停止:2.5[-]、モータ逆転:0[-]

Stateflow の動作仕様を次とする。

【Stateflow の動作仕様】

(1) 通常動作

Simulink からの入力信号(in)

Stateflow の出力信号(out)

in==1 の場合、out = 5　　（モータ正転）

in==0 の場合、out = 2.5　　（モータ停止）

in==-1 の場合、out = 0　　(モータ逆転)

(2) 異常時の出力

緊急停止 SW(EM)の値が変化すると、

out = 2.5　（モータ停止）

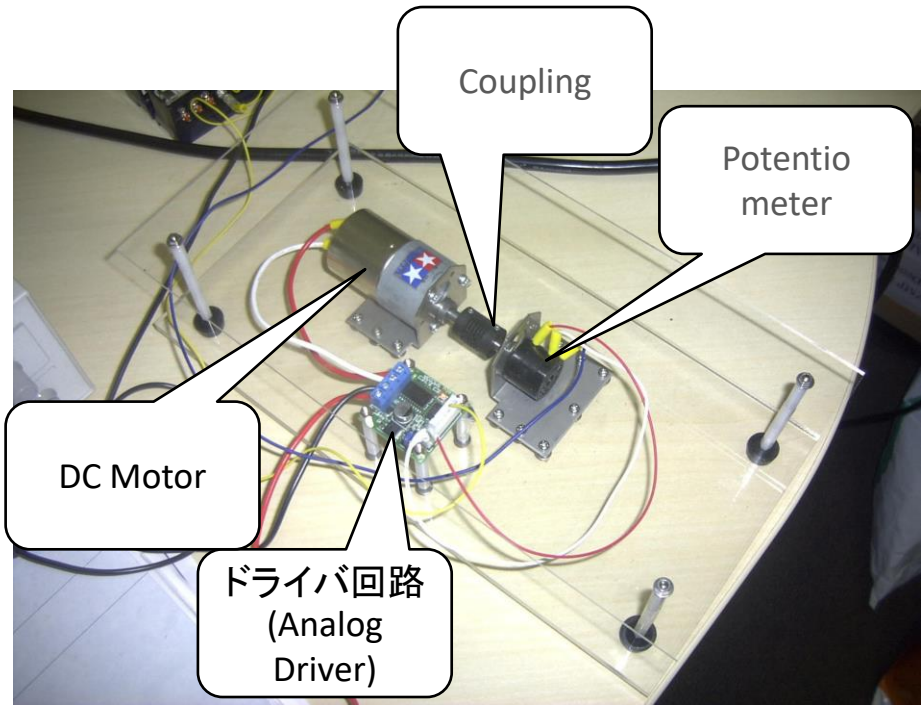


図 6-12　図 6-5 の一部拡大[3]

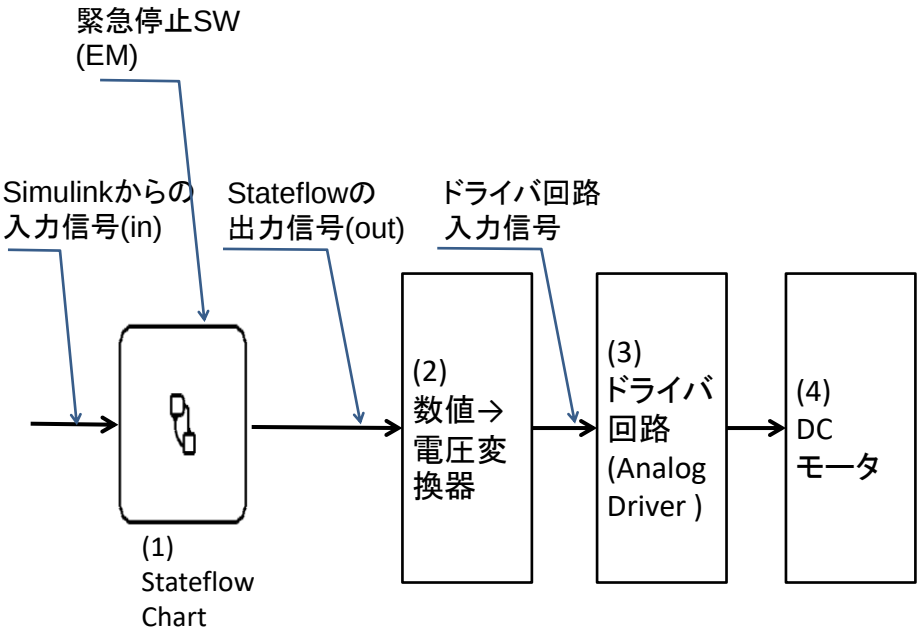


図 6-13　DC モータとドライバ回路及び Stateflow の関係

6.4.2　演習

【演習の手順】

[1] テンプレートファイル(exam_basic_template.mdl)を開いてください。テンプレートファイルには、Stateflow チャート以外に必要な情報は全て登録されています。

図 6-14/(1) Simulink からの入力信号は、マニュアルによるダブルクリックで-1,0,1,-1,…と切り替わります。

図 6-14/ (2) マニュアルスイッチにより、緊急停止 SW の入力を 0/1 で切り替えます。

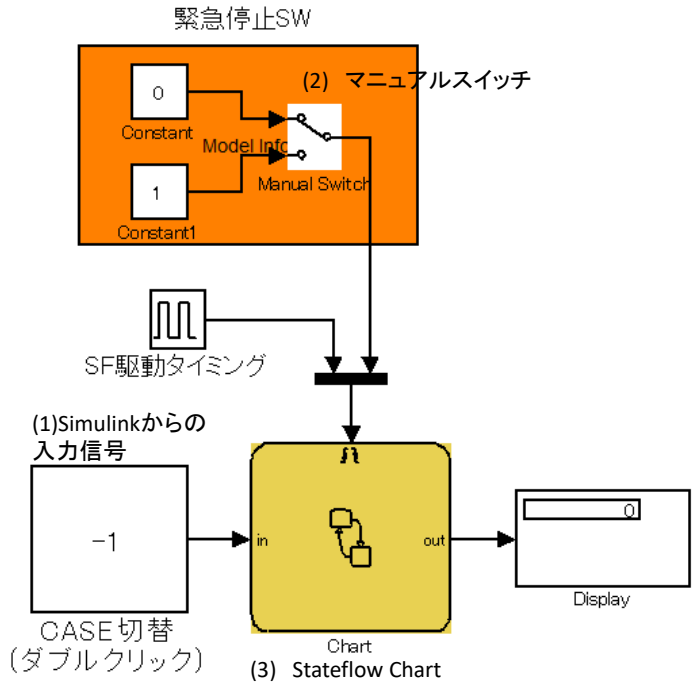


図 6-14　テンプレートファイル(exam_basic_template.mdl)

[2]　下記のモデル(図 6-15)を Stateflow Chart に作成してください。ここで必要なオブジェクトを図 6-16 に示します。

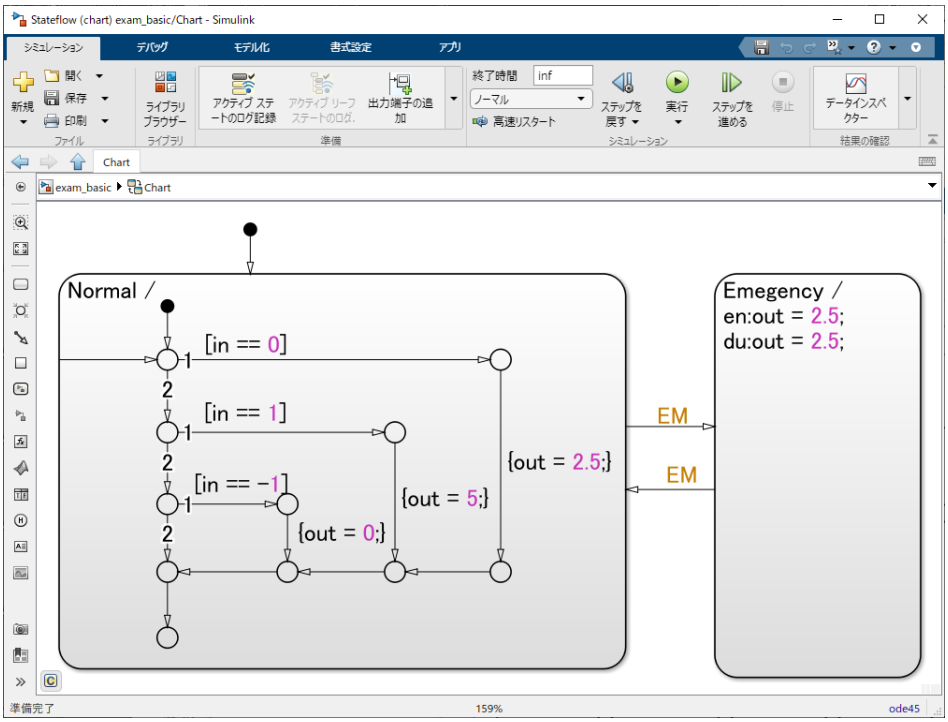


図 6-15　Stateflow Chart に記載するロジック

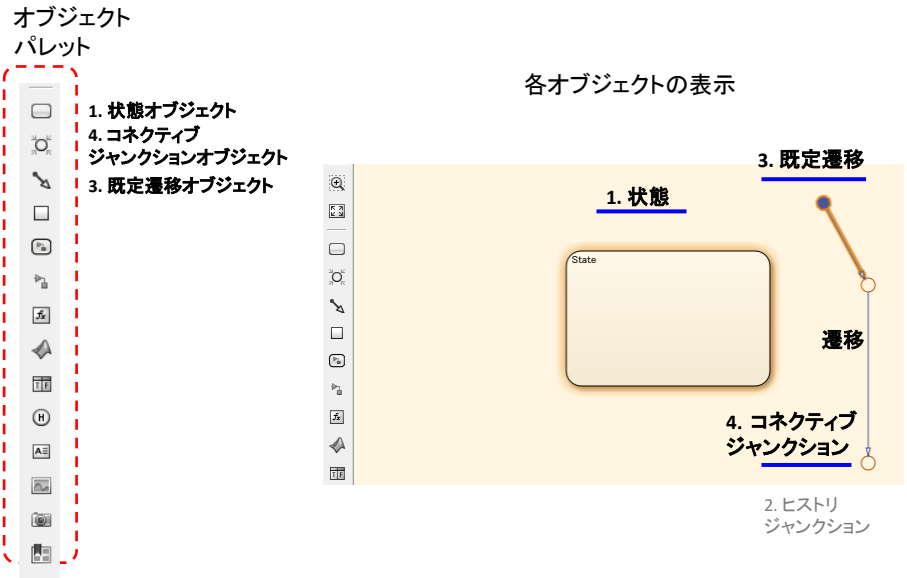


図 6-16　必要なオブジェクト

(3)　シミュレーションを実行し、下記表の仕様が満たされたか'ＯＫ？'カラムに記入してください。

表 6-1　動作チェック表

項目	仕様詳細	OK？
#1. 通常動作	Simulinkからの入力信号(in) Simulinkへの出力信号(out) in==1の場合、out = 5　（モータ正転） in==0の場合、out = 2.5　（モータ停止） in==-1の場合、out = 0　(モータ逆転)	
#2. 異常時の出力	緊急停止SWの値が変化すると、 out = 2.5　（モータ停止）	

6.4.3　モデルの説明

回答例のモデルを図 6-17/図 6-18 に示します。

図 6-17/(1)SF 駆動タイミングは表 6-2 の Event/RUN に相当します。RUN の ON/OFF

切り替え時の両方(立ち上がり/立下がり)のタイミングで Chart を駆動させるイベントが発行されます。図 6-17/(1)は 0.5 秒毎に ON/OFF するパルスなので、SF 駆動タイミングは、その立ち上がり/立下がりのタイミング(0.5 秒毎)で ON/OFF が切り替わります。図 6-17/(2)は表 6-2 の EM に対応します。緊急停止 SW をマニュアルで切り替えたタイミングでイベント EM が発行されます。

図 6-18/(A)では、状態 Normal の内部に次のロジックがフローチャートで記されています。また状態 Normal がアクティブの場合に、本ロジックを繰り返すよう、(a)の遷移を加えています。

【ロジック】

```
if in == 0
   out = 2.5;
elseif in == 1
   out = 5;
else in == -1
   out = 0;
end
```

注：文法は MATLAB で記載

図 6-18/(B)ではイベント EM が発行される毎に、アクティブな状態が(A)状態 Normal から(C)状態 Emergency に切り替わります。

図 6-18/(C)では、状態 Emergency がアクティブになったタイミング(entry:en)で out=2.5 を実行し、状態 Emergency がアクティブになった以降のタイミング (during:du)で out=2.5 を実行します。語句の定義は表 7-14 を参照してください。

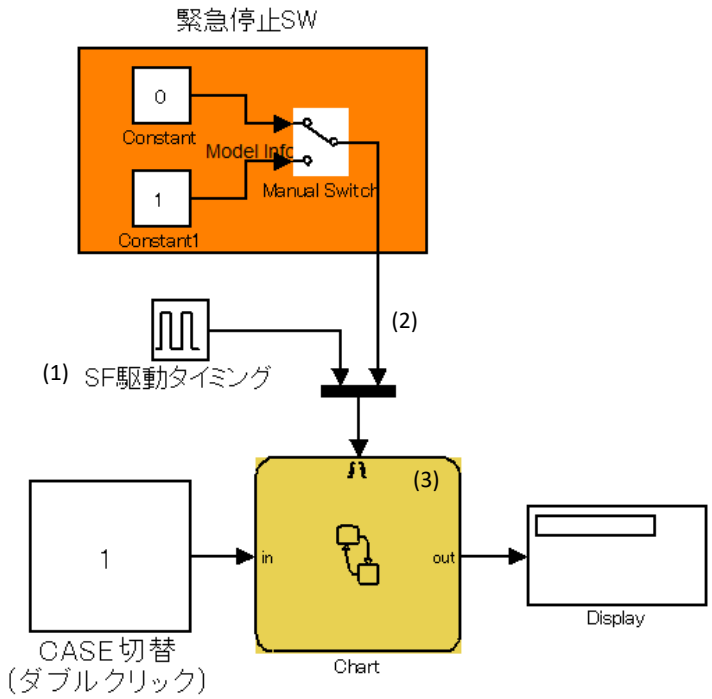


図 6-17　回答のモデル

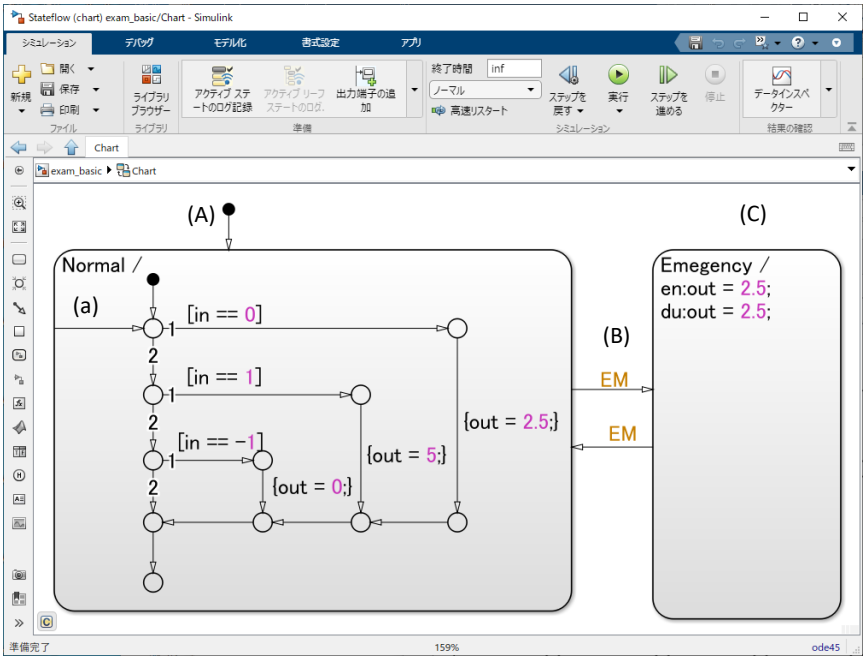


図 6-18　Stateflow Chart 内部のロジック

表 6-2　Event と Data の設定

【 Event 】

名前	スコープ	端子	トリガ
RUN	Simulinkから入力	1	両方
EM	Simulinkから入力	2	両方

【 Data 】

名前	
in	Input
out	Output

注：EventとDataの他パラメータはデフィルト値

参考文献

[1] Stateflow®ユーザーガイド R2012a, The MathWorks,Inc., 2012

[2] 三田宇洋 著:”就職に役立つ技術を学ぼう　MATLAB/Simulink を用いたロボットアームの制御 第 3 回”, pp. 66-68, ロボコンマガジン 2011 年 11 月号, 株式会社オーム社, 2011

[3] 三田宇洋,” 教育向け DC モータ制御のリアルタイムシミュレーション”, 技術資料 ,マスワークス ジャパン, 2011

[4] MathWorks,Stateflow ブローシャ,The MathWorks,Inc.,2012

[5] K.ジャムサ著、戸内順一訳,C ライブラリ,pp.127,マグロウヒルブック社,1986