

**CONTROL ALGORITHM MODELING  
GUIDELINES USING MATLAB<sup>®</sup>,  
Simulink<sup>®</sup>, and Stateflow<sup>®</sup>  
Version 3.0(和訳)**

**Japan MBD Automotive Advisory Board  
(JMAAB)  
2013 年 5 月 30 日**

## ■ 著作権について

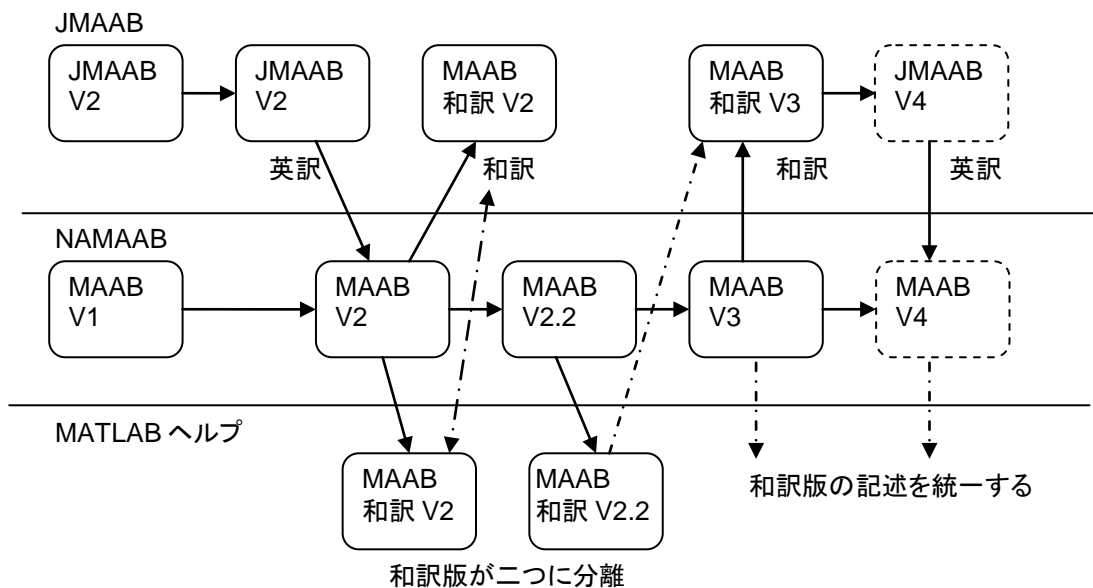
- 本ドキュメントの著作権は、JMAAB に帰属します。
- JMAAB は、本文書の内容に関し、いかなる保証もするものではありません。万一本文書を利用して不具合等があった場合でも、JMAAB は一切責任を負いかねます。また、本文書に記載されている事項は予告なしに変更または廃止されることがありますので、あらかじめご了承ください。

## ■ 本ドキュメントの取扱いについて

- 本文書は、非営利目的、または利用者内部で使用する場合に限り、複製が可能です。また、本文書を引用する場合は、本文書からの引用であることを明示し、引用された著作物の題号や著作者名を明示する等の引用の要件を満たす必要があります。
- 本成果物については、JMAAB ホームページ (<http://jmaab.mathworks.jp/>) を参照下さい。
- その他のお問い合わせは、JMAAB 事務局 ([jmaab-office@mathworks.co.jp](mailto:jmaab-office@mathworks.co.jp)) へご連絡下さい。

## ■ 本ドキュメントにおける注意事項

- 本文書は、“CONTROL ALGORITHM MODELING GUIDELINES USING MATLAB®, Simulink® and Stateflow® Version 3.0”を和訳したものである。
- Version2.0 は読みやすくなる意識を行ったが、MATLAB のヘルプにマスワークス社が原文を直訳した異なる和訳版が提供されていた。和訳が二つ存在する事は、ユーザーにとってメリットにならないので、より多くの人が目に触れる MATLAB のヘルプに掲載された和訳を採用することにした。本 Version3.0 は、MATLAB R2013a に掲載されているマスワークス社の V2.2 に表現を統一し、Version3.0 の新ルールについては、マスワークス社から和訳を先行入手し全体をまとめました。一部ガイドラインの内容が理解しがたい内容となっておりますが、原文に忠実に訳されています。解りにくいルールについては、Version4.0 で表現を変更し、マスワークスのヘルプを含め、全体の仕組みに正しく反映されるようにする予定です。あらかじめご了承ください。



<b>CONTROL ALGORITHM MODELING GUIDELINES USING MATLAB®, Simulink®, and Stateflow® .....</b>	<b>1</b>
<b>1. 更新履歴 .....</b>	<b>6</b>
<b>2. はじめに.....</b>	<b>7</b>
2.1. ガイドラインとは.....	7
2.2. Notes on version 3.0.....	7
2.3. ガイドラインのテンプレート.....	7
2.3.1. ID .....	8
2.3.2. タイトル.....	8
2.3.3. 重要度.....	8
2.3.4. 範囲.....	8
2.3.5. MATLAB バージョン .....	9
2.3.6. 前提条件.....	9
2.3.7. 記述内容.....	9
2.3.8. 関連.....	9
2.3.9. 更新履歴.....	9
2.4. 根拠.....	9
2.5. ガイドラインの使用 .....	9
2.5.1. ガイドラインの構成.....	9
2.5.2. Masked Subsystems and Readability Rules .....	10
<b>3. Software Environment.....</b>	<b>11</b>
3.1. General Guidelines.....	11
3.1.1. na_0026: Consistent software environment .....	11
3.1.2. na_0027: Use of only standard library blocks .....	11
<b>4. 命名規則 .....</b>	<b>12</b>
4.1. 一般規則.....	12
4.1.1. ar_0001: Filenames .....	12
4.1.2. ar_0002: Directory names .....	12
4.1.3. na_0035: Adoption of naming conventions .....	13
4.2. モデル内の規則 .....	13
4.2.1. jc_0201: Usable characters for Subsystem names .....	13
4.2.2. jc_0211: Usable characters for Inport blocks and Outport blocks .....	14
4.2.3. jc_0221: Usable characters for signal line name .....	14
4.2.4. na_0030: Usable characters for Simulink Bus names.....	15
4.2.5. jc_0231: Usable characters for block names.....	15
4.2.6. na_0014: Use of local language in Simulink and Stateflow .....	16
<b>5. モデルアーキテクチャ .....</b>	<b>18</b>
5.1. Simulink と Stateflow の使い分け .....	18
5.1.1. na_0006: Guidelines for mixed use of Simulink and Stateflow .....	18
5.1.2. na_0007: Guidelines for use of Flow Charts, Truth Tables and State Machines.....	22
5.2. サブシステムの階層構造.....	22
5.2.1. db_0143: Similar block types on the model levels .....	22
5.2.2. db_0144: Usage of Subsystems .....	24
5.2.3. db_0040: Model hierarchy.....	25
5.2.4. na_0037: Use of single variable variant conditionals .....	25
5.2.5. na_0020: Number of inputs to variant subsystems.....	26
5.2.6. na_0036: Default variant.....	26
5.3. J-MAAB Model Architecture Decomposition .....	27
5.3.1. jc_0301: Controller model.....	27
5.3.2. jc_0311: Top layer / root level.....	27
5.3.3. jc_0321: Trigger layer.....	28
5.3.4. jc_0331: Structure layer.....	28
5.3.5. jc_0341: Data flow layer.....	29
<b>6. モデル コンフィギュレーション オプション .....</b>	<b>31</b>

6.1.1. jc_0011: Optimization parameters for Boolean data types.....	31
6.1.2. jc_0021: Model diagnostic settings.....	31
<b>7. Simulink.....</b>	<b>32</b>
7.1. Diagram Appearance.....	32
7.1.1. na_0004: Simulink model appearance.....	32
7.1.2. db_0043: Simulink font and font size.....	33
7.1.3. db_0042: Port block in Simulink models.....	33
7.1.4. na_0005: Port block name visibility in Simulink models.....	34
7.1.5. jc_0081: Icon display for Port block.....	35
7.1.6. jm_0002: Block resizing.....	35
7.1.7. db_0142: Position of block names.....	36
7.1.8. jc_0061: Display of block names.....	36
7.1.9. db_0146: Triggered, enabled, conditional Subsystems.....	37
7.1.10. db_0140: Display of basic block parameters.....	38
7.1.11. jm_0013: アノテーションの影.....	38
7.1.12. db_0032: Simulink signal appearance.....	39
7.1.13. db_0141: Signal flow in Simulink models.....	39
7.1.14. jc_0171: Maintaining signal flow when using Goto and From blocks.....	40
7.1.15. na_0032: Use of Merge Blocks.....	40
7.1.16. jm_0010: Port block names in Simulink models.....	41
7.1.17. jc_0281: Naming of Trigger Port block and Enable Port block.....	41
7.2. Signals.....	42
7.2.1. na_0008: Display of labels on signals.....	42
7.2.2. na_0009: Entry versus propagation of signal labels.....	43
7.2.3. db_0097: Position of labels for signals and busses.....	44
7.2.4. db_0081: Unconnected signals, block inputs and block outputs.....	44
7.3. Block Usage.....	45
7.3.1. na_0003: Simple logical expressions in If Condition block.....	45
7.3.2. na_0002: Appropriate implementation of fundamental logical and numerical operations.....	46
7.3.3. jm_0001: Prohibited Simulink standard blocks inside controllers.....	47
7.3.4. hd_0001: Prohibited Simulink sinks.....	48
7.3.5. na_0011: Scope of Goto and From blocks.....	48
7.3.6. jc_0141: Usage of the Switch block.....	49
7.3.7. jc_0121: Use of the Sum block.....	50
7.3.8. jc_0131: Use of Relational Operator block.....	51
7.3.9. jc_0161: Use of Data Store Read/Write/Memory blocks.....	51
7.4. Block Parameters.....	52
7.4.1. db_0112: Indexing.....	52
7.4.2. na_0010: Grouping data flows into signals.....	52
7.4.3. db_0110: Tunable parameters in basic blocks.....	53
7.5. Simulink patterns.....	54
7.5.1. na_0012: Use of Switch vs. If-Then-Else Action Subsystem.....	54
7.5.2. db_0114: Simulink patterns for If-then-else-if constructs.....	54
7.5.3. db_0115: Simulink patterns for case constructs.....	55
7.5.4. na_0028: Use of If-Then-Else Action Subsystem to Replace Multiple Switches.....	56
7.5.5. db_0116: Simulink patterns for logical constructs with logical blocks.....	57
7.5.6. db_0117: Simulink patterns for vector signals.....	58
7.5.7. jc_0351: Methods of initialization.....	60
7.5.8. jc_0111: Direction of Subsystem.....	61
<b>8. Stateflow.....</b>	<b>62</b>
8.1. Chart Appearance.....	62
8.1.1. db_0123: Stateflow port names.....	62
8.1.2. db_0129: Stateflow transition appearance.....	62
8.1.3. db_0137: States in state machines.....	63
8.1.4. db_0133: Use of patterns for Flowcharts.....	63
8.1.5. db_0132: Transitions in Flowcharts.....	64
8.1.6. jc_0501: Format of entries in a State block.....	65
8.1.7. jc_0511: Setting the return value from a graphical function.....	66
8.1.8. jc_0531: Placement of the default transition.....	67
8.1.9. jc_0521: Use of the return value from graphical functions.....	67
8.2. Stateflow data and operations.....	68
8.2.1. na_0001: Bitwise Stateflow operators.....	68

8.2.2. jc_0451: Use of unary minus on unsigned integers in Stateflow.....	69
8.2.3. na_0013: Comparison operation in Stateflow .....	70
8.2.4. db_0122: Stateflow and Simulink interface signals and parameters .....	71
8.2.5. db_0125: Scope of internal signals and local auxiliary variables .....	71
8.2.6. jc_0481: Use of hard equality comparisons for floating point numbers in Stateflow .....	72
8.2.7. jc_0491: Reuse of variables within a single Stateflow scope .....	73
8.2.8. jc_0541: Use of tunable parameters in Stateflow .....	74
8.2.9. db_0127: MATLAB commands in Stateflow.....	74
8.2.10. jm_0011: Pointers in Stateflow .....	75
8.3. Events.....	75
8.3.1. db_0126: Scope of events.....	75
8.3.2. jm_0012: Event broadcasts .....	76
8.4. Statechart Patterns .....	77
8.4.1. db_0150: State machine patterns for conditions.....	77
8.4.2. db_0151: State machine patterns for transition actions .....	78
8.5. Flowchart Patterns.....	78
8.5.1. db_0148: Flowchart patterns for conditions.....	78
8.5.2. db_0149: Flowchart patterns for condition actions.....	80
8.5.3. db_0134: Flowchart patterns for If constructs .....	81
8.5.4. db_0159: Flowchart patterns for case constructs.....	82
8.5.5. db_0135: Flowchart patterns for loop constructs.....	83
8.6. State chart architecture .....	84
8.6.1. na_0038: Levels in Stateflow charts .....	84
8.6.2. na_0039: Use of Simulink in Stateflow charts .....	85
8.6.3. na_0040: Number of states per container.....	86
8.6.4. na_0041: Selection of function type .....	86
8.6.5. na_0042: Location of Simulink functions.....	87
<b>9. Enumerated Data .....</b>	<b>88</b>
9.1.1. na_0033: Enumerated Types Usage.....	88
9.1.2. na_0031: Definition of default enumerated value.....	88
<b>10. MATLAB Functions .....</b>	<b>89</b>
10.1. MATLAB Function Appearance.....	89
10.1.1. na_0018: Number of nested if/else and case statement.....	89
10.1.2. na_0019: Restricted Variable Names.....	89
10.1.3. na_0025: MATLAB Function Header .....	89
10.2. MATLAB Function Data and Operations .....	90
10.2.1. na_0034: MATLAB Function block input/output settings.....	90
10.2.2. na_0024: Global Variables.....	91
10.3. MATLAB Function Patterns.....	91
10.3.1. na_0022: Recommended patterns for Switch / Case statements .....	91
10.4. MATLAB Function Usage.....	92
10.4.1. na_0016: Source lines of MATLAB Functions .....	92
10.4.2. na_0017: Number of called function levels.....	92
10.4.3. na_0021: Strings .....	93
<b>11. 根拠、関連まとめ .....</b>	<b>94</b>
11.1. 根拠分類.....	94
11.2. 一覧表.....	94
<b>12. 付録 A: 自動化ツールに対する提言 .....</b>	<b>97</b>
<b>13. 付録 B: ユーザによるガイドライン追加時の注意点.....</b>	<b>98</b>
<b>14. 付録 C: フローチャートの使用例.....</b>	<b>99</b>
<b>15. 付録 D: Stateflow オブジェクトクイックリファレンス.....</b>	<b>104</b>
<b>16. 付録 E: 参考資料.....</b>	<b>105</b>

## 1.更新履歴

Date	Change
02.04.2001	Initial document Release, Version 1.00
04.27.2007	Version 2.00 Update release
07.30.2011	Version 2.20 Update release
08.31.2012	Version 3.0 Update release
05.30.2013	Version 3.0 日本語化

## 2.はじめに

### 2.1. ガイドラインとは...

自動車用制御装置のモデルを運用する上で、作成者と使用者の間に容易に共通の理解が得られるように、Simulink / Stateflow モデルの記述について重要な基本的なルールを規定したものである。

主に、以下のことを目的として制定した。

- システム結合の容易化
- インターフェースの明確化
- モデル、コード、ドキュメントの体裁統一
- 再利用性の向上
- 可読性の確保
- モデル交換の容易化
- 簡潔かつ効率的なプロセスの実現
- 完成度が高いドキュメントの作成
- 仕様理解の容易化
- 仕様変更の容易化
- モデル作成者、使用者間の協調
- 先行開発から量産開発へのスムーズな移行

### 2.2. Notes on version 3.0

The current version of this document, 3.0, supports MATLAB releases R2007b through R2011b. Version 3.0 references rules from the NASA Orion style guidelines (<http://www.mathworks.com/aerospace-defense/standards/nasa.html>). Rules that are referenced from the NASA Orion guideline are noted with a “See also” field that provides the original rule number.

### 2.3. ガイドラインのテンプレート

ガイドラインは以下のテンプレートに従って作成されている。独自でガイドラインを作成する場合も、このテンプレートの使用を推奨する。

ID: タイトル	XX_nnnn: ガイドラインのタイトル(短く、他と重複しない)
重要度	必須 / 強く推奨 / 推奨 のいずれか 1 つ
範囲	MAAB / NAMAAB / JMAAB / 会社名 (社内ルールを追加する場合)
MATLAB バージョン	ALL RX, RY, RZ RX 以降 RX 以前 RX から RY
前提条件	本ガイドラインの前提となるガイドラインへのリンク(ID+タイトル)
関連	参考となるガイドライン ID
記述内容	ガイドラインの説明(テキストおよびイメージ)
更新履歴	最終変更時のバージョン番号

注意: このテンプレートは、ガイドラインを正しく理解するため、最低限必要な項目を表にしたものである。既存の項目のいずれにも重複しない限り、新たな項目をこのテンプレートに追加することができる。

参考英語標記のテンプレート

ID: Title	XX_nnnn: Title of the guideline (unique, short)
Priority	One of mandatory / strongly recommended / recommended
Scope	MAAB, NA-MAAB, J-MAAB, Specific Company (for optional local company usage)
MATLAB® Version	all RX, RY, RZ RX and earlier



	RX and later RX through RY
Prerequisites	Links to guidelines, which are prerequisite to this guideline (ID+title)
Description	Description of the guideline (text, images)
See also	Links to guidelines
Last Change	Version number of last change

注意: このテンプレートは、ガイドラインを正しく理解するため、最低限必要な項目を表にしたものである。既存の項目のいずれにも重複しない限り、新たな項目をこのテンプレートに追加することができる。

### 2.3.1. ID

ID は、アルファベットの小文字 2 文字(ガイドライン著者を表す)と、4 桁の数字をアンダースコアでつないだものである。ID は、永久固定とし変更されることはなく、ガイドラインを参照する時に使用される。**na**、**jp**、**jc**、**eu** は MAAB ガイドラインで使用する。また、**db**、**jm**、**hd**、**ar** は既に使用されている。独自にガイドラインを追加する場合は、これら 2 文字で始まる ID を使用してはならない。

### 2.3.2. タイトル

タイトルは、そのガイドラインを短く説明したユニークなものである。タイトルは、ガイドラインを参照する時に使用される。

### 2.3.3. 重要度

重要度は、“必須”、“強く推奨”、“推奨”のいずれかに分類される。重要度は重要性を記述しているだけでなく、違反した場合に起こり得る結果の重大性も考慮している。

必須	強く推奨	推奨
<b>定義</b>		
すべての会社が認める必須のガイドライン  すべての会社が 100% 準拠するガイドライン	優れた方法であることは認めるが、レガシ モデルではガイドラインに 100% 準拠できないガイドライン  モデルはこれらのガイドラインに可能な限り準拠すべきであるが、100% 準拠する必要はない	モデル ダイアグラムの外観の改善には推奨されるが、モデルの実行には重要でないガイドライン  準拠が望ましいが、必須ではないガイドライン
<b>結果: ガイドラインに違反すると...</b>		
重要な項目が欠落する  モデルが正常に動作しない可能性がある	品質と外観が損なわれる  保守性、移植性、再利用性に悪影響を与える可能性がある	外観が他のプロジェクトに準拠しない
<b>適用免除ポリシー: ガイドラインに故意に違反した場合は...</b>		
理由を文書で提出しなければならない		

### 2.3.4. 範囲:

範囲は、以下のいずれかに設定される。

- MAAB: J-MAAB、NA-MAAB で合意されたガイドライン



- J-MAAB: Japan MBD Automotive Advisory Board でのみ合意されたガイドライン
- NA-MAAB: North America MATLAB Automotive Advisory Board でのみ合意されたガイドライン

MAAB には、J-MAAB と NA-MAAB というサブグループがある。

“J-MAAB”は、日本の自動車メーカーとサプライヤーのグループである。

“NA-MAAB”は北米とヨーロッパの自動車メーカーとサプライヤーのグループである。

### 2.3.5. MATLAB バージョン

全ての MATLAB バージョンをサポートしたガイドラインであるが、特定のバージョンのみサポートするガイドラインもある。この項目には以下の 5 つのフォーマットでバージョン情報が記述される。

- ALL: 全ての MATLAB バージョン
- RX、RY、RZ: 特定の MATLAB バージョン
- RX 以前: RX 以前の MATLAB バージョン
- RX 以降: RX 以降の MATLAB バージョン
- RX から RY: RX から RY の MATLAB バージョン

### 2.3.6. 前提条件

前提条件は、このガイドラインの前提条件となるガイドラインの ID とタイトルが記述される。ID とタイトルは、リンクが設定されている。

### 2.3.7. 記述内容

記述内容は、ガイドラインの詳細な内容が図表等を用いて記述されている。

### 2.3.8. 関連

参考となるガイドライン ID を記載する。

MAAB ガイドライン以外に下記のガイドラインを参考に行っている

- Modeling Guidelines for Code Generation(cgsl\_)
- Modeling Guidelines for High-Integrity Systems(hisl\_)
- Orion GN&C MATLAB/Simulink Standards(ek\_, hyl\_, im\_, jh\_)
 

Ver3 では、NASA オリオン・スタイル・ガイドラインの番号が関連として加わりました。  
<http://www.mathworks.com/aerospace-defense/standards/nasa.html>).

### 2.3.9. 更新履歴

更新履歴には、最終変更した際のバージョン番号が記述される。

## 2.4. 根拠

ガイドラインは、下記 5 つの何れかの根拠が対応します。

1. 可読性の向上
2. プロセス / ワークフローに効果的
3. 解析とシミュレーションが効率的
4. 検証が効率的
5. 組込みコード生成に効果的 / 効率的

注意: 本ガイドラインでは、根拠を個別に記載せず、「11. 根拠、関連まとめ」に記載した。

## 2.5. ガイドラインの使用

本ガイドラインの構成を次項に記述する。また、ガイドラインチェッカーの情報を付録 A に記述する。

### 2.5.1. ガイドラインの構成

本ガイドラインの前半は、命名規則とモデルアーキテクチャに関して記述され、すべてのモデルに適応できる基本ガイドラインである。後半は、Simulink、Stateflow 個別のガイドラインである。いくつかのガイドラインは、他のガイドラインを前提としており、それらはリンクにより呼び出される。

### 2.5.2. Masked Subsystems and Readability Rules

If users do not view the content of masked subsystems within a model, the guidelines for readability are not applicable.

## 3. Software Environment

### 3.1. General Guidelines

#### 3.1.1. na\_0026: Consistent software environment

ID: タイトル	na_0026: Consistent software environment
重要度	推奨
Scope	NA-MAAB
MATLAB Version	See Description
前提条件	
説明	<p>ソフトウェア開発が行われている間は、プロジェクト全体において一貫したソフトウェア環境を利用することをお勧めします。特に制限はしませんが、ここでいうソフトウェアとは以下を指します。</p> <ul style="list-style-type: none"><li>● MATLAB</li><li>● Simulink</li><li>● C Compiler (シミュレーション用)</li><li>● C Compiler (ターゲットハードウェア用)</li></ul> <p>一貫したソフトウェア環境とは、同じバージョンのソフトウェアが、プロジェクト全体に亘って利用されることを意味します。バージョン番号は、ソフトウェアに対して適用されるいかなるパッチまたは拡張子にも適用されます。</p>
See also	jh_0042: Required software
最終更新	V3.00

#### 3.1.2. na\_0027: Use of only standard library blocks

ID: タイトル	na_0027: Use of only standard library blocks
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>モデルを作成する際、Simulink ブロックのサブセットを指定したほうが良いでしょう。ブロックリストには、サードパーティ等によって作られたカスタムブロックライブラリを含むことができます。モデルは、これらのブロックのみを使って作成した方が良いでしょう。</p> <p>適合しないブロックでも、開発で利用することができます。もし適合しないブロックが利用されている場合、色分けやアイコン利用、注記などで印をつけておきましょう。尚、これらのブロックは、量産コード生成に利用される前に削除しておかなければいけません。</p>
See also	hyl_0201: Use of standard library blocks only
最終更新	V3.00

## 4.命名規則

### 4.1. 一般規則

#### 4.1.1. ar\_0001: Filenames

ID: タイトル	ar_0001: ファイル名に使用できる文字
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>ファイル名は以下の制約に従います。</p> <p><b>有効な形式</b> filename = name.extension</p> <ul style="list-style-type: none"><li>name: 先頭に数字は不可、空白文字は不可</li><li>extension(拡張子): 空白文字は不可</li></ul> <p><b>一意性</b></p> <ul style="list-style-type: none"><li>親プロジェクトディレクトリ内のすべてのファイル名は重複してはならない</li><li>C/C++またはMATLABキーワードと同じ文字を使ってはならない。</li></ul> <p><b>使用可能な文字</b></p> <p>名前: abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789_</p> <p>拡張子: abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789</p> <p><b>アンダースコア</b></p> <p>名前:</p> <ul style="list-style-type: none"><li>部分の区切りにアンダースコアは使用可能</li><li>アンダースコアの連続使用は不可</li><li>先頭にはアンダースコアは使用不可</li><li>末尾にはアンダースコアは使用不可</li></ul> <p>拡張子: アンダースコアを使用不可</p>
最終更新	V3.00

#### 4.1.2. ar\_0002: Directory names

ID: タイトル	ar_0002: ディレクトリ名に使用できる文字
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	<p>ディレクトリ名は以下の制約に従います。</p> <p><b>有効な形式</b> directory name = name name: 先頭に数字は不可、空白文字は不可</p> <p><b>一意性</b> 親プロジェクトディレクトリ内のすべてのディレクトリ名</p>

	<p>使用可能な文字</p> <p>name:</p> <p>abcdefghijklmnopqrstuvwxyz</p> <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ</p> <p>0123456789_</p> <p><b>アンダースコア</b></p> <p>name:</p> <ul style="list-style-type: none"> <li>• 部分の区切りにアンダースコアは使用可能</li> <li>• アンダースコアの連続使用は不可</li> <li>• 先頭にはアンダースコアは使用不可</li> <li>• 末尾にはアンダースコアは使用不可</li> </ul>
説明	
最終更新	V1.00

#### 4.1.3. na\_0035: Adoption of naming conventions

ID: タイトル	na_0035: Adoption of naming conventions
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>命名規約の適用を推奨します。命名規約は、ブロック、シグナル、パラメータやデータタイプのガイダンスになります。命名規約は、よく次のような問題をカバーします。</p> <p>プログラミング言語と関連ツールの整合性:</p> <ul style="list-style-type: none"> <li>• 長さ</li> <li>• シンボルの利用</li> </ul> <p>可読性:</p> <ul style="list-style-type: none"> <li>• アンダーラインの利用</li> <li>• 大文字の利用</li> </ul> <p>エンコーディング情報:</p> <ul style="list-style-type: none"> <li>• 意味のある名前の利用</li> <li>• スタンダードな略語と頭字語</li> <li>• データタイプ</li> <li>• 単位系</li> <li>• データのオーナーシップ</li> <li>• メモリータイプ</li> </ul>
最終更新	V3.00

## 4.2. モデル内の規則

#### 4.2.1. jc\_0201: Usable characters for Subsystem names

ID: タイトル	jc_0201: Subsystem の名前に使用できる文字
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>すべてのSubsystem ブロックの名前は以下の制約に従います。</p> <p><b>有効な形式</b></p>

	name: <ul style="list-style-type: none"> <li>先頭に数字は不可</li> <li>空白は不可</li> <li>改行を含むことはできません</li> </ul> <b>使用可能な文字</b> name: abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789_ <b>アンダースコア</b> name: <ul style="list-style-type: none"> <li>部分の区切りにアンダースコアは使用可能</li> <li>アンダースコアの連続使用は不可</li> <li>先頭にはアンダースコアは使用不可</li> <li>末尾にはアンダースコアは使用不可</li> </ul>
最終更新	V2.20

#### 4.2.2. jc\_0211: Usable characters for Inport blocks and Outport blocks

<b>ID: タイトル</b>	<b>jc_0211: Inport ブロック / Outport ブロックに使用できる文字</b>
<b>重要度</b>	強く推奨
<b>Scope</b>	MAAB
<b>MATLAB Version</b>	All
<b>前提条件</b>	
<b>説明</b>	Inport ブロックとOutput ブロックの名前は以下の制約に従います。 <b>有効な形式</b> name: <ul style="list-style-type: none"> <li>先頭に数字は不可</li> <li>空白は不可</li> <li>改行を含むことはできません</li> </ul> <b>使用可能な文字</b> name: abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789_ <b>アンダースコア</b> name: <ul style="list-style-type: none"> <li>部分の区切りにアンダースコアは使用可能</li> <li>アンダースコアの連続使用は不可</li> <li>先頭にはアンダースコアは使用不可</li> <li>末尾にはアンダースコアは使用不可</li> </ul>
最終更新	V2.20

#### 4.2.3. jc\_0221: Usable characters for signal line name

<b>ID: タイトル</b>	<b>jc_0221: 信号線の名前に使用できる文字</b>
<b>重要度</b>	強く推奨
<b>Scope</b>	MAAB
<b>MATLAB Version</b>	All
<b>前提条件</b>	
<b>説明</b>	名前付き信号の制約を示します。

	<p>有効な形式</p> <p>name:</p> <ul style="list-style-type: none"> <li>先頭に数字は不可</li> <li>空白は不可</li> <li>制御文字は不可</li> <li>改行を含むことはできません</li> </ul> <p>使用可能な文字</p> <p>name:</p> <p>abcdefghijklmnopqrstuvwxyz  ABCDEFGHIJKLMNOPQRSTUVWXYZ  0123456789_</p> <p>アンダースコア</p> <p>name:</p> <ul style="list-style-type: none"> <li>部分の区切りにアンダースコアは使用可能</li> <li>アンダースコアの連続使用は不可</li> <li>先頭にはアンダースコアは使用不可</li> <li>末尾にはアンダースコアは使用不可</li> </ul>
最終更新	V2.20

#### 4.2.4. na\_0030: Usable characters for Simulink Bus names

ID: タイトル	na_0030: Simulink Bus に使える名前
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>すべての Simulink バス名は、以下の制約に従わなければなりません:</p> <p>フォーム名:</p> <ul style="list-style-type: none"> <li>数字から始めてはならない</li> <li>ブランクスペースがあってはならない</li> <li>キャリッジリターンは禁止</li> </ul> <p>利用可能な文字:</p> <p>abcdefghijklmnopqrstuvwxyz  ABCDEFGHIJKLMNOPQRSTUVWXYZ  0123456789_</p> <p>アンダーライン名:</p> <ul style="list-style-type: none"> <li>名前のパートを分けるために利用可能</li> <li>連続しての利用はできない</li> <li>アンダーラインでの開始はできない</li> <li>アンダーラインで終わることはできない</li> </ul>
See Also	jh_0040: Usable characters for Simulink Bus Names
最終更新	V3.00

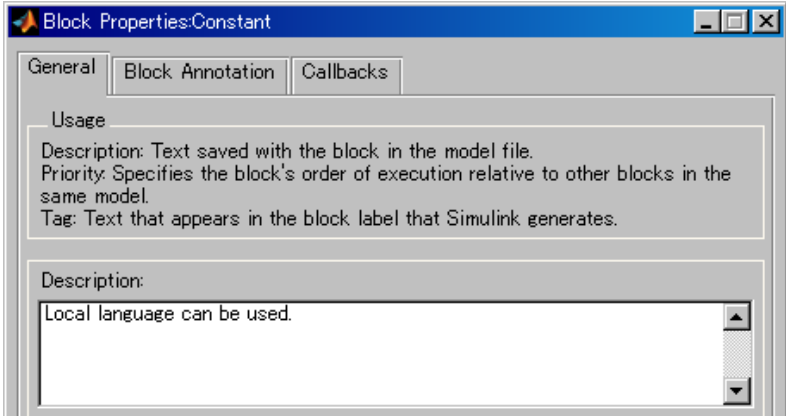
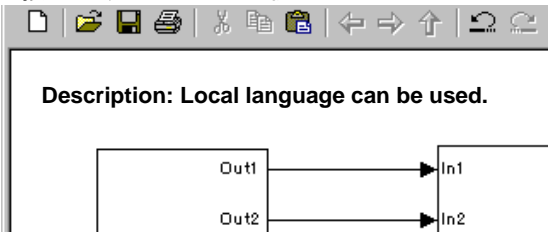
#### 4.2.5. jc\_0231: Usable characters for block names

ID: タイトル	jc_0231: ブロックの名前に使用できる文字
重要度	強く推奨
Scope	MAAB



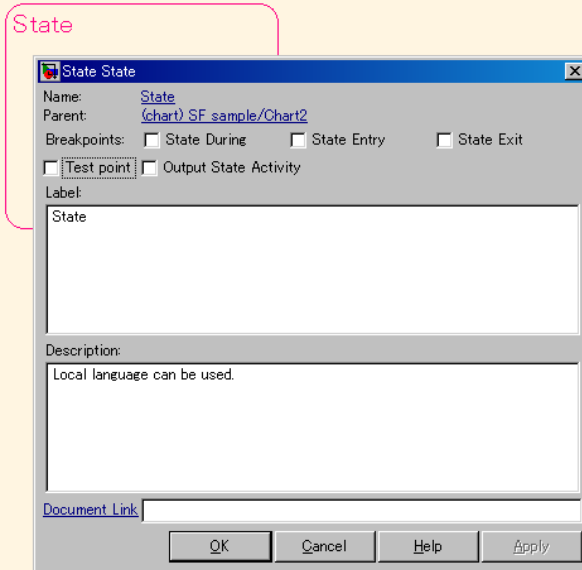
MATLAB Version	All
前提条件	<a href="#">jc_0201: Usable characters for Subsystem names</a>
説明	<p>すべてのブロックの名前は以下の制約に従います。</p> <p>有効な形式 name:</p> <ul style="list-style-type: none"> <li>先頭に数字は不可</li> <li>ブロック名の先頭に空白は不可</li> <li>全角文字は不可</li> <li>改行は可</li> </ul> <p>使用可能な文字 name: a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 _</p> <p>メモ: このルールはSubsystem ブロックには適用されません。</p>
最終更新	V2.00

#### 4.2.6. na\_0014: Use of local language in Simulink and Stateflow

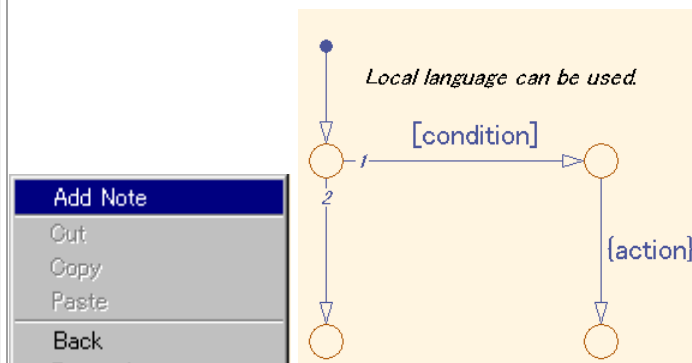
ID: タイトル	na_0014: Simulink / Stateflow でローカル言語を使用できる箇所
重要度	強く推奨
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>ローカル言語は説明フィールドでのみ使用します。説明フィールドは、コード生成やシミュレーションに影響しないテキスト入力部分です。説明フィールドの例には、[ブロックプロパティ] ダイアログボックスの[説明] フィールドがあります。</p> <p>Simulink例</p> <ul style="list-style-type: none"> <li>[ブロックプロパティ] ダイアログボックスの[説明] フィールド</li> </ul>  <ul style="list-style-type: none"> <li>モデルに直接入力するテキスト注釈</li> </ul> 

## Stateflow例

- チャートや状態プロパティの[説明] フィールド



- [メモの追加] を使用して追加した注釈の説明



メモ: Simulink では、ローカル言語が含まれたモデルを、文字エンコード方式が異なるシステム上で開けない可能性があります。したがって、ローカル文字を使用して国外とモデルを交換する場合は注意が必要です。

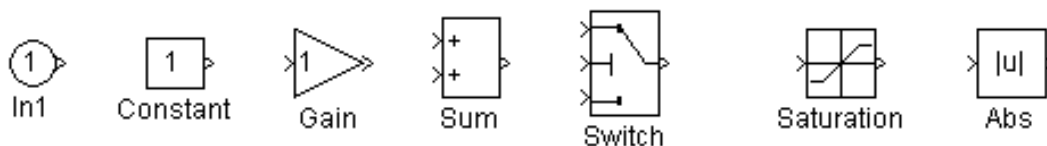
最終更新

V2.00

## 5.モデルアーキテクチャ

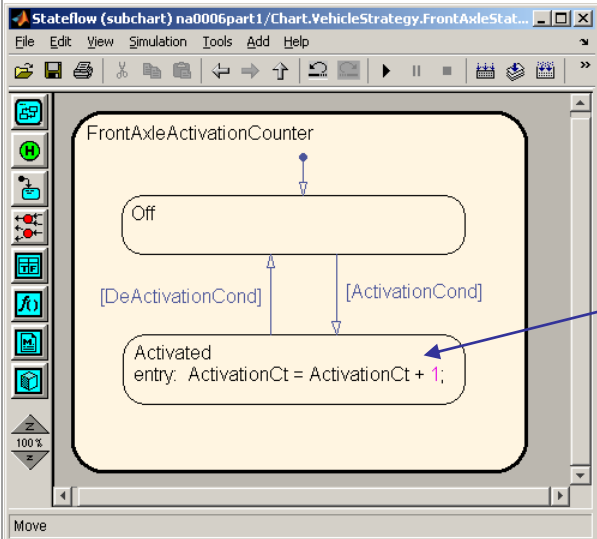
### 基本ブロック

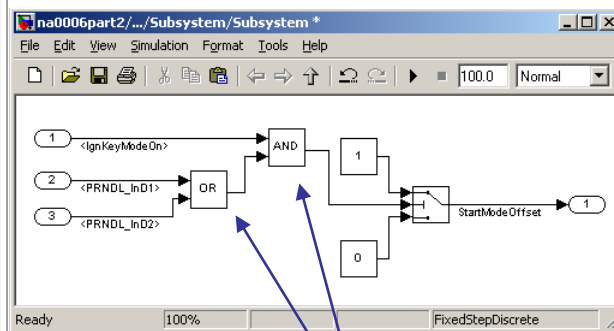
本ガイドラインでは、標準 Simulink ライブラリのビルトインブロックを“基本ブロック”としている。  
基本ブロックの例を以下に示す。



## 5.1. Simulink と Stateflow の使い分け

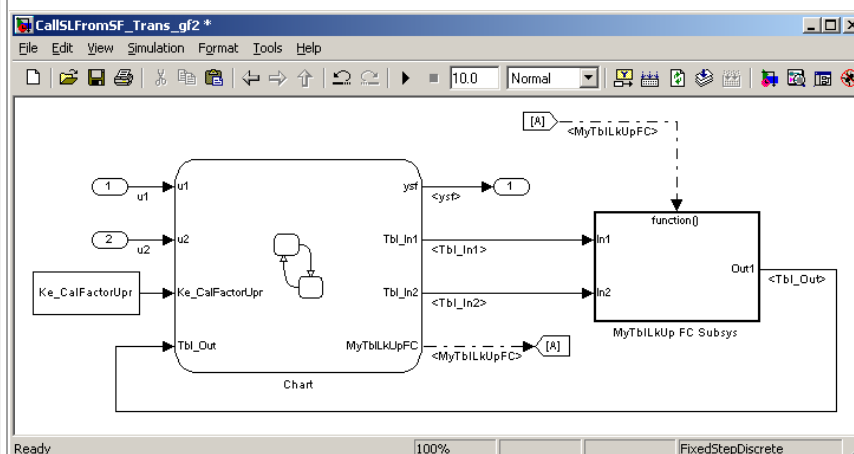
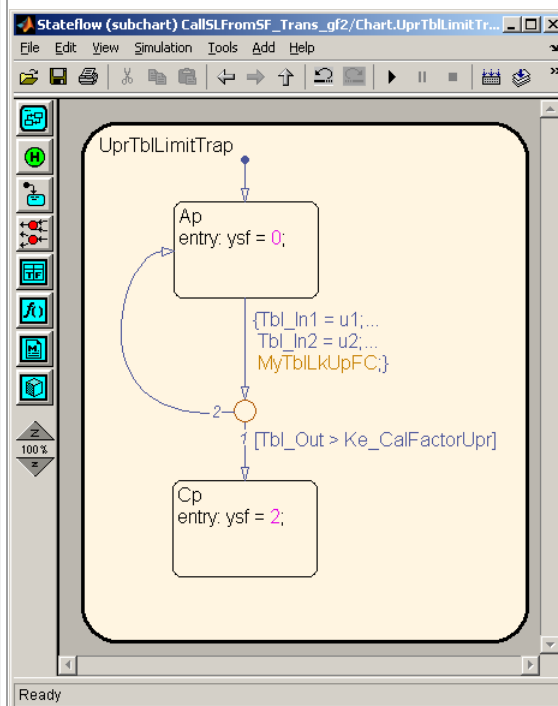
### 5.1.1. na\_0006: Guidelines for mixed use of Simulink and Stateflow

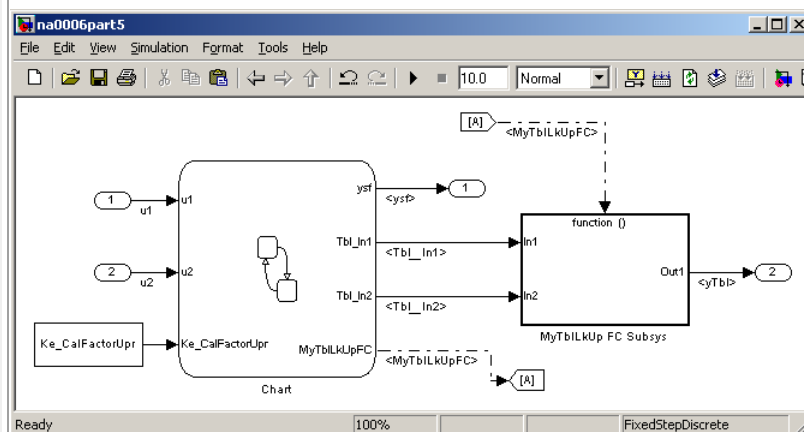
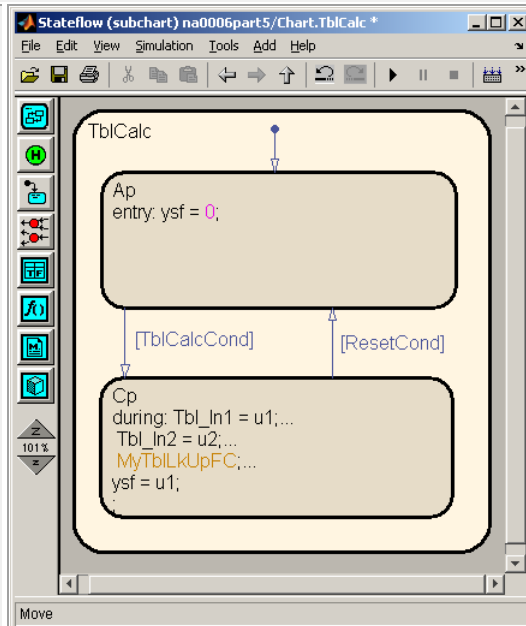
ID: タイトル	na_0006: Simulink と Stateflow を組み合わせ方法
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>制御アルゴリズムの特定部分のモデリングにSimulink を使用するかStateflow を使用するかは、モデル化する動作の特性によって決まります。</p> <ul style="list-style-type: none"> <li>関数が主として複雑な論理演算を必要とする場合は、Stateflow ダイアグラムを使用します。</li> </ul> <p>Stateflow ダイアグラムを使用して、モーダル論理を実装します。ここで、現時点で実行される制御関数は“過去と現在の論理状態”の組み合わせによって決まります。</p> <ul style="list-style-type: none"> <li>関数が主として数値演算を含む場合は、Simulink 機能を使用します。</li> </ul> <p><b>特記</b></p> <ul style="list-style-type: none"> <li>関数の主要特性が論理であっても、単純な数値計算を実行してロジックをサポートする場合は、Stateflow アクション言語を使用して単純な数値関数を実装します。</li> </ul>  <p>Embedded simple math operation</p> <ul style="list-style-type: none"> <li>関数の主要特性が数値であっても、単純な論理演算を実行して計算をサポートする場合は、Simulink ブロックを使用して単純な論理関数を実行します。</li> </ul>



Embedded simple  
logic operations

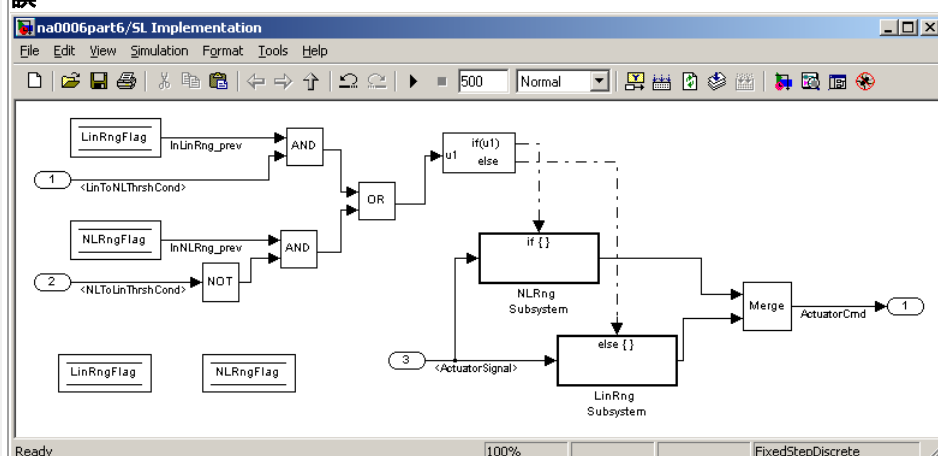
- 関数の主要特性が論理で、ロジックのサポートに複雑な数値計算をしなければならない場合は、Simulink サブシステムを使用して数値計算を実装します。Stateflow ソフトウェアは関数呼び出しを使用してサブシステムの実行を呼び出します。

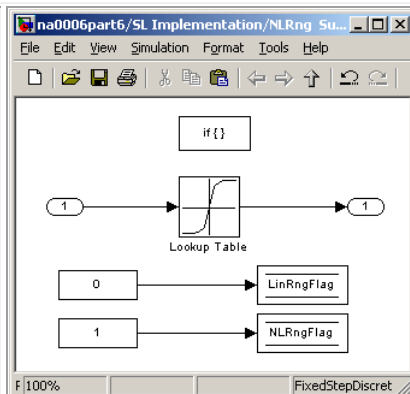




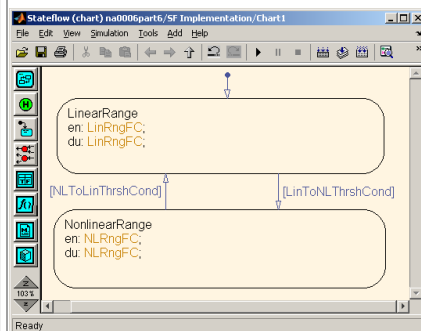
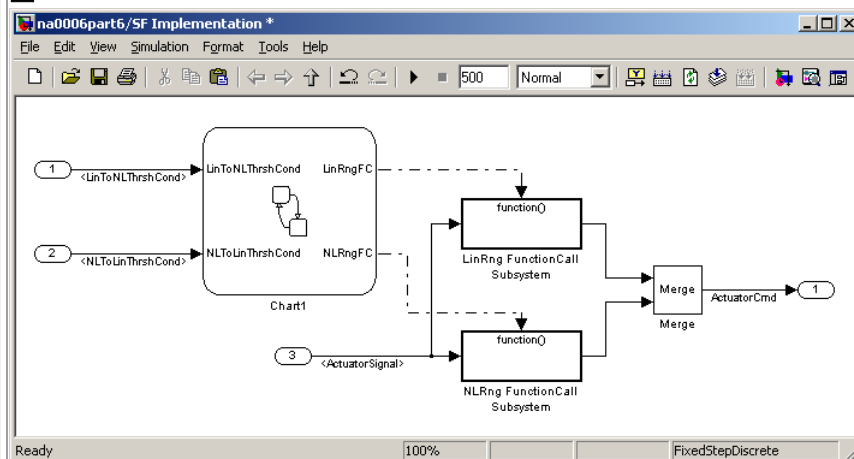
- Stateflow 製品を使用して、モーダル論理を実装します。ここで、現時点で実行される制御関数は“過去と現在の論理状態”の組み合わせによって決まります（たとえば**フラグ**の保存のように、論理条件テストの結果をSimulink モデルに保存する必要がある場合は、これがモーダル論理の存在を示し、Stateflow ソフトウェアでモデル化する必要があります）。

## 誤



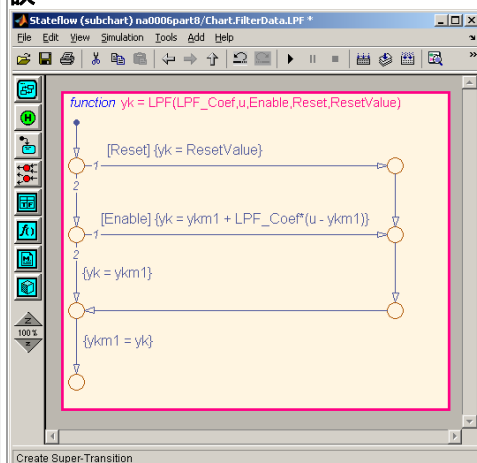


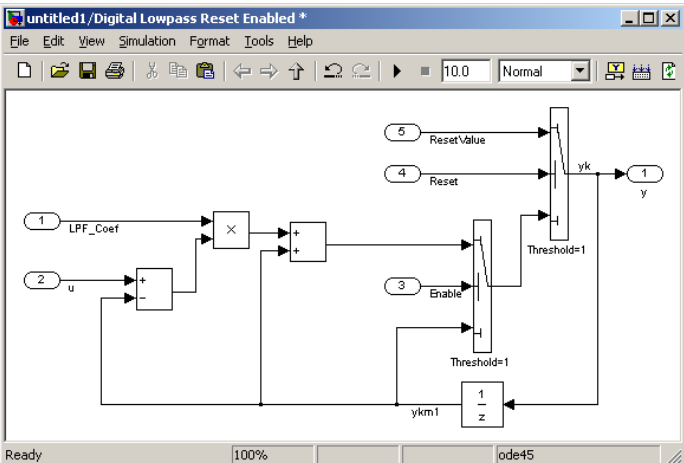
正



- Simulink を使用して、次のような連続値状態を含む数値表現を実現します。  
差分方程式、積分、微分およびフィルター。

誤



	<p>正</p> 
最終更新	V2.00

### 5.1.2. na\_0007: Guidelines for use of Flow Charts, Truth Tables and State Machines

ID: タイトル	na_0007: フローチャート / 真理値表 / ステートマシンの使用方法
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">na_0006: Guidelines for Mixed use of Simulink and Stateflow</a>
説明	<p>制御アルゴリズムの特定部分のモデリングに Stateflow 内で、フローチャートを使用するかステートチャートを使用するかは、モデル化する動作の特性によって決まります。</p> <ul style="list-style-type: none"> <li>関数セグメントの主要特性が演算モードや離散値状態の計算の場合は、ステートチャートを使用します。その例は次のとおりです。 <ul style="list-style-type: none"> <li>合格・不合格、中止・競合の状態の診断モデル</li> <li>制御アルゴリズムの異なる演算モードを計算するモデル</li> </ul> </li> <li>関数セグメントの主要特性が if-then-else ステートメントを含む場合は、フローチャートまたは真理値表を使用します。</li> </ul> <p>特記</p> <p>関数セグメントの主要特性がモードまたは状態の計算である場合でも、if-then-else ステートメントが必要な場合は、ステートチャート内の状態にフローチャートを追加します (「フローチャート パターン」を参照)。</p>
最終更新	V2.00

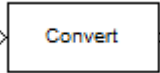


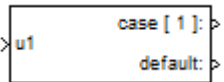




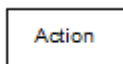
## 5.2. サブシステムの階層構造

### 5.2.1. db\_0143: Similar block types on the model levels

ID: タイトル	db_0143: 各モデル階層で使えるブロックタイプ
重要度	強く推奨
Scope	NA-MAAB



MATLAB Version	All																														
前提条件																															
説明	<p>モデルを不連続単位に分割できるようにするには、モデルのすべてのレベルが同じタイプのビルディング ブロック (つまりサブシステムのみ、または「基本ブロック」のみ) を使用して設計されなければなりません。このガイドラインでリストされているブロックは、信号の経路指定に使用されます。これらはモデルの任意のレベルに配置できます。</p> <p>どのモデル レベルでも使用できるブロック</p> <table> <tr> <th>ブロック</th><th>例</th></tr> <tr> <td>Inport</td><td></td></tr> <tr> <td>Outport</td><td></td></tr> <tr> <td>Mux</td><td></td></tr> <tr> <td>Demux</td><td></td></tr> <tr> <td>Bus Selector</td><td></td></tr> <tr> <td>Bus Creator</td><td></td></tr> <tr> <td>Selector</td><td></td></tr> <tr> <td>Ground</td><td></td></tr> <tr> <td>Terminator</td><td></td></tr> <tr> <td>From</td><td></td></tr> <tr> <td>Goto</td><td></td></tr> <tr> <td>Merge</td><td></td></tr> <tr> <td>Unit Delay</td><td></td></tr> <tr> <td>Rate Transition</td><td></td></tr> </table>	ブロック	例	Inport		Outport		Mux		Demux		Bus Selector		Bus Creator		Selector		Ground		Terminator		From		Goto		Merge		Unit Delay		Rate Transition	
ブロック	例																														
Inport																															
Outport																															
Mux																															
Demux																															
Bus Selector																															
Bus Creator																															
Selector																															
Ground																															
Terminator																															
From																															
Goto																															
Merge																															
Unit Delay																															
Rate Transition																															

	Data Type Conversion	
	Data Store Memory	
	If	
	Case	
	Function-Call Generator	
	Function-Call Split	
	Trigger <sup>(1)</sup>	
	Enable <sup>(2)</sup>	
	Action port <sup>(3)</sup>	
Note	<p>1)Action 端子はモデルのルート レベルでは利用できません。</p> <p>2)R2011b からは、Enable ブロックをモデルのルート レベルで利用できます。</p> <p>3)R2009a からは、Trigger ブロックをモデルのルート レベルで利用できます。</p> <p>メモ: Trigger ブロックまたは Enable ブロックがモデルのルート レベルに配置された場合は、モデルはスタンドアロン モードでシミュレーションしません。モデルは Model ブロックを使用して参照されなければなりません。</p>	
最終更新	V2.20	

### 5.2.2. db\_0144: Usage of Subsystems

ID: タイトル	db_0144: サブシステムの使用方法
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>Simulink ブロック線図に示されているアルゴリズムの機能的分解またはその一部分に基づいて、ブロック線図のブロックをサブシステムにグループ化しなければなりません。</p> <p>ブロック線図内のスペースを節約することを主目的としてブロックをサブシステムにグループ化することは、避けなければなりません。ブロック線図の各サブシステムは、モデルまたはサブモデルの目的を果たすために必要な機能の単位を表します。ブロックは、動作のバリエーションまたはタイミングに基づいてグループ化することもできます。</p>

	読みやすくするためにサブシステムを作成する必要がある場合は、バーチャル サブシステムを使用しなければなりません。
最終更新	V2.20

### 5.2.3. db\_0040: Model hierarchy

ID: タイトル	db_0040: モデルの階層構造
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	モデルの階層構造は、制御システムの機能構造に対応していなければなりません。
最終更新	V2.00

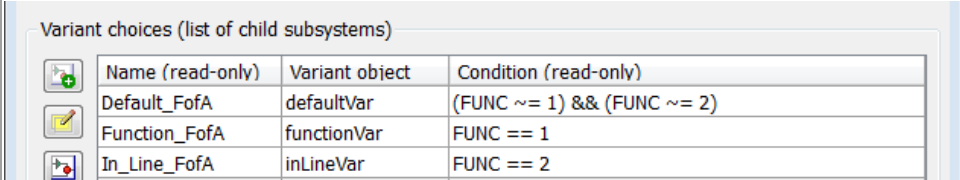
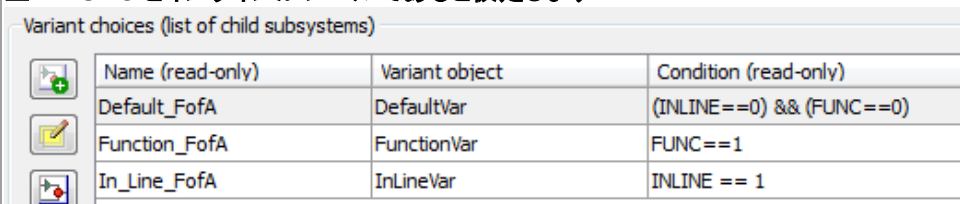
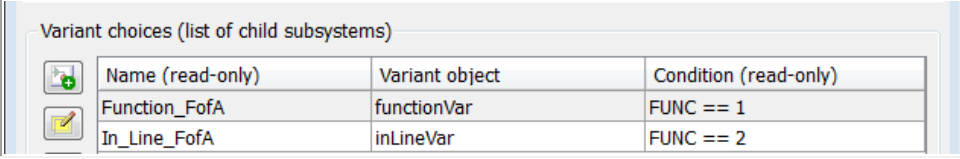
### 5.2.4. na\_0037: Use of single variable variant conditionals

ID: タイトル	na_0037: Use of single variable variant conditionals																
重要度	推奨																
Scope	NA-MAAB																
MATLAB Version	All																
前提条件																	
説明	<p>バリエーション条件の記述は、単一変数かつ複合条件か、複数変数かつ単一条件の組み合わせで構成されていなければなりません。 デフォルトバリエーションは、2 つ目のルールでは例外となります。</p> <p>正しい例： 行ごとに一つの条件 複数変数 (INLINE/FUNCTION)</p> <div><p>Variant choices (list of child subsystems)</p><table><tr><td></td><td>Name (read-only)</td><td>Variant object</td><td>Condition (read-only)</td></tr><tr><td></td><td>Default_FofA</td><td>DefaultVar</td><td>(INLINE==0) &amp;&amp; (FUNC==0)</td></tr><tr><td></td><td>Function_FofA</td><td>FunctionVar</td><td>FUNC == 1</td></tr><tr><td></td><td>In_Line_FofA</td><td>InLineVar</td><td>INLINE == 1</td></tr></table></div>		Name (read-only)	Variant object	Condition (read-only)		Default_FofA	DefaultVar	(INLINE==0) && (FUNC==0)		Function_FofA	FunctionVar	FUNC == 1		In_Line_FofA	InLineVar	INLINE == 1
		Name (read-only)	Variant object	Condition (read-only)													
		Default_FofA	DefaultVar	(INLINE==0) && (FUNC==0)													
		Function_FofA	FunctionVar	FUNC == 1													
	In_Line_FofA	InLineVar	INLINE == 1														
	<p>正しい例： 単一変数で複合条件</p> <div><p>Variant choices (list of child subsystems)</p><table><tr><td></td><td>Name (read-only)</td><td>Variant object</td><td>Condition (read-only)</td></tr><tr><td></td><td>AutoTrans</td><td>autoTrans</td><td>(transType == 3)    (transType == 4)    (transType == 5)</td></tr><tr><td></td><td>Default_4speed</td><td>defaultTrans</td><td>(transType ~= 3) &amp;&amp; (transType ~= 4) &amp;&amp; (transType ~= 5) &amp;&amp; (transType ~= 0)</td></tr><tr><td></td><td>ManualTrans</td><td>manualTrans</td><td>(transType == 0)</td></tr></table></div>		Name (read-only)	Variant object	Condition (read-only)		AutoTrans	autoTrans	(transType == 3)    (transType == 4)    (transType == 5)		Default_4speed	defaultTrans	(transType ~= 3) && (transType ~= 4) && (transType ~= 5) && (transType ~= 0)		ManualTrans	manualTrans	(transType == 0)
	Name (read-only)	Variant object	Condition (read-only)														
	AutoTrans	autoTrans	(transType == 3)    (transType == 4)    (transType == 5)														
	Default_4speed	defaultTrans	(transType ~= 3) && (transType ~= 4) && (transType ~= 5) && (transType ~= 0)														
	ManualTrans	manualTrans	(transType == 0)														
	<p>正しくない例： 複数変数で複合条件</p> <div><p>Variant choices (list of child subsystems)</p><table><tr><td></td><td>Name (read-only)</td><td>Variant object</td><td>Condition (read-only)</td></tr><tr><td></td><td>AutoTrans</td><td>incorrect_1</td><td>(INLINE==0) &amp;&amp; (transType == 3)</td></tr><tr><td></td><td>Default_4speed</td><td>incorrectDefault</td><td>((((INLINE==0) &amp;&amp; (transType == 3))==0) &amp;&amp; (FUNC == 0) &amp;&amp; (transType ~= 2)</td></tr><tr><td></td><td>ManualTrans</td><td>incorrect_2</td><td>(FUNC == 1)    (transType == 2)</td></tr></table></div>		Name (read-only)	Variant object	Condition (read-only)		AutoTrans	incorrect_1	(INLINE==0) && (transType == 3)		Default_4speed	incorrectDefault	((((INLINE==0) && (transType == 3))==0) && (FUNC == 0) && (transType ~= 2)		ManualTrans	incorrect_2	(FUNC == 1)    (transType == 2)
	Name (read-only)	Variant object	Condition (read-only)														
	AutoTrans	incorrect_1	(INLINE==0) && (transType == 3)														
	Default_4speed	incorrectDefault	((((INLINE==0) && (transType == 3))==0) && (FUNC == 0) && (transType ~= 2)														
	ManualTrans	incorrect_2	(FUNC == 1)    (transType == 2)														
Note	条件の記述では列挙型変数の使用が推奨されます。本例では可読性を改善するために、数値が使われました。.																
See also	na_0036 Default variant																
最終更新	V3.00																

### 5.2.5. na\_0020: Number of inputs to variant subsystems

ID: タイトル	na_0020: Number of inputs to variant subsystems
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	バリエーションサブシステムでは、各サブシステムの入力数が同じである必要があります。しかし、サブシステムが入力の全てを使うというわけではないかもしれません。この場合、Terminator ブロックで使っていない入力を終了してください。
最終更新	V3.00

### 5.2.6. na\_0036: Default variant

ID: タイトル	na_0036 Default variant
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	<a href="#">na_0037 Use of single variable variant conditionals</a>
説明	<p>全てのバリエーションサブシステムでは、1つのサブシステムが常に選ばれるようにモデルを作成しなければなりません。これは、以下のうち1つを実行することによって達成可能です。</p> <ul style="list-style-type: none"> <li>・デフォルトバリエーションを使うこと。</li> <li>・徹底的にバリエーションのすべての可能な値をカバーする条件を定めること。例えば、ブーリアンの True/False 値を定めることなど。</li> </ul> <p><b>正：</b></p>  <p><b>正： FUNC とインラインがブールであると仮定します</b></p>  <p><b>正しくないケース： FUNC が1, 2と等しくない場合、アクティブなサブシステムはありません</b></p> 
最終更新	V3.00

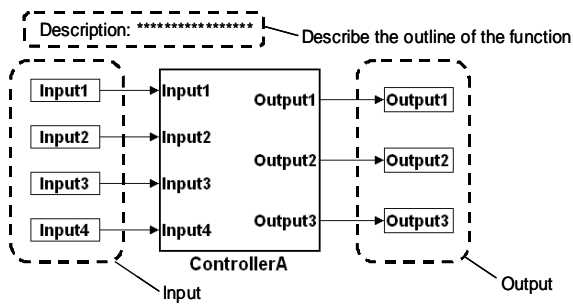
## 5.3. J-MAAB Model Architecture Decomposition

### 5.3.1. jc\_0301: Controller model

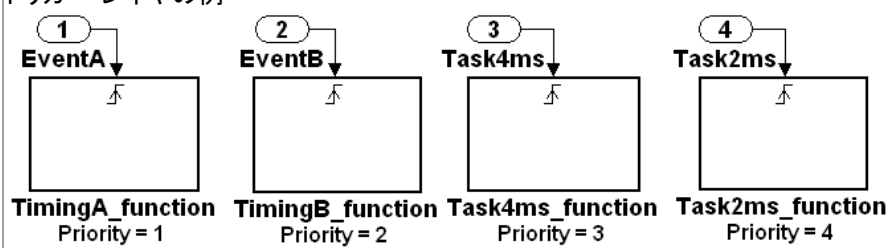
ID: タイトル	jc_0301: コントローラ モデル
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>コントロール モデルは、以下の階層構造を使用して構成されます。各レイヤの詳細は、対応するルールで提供されます。</p> <p>トップレイヤ (ルートレベル) 「jc_0311: Top layer/root level」</p> <p>トリガー レイヤ「jc_0321: Trigger layer」</p> <p>構造体レイヤ jc_0331: Structure layer</p> <p>データフロー レイヤ「jc_0341: Data flow layer」</p> <p>トリガー レベルの使用はオプションです。次の図で、タイプ A はトリガー レベルの使用を示しており、タイプ B はトリガー レベルなしのモデルを示しています。</p> <p>コントローラ モデル</p>
最終更新	V2.00

### 5.3.2. jc\_0311: Top layer / root level

ID: タイトル	jc_0311: トップレイヤ/ルートレベル
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	

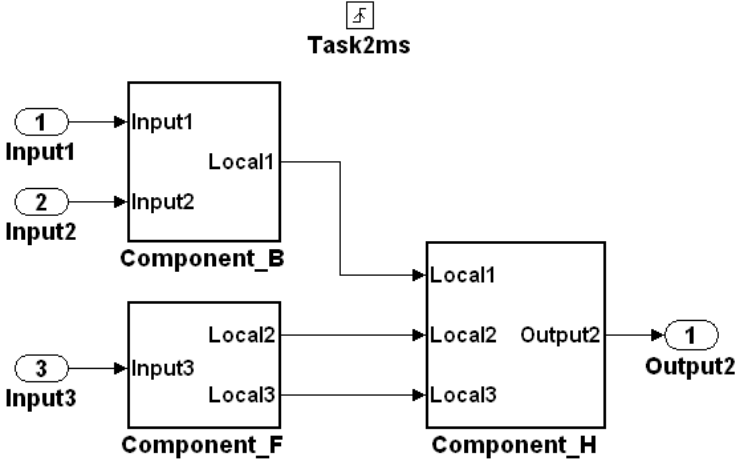
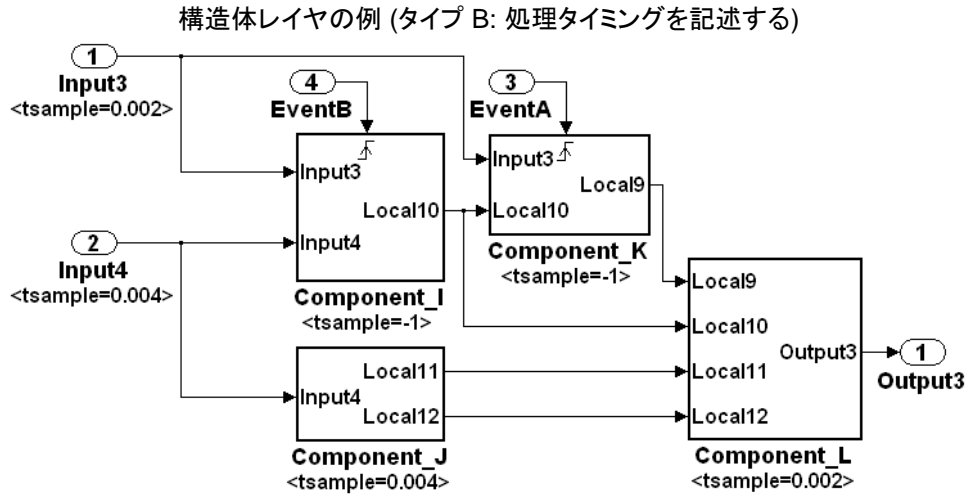
説明	<p>トップ レイヤに記述する項目は以下のとおりです。</p> <p>概要: モデルの機能概要の説明</p> <p>入力: 入力変数</p> <p>出力: 出力変数</p> <p>トップ レイヤの例</p> 
	最終更新 V2.00

### 5.3.3. jc\_0321: Trigger layer

ID: タイトル	jc_0321: トリガー レイヤ
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>トリガー レイヤでは、Triggered Subsystem ブロックまたは Function-Call Subsystem ブロックを使用して処理タイミングを指定します。</p> <ul style="list-style-type: none"> <li>● ブロックは、必要に応じて重要度を設定します。</li> <li>● 重要度の値は、ブロック注釈として表示しなければなりません。ブロックを開かなくても重要度ベースの順序を理解できるようにしてください。</li> </ul> <p>トリガー レイヤの例</p> 
最終更新	V2.00

### 5.3.4. jc\_0331: Structure layer

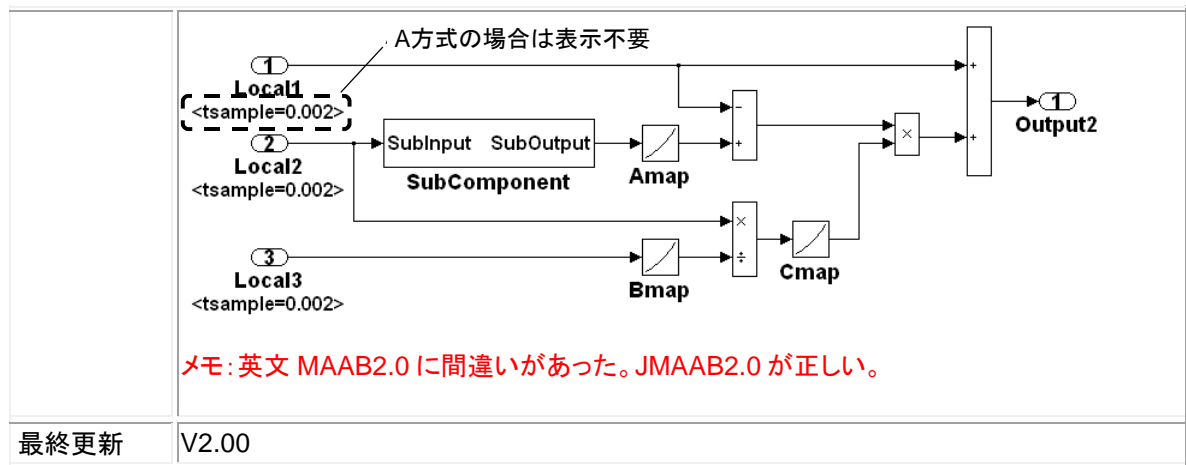
ID: タイトル	jc_0331: 構造レイヤ
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	

	<ul style="list-style-type: none"> <li>● 構造体レイヤには以下を記述します。 <ul style="list-style-type: none"> <li>－ タイプ B の場合は、Inport ブロックまたは Subsystem ブロックでサンプル時間を指定し、サブシステムの作業時間を定義します。</li> <li>－ タイプ B の場合は、ブロック注釈を使用し、サンプル時間を表示し、サブシステムの作業時間を明確にします。</li> </ul> </li> <li>● 構造体レイヤのサブシステムは Atomic Subsystem です。</li> </ul> <p>構造体レイヤの例 (タイプ A: 処理タイミングを記述しない)</p>  <p>構造体レイヤの例 (タイプ B: 処理タイミングを記述する)</p> 
最終更新	V2.00

### 5.3.5. jc\_0341: Data flow layer

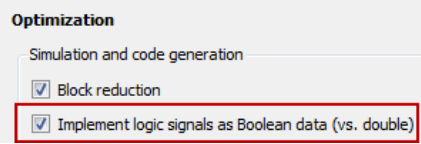
ID: タイトル	jc_0341: データフローレイヤ
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>以下の例に従ってデータフローレイヤを記述します。</p> <p><b>タイプ B の場合は</b>、Inport ブロックでブロック注釈を使用し、そのサンプル時間を表示して、信号の実行タイミングを明確にします。</p> <p>データフローレイヤの例</p>





## 6.モデル コンフィギュレーション オプション

### 6.1.1. jc\_0011: Optimization parameters for Boolean data types

ID: タイトル	jc_0011: 論理信号に対する最適化パラメータ設定
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">na_0002: Appropriate implementation of fundamental logical and numerical operations</a>
説明	<p>Boolean データ型の最適化オプションを有効 (オン) にしなければなりません。コンフィギュレーション パラメータ ダイアログ ボックスで、[最適化] ペインの [シミュレーションとコード生成] の下にある [boolean データ (対 double) として論理信号を処理] を選択します。</p> 
最終更新	V2.20

### 6.1.2. jc\_0021: Model diagnostic settings

ID: タイトル	jc_0021: 診断パラメータ設定
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>以下の診断を有効にしなければなりません。有効にした診断は warning または error に設定されます。診断オプションを none に設定することはできません。リストにない診断は、任意の値 (none、warning、または error) に設定できます。</p> <p><b>ソルバーの診断</b></p> <ul style="list-style-type: none"><li>● 代数ループ</li><li>● 代数ループの最小化</li></ul> <p><b>サンプル時間の診断</b></p> <ul style="list-style-type: none"><li>● マルチタスク レート変換</li></ul> <p><b>データ有効性の診断</b></p> <ul style="list-style-type: none"><li>● [Inf または NaN のブロック出力]</li><li>● データ ストア名の重複</li></ul> <p><b>接続性</b></p> <ul style="list-style-type: none"><li>● 接続されていないブロックの入力端子</li><li>● 接続されていないブロックの出力端子</li><li>● 接続されていないライン</li><li>● ルートの Outport ブロックの未定義のバス オブジェクト</li><li>● Mux ブロックをバス信号の作成に使用</li><li>● 無効な関数呼び出し接続</li><li>● 要素名の不一致</li></ul>
最終更新	V2.00

## 7.Simulink

### 7.1. Diagram Appearance

#### 7.1.1. na\_0004: Simulink model appearance

ID: タイトル	na_0004 :Simulink モデルの表示設定	
重要度	推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	<p>モデルの外観の設定は、モデルのリリース時には以下のガイドラインに従います。 この設定は開発プロセス中に変更できます。</p>	
	表示オプション	設定
	モデル ブラウザー	無効
	画面の色	白
	ステータス バー	有効
	ツール バー	有効
	ズーム倍率	標準 (100%)
	ブロック表示オプション	設定
	背景色	白
	前景色	黒
	実行コンテキスト インジケータ	無効
	ライブラリ リンクの表示	なし
	線形化インジケータ	有効
	Model/ブロック I/O の不一致	無効
	Model ブロックのバージョン	無効
	サンプル時間の色分け	無効
	並べ替え順序	無効
	信号の表示オプション	設定
	端子のデータ型	無効
	信号の次元	無効
	ストレージ クラス	無効
	テスト ポイント インジケータ	有効
	ビューアー インジケータ	有効
	非スカラー ラインを太く表示	有効
最終更新	V2.00	

### 7.1.2. db\_0043: Simulink font and font size

ID: タイトル	db_0043: モデルで使用するフォントとフォントサイズ
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>モデル内のフリー テキスト注釈以外のテキスト要素 (ブロック名、ブロック注釈、信号ラベルなど) はすべて、同じフォント スタイルとフォント サイズにしなければなりません。フォント スタイルとフォント サイズは読みやすさに配慮して選択してください。</p> <p>メモ: 選択したフォントは、移植可能 (たとえば Simulink または Stateflow の既定のフォント) であるか、プラットフォーム間で変換可能 (Arial、Helvetica 12pt など) でなければなりません。</p>
最終更新	V2.00


### 7.1.3. db\_0042: Port block in Simulink models

ID: タイトル	db_0042: Inport ブロック / Outport ブロックの使用方法
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>Simulink モデルでは、端子は以下のルールに従わなければなりません。</p> <ul style="list-style-type: none"> <li>● Inport ブロックをダイアグラムの左側に配置します。ブロックは、信号の交差を避けるために移動することができます。</li> <li>● Outport ブロックをダイアグラムの右側に配置します。ブロックは、信号の交差を避けるために移動することができます。</li> <li>● 必要に応じて、サブシステム レベルで重複した Inport ブロックを使用できますが、できれば避けてください。 <ul style="list-style-type: none"> <li>➤ ルートレベルでは重複した Inport ブロックを使用しないでください。</li> </ul> </li> </ul> <p><b>【正】</b></p>


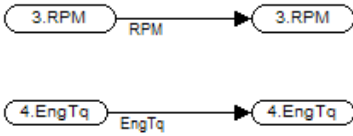
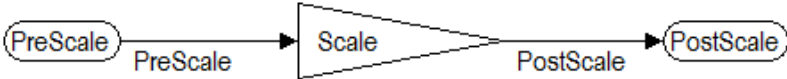
	<p><b>【誤】</b></p> <p>誤りのあるモデルについての注釈</p> <ul style="list-style-type: none"> <li>• Inport 2 は、フィードバック ループ ラインと交差しないように移動した方が良い。</li> <li>• Output 1 は、ダイアグラムの右側に移動した方が良い。</li> </ul>
最終更新	V2.00

#### 7.1.4. na\_0005: Port block name visibility in Simulink models

ID: タイトル	na_0005: Inport ブロック / Outport ブロック名の表示
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>項目によっては、すべての組織の内部プロセスに適用できる 1 つの方法を定義することは不可能ですが、少なくとも組織内では 1 つの一貫した方法に従うことが重要です。ガイドラインを適用する組織は、以下のいずれかのオプションを実行しなければなりません。</p> <p>以下のいずれかの方法を適用してください。</p> <ol style="list-style-type: none"> <li>1. Inport または Outport ブロックの名前を表示します ([書式]、[ブロック名の非表示] は不可)。</li> </ol> <ol style="list-style-type: none"> <li>2. Inport または Outport ブロックの名前を非表示にしなければなりません ([書式]、[ブロック名の非表示] を使用)。 例外: ライブラリ サブシステム ブロック内では名前を非表示にすることはできません。</li> </ol> <p>正:信号ラベルの使用</p>

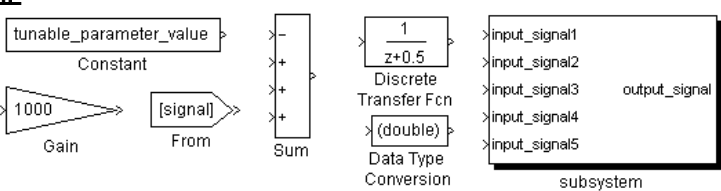
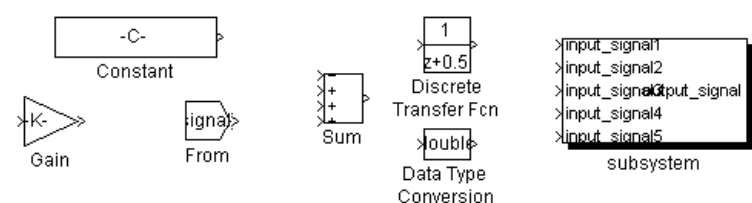
	
最終更新	V2.20

#### 7.1.5. jc\_0081: Icon display for Port block

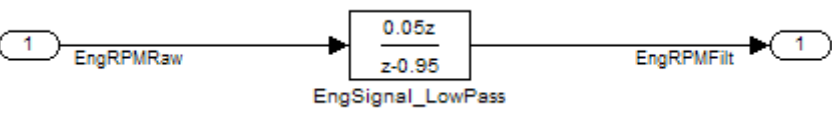
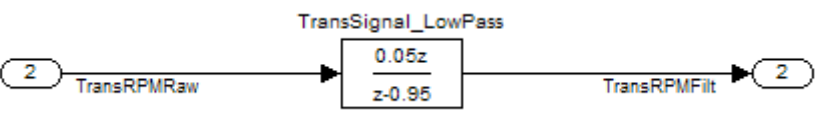
ID: タイトル	jc_0081: Inport ブロック / Outport ブロックの“アイコンの表示”
重要度	推奨
Scope	MAAB
MATLAB Version	R14 and later
前提条件	
説明	<p>アイコンの表示設定は、Inport ブロックと Outport ブロックの端子番号に設定しなければなりません。</p> <p><b>正</b></p>  <p><b>誤</b></p>  <p><b>誤</b></p> 
最終更新	V2.20

#### 7.1.6. jm\_0002: Block resizing

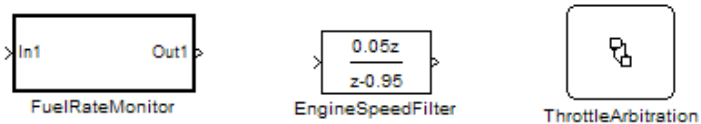
ID: タイトル	jm_0002: ブロックのサイズ調整
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>モデル内のすべてのブロックは、アイコンが完全に表示されて確認できるサイズに調整しなければなりません。特に、アイコン内に表示されるテキスト (調整可能なパラメータ、ファイル名、方程式など) を読みやすくしなければなりません。</p> <p>このガイドラインでは、さまざまなアイコンがあるブロックや、入出力変数があるブロックのサイズを調整しなければなりません。ただし、サブシステム ブロック内の入力名や出力名がすべて判読できるように、サブシステム ブロックのアイコンをサイズ調整することは合理的でなかったり望ましくない場合があります。そのような場合は、マスクを使用するか、アイコンに関連付けられているサブシステム内の名前を非表示にして、名前を隠してもかまいません。これを行う場合は、サブシステム ブロックの近くで、そのブロックに出入りする信号線にラベルを明記する必要があります。</p>

	<p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.00

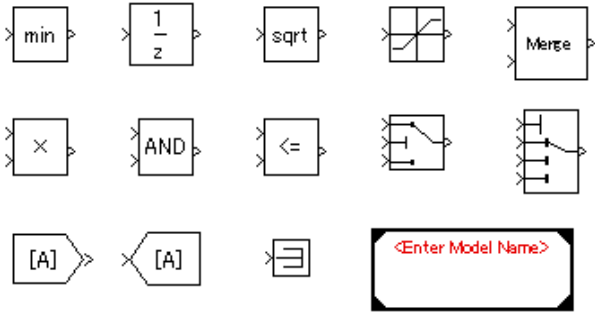
### 7.1.7. db\_0142: Position of block names

ID: タイトル	db_0142: ブロック名の位置
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>ブロック名を表示する場合は、ブロックの下に配置します。</p> <p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.00

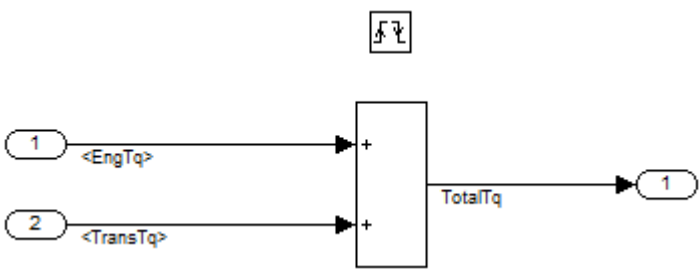
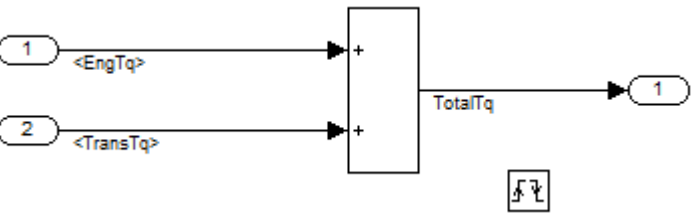
### 7.1.8. jc\_0061: Display of block names

ID: タイトル	jc_0061: ブロック名の表示
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>● 説明的なブロック名である場合は、ブロック名を表示します。</p> <p>例</p>  <p>● ブロックの機能が既知である場合やブロックの外観からわかる場合は、ブロック名を</p>



	<p>表示しません。</p> 
最終更新	V2.00

### 7.1.9. db\_0146: Triggered, enabled, conditional Subsystems

ID: タイトル	db_0146: 条件付き Subsystem 内のブロック配置
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>サブシステムを条件付または反復的に定義するブロックは、サブシステム ダイアグラムの最上位の定位置に配置しなければなりません。ブロックは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• Enable</li> <li>• For Iterator</li> <li>• Action Port</li> <li>• Switch Case Action</li> <li>• Trigger</li> <li>• While Iterator</li> </ul> <p>メモ: Action Port は、If ブロックおよび Case ブロックと関連付けられています。Trigger 端子は、function-call ブロックでもあります。</p> <p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.20

### 7.1.10. db\_0140: Display of basic block parameters

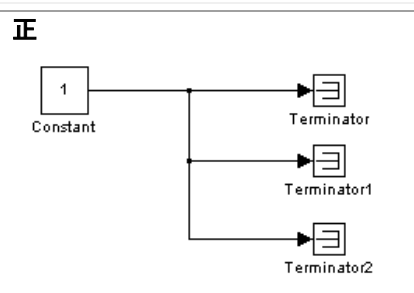
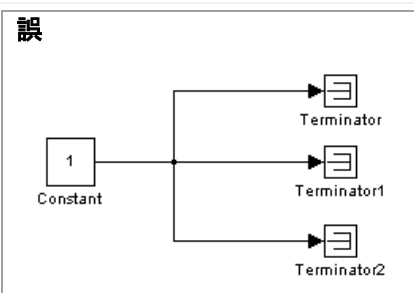
ID: タイトル	db_0140: ブロックパラメータの表示
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>既定値から変更された重要なブロック パラメータは表示されなければなりません。          メモ: 属性文字列は、ブロック パラメータの表示をサポートする 1 つの方法です。[ブロック注釈] タブで、必要な属性情報を追加することができます。          R2011b では、基本ブロックのマスキングは、情報を表示する方法としてサポートされています。この方法は、ベース アイコンが区別できる場合に使用できます。</p> <p>正</p> <p>誤: マスクされたブロック</p> <p>サブシステムではなく、基本ブロックをマスク化して、表示機能を使用する</p>
最終更新	V2.20

### 7.1.11. jm\_0013: アノテーションの影

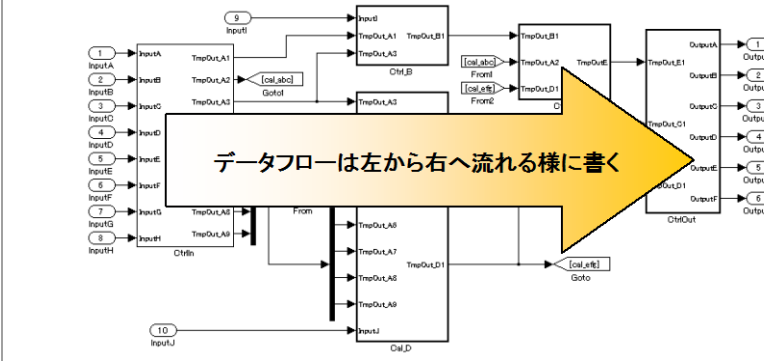
ID: タイトル	jm_0013: アノテーションの影
	削除

最終更新	V3.0
------	------

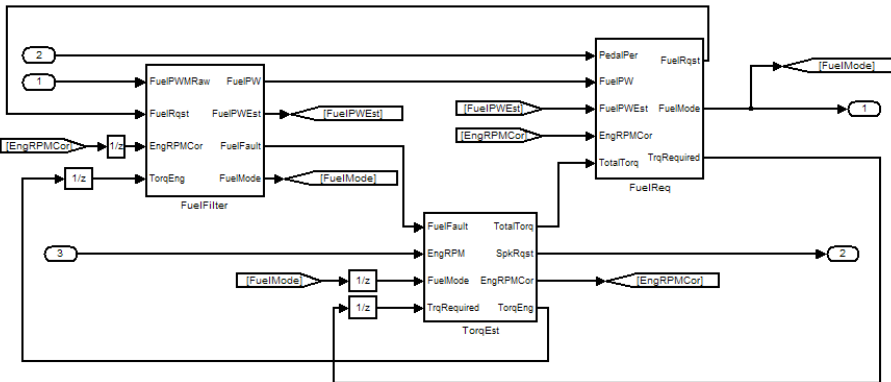
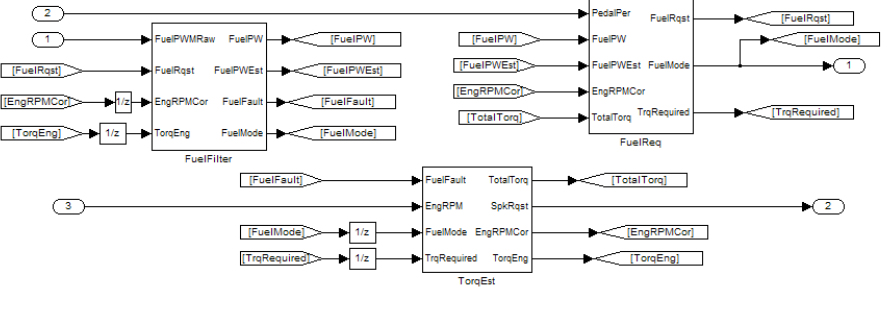
### 7.1.12. db\_0032: Simulink signal appearance

ID: タイトル	db_0032: 信号線の結線
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>信号線</p> <ul style="list-style-type: none"> <li>可能な限り交差させない</li> <li>直角で曲がる</li> <li>他の信号線と重ねてはならない</li> <li>ブロックを横切ってはならない</li> <li>1つの分岐点で複数の線分岐しない</li> </ul> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p><b>正</b></p>  </div> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p><b>誤</b></p>  </div> </div>
最終更新	V2.00

### 7.1.13. db\_0141: Signal flow in Simulink models

ID: タイトル	db_0141: Simulink モデルのデータフロー
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>モデル内の信号フローは左から右に向かって記述します。</p> <p>例外: フィードバック ループ 連続ブロックまたはサブシステムは左から右へと配置します。</p> <p>例外: フィードバック ループ 並列ブロックまたはサブシステムは上から下へと配置します。</p> <div style="text-align: center;">  <p>データフローは左から右へ流れる様を書く</p> </div>
最終更新	V2.00

### 7.1.14. jc\_0171: Maintaining signal flow when using Goto and From blocks

ID: タイトル	jc_0171: Goto ブロック / From ブロック使用時の結線
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>サブシステム間の信号フローの描画を維持しなければなりません。</p> <ul style="list-style-type: none"> <li>以下の場合、Goto ブロックおよび From ブロックを使用できます。 <ul style="list-style-type: none"> <li>接続されたサブシステム間に少なくとも 1 つの信号線がある。</li> <li>フィードフォワードおよびフィードバック ループで接続されているサブシステムに、それぞれの方向に少なくとも 1 本の信号線がある。</li> </ul> </li> <li>Goto ブロックおよび From ブロックを使用して、バスを作成する、または入力を Merge ブロックに接続することは、この規則の例外です。</li> </ul> <p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.20

### 7.1.15. na\_0032: Use of Merge Blocks

ID: タイトル	na_0032: マージブロックの使い方
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	None
説明	<p>マージブロックを利用する際は、</p> <ul style="list-style-type: none"> <li>マージブロックに入っている信号は、他のどのブロックにも分岐してはならな</li> </ul>

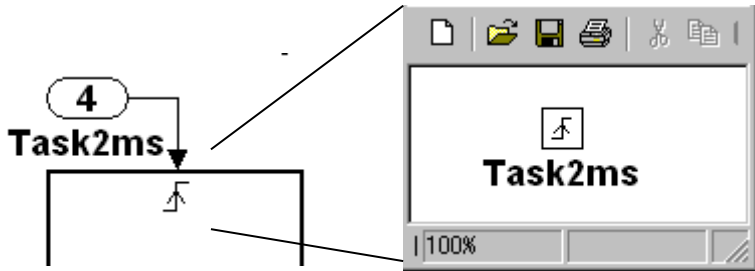
	<p>い。</p> <ul style="list-style-type: none"> <li>・ バスでマージブロックを利用するとき : <ul style="list-style-type: none"> <li>o 全てのバスは同一でなければならない。これは次を含みます。 <ul style="list-style-type: none"> <li>・ 要素数</li> <li>・ エLEMENT名</li> <li>・ 要素の順番</li> <li>・ 要素データタイプ</li> <li>・ 要素サイズ</li> </ul> </li> <li>o バスは全てバーチャルか、または全てノンバーチャルでなければならない。</li> <li>o マージブロックに入っている全てのバス信号線は、他のどのブロックにも分岐してはならない。</li> </ul> </li> </ul>
See Also	jh_0109: Merge blocks
最終更新	V3.00

#### 7.1.16. jm\_0010: Port block names in Simulink models

ID: タイトル	<b>jm_0010: Inport ブロック / Outport ブロックの名前</b>
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0042: Ports in Simulink models</a> <a href="#">na_0005: Port block name visibility in Simulink models</a>
説明	<p>項目によっては、すべての組織の内部プロセスに 1 つの方法を定義することはできないかもしれませんが、組織内では 1 つの一貫した方法に従ってください。ガイドラインを適用する組織は、以下の "いずれか" のオプションを実行しなければなりません。</p> <ul style="list-style-type: none"> <li>● Inport ブロックと Outport ブロックの名前を、対応する信号またはバスの名前と一致させなければなりません。</li> </ul> <p>例外:</p> <ul style="list-style-type: none"> <li>➤ Inport ブロック、Outport ブロック、その他のブロックの組み合わせが同じブロック名の場合は、Inport ブロックと Outport ブロックにサフィックスまたはプレフィックスを使用します。</li> <li>➤ よく使用されるサフィックス/プレフィックスは、Inport ブロックでは _in、Outport ブロックでは _out です。</li> <li>➤ 端子に使用するサフィックスやプレフィックスは任意ですが、一貫したプレフィックスを選択しなければなりません。</li> <li>➤ 一般的な機能をカプセル化するライブラリ ブロックと再利用可能なサブシステム。</li> </ul> <ul style="list-style-type: none"> <li>● Inport ブロックと Outport ブロックの名前が非表示の場合は、ブロックに一貫した命名基準を適用してください。推奨される命名方法としては、既定の名前のままにする (たとえば Out1)、関連付けられた信号と同じ名前にする、または関連付けられた信号の名前を短縮形にする、などがあります。</li> </ul>
最終更新	V2.00

#### 7.1.17. jc\_0281: Naming of Trigger Port block and Enable Port block

ID: タイトル	<b>jc_0281: Trigger Port ブロック / Enable Port ブロックの名前</b>
重要度	強く推奨
Scope	J-MAAB
MATLAB Version	All

前提条件	
説明	<p>Trigger Port ブロックと Enable Port ブロックの場合は、サブシステムをトリガーする信号のブロック名と一致させます。</p> 
最終更新	V2.00

## 7.2. Signals

信号は、スカラー、ベクトル、バスのいずれの場合もあります。データを含むものも制御フローを含むものもあります。

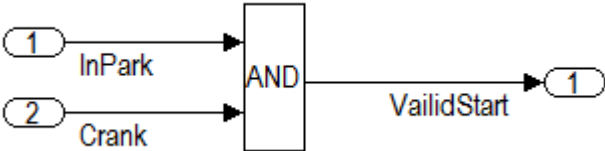
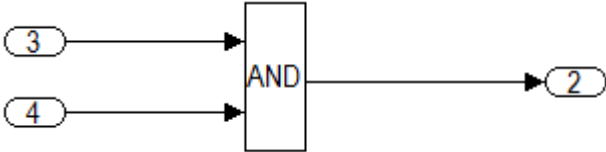
信号ラベルは、Simulink ブロック線図でモデルの機能をわかりやすくするために使用します。さらに、シミュレーションおよびコード生成で使用する変数名を管理するためにも使用できます。信号名は (信号の発生時に) 一度だけ入力します。また、信号名をモデルの別の場所に表示する場合があります。そのような場合、その信号が機能的に変換されるまで信号名を継承しなければなりません (信号は積分器を通すことで、機能的に変換されます。入れ子になったサブシステムへのインポートへ通しても、信号は機能的に変換されません)。名前を付けた信号が機能的に変換されたら、新しい名前をそれに関連付けてください。

別に明確に指定されていない限り、「信号」のガイドラインはあらゆる種類の信号に適用されます。

Simulink モデルでの信号の表現の詳細は、Simulink ドキュメンテーションの「信号の取り扱い」を参照してください。

### 7.2.1. na\_0008: Display of labels on signals

ID: タイトル	na_0008: 信号のラベルの表示
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>以下のブロックの出力信号には、ラベルを表示しなければなりません。</p> <ul style="list-style-type: none"> <li>• Inport ブロック</li> <li>• From ブロック (ブロック アイコンの例外適用 – 以下のメモを参照)</li> <li>• Subsystem ブロックまたは Stateflow chart ブロック (ブロック アイコンの例外適用)</li> <li>• Bus Selector ブロック (ツールがこれを強制)</li> <li>• Demux ブロック</li> <li>• Selector ブロック</li> <li>• Data Store Read ブロック (ブロック アイコンの例外適用)</li> <li>• Constant ブロック (ブロック アイコンの例外適用)</li> </ul> <p>以下の接続先ブロックに (直接、または変換されない演算を実行する基本ブロックを通じて) 接続している信号には、ラベルを表示しなければなりません。</p> <ul style="list-style-type: none"> <li>• Outport ブロック</li> <li>• Goto ブロック</li> <li>• Data Store Write ブロック</li> <li>• Bus Creator ブロック</li> <li>• Mux ブロック</li> <li>• Subsystem ブロック</li> <li>• Chart ブロック</li> </ul>

	<p>メモ: ブロックアイコンの例外 (呼び出された場合にのみ適用): 出力ブロックのアイコン表示に信号ラベルが含まれている場合は、接続先ベースのルールにより信号ラベルが他の場所で必要な場合を除いて、接続している信号にラベルを表示する必要はありません。</p> <p>正</p>  <p>誤</p> 
最終更新	V2.20

### 7.2.2. na\_0009: Entry versus propagation of signal labels

ID: タイトル	na_0009: 信号ラベルの登録と伝播
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">na_0008: Display of labels on signals</a>
説明	<p>信号にラベルがある場合は、そのラベルがそこで作成されるか (信号に直接入力)、実際のソースから伝播されるか (小なり記号 (&lt;) を使用してモデルの他の場所から継承) を以下のルールで定義します。</p> <ul style="list-style-type: none"> <li>● 以下の場合は、表示されるすべての信号ラベルを信号に "入力" しなければなりません。 <ul style="list-style-type: none"> <li>➢ モデルのルート (トップ) レベルの Inport からの出力信号</li> <li>➢ 変換される演算を実行する基本ブロックの出力信号 (このルールのみを解釈する目的で、Bus Creator ブロック、Mux ブロック、および Selector ブロックは、変換される演算を実行するブロック間に含まれていると見なされます)。</li> </ul> </li> <li>● 以下の場合は、表示されるすべての信号ラベルを信号に "伝播" しなければなりません。 <ul style="list-style-type: none"> <li>➢ 入れ子になったサブシステムの Inport ブロックから出力する</li> </ul> <p>例外: 入れ子になったサブシステムがライブラリ サブシステムの場合は、ライブラリブロックを再利用できるように Inport からの信号にラベルを入力できます。</p> <ul style="list-style-type: none"> <li>➢ 変換されない演算を実行する基本ブロックから出力する</li> <li>➢ Subsystem または Stateflow Chart ブロックの出力信号</li> </ul> <p>例外: 接続がライブラリ サブシステム ブロック インスタンスの出力からの場合は、ライブラリブロックを再利用できるように新しいラベルを信号に入力できます。</p> </li> </ul>

最終更新	V2.00

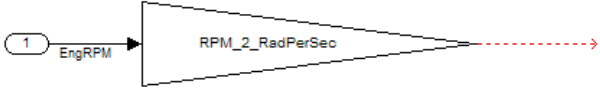
### 7.2.3. db\_0097: Position of labels for signals and busses

ID: タイトル	db_0097: 信号とバスのラベルの位置
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>ラベルは、対応する信号と視覚的に関連付けなければならず、他のラベル、信号、ブロックと重なってはなりません。</p> <p>ラベルは信号線の下に対応するソースまたは接続先ブロックの近くに配置します。</p>
最終更新	V2.00

### 7.2.4. db\_0081: Unconnected signals, block inputs and block outputs

ID: タイトル	db_0081: 接続の信号、ブロック入力、ブロック出力
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>システムは以下を含んではなりません。</p> <ul style="list-style-type: none"> <li>● 未接続のサブシステムまたは基本ブロック入力</li> <li>● 未接続のサブシステムまたは基本ブロック出力</li> <li>● 未接続の信号線</li> </ul> <p>さらに、</p> <ul style="list-style-type: none"> <li>● それ以外の未接続の入力は Ground ブロックに接続します。</li> <li>● それ以外の未接続の出力は Terminator ブロックに接続します。</li> </ul> <p><b>Correct</b></p> <p><b>Incorrect</b></p>



	
最終更新	V2.00

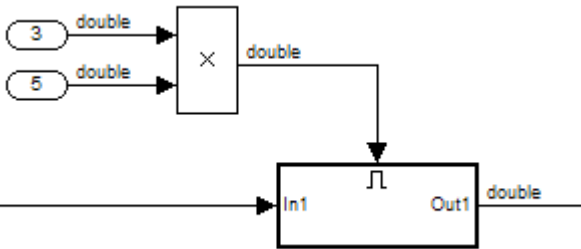
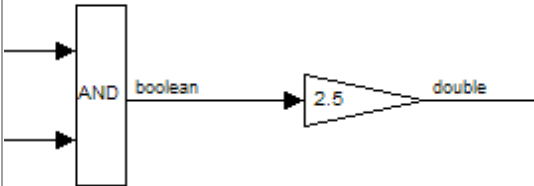
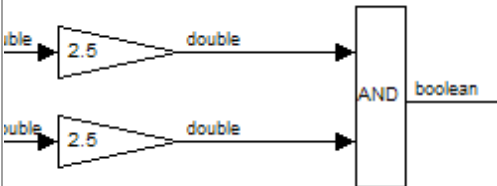
## 7.3. Block Usage

### 7.3.1. na\_0003: Simple logical expressions in If Condition block

ID: タイトル	na_0003: If 条件ブロックの単純な論理式		
重要度	必須		
Scope	MAAB		
MATLAB Version	All		
前提条件			
説明	<p>式内の一次式が 2 つ以下の場合は、論理演算ブロックで構築する代わりに、If 条件ブロック内で論理式を実行できます。一次式は以下のいずれかとして定義されます。</p> <ul style="list-style-type: none"><li>• 入力</li><li>• 定数</li><li>• 定数パラメータ</li><li>• かっこで囲んだ式。次の演算子のうち 0 個または 1 個のインスタンスを除き、演算子を含みません <code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , <code>&gt;=</code> , <code>~=</code> , <code>==</code> , <code>~</code> . (以下の例を参照)。</li></ul> <p>例外</p> <p>以下の両方を満たしている場合は、論理式に一次式を 3 つ以上含めることができます。</p> <ul style="list-style-type: none"><li>• 一次式がすべて入力である</li><li>• 論理演算子が 1 つのタイプだけである</li></ul> <p>許容可能な例外の例</p> <ul style="list-style-type: none"><li>• <code>u1    u2    u3    u4    u5</code></li><li>• <code>u1 &amp;&amp; u2 &amp;&amp; u3 &amp;&amp; u4</code></li></ul> <p>一次式の例</p> <ul style="list-style-type: none"><li>• <code>u1</code></li><li>• <code>5</code></li><li>• <code>K</code></li><li>• <code>(u1 &gt; 0)</code></li><li>• <code>(u1 &lt;= G)</code></li><li>• <code>(u1 &gt; U2)</code></li><li>• <code>(~u1)</code></li><li>• <code>(EngineState.ENGINE_RUNNING)</code></li></ul> <p>許容可能な論理式の例</p> <ul style="list-style-type: none"><li>• <code>u1    u2</code></li><li>• <code>(u1 &gt; 0) &amp;&amp; (u1 &lt; 20)</code></li><li>• <code>(u1 &gt; 0) &amp;&amp; (u2 &lt; u3)</code></li><li>• <code>(u1 &gt; 0) &amp;&amp; (~u2)</code></li><li>• <code>(EngineState.ENGINE_RUNNING &gt; 0) &amp;&amp; (PRNDLState.PRNDL_PARK)</code></li></ul> <p>メモ: この例では、<code>EngineState.ENGINE_RUNNING</code> および <code>PRNDLState.PRNDL_PARK</code> は、列挙型のリテラルです。</p> <p>許容不可能な論理式の例</p>		
	<code>u1 &amp;&amp; u2    u3</code>	(一次式が多すぎる)	












	$u1 \ \&\& \ (u2 \    \ u3)$	(一次式内に許容不可能な演算子がある)
	$(u1 > 0) \ \&\& \ (u1 < 20) \ \&\& \ (u2 > 5)$	(入力でない一次式が多すぎる)
	$(u1 > 0) \ \&\& \ ((2 * u2) > 6)$	(一次式内に許容不可能な演算子がある)
最終更新	V2.20	

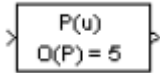
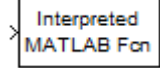

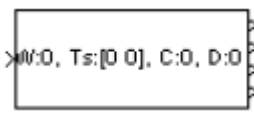
### 7.3.2. na\_0002: Appropriate implementation of fundamental logical and numerical operations

ID: タイトル	na_0002: 基本的な論理演算と数値演算の適切な実装
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>● 数値演算を実行するためのブロックを、論理演算の実行に使用してはなりません。 誤</p>  <p>● 論理出力を、数値入力を処理するブロックの入力に直接接続してはなりません。 ● 論理式フラグメントの結果を数値演算子で処理してはなりません。 ● 論理演算用のこのガイドラインは、さらに列挙データ型に当てはまります。(This guideline for logical operations also applies to enumerated data types.) 誤</p>  <p>● 論理演算を実行するためのブロックを、数値演算の実行に使用してはなりません。 ● 数値出力を、論理入力を処理するブロックの入力に接続してはなりません。 誤</p> 

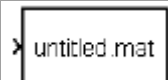
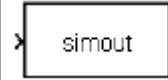

最終更新	V3.00
------	-------

### 7.3.3. jm\_0001: Prohibited Simulink standard blocks inside controllers

ID: タイトル	jm_0001: コントローラ内で禁止されている Simulink 標準ブロック	
重要度	必須	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	<ul style="list-style-type: none"> <li>● コントローラ モデルは離散ブロックから設計しなければなりません。</li> <li>● MathWorks "Simulink ブロックのデータ型サポート" テーブルは、量産コード生成をサポートするブロックの一覧を示します。「Simulink ブロックのデータ型サポート」を参照してください。 <ul style="list-style-type: none"> <li>➢ 「コード生成のサポート」と記載されているブロックを使用します。</li> <li>➢ 「量産コードには非推奨」と記載されているブロックは使用しないでください。表の脚注 4 を参照してください。</li> </ul> </li> <li>● 上述の規則で定義されたブロックに加えて、以下のブロックも使用しないでください。以下のソースの使用は "禁止" されています。</li> </ul>	
	<b>Sources are not allowed:</b>	
	Sine Wave	
	Pulse Generator	
	Random Number	
	Uniform Random Number	
	Band-Limited White Noise	
	<p>以下の追加ブロックの使用は "禁止" されています。MAAB スタイル ガイド グループは以下のブロックの使用を推奨していません。このリストは個々の会社ごとに拡張できます。</p>	
	Slider Gain	
	Manual Switch	
	Complex to Magnitude-Angle	
	Magnitude-Angle to Complex	
	Complex to Real-Imag	
	Real-Imag to Complex	

	Polynomial	
	MATLAB Fcn <sup>(1)</sup>	
	Goto Tag Visibility	
	Probe	
Notes	(1) R2011a では、MATLAB Fcn ブロックは名前が Interpreted MATLAB Function ブロックに変更されました。	
最終更新	V2.20	

#### 7.3.4. hd\_0001: Prohibited Simulink sinks

ID: タイトル	<b>hd_0001: Simulink Sinks カテゴリの使用禁止ブロック</b>	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	<p>コントローラ モデルは離散ブロックから設計しなければなりません。</p> <p>以下の Sink ブロックの使用は "禁止" されています。</p>	
	To File	
	To Workspace	
	Stop Simulation	
Note	<p>メモ: Simulink Scope ブロックおよび Display ブロックは、モデル ダイアグラムで使用できます。データ ログの記録と参照の要件に対しては、Simulink の信号ログおよび信号とスコープのマネージャーの使用を考慮してください。</p>	
最終更新	V2.20	

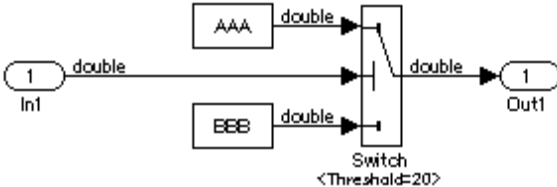
#### 7.3.5. na\_0011: Scope of Goto and From blocks

ID: タイトル	<b>na_0011: Goto ブロックと From ブロックの範囲</b>	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	<p>信号フローに対しては、以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>From ブロックと Goto ブロックではローカル範囲を使用しなければなりません。</li> </ul>	

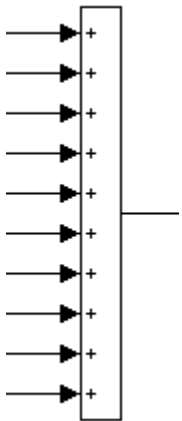
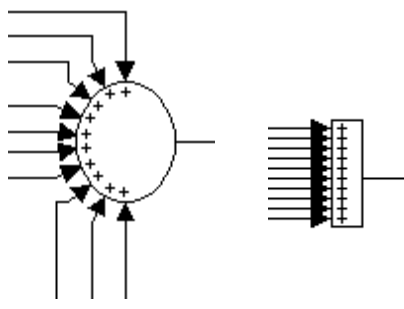
	<div data-bbox="459 141 1214 185" data-label="Text"> <p>メモ: コントロール フロー信号ではグローバル範囲を使用できます。</p> </div> <div data-bbox="419 190 1094 219" data-label="Text"> <p>コントロール フロー信号は、以下のブロックからの出力です。</p> </div> <div data-bbox="419 224 1276 318" data-label="List-Group"> <ul style="list-style-type: none"> <li>● Function-call Generator</li> <li>● If ブロックと Case ブロック</li> <li>● MATLAB ブロックおよび Stateflow ブロックからの関数呼び出しの出力</li> </ul> </div> <div data-bbox="419 322 1418 387" data-label="Text"> <p>コントロール フロー信号は、Simulink モデルを更新した後のモデルでは、破線で示されます。</p> </div> <div data-bbox="419 392 1209 1077" data-label="Diagram"> </div> <tr> <td>最終更新</td><td>V2.20</td></tr>	最終更新	V2.20
最終更新	V2.20		

### 7.3.6. jc\_0141: Usage of the Switch block

ID: タイトル	jc_0141: Switch ブロックの使用方法
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<div data-bbox="419 1568 1406 1668" data-label="List-Group"> <ul style="list-style-type: none"> <li>● Switch 条件の入力 2 は論理値でなければなりません。</li> <li>● ブロック パラメータの [1 番目の入力が通過する条件] は <math>u2 \sim 0</math> に設定しなければなりません。</li> </ul> </div> <div data-bbox="419 1702 1350 1912" data-label="Diagram"> <p>正</p> </div>

	<div data-bbox="443 145 1050 465"> <p><b>Function Block Parameters: Switch</b></p> <p>Switch</p> <p>Pass through input 1 when input 2 satisfies the selected criterion; otherwise, pass through input 3. The inputs are numbered top to bottom (or left to right). The input 1 pass-through criteria are input 2 greater than or equal, greater than, or not equal to the threshold. The first and third input ports are data ports, and the second input port is the control port.</p> <p>Main   Signal Data Types</p> <p>Criteria for passing first input: <math>u2 \sim 0</math></p> <p>Threshold: 0</p> <p><math>u2 \geq \text{Threshold}</math></p> <p><math>u2 &gt; \text{Threshold}</math></p> <p><math>u2 \sim 0</math></p> </div> <div data-bbox="443 474 1018 913"> <p><b>誤</b></p>  <p>Switch &lt;Threshold=20&gt;</p> <div data-bbox="459 728 992 913"> <p>Main   Signal Data Types</p> <p>Criteria for passing first input: <math>u2 \geq \text{Threshold}</math></p> <p>Threshold: 20</p> </div> </div>
最終更新	V2.20

### 7.3.7. jc\_0121: Use of the Sum block

ID: タイトル	jc_0121: Sum ブロックの使用方法
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>Sum ブロックの条件</p> <ul style="list-style-type: none"> <li>● 四角形を使用する。</li> <li>● 入力信号が重ならないようにサイズ調整する。</li> </ul> <div data-bbox="443 1489 1343 2016"> <div data-bbox="443 1489 858 2016"> <p><b>正</b></p>  </div> <div data-bbox="865 1489 1343 2016"> <p><b>誤</b></p>  </div> </div> <p>フィードバック ループでは丸形を使用できます。</p> <ul style="list-style-type: none"> <li>● 入力数は 3 つまでにしてください。</li> </ul>

	<ul style="list-style-type: none"> <li>● 入力は 90°、180°、270°の位置に合わせます。</li> <li>● 出力は 0°の位置に合わせます。</li> </ul>
	<div> <div> <p><b>正</b></p> </div> <div> <p><b>誤</b></p> </div> </div>
最終更新	V2.00

### 7.3.8. jc\_0131: Use of Relational Operator block

ID: タイトル	jc_0131: Relational Operator ブロックの使用方法
重要度	推奨
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>関係演算子を信号と定数値の比較に使用する場合は、定数入力を 2 番目の (下の) 入力にします。</p> <div> <div> <p><b>正</b></p> </div> <div> <p><b>誤</b></p> </div> </div>
最終更新	V2.00

### 7.3.9. jc\_0161: Use of Data Store Read/Write/Memory blocks

ID: タイトル	jc_0161: Data Store Read/Write/Memory ブロックの使用方法
重要度	強く推奨

Scope	J-MAAB
MATLAB Version	All
前提条件	<a href="#">jc_0341: Data flow layer</a>
説明	<div> <div>Data Store Read</div> <div>   Data Store Read </div> <div>Data Store Write</div> <div>   Data Store Write </div> <div>Data Store Memory</div> <div>   Data Store Memory </div> </div> <p>Data Store Memory、Data Store Read、および Data Store Write ブロックの条件</p> <ul style="list-style-type: none"> <li>データフローレイヤでは使用禁止</li> <li>異なる速度で実行しているサブシステム間で使用可能</li> </ul>
最終更新	V2.00

## 7.4. Block Parameters

### 7.4.1. db\_0112: Indexing

ID: タイトル	<b>db_0112: インデックスの使用方法</b>
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>すべてのブロックに一貫したベクトル インデックス方法を使用する。 可能な場合は、0 ベースのインデックスを使用してコード効率を改善します。ただし、MATLAB ブロックは 0 ベースのインデックスをサポートしないため、MATLAB ブロックを含むモデルでは、1 ベースのインデックスを使用できます。</p>
See Also	<ul style="list-style-type: none"> <li>cgs1_0101: Zero-based indexing 「0 ベースのインデックス」</li> <li>his1_0021: Consistent vector indexing 「一貫したベクトル インデックス方法」</li> </ul>
最終更新	V2.20

### 7.4.2. na\_0010: Grouping data flows into signals

ID: タイトル	<b>na_0010: ベクトル信号 / バス信号の使用方法</b>
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p><b>ベクトル</b> ベクトルを構成する個々のスカラー信号は、共通する機能、データ型、次元、および単位をもっていなければなりません。ベクトル信号の最も一般的な例には、場所のインデックスが付いた配列にグループ化されたセンサー データまたはアクチュエータ データがあります。Mux ブロックの出力は常にベクトルでなければなりません。Mux ブロックへの入力には常にスカラーでなければなりません。</p> <p><b>バス</b> 前述のような、ベクトルとしての条件を満たさない信号は、バス信号にしかグループ化でき</p>



ません。Bus Selector ブロックは、バス信号入力でのみ使用し、ベクトル信号からスカラー信号を抽出するためには使用しないでください。

例

次に、ベクトル信号の例を示します。

ベクトルのタイプ	サイズ
行ベクトル	[1 n]
列ベクトル	[n 1]
ホイール速度サブシステム	[1 ホイール数]
シリンダー ベクトル	[1 シリンダー数]
2 次元座標に基づく位置ベクトル	[1 2]
3 次元座標に基づく位置ベクトル	[1 3]

次に、バス信号の例を示します。

バスのタイプ	要素
センサー バス	カベクトル [Fx, Fy, Fz]
	位置
	ホイール速度ベクトル [ $\Theta_x$ , $\Theta_y$ , $\Theta_z$ , $\Theta_r$ ]
	加速
	圧力
コントローラ バス	センサー バス
	アクチュエータ バス
シリアル データ バス	冷却水温度
	エンジン速度、助手席ドア開

最終更新

V2.00

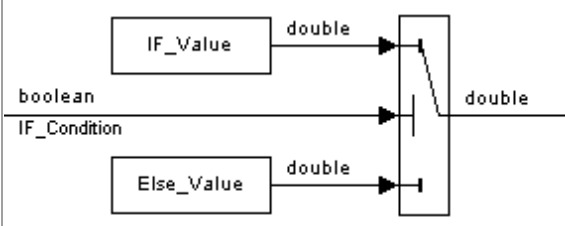
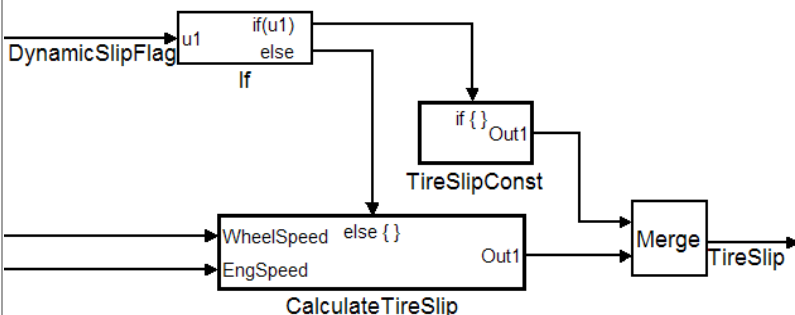
### 7.4.3. db\_0110: Tunable parameters in basic blocks

ID: タイトル	db_0110: 基本ブロック内の調整可能なパラメータ									
重要度	強く推奨									
Scope	MAAB									
MATLAB Version	All									
前提条件										
説明	<p>パラメータを調整可能にするには、そのパラメータを次のようにしてブロック ダイアログ フィールドに入力します。</p> <ul style="list-style-type: none"><li>● 式なしで入力</li><li>● データ型を変換せずに入力</li><li>● 行または列を選択せずに入力</li></ul> <p><b>正</b></p> <table><tr><td>tunable_parameter_value</td><td>tunable_parameter_vector</td><td>tunable_parameter_array</td></tr></table> <p><b>誤</b></p> <table><tr><td>tunable_parameter_value*2</td><td>tunable_parameter_vector*3</td><td>tunable_parameter_array*3</td></tr><tr><td>int16(tunable_parameter_value)</td><td>tunable_parameter_vector(2)</td><td>tunable_parameter_array(1,1)</td></tr></table>	tunable_parameter_value	tunable_parameter_vector	tunable_parameter_array	tunable_parameter_value*2	tunable_parameter_vector*3	tunable_parameter_array*3	int16(tunable_parameter_value)	tunable_parameter_vector(2)	tunable_parameter_array(1,1)
tunable_parameter_value	tunable_parameter_vector	tunable_parameter_array								
tunable_parameter_value*2	tunable_parameter_vector*3	tunable_parameter_array*3								
int16(tunable_parameter_value)	tunable_parameter_vector(2)	tunable_parameter_array(1,1)								
最終更新	V2.20									

## 7.5. Simulink patterns

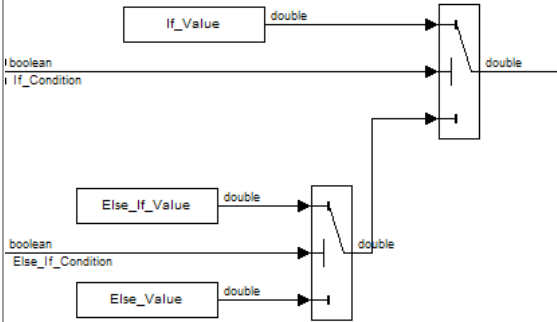
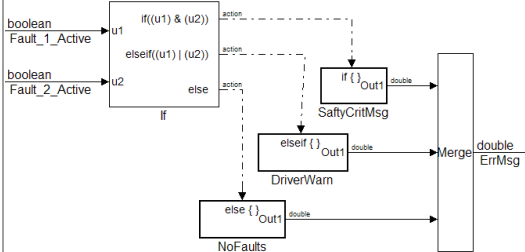
The following rules illustrate sample patterns used in Simulink diagrams. As such, they would normally be part of a much larger Simulink diagram.

### 7.5.1. na\_0012: Use of Switch vs. If-Then-Else Action Subsystem

ID: タイトル	na_0012: Switch と If-Then-Else Action Subsystem の使い分け
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>関連する then アクションと else アクションに定数値の代入のみが含まれる場合、単純な if-then-else 構造体のモデル化には Switch ブロックを使用してください。</p>  <p>if-then-else Action Subsystem は以下のような場合に使用します。</p> <ul style="list-style-type: none"> <li>● 関連する then アクションまたは else アクションに複雑な計算が必要な場合、if-then-else 構造体のモデル化に使用してください。これにより、シミュレーションの効率だけでなく、生成されるコードの効率も最大限に向上します (テーブル ルックアップなどの基本ブロックでも、かなり複雑な計算が必要になる場合があることに注意してください)。</li> </ul>  <p>例えば以下の場合</p> <ul style="list-style-type: none"> <li>● ゼロ除算などの望ましくない数値計算を避けることである場合</li> <li>● 明示的または暗黙的な then アクションまたは else アクションが、関連する出力値を切り替えて使用する場合</li> </ul> <p>その他の場合、then または else アクションの計算の複雑度、および Simulink シミュレーションおよびコード生成エンジンの能力により、適切な構文が決定されます。</p> <p>このことは、より複雑な入れ子およびカスケードの if-then-else 構造体および case 構造体の実装にも適用されます。</p>
最終更新	V2.00

### 7.5.2. db\_0114: Simulink patterns for If-then-else-if constructs

ID: タイトル	db_0114: If-then-else-if 構文の Simulink パターン
重要度	強く推奨
Scope	MAAB
MATLAB	All

Version		
前提条件		
説明	Simulink モデル内の If-then-else-if 構文には以下のパターンを使用します。	
	等価な機能	Simulink パターン
	<p>ブロックを含む if then else if</p> <pre>if (If_Condition) {   output_signal = If_Value; } else if (Else_If_Condition) {   output_signal =   Else_If_Value; } else {   output_signal =   Else_Value; }</pre>	
	等価な機能	Simulink パターン
	<p>if/then/else サブシステムを 含む if then else if</p> <pre>if(Fault_1_Active &amp;   Fault_2_Active) {   ErrMsg = SaftyCrit; } else if (Fault_1_Active     Fault_2_Active) {   ErrMsg = DriveWarn; } else {   ErrMsg = NoFaults; }</pre>	
最終更新	V2.00	

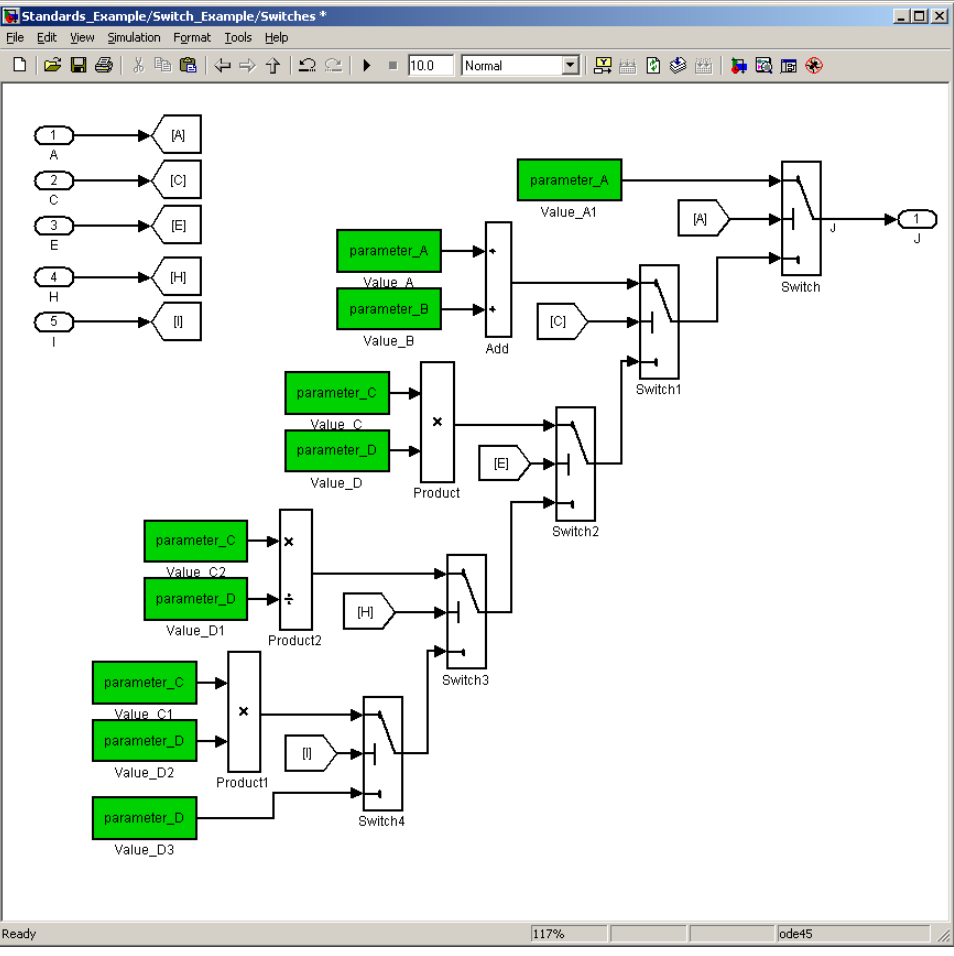
### 7.5.3. db\_0115: Simulink patterns for case constructs

ID: タイトル	db_0115: case 構文の Simulink パターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	Simulink モデル内の case 構文には以下のパターンを使用します。	
	等価な機能	Simulink パターン

	Switch Case を含む case <pre> switch (PRNDL_Enum) { case 1     TqEstimate = ParkV;     break; case 2     TqEstimate = RevV;     break; default     TqEstimate = NeutralV;     break; } </pre>	
最終更新	V2.20	

#### 7.5.4. na\_0028: Use of If-Then-Else Action Subsystem to Replace Multiple Switches

ID: タイトル	na_0028: Use of If-Then-Else Action Subsystem to Replace Multiple Switches
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	<a href="#">na_0012: Use of Switch vs. If-Then-Else Action Subsystem</a> <a href="#">db_0114: Simulink パターン s for If-then-else-if constructs</a>
説明	<p>Switch ブロックによる条件分岐の使用は、制限されなければならない； 通常、3 段に限られている</p> <p>3 段以上の場合、代わりに If-Then-else アクションサブシステムを使用してください。</p> <p>誤</p>

	
See also	bn_0003: Use of If-Then-Else Action Subsystem to Replace Multiple Switches
最終更新	V3.00

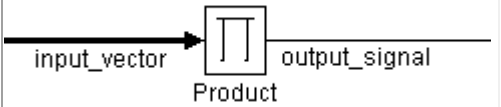
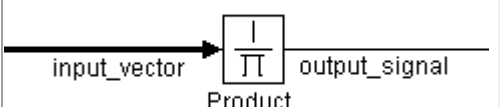
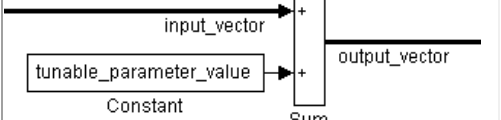
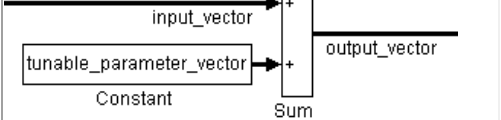
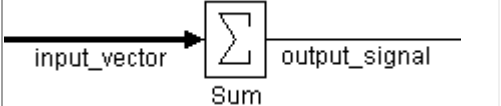
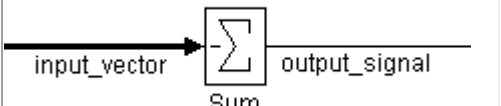
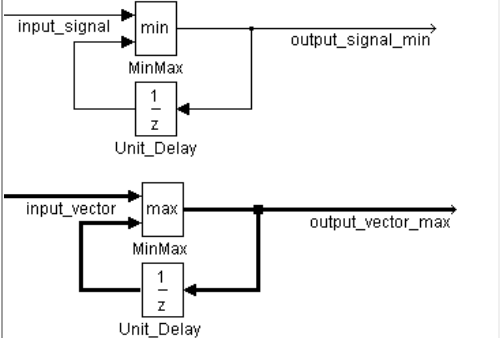
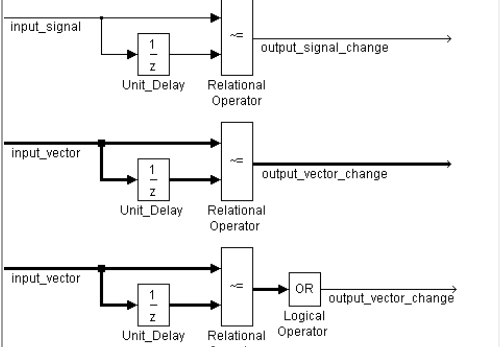
#### 7.5.5. db\_0116: Simulink patterns for logical constructs with logical blocks

ID: タイトル	db_0116: 論理ブロックをもつ論理構文の Simulink パターン
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	

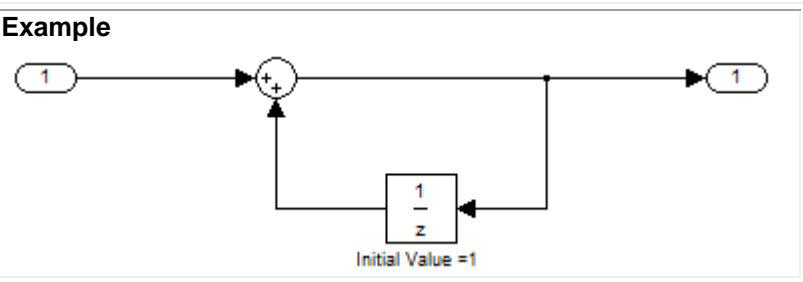
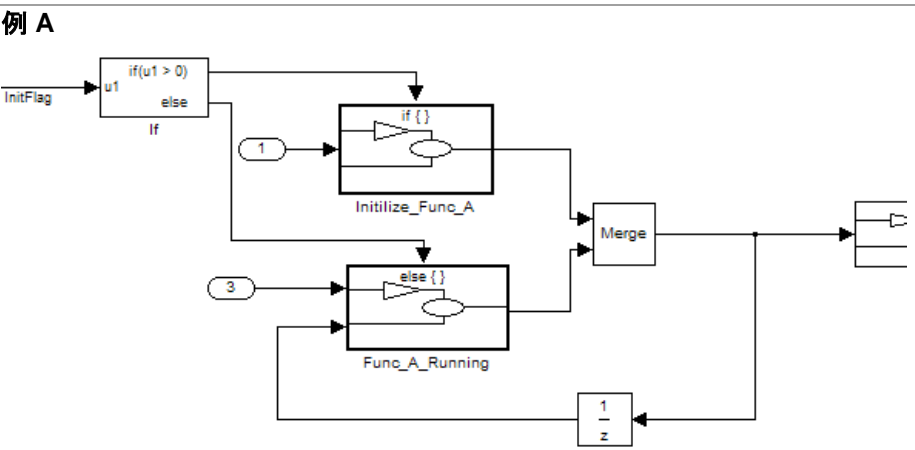
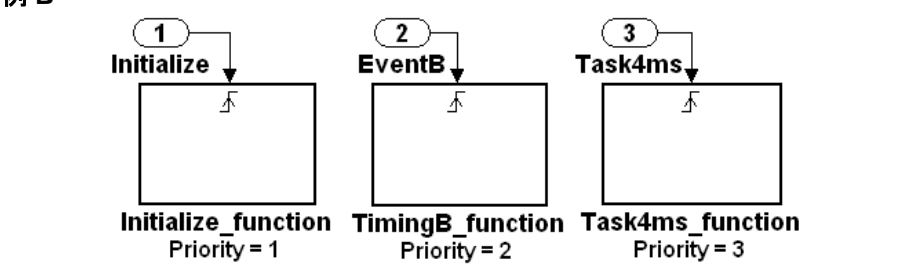
説明	Simulink 内の論理組み合わせには以下のパターンを使用します。	
	等価な機能	Simulink パターン
	論理信号の組み合わせ: 乗法	
最終更新	V1.00	

#### 7.5.6. db\_0117: Simulink patterns for vector signals

ID: タイトル	db_0117: ベクトル信号の Simulink パターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	Simulink は、ベクトル化可能なモデリング言語で、ベクトル データの直接処理が可能です。Simulink モデル内のベクトル信号には以下のパターンを使用します。	
	等価な機能	Simulink パターン
	ベクトル ループ: <pre>for (i=0; i&lt;input_vector_size; i++) {   output_vector(i) = input_vector(i) *   tunable_parameter_value; }</pre>	
	ベクトル ループ: <pre>for (i=0; i&lt;input_vector_size; i++) {   output_vector(i) = input_vector(i) *   tunable_parameter_vector(i); }</pre>	

}	
ベクトル ループ: output_signal = 1; for (i=0; i<input_vector_size; i++) { output_signal = output_signal * input_vector(i); }	
ベクトル ループ: output_signal = 1; for (i=0; i<input_vector_size; i++) { output_signal = output_signal / input_vector(i); }	
ベクトル ループ: for (i=0; i<input_vector_size; i++) { output_vector(i) = input_vector(i) + tunable_parameter_value; }	
ベクトル ループ: for (i=0; i<input_vector_size; i++) { output_vector(i) = input_vector(i) + tunable_parameter_vector(i); }	
ベクトル ループ: output_signal = 0; for (i=0; i<input_vector_size; i++) { output_signal = output_signal + input_vector(i); }	
ベクトル ループ: output_signal = 0; for (i=0; i<input_vector_size; i++) { output_signal = output_signal - input_vector(i); }	
経時変化する信号またはベクトルの最小値または最大値:	
信号またはベクトルの変更イベント:	

### 7.5.7. jc\_0351: Methods of initialization

ID: タイトル	jc_0351: 初期化方法
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0140: Display of block parameters</a>
説明	<p><b>単純な初期化</b></p> <ul style="list-style-type: none"> <li>● 初期値フィールドをもつ Unit Delay などのブロックは、単純な初期値の設定に使用できます。</li> <li>● 初期値の表示が必要かどうかを判断するには、MAAB ガイドライン「db_0140: Display of basic block parameters」を参照してください。</li> </ul> <p><b>Example</b></p>  <p><b>計算が必要な初期化</b></p> <p>複雑な初期化には、以下のルールが適用されます。</p> <ul style="list-style-type: none"> <li>● 初期化は別個のサブシステムで実行しなければなりません。</li> <li>● 初期化サブシステムには、そのサブシステムが初期化を実行していることを示す名前が必要です。</li> </ul> <p>複雑な初期化は、ローカル レベル (例 A) またはグローバル レベル (例 B)、あるいはローカルとグローバルの組み合わせのレベルで実行できます。</p> <p><b>例 A</b></p>  <p><b>例 B</b></p>  <p>または</p> <p>Or</p>



最終更新	V2.00

### 7.5.8. jc\_0111: Direction of Subsystem

ID: タイトル	jc_0111: サブシステムの方
重要度	強く推奨
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>サブシステムの方を逆にするにはできません。</p> <p><b>正</b></p> <p><b>誤</b></p>
最終更新	V2.00

## 8.Stateflow

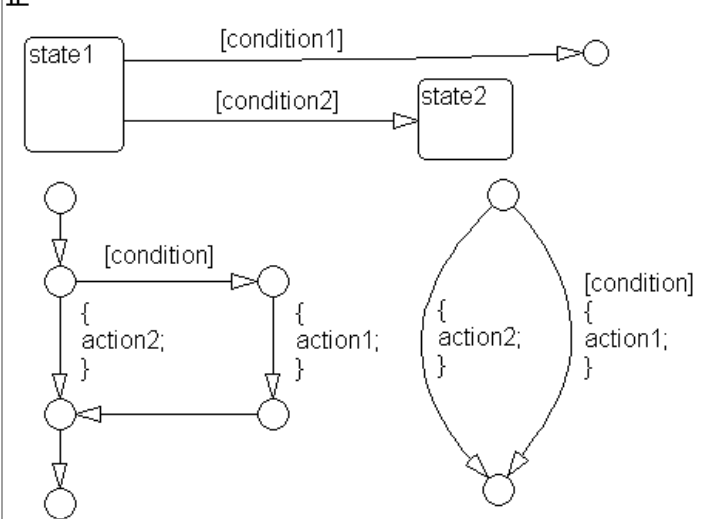
Stateflow® チャートの外観、データと演算、イベント、ステートチャート パターン、およびフローチャート パターンのガイドラインについての説明

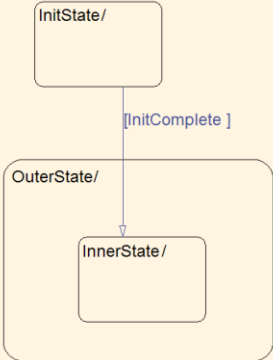
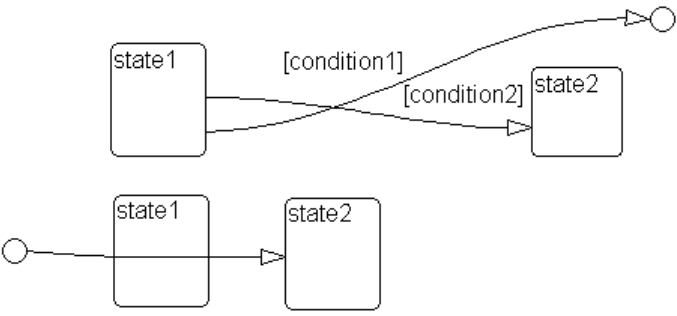
### 8.1. Chart Appearance

#### 8.1.1. db\_0123: Stateflow port names

ID: タイトル	db_0123: Stateflow 端子の名前
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	Stateflow の入力または出力の名前は、対応する信号と同じでなければなりません。 例外: 再利用可能な Stateflow ブロックは異なる端子名をもつ場合があります。
最終更新	V1.00

#### 8.1.2. db\_0129: Stateflow transition appearance

ID: タイトル	db_0129: Stateflow 遷移の外観
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>Stateflow での遷移では以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● 可能な限り交差させない。</li> <li>● 他の信号線と重ねてはならない。</li> <li>● 状態、ジャンクション、テキスト フィールドを交差させない。</li> <li>● 内部状態への遷移では Stateflow での遷移が可能。</li> </ul> <p>遷移ラベルは、対応する遷移に視覚的に関連付けることができます。</p> <p>正</p>  <p>正: 遷移がステートの境界線に交差して、サブステートを接続する</p>

	 <p>誤 遷移が相互に交差し、遷移がステートを通して交差する</p> 
最終更新	V2.20

### 8.1.3. db\_0137: States in state machines


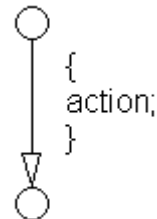
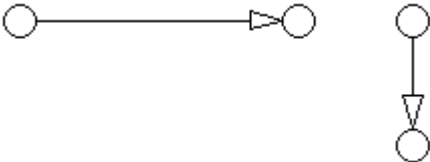
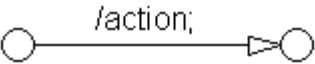
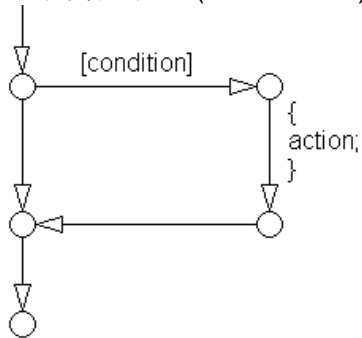
ID: タイトル	db_0137: ステート マシンのステート
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0149: Flowchart patterns for condition actions</a>
説明	<p>サブステートのステート マシンでは以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● 少なくとも 2 つの排他的ステートが存在する。</li> <li>● ステートは 1 つのみのサブステートをもつことはできない。</li> <li>● 排他的ステートが存在するすべての階層レベルの初期ステートは、既定遷移により明確に定義されている。</li> </ul>
最終更新	V3.00

### 8.1.4. db\_0133: Use of patterns for Flowcharts

ID: タイトル	db_0133: フローチャートのパターンの使用
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>フローチャートは、フローチャートのパターン (たとえば、if-then-else、for ループなど) を使用して作成されます。その場合、以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● データフローは、上から下への方向。</li> <li>● パターンは空の遷移と接続される。</li> </ul>

最終更新	V2.20

### 8.1.5. db\_0132: Transitions in Flowcharts

ID: タイトル	db_0132: フローチャートの遷移
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>フローチャートでの遷移には以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● 条件は水平軸に描画される。</li> <li>● アクションは垂直軸に描画される。</li> <li>● ループ 構文は、この規則に対する意図的な例外。</li> <li>● 遷移には、条件、条件アクション、または空の遷移がある。</li> </ul> <p>条件付き遷移:</p>  <p>条件アクション付き遷移:</p>  <p>空の遷移:</p>  <p>遷移アクションは、フローチャートでは使用されません。遷移アクションは、ステート マシンの状態間の遷移で使用する場合にのみ有効です。遷移アクションは、有効な状態間の遷移でしか有効にならないため、それ以外の場合には無効です。</p> <p>遷移アクション:</p>  <p>流れ図の最後のジャンクションを除く、各ジャンクションで、1 件の条件なし遷移が開始されます。各判定点 (ジャンクション) には既定のパスが必要です。</p>  <p>コメント付きの遷移:</p>

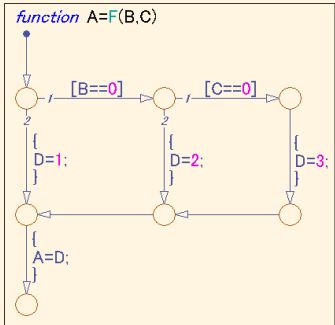
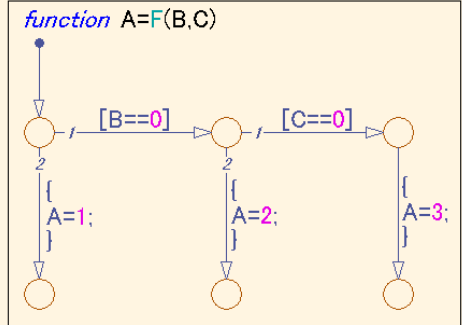
最終更新	V2.00

#### 8.1.6. jc\_0501: Format of entries in a State block

ID: タイトル	jc_0501: State ブロック内の記述ルール
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>改行には次の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● entry (en)、during (du) および exit (ex) ステートメントの完了“:”の後に改行する。</li> <li>● 代入ステートメントの完了 ";" の後に改行する。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>正</p> </div>

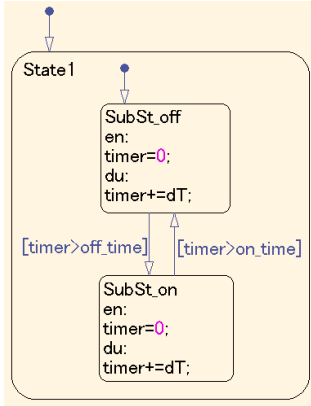
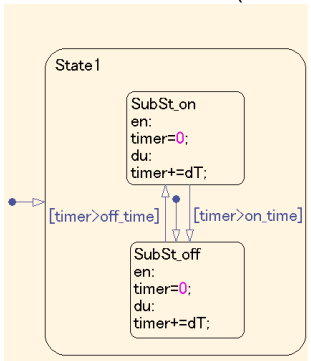
	<p>誤</p> <p>en、du、および ex の後に新しい行を改行していない。</p> <pre> State en:entry_value=1; during_value=0; du:entry_value=0; during_value=1; ex:exit_value=2; </pre> <p>誤</p> <p>代入ステートメントの完了 ";" の後に新しい行を改行していない。</p> <pre> State en:entry_value=1;during_value=0;du:entry_value=0; during_value=1;ex:exit_value=2; </pre>
最終更新	V2.00

### 8.1.7. jc\_0511: Setting the return value from a graphical function

ID: タイトル	jc_0511: グラフィカル関数からの戻り値の設定
重要度	必須
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<p>グラフィカル関数の戻り値は、1 か所のみで設定しなければなりません。</p> <p>正</p> <p>戻り値 A は 1 か所で設定されます。</p>  <p>誤</p> <p>戻り値 A は複数の場所で設定されます。</p> 

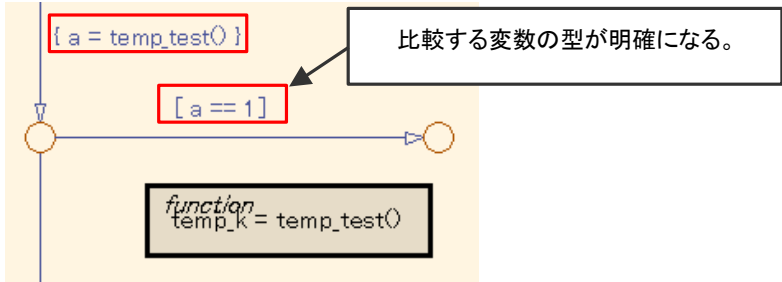
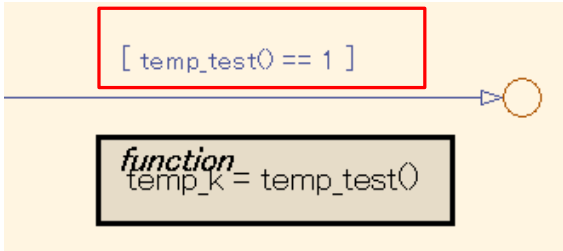
最終更新	V2.00
------	-------

### 8.1.8. jc\_0531: Placement of the default transition

ID: タイトル	jc_0531: デフォルト遷移の配置
重要度	推奨
Scope	J-MAAB
MATLAB Version	All
前提条件	
説明	<ul style="list-style-type: none"> <li>● デフォルト遷移は、ステートの一番上で接続されます。</li> <li>● デフォルト遷移の遷移先ステートは、同じ階層内のその他のステートの上に配置されます。</li> </ul> <p>正</p> <ul style="list-style-type: none"> <li>● デフォルト遷移は、ステートの一番上で接続されます。</li> <li>● デフォルト遷移の遷移先ステートは、同じ階層内のその他のステートの上に配置されます。</li> </ul>  <p>誤</p> <ul style="list-style-type: none"> <li>● デフォルト遷移は、ステート (ステート 1) の横で接続されます。</li> <li>● デフォルト遷移の遷移先ステートは、同じ階層内のその他のステートよりも下になります (SubSt_off)。</li> </ul> 
最終更新	V2.00

### 8.1.9. jc\_0521: Use of the return value from graphical functions

ID: タイトル	jc_0521: グラフィカル関数からの戻り値の使用
重要度	推奨
Scope	J-MAAB
MATLAB Version	All

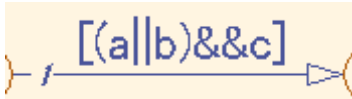
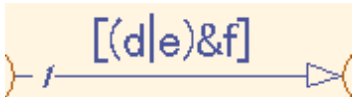

前提条件	
説明	<p>グラフィカル関数の戻り値は、比較演算では直接使用しないでください。</p> <p><b>正</b>  中間変数は、関数 temp_test からの戻り値を中間変数 a に代入した後に、条件式で使います。</p>  <p><b>誤</b>  関数 temp_test の戻り値は、条件式で使用されます。</p> 
最終更新	V2.00

## 8.2. Stateflow data and operations

### 8.2.1. na\_0001: Bitwise Stateflow operators


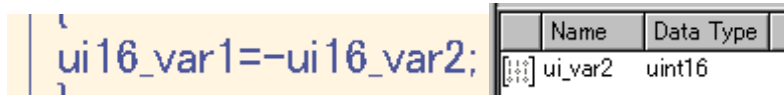
ID: タイトル	na_0001: ビット単位の Stateflow 演算子
重要度	強く推奨
Scope	MAAB
前提条件	
説明	<p>ビット単位の Stateflow 演算子 (&amp;   ^) は、ビット単位の演算が必要な場合を除き、Stateflow チャートでは使用しないでください。</p> <p>ビット単位の演算を有効にするには、以下の手順に従います。</p> <ol style="list-style-type: none"> <li>1. [ファイル]、[チャート プロパティ] を選択します。</li> <li>2. [C 言語のビット演算が可能] を選択します。</li> </ol>



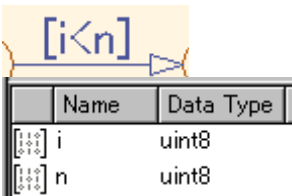
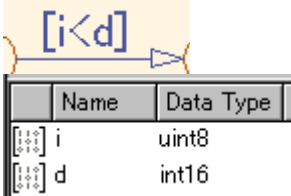
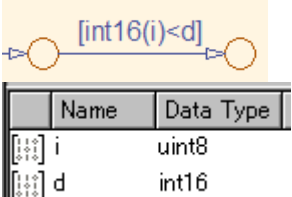
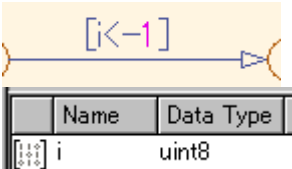
	<div> <b>Chart: C_Bit_Operations</b>  Name: <a href="#">C Bit Operations</a>  Machine: <a href="#">(machine) na_0001</a>  State Machine Type: <span>Classic</span>  Update method: <span>Inherited</span> Sample Time: <input type="text"/>  <input checked="" type="checkbox"/> Enable C-bit operations  <input checked="" type="checkbox"/> User specified state/transition execution order  <input type="checkbox"/> Export Chart Level Graphical Functions (Make Global)  <input checked="" type="checkbox"/> Use Strong Data Typing with Simulink I/O  <input type="checkbox"/> Execute (enter) Chart At Initialization  <input type="checkbox"/> Initialize Outputs Every Time Chart Wakes Up  <input type="checkbox"/> Enable Super Step Semantics  <input checked="" type="checkbox"/> Support variable-size arrays  Debugger breakpoint: <input type="checkbox"/> On chart entry <input type="checkbox"/> Lock Editor  Description: </div>																
	<div> <p><b>正</b> 論理演算には <code>&amp;&amp;</code> と <code>  </code> を使用してください。</p> <div>  <table border="1"> <thead> <tr> <th>Name</th><th>Data Type</th></tr> </thead> <tbody> <tr> <td>a</td><td>boolean</td></tr> <tr> <td>b</td><td>boolean</td></tr> <tr> <td>c</td><td>boolean</td></tr> </tbody> </table> </div> <p>ビット演算には <code>&amp;</code> と <code> </code> を使用してください。</p> <div>  <table border="1"> <thead> <tr> <th>Name</th><th>Data Type</th></tr> </thead> <tbody> <tr> <td>d</td><td>uint8</td></tr> <tr> <td>e</td><td>uint8</td></tr> <tr> <td>f</td><td>uint8</td></tr> </tbody> </table> </div> </div>	Name	Data Type	a	boolean	b	boolean	c	boolean	Name	Data Type	d	uint8	e	uint8	f	uint8
Name	Data Type																
a	boolean																
b	boolean																
c	boolean																
Name	Data Type																
d	uint8																
e	uint8																
f	uint8																
	<div> <p><b>誤</b> 論理演算には <code>&amp;</code> と <code> </code> を使用してください。</p> <div>  <table border="1"> <thead> <tr> <th>Name</th><th>Data Type</th></tr> </thead> <tbody> <tr> <td>a</td><td>boolean</td></tr> <tr> <td>b</td><td>boolean</td></tr> <tr> <td>c</td><td>boolean</td></tr> </tbody> </table> </div> </div>	Name	Data Type	a	boolean	b	boolean	c	boolean								
Name	Data Type																
a	boolean																
b	boolean																
c	boolean																
Rational	<div> <input checked="" type="checkbox"/> 可読性  <input type="checkbox"/> ワークフロー  <input type="checkbox"/> Simulation <input checked="" type="checkbox"/> 検証と確認  <input checked="" type="checkbox"/> コード生成 </div>																
最終更新	V2.20																

### 8.2.2. jc\_0451: Use of unary minus on unsigned integers in Stateflow

ID: タイトル jc\_0451: Stateflow の符号なし整数での単項マイナスの使用

重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>符号なし整数には単項マイナスを実行しないでください。</p> <p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.00

### 8.2.3. na\_0013: Comparison operation in Stateflow

ID: タイトル	na_0013: Stateflow での比較演算
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<ul style="list-style-type: none"> <li>比較は、同じデータ型の変数間のみで行わなければなりません。</li> <li>異なるデータ型の変数間で比較を行う場合、一致するデータ型に変数を明示的にキャストしなければなりません。</li> </ul> <p><b>正</b> "i" と "n" の同じデータ型</p>  <p><b>誤</b> "i" と "n" の異なるデータ型</p>  <p><b>正</b></p>  <p><b>誤</b></p>  <ul style="list-style-type: none"> <li>符号なしの整数と負数間の比較は行わないでください。</li> </ul>

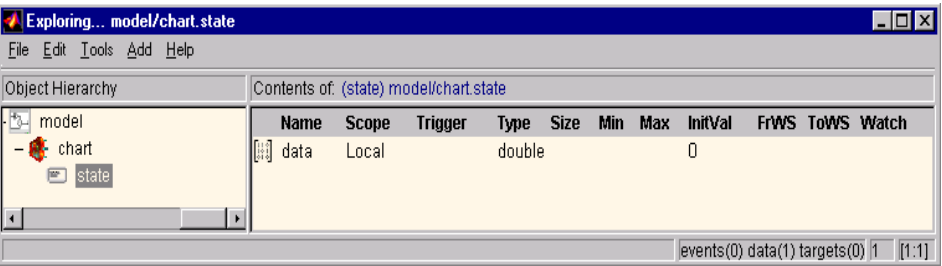
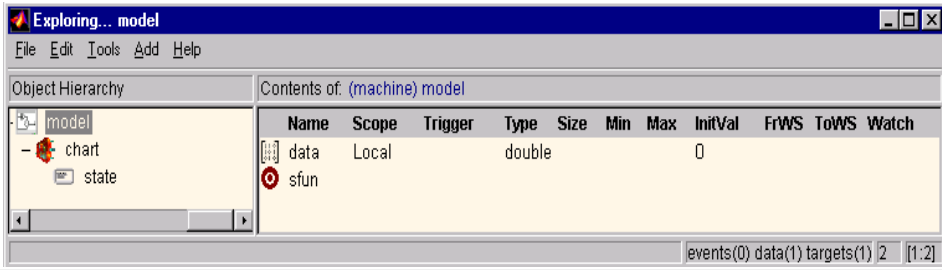
最終更新	V2.10
------	-------

#### 8.2.4. db\_0122: Stateflow and Simulink interface signals and parameters

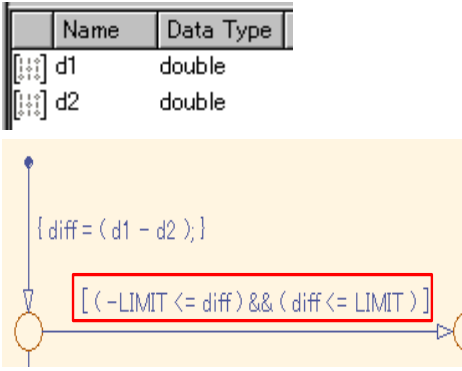
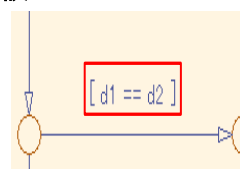
ID: タイトル	db_0122: Stateflow と Simulink を接続する信号とパラメータ
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>チャートは Simulink で厳密なデータ型指定を使用しますが、[Simulink I/O で厳密な型指定] パラメータを選択することが必要になります。</p> 
最終更新	V2.00

#### 8.2.5. db\_0125: Scope of internal signals and local auxiliary variables

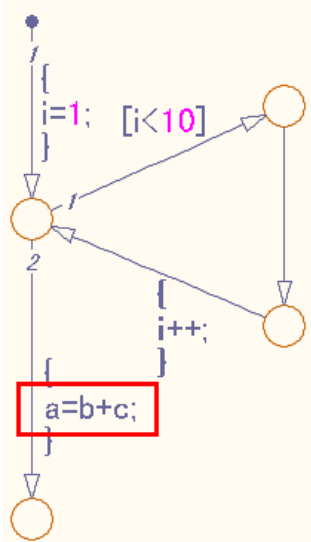
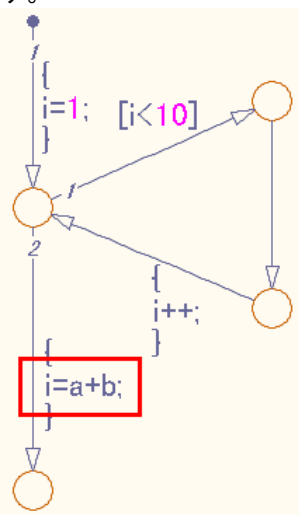
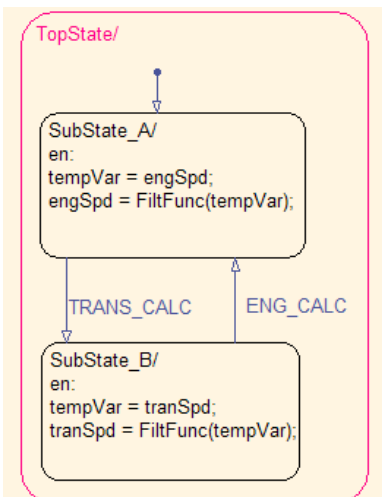
ID: タイトル	db_0125: Stateflow のローカル変数
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>内部信号とローカル補助変数は、Stateflow の "ローカル データ" で、以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● Stateflow ブロックのすべてのローカル データは、チャートレベルで、またはオブジェクト階層の下で定義しなければならない。</li> <li>● マシンレベルでローカル変数が存在することはない (つまり、異なるチャートのローカル データ間のやり取りはない)。</li> <li>● パラメータと定数は、マシンレベルで使用できる。</li> </ul>

最終更新	<p><b>正</b></p> 
	<p><b>誤</b></p> 
最終更新	V2.00

## 8.2.6. jc\_0481: Use of hard equality comparisons for floating point numbers in Stateflow

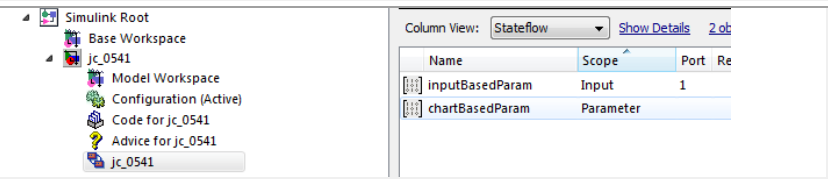
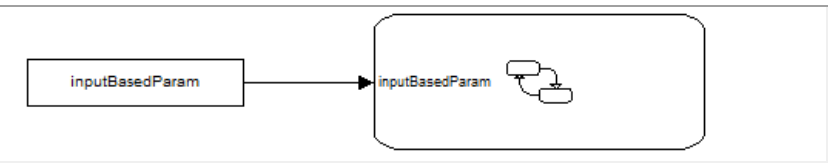
ID: タイトル	jc_0481: Stateflow での浮動小数点数の硬判定等価比較の使用
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<ul style="list-style-type: none"> <li>● 2つの浮動小数点数では、硬判定等価比較 (Var1 == Var2) を使用しないでください。</li> <li>● 硬判定比較が必要な場合は、許容誤差を定義して、比較で使用しなければなりません (例では LIMIT)。</li> <li>● 硬判定等価比較は、整数データ型間で実行できます。</li> </ul>
	<p><b>正</b></p>  <p><b>誤</b></p> 
最終更新	V2.00

## 8.2.7. jc\_0491: Reuse of variables within a single Stateflow scope

ID: タイトル	jc_0491: Stateflow における変数の再利用																							
重要度	推奨																							
Scope	MAAB																							
MATLAB Version	All																							
前提条件																								
説明	<p>1 つの Stateflow ステート内で同じ変数が複数の意味 (用途) をもつことはできません。</p> <div><div><p><b>正</b> ループ カウンターの変数は、ループ カウンター以外には使用してはなりません。</p></div><div><p><b>誤</b> 変数 i の意味は、ループ カウンターのインデックスから a+b の合計に変わります。</p></div></div>																							
	<p><b>正</b> tempVar は SubState_A と SubState_B の両方でローカル スコープとして定義されます。</p>  <div><p>Contents of: jc_0491/Chart/TopState/SubState_A</p><table><tr><th>Name</th><th>Scope</th><th>Port</th><th>Data Type</th><th>Mode</th><th>Data Type</th></tr><tr><td>tempVar</td><td>Local</td><td></td><td>Built-in</td><td></td><td>int32</td></tr></table><p>Contents of: jc_0491/Chart/TopState/SubState_B</p><table><tr><th>Name</th><th>Scope</th><th>Port</th><th>Data Type</th><th>Mode</th><th>Data Type</th></tr><tr><td>tempVar</td><td>Local</td><td></td><td>Built-in</td><td></td><td>int32</td></tr></table></div>	Name	Scope	Port	Data Type	Mode	Data Type	tempVar	Local		Built-in		int32	Name	Scope	Port	Data Type	Mode	Data Type	tempVar	Local		Built-in	
Name	Scope	Port	Data Type	Mode	Data Type																			
tempVar	Local		Built-in		int32																			
Name	Scope	Port	Data Type	Mode	Data Type																			
tempVar	Local		Built-in		int32																			

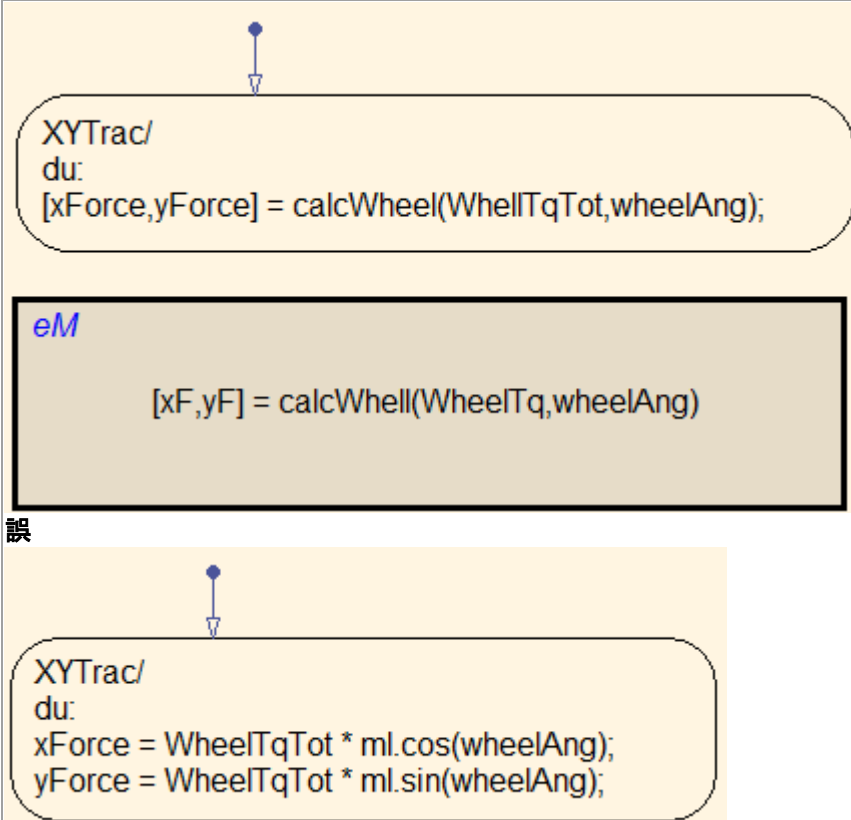
最終更新	V2.20
------	-------

### 8.2.8. jc\_0541: Use of tunable parameters in Stateflow

ID: タイトル	jc_0541: Stateflow での調整可能パラメータの使用
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>次のいずれかの方法で、Stateflow チャート内に調整可能なパラメータを作成します。</p> <ul style="list-style-type: none"> <li>● Stateflow チャート内のパラメータと、ベース ワークスペース内の対応するパラメータを定義する。</li> <li>● 調整可能なパラメータを Stateflow チャートへの入力として含める。このパラメータは、ベース ワークスペース内で定義しなければなりません。</li> </ul> <div> <div>ベース ワークスペースの定義</div>  </div> <div> <div>Stateflow チャートの定義</div>  </div> <div> <div>Stateflow チャート</div>  </div>
最終更新	V2.20

### 8.2.9. db\_0127: MATLAB commands in Stateflow

ID: タイトル	db_0127: MATLAB のコマンド Stateflow
重要度	必須
Scope	MAAB
MATLAB Version	All
前提条件	
説明	<p>Stateflow チャート内では、.ml 構文を使用しないでください。</p> <p>個々の会社が、MATLAB 関数の使用について決定しなければなりません。許可されている場合は、MATLAB 関数は MATLAB Function ブロックを通してアクセスされなければなりません。</p> <p><b>正</b></p>

	 <p>XYTrac/ du: [xForce,yForce] = calcWheel(WheelTqTot,wheelAng);</p> <p><i>eM</i></p> <p>[xF,yF] = calcWhell(WheelTq,wheelAng)</p> <p>誤</p> <p>XYTrac/ du: xForce = WheelTqTot * ml.cos(wheelAng); yForce = WheelTqTot * ml.sin(wheelAng);</p>
Note	コード生成 supports a limited subset of the MATLAB functions. For a complete list of the supported function, see the MathWorks documentation.
最終更新	V2.20

#### 8.2.10. jm\_0011: Pointers in Stateflow

ID: タイトル	<b>jm_0011: Stateflow のポインター</b>
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	
説明	Stateflow ダイアグラムでは、カスタム コード変数のポインターは使用できません。
最終更新	V1.00

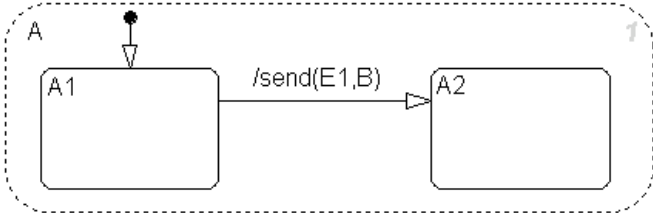
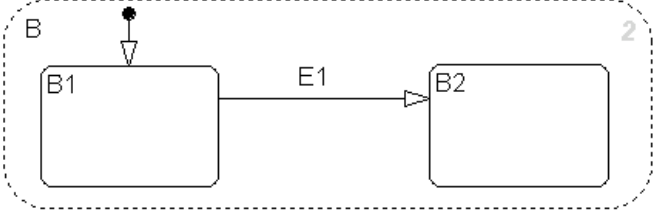
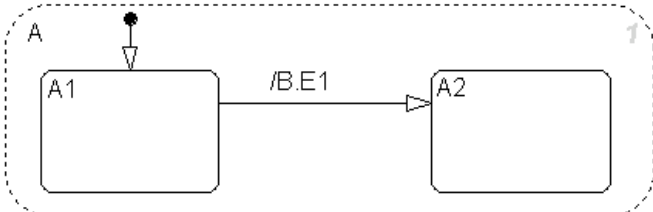
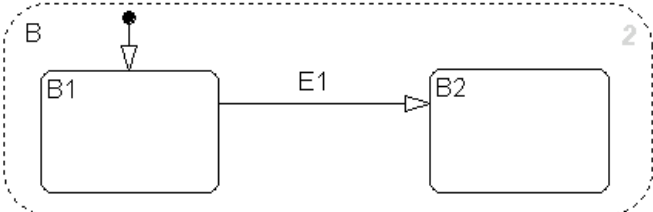
## 8.3. Events

#### 8.3.1. db\_0126: Scope of events

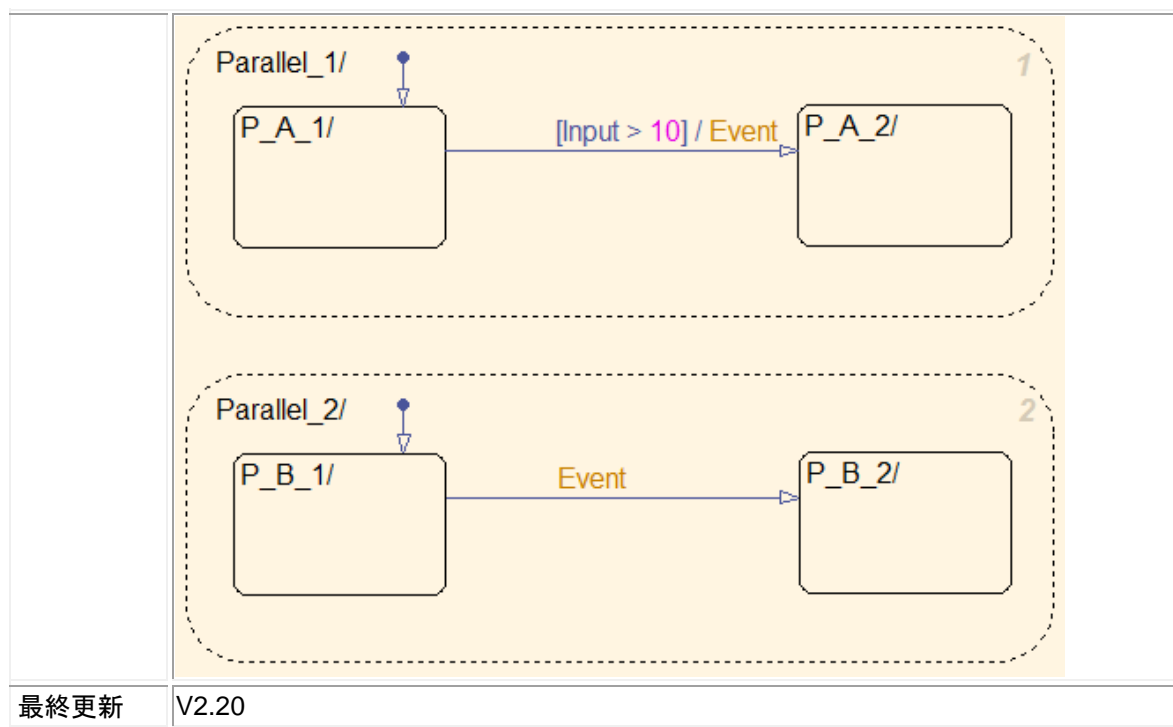
ID: タイトル	<b>db_0126: イベントの定義方法</b>
重要度	必須
Scope	MAAB
MATLAB Version	Pre R2009b
前提条件	
説明	<p>Stateflow のイベントには、以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● チャートのすべてのイベントは、チャート レベルまたはそれ以下のレベルで定義しな</li> </ul>

	<p>ければならない。</p> <ul style="list-style-type: none"> <li>● マシンレベルのイベントはない (つまり異なるチャート間でローカル イベントとのやり取りはない)。</li> </ul>
最終更新	V2.20

### 8.3.2. jm\_0012: Event broadcasts

ID: タイトル	<b>jm_0012: イベント ブロードキャスト</b>
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0126: Scope of events</a>
説明	<p>Stateflow のイベント ブロードキャストには、以下の規則が適用されます。</p> <ul style="list-style-type: none"> <li>● 使用できるイベント ブロードキャストの種類は、直接イベント ブロードキャストのみ。</li> <li>● send 構文または修飾イベント名は、イベントを特定のステートにするために使用する。</li> <li>● イベントを複数のステートにするには、複数の send ステートメントを使用しなければならない。</li> </ul> <p>正:Send 構文の使用例</p>   <p>正:修飾イベント名の使用例</p>   <p>誤:非直接イベントの使用</p>





## 8.4. Statechart Patterns

### 8.4.1. db\_0150: State machine patterns for conditions

ID: タイトル	db_0150: ステートの遷移条件パターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	Stateflow ステート マシン内の条件には次のパターンが使用されます。	
	等価な機能	ステートマシンパターン
	1つの条件  (condition)	
	条件 3 つまで、省略表記: (この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)  (condition1 && condition2) (condition1    condition2)	

	<p>2 つ以上の条件、複数行表記: サブ条件は、かっこで囲まれた、 すべて同じ型の論理演算のセット です。 (この形式では異なる演算子は使 用できません。代わりにサブ条件 を使用してください)</p> <p>(condition1 ... &amp;&amp; condition2 ... &amp;&amp; condition3) (condition1 ...    condition2 ...    condition3)</p>	
最終更新	V2.20	

#### 8.4.2. db\_0151: State machine patterns for transition actions

ID: タイトル	db_0151: ステートマシンの遷移アクションパターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	遷移アクションのステート マシン パターン	
	等価な機能	ステートマシンパターン
	<p>1 つの遷移アクション: <i>action</i>;</p> <p>2 つ以上の遷移アクシ ョン、複数行表記: (1 行に 2 つ以上の遷移 アクションを含めるこ とはできません) <i>action1</i>; <i>action2</i>; <i>action3</i>;</p>	
最終更新	V2.20	

## 8.5. Flowchart Patterns

The following rules illustrate sample patterns used in flow charts. As such they would normally be part of a much larger Stateflow diagram.

#### 8.5.1. db\_0148: Flowchart patterns for conditions

ID: タイトル	db_0148:フローチャート の遷移条件パターン
重要度	強く推奨
Scope	MAAB
MATLAB	All

Version											
前提条件											
説明	<p>Stateflow フローチャート内の条件には以下のパターンを使用します。</p> <table border="1"> <thead> <tr> <th>等価な機能</th><th>フローチャートパターン</th></tr> </thead> <tbody> <tr> <td> <p>1つの条件</p> <p><code>[condition]</code></p> </td><td>   <pre> graph LR     Start(( )) -- "[condition]" --&gt; End(( ))           </pre> <p><code>/* comment */</code>  <code>[condition]</code></p> </td></tr> <tr> <td> <p>条件 3 つまで、省略表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 &amp;&amp; condition2 &amp;&amp; condition3]</code>  <code>[condition1    condition2    condition3]</code></p> </td><td>   </td></tr> <tr> <td> <p>2 つ以上の条件、複数行表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 ... &amp;&amp; condition2 ... &amp;&amp; condition3]</code>  <code>[condition1 ...    condition2 ...    condition3]</code></p> </td><td>   </td></tr> <tr> <td> <p>サブ条件付きの条件:</p> <p>(サブ条件の接続には異なる論理演算子は使用できません。かつこの使用が必須です)</p> <p><code>[(condition1a    condition1b) ... &amp;&amp; (condition2a    condition2b) ... &amp;&amp; condition3]</code>  <code>[(condition1a &amp;&amp; condition1b) ...    (condition2a &amp;&amp; condition2b) ...    condition3]</code></p> </td><td>   </td></tr> </tbody> </table>	等価な機能	フローチャートパターン	<p>1つの条件</p> <p><code>[condition]</code></p>	 <pre> graph LR     Start(( )) -- "[condition]" --&gt; End(( ))           </pre> <p><code>/* comment */</code>  <code>[condition]</code></p>	<p>条件 3 つまで、省略表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 &amp;&amp; condition2 &amp;&amp; condition3]</code>  <code>[condition1    condition2    condition3]</code></p>	 	<p>2 つ以上の条件、複数行表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 ... &amp;&amp; condition2 ... &amp;&amp; condition3]</code>  <code>[condition1 ...    condition2 ...    condition3]</code></p>	 	<p>サブ条件付きの条件:</p> <p>(サブ条件の接続には異なる論理演算子は使用できません。かつこの使用が必須です)</p> <p><code>[(condition1a    condition1b) ... &amp;&amp; (condition2a    condition2b) ... &amp;&amp; condition3]</code>  <code>[(condition1a &amp;&amp; condition1b) ...    (condition2a &amp;&amp; condition2b) ...    condition3]</code></p>	 
等価な機能	フローチャートパターン										
<p>1つの条件</p> <p><code>[condition]</code></p>	 <pre> graph LR     Start(( )) -- "[condition]" --&gt; End(( ))           </pre> <p><code>/* comment */</code>  <code>[condition]</code></p>										
<p>条件 3 つまで、省略表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 &amp;&amp; condition2 &amp;&amp; condition3]</code>  <code>[condition1    condition2    condition3]</code></p>	 										
<p>2 つ以上の条件、複数行表記:</p> <p>(この形式では異なる論理演算子は使用できません。代わりにサブ条件を使用してください)</p> <p><code>[condition1 ... &amp;&amp; condition2 ... &amp;&amp; condition3]</code>  <code>[condition1 ...    condition2 ...    condition3]</code></p>	 										
<p>サブ条件付きの条件:</p> <p>(サブ条件の接続には異なる論理演算子は使用できません。かつこの使用が必須です)</p> <p><code>[(condition1a    condition1b) ... &amp;&amp; (condition2a    condition2b) ... &amp;&amp; condition3]</code>  <code>[(condition1a &amp;&amp; condition1b) ...    (condition2a &amp;&amp; condition2b) ...    condition3]</code></p>	 										

	<p>視覚的に分離している条件:</p> <p>(この形式は、前のパターンと組み合わせることができます)</p> <p><code>[condition1 &amp;&amp; condition2]</code>  <code>[condition1    condition2]</code></p>	
最終更新	V2.20	

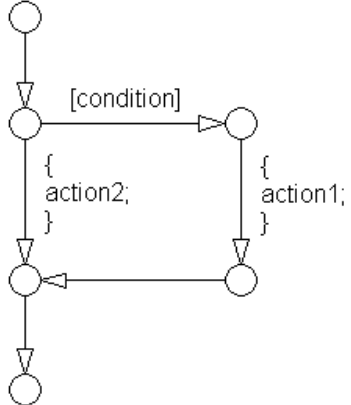
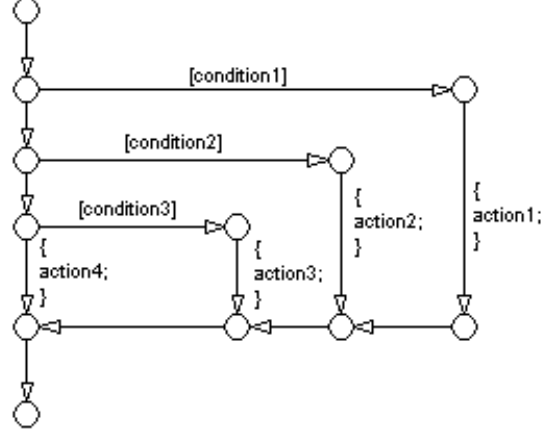
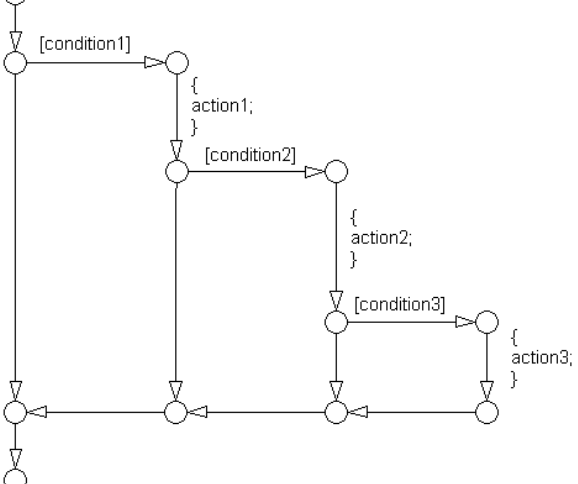
### 8.5.2. db\_0149: Flowchart patterns for condition actions

ID: タイトル	db_0149:フローチャート の条件アクションパターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件		
説明	Stateflow フローチャート内の条件アクションには以下のパターンを使用します。	
	等価な機能	フローチャートパターン
	<p>1 つの条件アクション:</p> <p><code>action;</code></p>	
説明	<p>2 つ以上の条件アクション、複数行表記:</p> <p>(1 行に 2 つ以上の条件アクションを含めることはできません)</p> <p><code>action1; ...</code>  <code>action2; ...</code>  <code>action3; ...</code></p>	

	<p>視覚的に分離している条件アクション:</p> <p>(この形式は、前のパターンと組み合わせることができます)</p> <pre> action1a; action1b; action2; action3; </pre>	
最終更新	V2.20	

### 8.5.3. db\_0134: Flowchart patterns for If constructs

ID: タイトル	db_0134: If 構文のフローチャート パターン	
重要度	強く推奨	
Scope	MAAB	
MATLAB Version	All	
前提条件	<a href="#">db_0148: Flowchart patterns for conditions</a> <a href="#">db_0149: Flowchart patterns for condition actions</a>	
説明	Stateflow フローチャート内の If 構文には以下のパターンを使用します。	
	<p>等価な機能</p> <pre> IF THEN if (condition){     action; } </pre>	<p>フローチャートパターン</p>

	<p><b>IF THEN ELSE</b>  if (condition) {    action1;  }  else {    action2;  }</p>	
	<p><b>IF THEN ELSE IF</b>  if (condition1) {    action1;  }  else if (condition2) {    action2;  }  else if (condition3) {    action3;  }  else {    action4;  }</p>	
	<p><b>Cascade of IF THEN</b>  if (condition1) {    action1;    if (condition2) {      action2;      if (condition3) {        action3;      }    }  }</p>	
最終更新	V1.00	

#### 8.5.4. db\_0159: Flowchart patterns for case constructs

ID: タイトル	db_0159: case 構文のフローチャート パターン
重要度	強く推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0148: Flowchart patterns for conditions</a> <a href="#">db_0149: Flowchart patterns for condition actions</a>
説明	Stateflow フローチャート内の case 構文には以下のパターンを使用しなければなりません。

	等価な機能	フローチャートパターン
	<p>排他的選択をもつ case</p> <pre> selection = ...; switch (selection) {   case 1:     action1;     break;   case 2:     action2;     break;   case 3:     action3;     break;   default:     action4; } </pre>	
	<p>排他的条件をもつ case</p> <pre> conditions c1 = condition1; c2 = condition2; c3 = condition3; if (c1 &amp;&amp; !c2 &amp;&amp; !c3) {   action1; } else if (!c1 &amp;&amp; c2 &amp;&amp; !c3) {   action2; } else if (!c1 &amp;&amp; !c2 &amp;&amp; c3) {   action3; } else {   action4; } </pre>	
最終更新	V1.00	

#### 8.5.5. db\_0135: Flowchart patterns for loop constructs

ID: タイトル	db_0135: ループ構文のフローチャート パターン
重要度	推奨
Scope	MAAB
MATLAB Version	All
前提条件	<a href="#">db_0148: Flowchart patterns for conditions</a> <a href="#">db_0149: Flowchart patterns for condition actions</a>
説明	The following patterns must be used to create Loops within Stateflow Flowcharts:

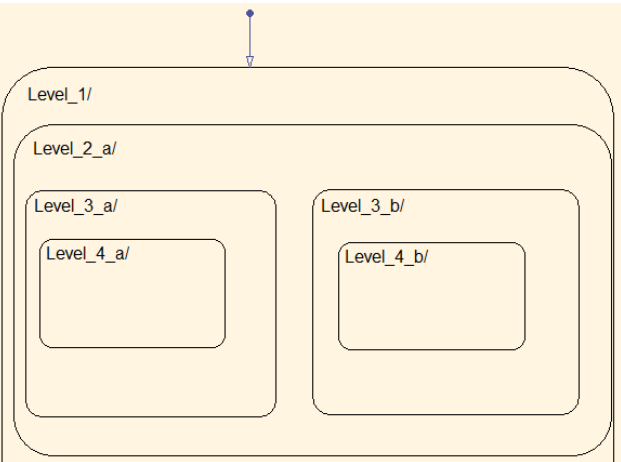
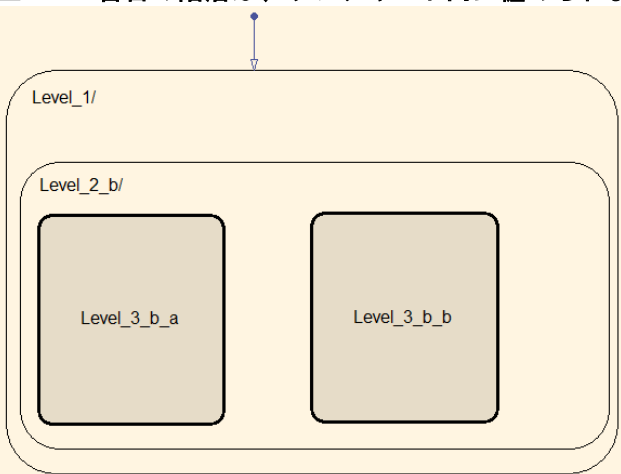
	等価な機能	フローチャートパターン
	<p>FOR ループ</p> <pre>for (index=0;index&lt;number_of_loops;index++) {   action; }</pre>	
	<p>WHILE ループ</p> <pre>while (condition) {   action; }</pre>	
	<p>DO WHILE ループ</p> <pre>do {   action; } while (condition);</pre>	
最終更新	V1.00	

## 8.6. State chart architecture

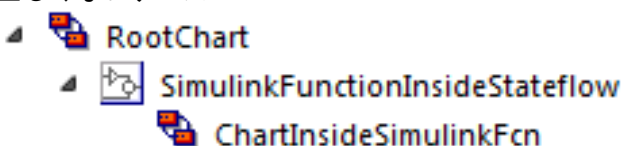
### 8.6.1. na\_0038: Levels in Stateflow charts

ID: タイトル	na_0038: Levels in Stateflow charts
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
Prerequisite	

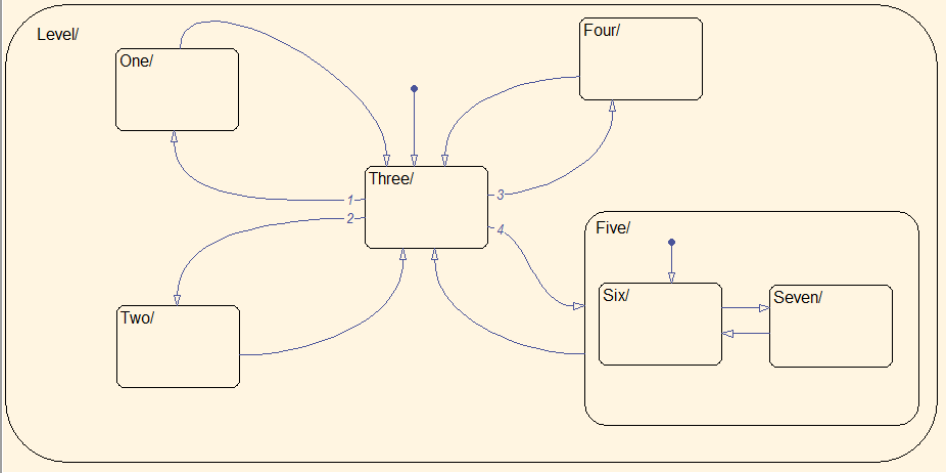


説明	<p>状態の階層化は限定されるべきです。一般的には、3階層です。さらなる階層が必要な場合、サブチャートを使ってください。</p> <p>正しくないケース： Level_4_a と Level_4_b は 3 の奥にネストされます。</p>  <p>正： 4 番目の階層は、サブチャート内に纏められます。</p> 
最終更新	V3.00

#### 8.6.2. na\_0039: Use of Simulink in Stateflow charts

ID: タイトル	na_0039: Use of Simulink in Stateflow charts
重要度	推奨
Scope	NA-MAAB
MATLAB Version	2010B and Later
Prerequisite	
説明	<p>Stateflow チャートに含まれる Simulink 関数の中に、Stateflow チャートを入れてはいけません。</p> <p>正しくないケース：</p> 
最終更新	V3.00

### 8.6.3. na\_0040: Number of states per container

ID: タイトル	na_0040: Number of states per container
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
Prerequisite	
説明	<p>状態に入れる状態数は制限されるべきです。通常 6 から 10 に制限されます。 数は、ダイアグラム内で見える状態に基づきます。</p> <p>正：</p> 
Note	A container is either a State, Box or root level chart.
最終更新	V3.00

### 8.6.4. na\_0041: Selection of function type

ID: タイトル	na_0041: Selection of function type
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
Prerequisite	
説明	<p>Stateflow は、3 種類の関数をサポートします： グラフィック、MATLAB、Simulink です。 適切な関数は、必要とされる作業に依存します：</p> <ul style="list-style-type: none"> <li>• Simulink <ul style="list-style-type: none"> <li>➢ 伝達関数</li> <li>➢ 積分器</li> <li>➢ ルックアップテーブル</li> </ul> </li> <li>• MATLAB <ul style="list-style-type: none"> <li>➢ 複雑な方程式</li> <li>➢ If Then Else ロジック</li> </ul> </li> <li>• グラフィック関数 <ul style="list-style-type: none"> <li>➢ If Then Else ロジック</li> </ul> </li> </ul>
最終更新	V3.00

### 8.6.5. na\_0042: Location of Simulink functions

ID: タイトル	na_0042: Location of Simulink functions
重要度	推奨
Scope	NA-MAAB
MATLAB Version	All
Prerequisite	<a href="#">na_0039: Use of Simulink in Stateflow charts</a>
説明	<p>Stateflow チャートに Simulink 関数を組み込むべきかどうか決める際、以下のいずれかの条件が好ましい。</p> <p>Simulink 関数が：</p> <ul style="list-style-type: none"> <li>➤ ローカル Chart データだけを使う、</li> <li>➤ ローカル Chart データと Simulink 入力を使う、</li> <li>➤ チャートの範囲内で複数の場所から呼ばれる、</li> <li>➤ 全ての時間ステップでは呼び出されない。</li> </ul>
最終更新	V3.00

## 9.Enumerated Data

### 9.1.1. na\_0033: Enumerated Types Usage

ID: タイトル	na_0033: Enumerated Types Usage
重要度	推奨
Scope	NA-MAAB
MATLAB Version	R2010b and later
前提条件	<a href="#">na_0002: Appropriate implementation of fundamental logical and numerical operations</a>
説明	<p>列挙型は、信号またはパラメータが整数の有限値で各値に名前が付けられる時に使用しなければなりません。名前はリテラルと呼ばれ、アルゴリズムや操作対象に応じた意味を持ちます。一般的に、これらのリテラルは操作モード、信号ステータス、ビルド変化、またはその変数が意味する離散的な値を表します。</p> <p>典型的な例は自動車のトランスミッションモードです（パーキング、リバース、ニュートラル、ドライブ、ロー）。</p> <p>プロジェクト内で、複数の列挙型で同じリテラルが定義されていないことを保証するコードビルドプロセスが提供されていなければなりません。</p>
See also	dm_0002: Enumerated type usage
最終更新	V3.00

### 9.1.2. na\_0031: Definition of default enumerated value

ID: タイトル	na_0031: Definition of default enumerated value
重要度	推奨
Scope	NA-MAAB
MATLAB Version	R2010b and later
前提条件	
説明	列挙型のデフォルト値は、常に明示的に定められなければならない。
最終更新	V3.00

## 10.MATLAB Functions

### 10.1. MATLAB Function Appearance

#### 10.1.1. na\_0018: Number of nested if/else and case statement

ID: タイトル	na_0018: Number of nested if/else and case statement
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	ネストされた if/else のレベルの数とケース文は、制限されなければなりません。通常、3レベルに制限されます。
See also	jr_0002: Number of nested if/else and case statement blocks
最終更新	V3.00

#### 10.1.2. na\_0019: Restricted Variable Names

ID: タイトル	na_0019: Restricted Variable Names
重要度	必須
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>MATLAB コードの可読性を改善するために、C の予約変数名（例えば const、TRUE、FALSE、infinity、nil、double、single、or enum）は、MATLAB Function コードで使用することは避けてください。これらは、MATLAB コードから C コードが生成された後、コンパイラと衝突するかもしれません。</p> <p>「conv」のような MATLAB 関数と衝突するような変数名を使用することは避けてください。</p>
Note	Reserved key words are defined in Simulink Coder > User's Guide > コード生成 > Configuration > Code Appearance.
See also	Derived from jh_0021: Restricted Variable Names
根拠	<div><input checked="" type="checkbox"/> 可読性</div> <div><input checked="" type="checkbox"/> 検証と確認</div> <div><input type="checkbox"/> ワークフロ</div> <div><input type="checkbox"/> コード生成</div> <div>—</div> <div><input type="checkbox"/> Simulation</div>
最終更新	V3.00

#### 10.1.3. na\_0025: MATLAB Function Header

ID: タイトル	na_0025: MATLAB Function Header
重要度	強く推奨
Scope	NA-MAAB
MATLAB	All

Version	
前提条件	
説明	<p>MATLAB Function ブロックは説明用ヘッダを持っていないけません。ヘッダ内容は以下の情報を含みますが、これに限られているわけではありません。</p> <ul style="list-style-type: none"> <li>・ 関数名</li> <li>・ 関数の説明</li> <li>・ 前提と制約</li> <li>・ 前バージョンからの変更点説明</li> <li>・ 入力と出力のリスト</li> </ul> <p>例 :</p> <pre>%% Function Name: NA_0025_Example_Header % % Description: An example of a header file % % Assumptions: None % % Inputs: %   List of input arguments % % Outputs: %   List of output arguments % % \$Revision: 3.0\$ % \$Author: MAAB\$ % \$Date: July 24,2012\$ % %-----</pre>
See also	jh_0073: eML Header version
最終更新	V3.00

## 10.2. MATLAB Function Data and Operations

### 10.2.1. na\_0034: MATLAB Function block input/output settings

ID: タイトル	<b>na_0034: MATLAB Function block input/output settings</b>
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>MATLAB Function ブロックへのすべての入力と出力は、Model Explorer 内かまたは関数の冒頭で明示的にデータタイプを定義しなければなりません。これはより厳しいデータタイプチェックを MATLAB Function ブロックに課し、assert 文を用いる必要性を無くします。</p>
See also	jh_0063: eML block input / output settings
最終更新	V3.00

### 10.2.2. na\_0024: Global Variables

ID: タイトル	na_0024: Global Variables
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>共通データにアクセスする好ましい方法は、信号線を使うことです。しかし、必要であれば、グローバルメモリを表すためにデータストアメモリはを用いても構いません。</p> <p>例: この例では、同じデータストアメモリ (ErrorFlag_DataStore) が 2 つの別々の MATLAB 関数内で書かれています。</p> <pre>function EngineFaultEvaluation(EngineData) %#codegen     global ErrorFlag_DataStore     if (EngineData.RPM_HIGH)         ErrorFlag_DataStore = bitor(ErrorFlag_DataStore,HIGHRPMFAULT);     end      if (EngineData.RPM_LOW)         ErrorFlag_DataStore = bitor(ErrorFlag_DataStore,LOWRPMFAULT);     end  end  function WheelFaultEvaluation(WheelData) %#codegen     global ErrorFlag_DataStore     if (WheelData.SlipHigh)         ErrorFlag_DataStore = bitor(ErrorFlag_DataStore,WHEELSLIP);     end      if (WheelData.SlipHigh)         ErrorFlag_DataStore = bitor(ErrorFlag_DataStore,LOWRPMFAULT);     end  end</pre>
See also	ek_0003: Global Variables
最終更新	V3.00

## 10.3. MATLAB Function Patterns

### 10.3.1. na\_0022: Recommended patterns for Switch / Case statements

ID: タイトル	na_0022: Recommended patterns for Switch / Case statements
重要度	必須
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>Switch/Case 文の Case 値には定数値を用いなければなりません。入力変数を Case 値に用いることはできません。</p> <p>正:</p>

	<pre> function outVar = NA_0022_Pass(SwitchVar) %#codegen     switch SwitchVar         case Case_1_Parameter % Parameter             outVar = 0;         case NA_0022.Case_2 % Enumerated Data type             outVar = 1;         case 3 % Hard Code Value             outVar = 2;         otherwise             outVar = 10;     end end </pre> <p>誤</p> <pre> function outVar = NA_0022_Fail(Case_1,Case_2,Case_3,SwitchVar) %#codegen     switch SwitchVar         case Case_1             outVar = 1;         case Case_2             outVar = 2;         case Case_3             outVar = 3;         otherwise             outVar = 10;     end end </pre>
See also	jh_0026: Switch / Case statement
最終更新	V3.00

## 10.4. MATLAB Function Usage

### 10.4.1. na\_0016: Source lines of MATLAB Functions

ID: タイトル	na_0016: Source lines of MATLAB Functions
重要度	必須
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>MATLAB 関数の行数は制限されるべきです（推奨行数は 60 行です）。この制限は、MATLAB Function ブロックおよび外部の M-ファイルに適用されます。</p> <p>サブ関数を使う場合、サブ関数の長さも 60 行に制限してください。</p>
See also	IM_0008: Source lines of eML
最終更新	V3.00

### 10.4.2. na\_0017: Number of called function levels

ID: タイトル	na_0017: Number of called function levels
----------	---



重要度	必須
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>サブ関数の階層数は通常は 3 階層に制限されなければなりません。単に外部の M-ファイル用ラッパーでない限り、MATLAB Function ブロックを最初の階層としてカウントします。</p> <p>これは MATLAB Function ブロックと、別々の M-ファイルの中で定義される関数を含みます。標準関数（例えば sqrt や log のような組込み機能）は階層数には含まれません。同様に、一般的に用いられる汎用関数は階層数から除外することができます。</p>
Note	Standard utility functions, such as built in functions like sqrt or log, are not included in the number of levels. Likewise, commonly used custom utility functions can be excluded from the number of levels.
See also	im_0009: Number of called function levels
最終更新	V3.00

#### 10.4.3. na\_0021: Strings

ID: タイトル	<b>na_0021: Strings</b>
重要度	強く推奨
Scope	NA-MAAB
MATLAB Version	All
前提条件	
説明	<p>文字列の使用はお勧めしません。 MATLAB Function ブロックは、文字列を文字配列として保存します。動的なメモリ配置が無いため、文字配列は異なる長さの文字列に合うようにサイズを調節することができません。文字列は Simulink がサポートしているデータ型ではないため、MATLAB Function ブロックからブロックの外に文字列データを渡すことができません。</p> <p>例えば、以下のコードではエラーが生じます：</p> <pre><i>name = 'rate_error'; %this creates a 1 x 10 character array</i> name = 'x_rate_error'; %this causes an error because the array size is now 1 x 12, not 1 x 10</pre>
Note	もし文字列を Switch/Case 文に使用したいなら、列挙型の利用を検討して下さい。
See also	jh_0024: Strings
最終更新	V3.00

## 11. 根拠、関連まとめ

### 11.1. 根拠分類

根拠には、ガイドラインを推奨する以下の理由の内、1つ以上が選択されている。

1. 可読性の向上
  - 可読性の確保
  - モデル、コード、ドキュメントの体裁統一
  - インターフェースの明確化
  - 完成度が高いドキュメントの作成
2. プロセス / ワークフローに効果的
  - 保守性の向上
  - 仕様変更の迅速化
  - 再利用性の向上
  - モデル交換の容易化
  - 移植性の向上
3. 解析とシミュレーションが効率的
  - シミュレーションの高速化
  - シミュレーション使用メモリの削減
  - 計測環境へのモデル接続の容易化
4. 検証が効率的
  - 機能要求に対するトレーサビリティの確保
  - テストの容易化
  - システム結合の容易化
  - インターフェースの明確化
5. 組込みコード生成に効果的 / 効率的
  - 仕様変更の容易化
  - 生成コードのロバスト性確保

### 11.2. 一覧表

章番号	Guideline ID	適応範囲	1	2	3	4	5	関連
3.1.1	na_0026	NA-MAAB	○				○	jh_0042
3.1.2	na_0027	NA-MAAB	○		○	○	○	hyl_0201
4.1.1.	ar_0001	MAAB	○	○	○		○	
4.1.2.	ar_0002	MAAB	○	○	○		○	
4.1.3.	na_0035	NA-MAAB	○	○	○		○	
4.2.1.	jc_0201	MAAB	○					
4.2.2.	jc_0211	MAAB	○					
4.2.3.	jc_0221	MAAB	○					
4.2.4.	na_0030	NA-MAAB	○					jh_0040
4.2.5.	jc_0231	MAAB	○					
4.2.6.	na_0014	J-MAAB	○					
5.1.1.	na_0006	MAAB	○	○	○	○	○	
5.1.2.	na_0007	MAAB	○	○	○	○	○	
5.2.1.	db_0143	NA-MAAB	○	○		○	○	
5.2.2.	db_0144	MAAB	○	○		○	○	
5.2.3.	db_0040	MAAB	○	○		○	○	
5.2.4.	na_0037	NA-MAAB	○		○		○	na_0036
5.2.5.	na_0020	NA-MAAB	○		○		○	
5.2.6.	na_0036	NA-MAAB						
5.3.1.	jc_0301	J-MAAB		○				

5.3.2.	jc_0311	J-MAAB		○				
5.3.3.	jc_0321	J-MAAB	○	○			○	
5.3.4.	jc_0331	J-MAAB	○	○			○	
5.3.5.	jc_0341	J-MAAB		○				
6.1.1.	jc_0011	MAAB		○			○	
6.1.2.	jc_0021	MAAB		○			○	
7.1.1.	na_0004	MAAB	○					
7.1.2.	db_0043	MAAB	○					
7.1.3.	db_0042	MAAB	○					
7.1.4.	na_0005	MAAB	○					
7.1.5.	jc_0081	MAAB	○					
7.1.6.	jm_0002	MAAB	○					
7.1.7.	db_0142	MAAB	○					
7.1.8.	jc_0061	MAAB	○					
7.1.9.	db_0146	MAAB	○					
7.1.10.	db_0140	MAAB	○					
7.1.11.	db_0032	MAAB	○					
7.1.12.	db_0141	MAAB	○					
7.1.13.	jc_0171	MAAB	○					
7.1.14.	na_0032	NA-MAAB		○			○	jh_0109
7.1.15.	jm_0010	MAAB	○					
7.1.16.	jc_0281	J-MAAB	○				○	
7.2.1.	na_0008	MAAB	○	○	○	○	○	
7.2.2.	na_0009	MAAB	○	○	○	○	○	
7.2.3.	db_0097	MAAB	○					
7.3.1.	na_0003	MAAB	○	○			○	
7.3.2.	na_0002	MAAB	○	○		○	○	
7.3.3.	jm_0001	MAAB	○	○	○	○	○	
7.3.4.	hd_0001	MAAB			○	○	○	
7.3.5.	na_0011	MAAB	○	○	○	○	○	
7.3.6.	jc_0141	MAAB	○	○		○	○	
7.3.7.	jc_0121	MAAB	○					
7.3.8.	jc_0131	J-MAAB	○					
7.3.9.	jc_0161	J-MAAB	○	○				
7.4.1.	db_0112	MAAB	○			○	○	cgsI_0101 hisI_0021
7.4.2.	na_0010	MAAB	○				○	
7.4.3.	db_0110	MAAB	○	○	○	○	○	
7.5.1.	na_0012	MAAB	○			○	○	
7.5.2.	db_0114	MAAB	○					
7.5.3.	db_0115	MAAB	○					
7.5.4.	na_0028	NA-MAAB	○					bn_0003
7.5.5.	db_0116	MAAB	○					
7.5.6.	db_0117	MAAB	○			○	○	
7.5.7.	jc_0351	MAAB	○				○	
7.5.8.	jc_0111	J-MAAB	○					
8.1.1.	db_0123	MAAB	○				○	
8.1.2.	db_0129	MAAB	○					
8.1.3.	db_0137	MAAB	○	○		○	○	
8.1.4.	db_0133	MAAB	○					

8.1.5.	db_0132	MAAB	○					
8.1.6.	jc_0501	MAAB	○					
8.1.7.	jc_0511	J-MAAB	○			○	○	
8.1.8.	jc_0531	J-MAAB	○					
8.1.9.	jc_0521	J-MAAB	○			○	○	
8.2.1.	na_0001	MAAB	○			○	○	
8.2.2.	jc_0451	MAAB				○	○	
8.2.3.	na_0013	MAAB			○	○	○	
8.2.4.	db_0122	MAAB			○	○	○	
8.2.5.	db_0125	MAAB	○				○	
8.2.6.	jc_0481	MAAB				○	○	
8.2.7.	jc_0491	MAAB	○			○	○	
8.2.8.	jc_0541	MAAB				○	○	
8.2.9.	db_0127	MAAB			○	○	○	
8.2.10.	jm_0011	MAAB	○			○	○	
8.3.1.	db_0126	MAAB	○	○	○	○	○	
8.3.2.	jm_0012	MAAB	○	○	○	○	○	
8.4.1.	db_0150	MAAB	○					
8.4.2.	db_0151	MAAB	○	○	○	○	○	
8.5.1.	db_0148	MAAB	○					
8.5.2.	db_0149	MAAB	○					
8.5.3.	db_0134	MAAB	○	○	○	○	○	
8.5.4.	db_0159	MAAB	○					
8.5.5.	db_0135	MAAB	○					
8.6.1.	na_0038	NA-MAAB	○					
8.6.2.	na_0039	NA-MAAB	○			○	○	
8.6.3.	na_0040	NA-MAAB	○			○	○	
8.6.4.	na_0041	NA-MAAB		○			○	
8.6.5.	na_0042	NA-MAAB	○	○				
9.1.1.	na_0033	NA-MAAB	○	○	○	○	○	dm_0002
9.1.2.	na_0031	NA-MAAB	○			○	○	
10.1.1.	na_0018	NA-MAAB	○				○	jr_0002
10.1.2.	na_0019	NA-MAAB	○			○		jh_0021
10.1.3.	na_0025	NA-MAAB	○	○		○	○	jh_0073
10.2.1.	na_0034	NA-MAAB	○	○		○	○	jh_0063
10.2.2.	na_0024	NA-MAAB	○		○	○	○	ek_0003
10.3.1.	na_0022	NA-MAAB			○	○	○	jh_0026
10.4.1.	na_0016	NA-MAAB	○	○		○	○	IM_0008
10.4.2.	na_0017	NA-MAAB	○			○		im_0009
10.4.3.	na_0021	NA-MAAB		○		○	○	jh_0024

## 12.付録 A: 自動化ツールに対する提言

以下の提言は、本ガイドラインの自動チェックツールを開発するすべての企業に向けたものである。本ガイドラインは、MAABによって作成された。MAABは、ツールベンダが本ガイドラインに反するモデル記述をチェックするチェッカーを作成することを期待する。また、MAABは、ツールベンダからのコンプライアンスマトリックスの提供を期待する。

コンプライアンスマトリックスは、以下の情報を含む。

- 対象とするドキュメントのタイトルとバージョン
- 各ガイドラインに関する以下の情報
  - ガイドラインの ID
  - ガイドラインのタイトル
  - 順守レベル
  - 詳細

ガイドラインの ID とタイトルは、本ドキュメントと同一であること。また、順守レベルは、以下のいずれかであること。

- 修正：ツールが、非順守部分をチェックし、自動または半自動で修正する
- チェック：ツールが、非順守部分をチェックし、明示化する。修正は、ユーザが行う
- 部分：ツールが、ガイドラインを部分的にチェックする  
詳細部で、ガイドライン内でチェック可能な部分とチェック不可能な部分を明確化する
- 無し：ツールでは、ガイドラインをチェックできない  
ユーザがマニュアルでチェックする推奨方法をツールベンダが提供することを期待する

## 13. 付録 B: ユーザによるガイドライン追加時の注意点

新しいガイドラインを追加する際に重要な項目は以下である。

- わかりやすく、明確であること
- 簡単に見たい項目を探せること
- 内容を最小限にすること

これらの項目を満たしているガイドラインは、わかりやすく使いやすい。

“わかりやすく、明確であること”とは...

ガイドラインの説明が正確で、はっきりとした文章で書かれ、簡潔に、モデル（またはモデルの一部）の特徴を定義しているということである。ルールを遵守するための手段（特別なオプションを選択する、あるボタンをクリックするなど）を記述することはガイドラインの利用者にとって非常に有益である。また、例や反例などをダイアグラムやスクリーンショット等を用いて記述することを推奨する。なお、ガイドラインの例外は内容があいまいになるため、最小限とすべきである。ガイドラインに多くの例外がある場合は、1つのガイドラインで多くのルールを規定しようとしている可能性があるため、“内容を最小限にすること”が必要である。

“簡単に見たい項目を探せること”とは...

明確で不変なタイトルとし、適切にカテゴリ化することである。タイトルは、ルールそのものではなく、ルールの対象を説明したものが良い。そうすることにより、将来に渡りタイトルを変更する必要がなく、簡単に検索することができる。ルールそのものはガイドラインの記述内容に入れるべきである。例えば、名前に使用できる文字列について取り上げている場合、タイトルは“名前に使用できる文字列”のようなものとし、記述内容に、実際に使用できる文字列と使用できない文字列を記述する。ガイドラインに前提条件がある場合、前提条件となるガイドラインはそのガイドラインよりも前に記述すべきである。（前提条件が別のセクションにある場合は、この限りではない。）

“内容を最小限にすること”とは...

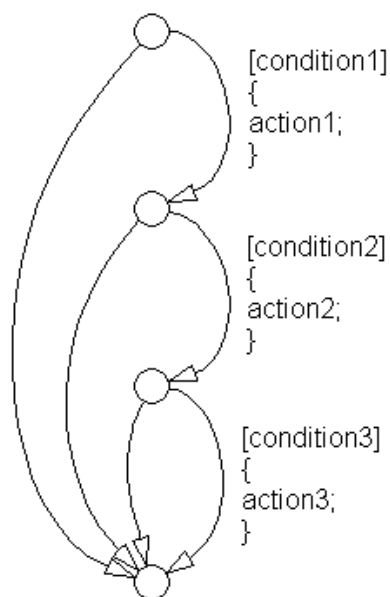
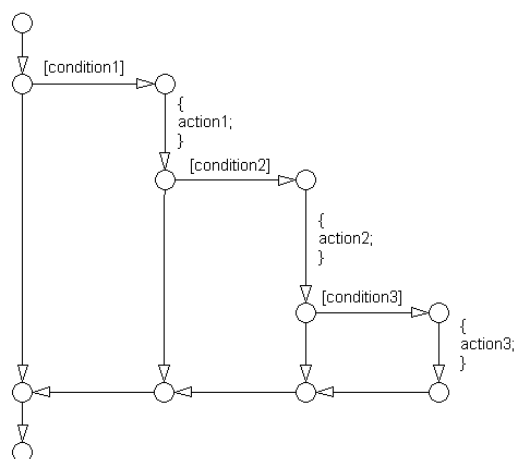
1つのガイドラインには1つの特性のみを記述するということである。記述する内容は最小限とすべきである。例えば、エラー防止と可読性について取り上げている1つの大きなガイドラインを作成するのではなく、エラー防止と可読性のそれぞれについて2つのガイドラインを作成すべきである。そして、必要であれば前提条件の項にもう一方のガイドラインを記述すればよい。また、最小限のガイドラインに比べ、大きなガイドラインでは、多くのユーザに受け入れられるために内容を妥協しなければならない場合がある。そのため、内容が弱く、具体性に欠け、ガイドラインとしてあまり役に立たない場合がある。最小限のガイドラインは、妥協により変更することも少なく、利用しやすい。

## 14. 付録 C: フローチャートの使用例

if 構文には以下のパターンを使用する。

直線で記述するフローチャートパターン	曲線で記述するフローチャートパターン
if-then 構文	
<pre> graph TD     Start(( )) --&gt; Decision{ }     Decision -- "[condition]" --&gt; ActionBlock["{ action; }"]     Decision --&gt; NextBlock[" "]     ActionBlock --&gt; NextBlock     NextBlock --&gt; End(( ))         </pre>	<pre> graph TD     Start(( )) --&gt; Decision{ }     Decision -- "[condition]" --&gt; ActionBlock["{ action; }"]     Decision --&gt; NextBlock[" "]     ActionBlock --&gt; NextBlock     NextBlock --&gt; End(( ))         </pre>
if-then-else 構文	
<pre> graph TD     Start(( )) --&gt; Decision{ }     Decision -- "[condition]" --&gt; Action1Block["{ action1; }"]     Decision --&gt; Action2Block["{ action2; }"]     Action1Block --&gt; Merge(( ))     Action2Block --&gt; Merge     Merge --&gt; End(( ))         </pre>	<pre> graph TD     Start(( )) --&gt; Decision{ }     Decision -- "[condition]" --&gt; Action1Block["{ action1; }"]     Decision --&gt; Action2Block["{ action2; }"]     Action1Block --&gt; Merge(( ))     Action2Block --&gt; Merge     Merge --&gt; End(( ))         </pre>
if-then-else-if 構文	
<pre> graph TD     Start(( )) --&gt; Decision1{ }     Decision1 -- "[condition1]" --&gt; Action1Block["{ action1; }"]     Decision1 --&gt; Decision2{ }     Decision2 -- "[condition2]" --&gt; Action2Block["{ action2; }"]     Decision2 --&gt; Decision3{ }     Decision3 -- "[condition3]" --&gt; Action3Block["{ action3; }"]     Decision3 --&gt; Action4Block["{ action4; }"]     Action1Block --&gt; Merge1(( ))     Action2Block --&gt; Merge1     Action3Block --&gt; Merge1     Action4Block --&gt; Merge1     Merge1 --&gt; End(( ))         </pre>	<pre> graph TD     Start(( )) --&gt; Decision1{ }     Decision1 -- "[condition1]" --&gt; Action1Block["{ action1; }"]     Decision1 --&gt; Decision2{ }     Decision2 -- "[condition2]" --&gt; Action2Block["{ action2; }"]     Decision2 --&gt; Decision3{ }     Decision3 -- "[condition3]" --&gt; Action3Block["{ action3; }"]     Decision3 --&gt; Action4Block["{ action4; }"]     Action1Block --&gt; Merge1(( ))     Action2Block --&gt; Merge1     Action3Block --&gt; Merge1     Action4Block --&gt; Merge1     Merge1 --&gt; End(( ))         </pre>

## if-then 構文のネスト

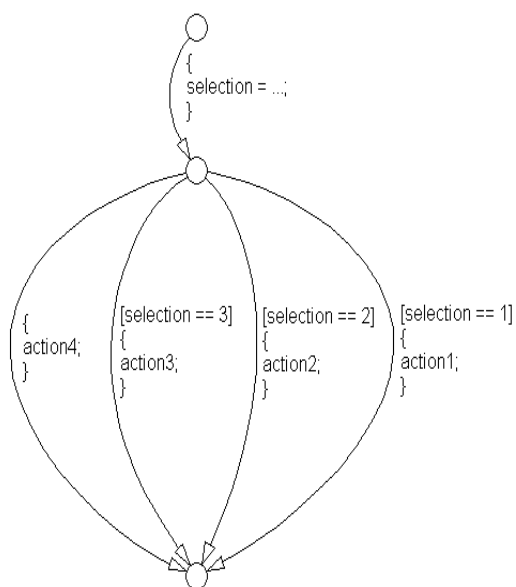
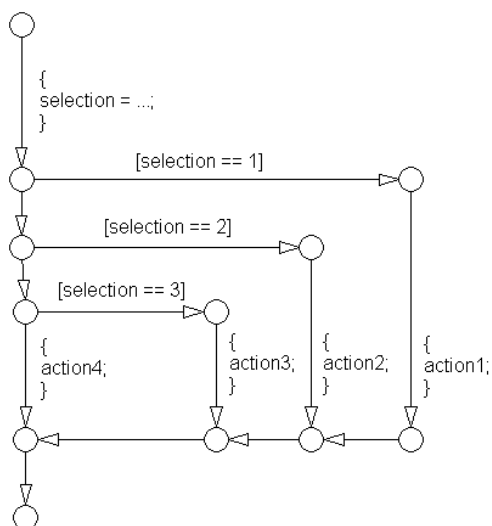


case 構文には以下のパターンを使用する。

### 直線で記述するフローチャートパターン

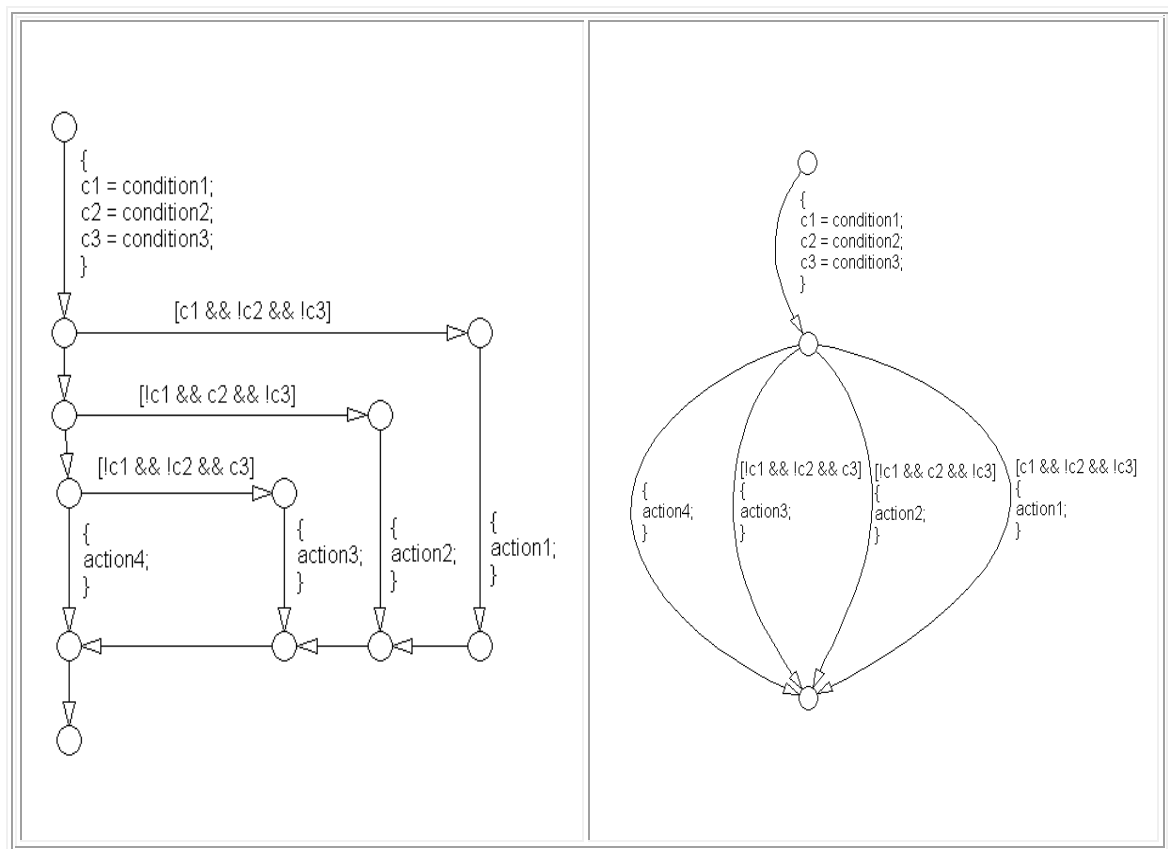
### 曲線で記述するフローチャートパターン

排他的選択による case 構文



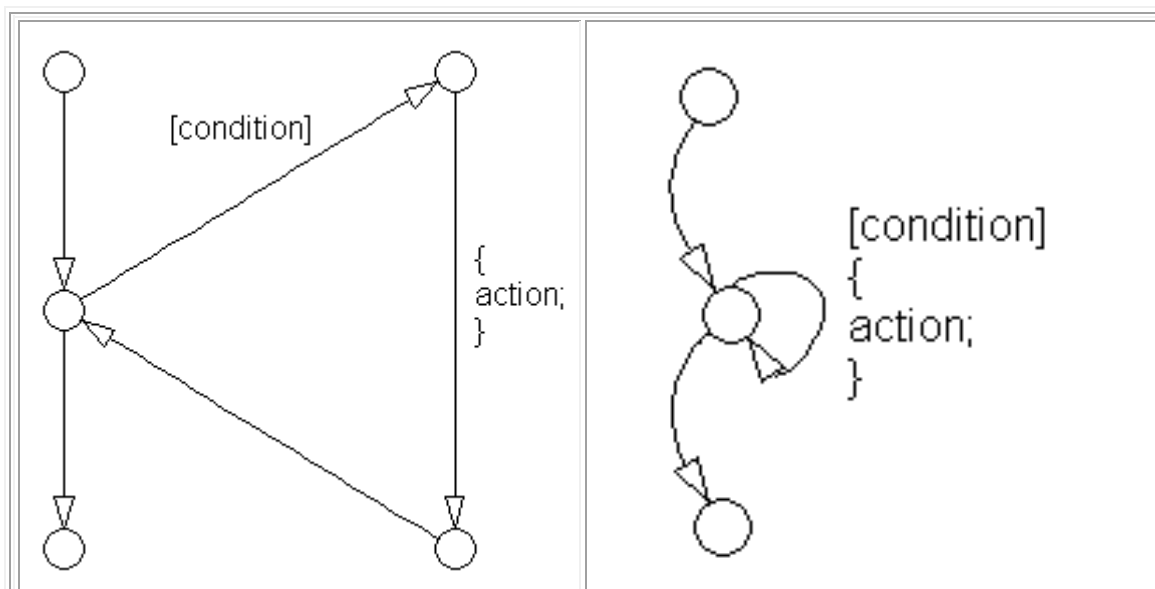
排他的条件による case 構文



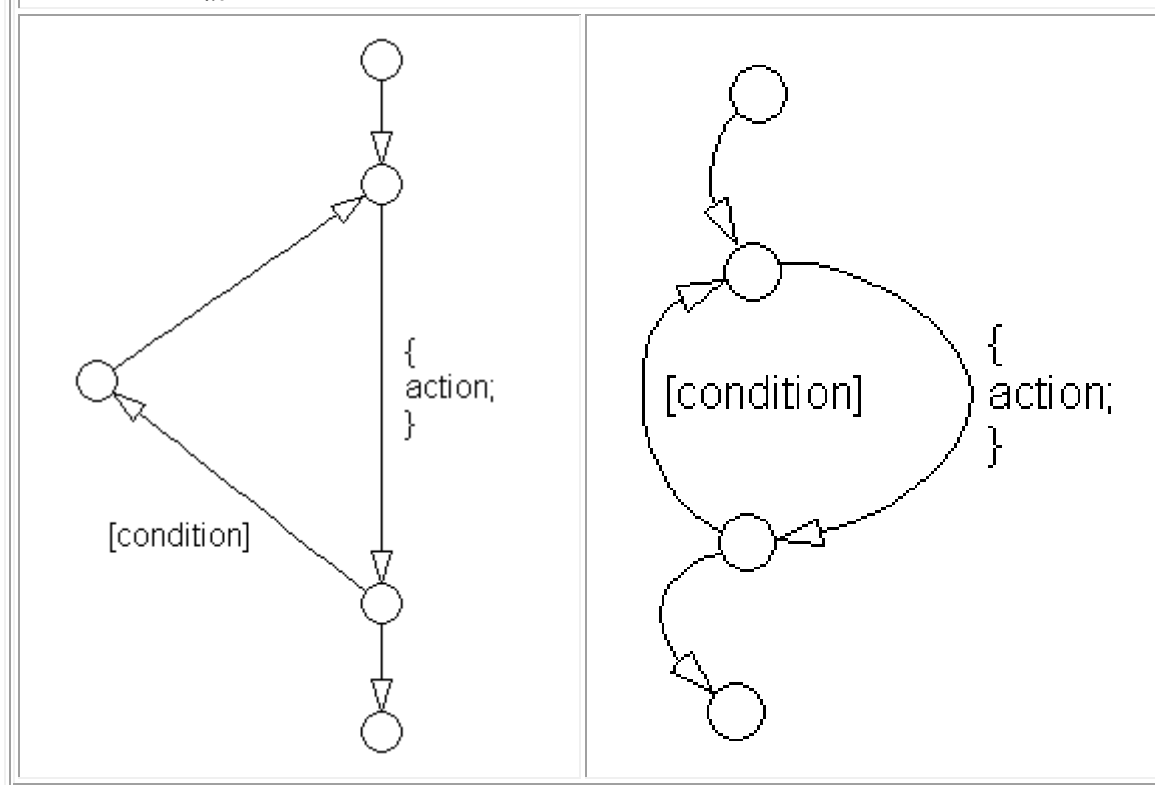


ループ構文には以下のパターンを使用する。

直線で記述するフローチャートパターン	曲線で記述するフローチャートパターン
for ループ構文	
<p>Flowchart for for loop using straight lines. It shows an initialization block <code>{ index = 0; }</code>, a loop condition <code>[index &lt; number_of_loops]</code>, a loop body with <code>{ action; }</code> and <code>{ index++; }</code>, and a final node.</p>	<p>Flowchart for for loop using curved lines. It shows an initialization block <code>{ index = 0; }</code>, a loop condition <code>[index &lt; number_of_loops]</code>, a loop body with <code>{ action; index++; }</code>, and a final node.</p>
while ループ構文	



do while ループ構文

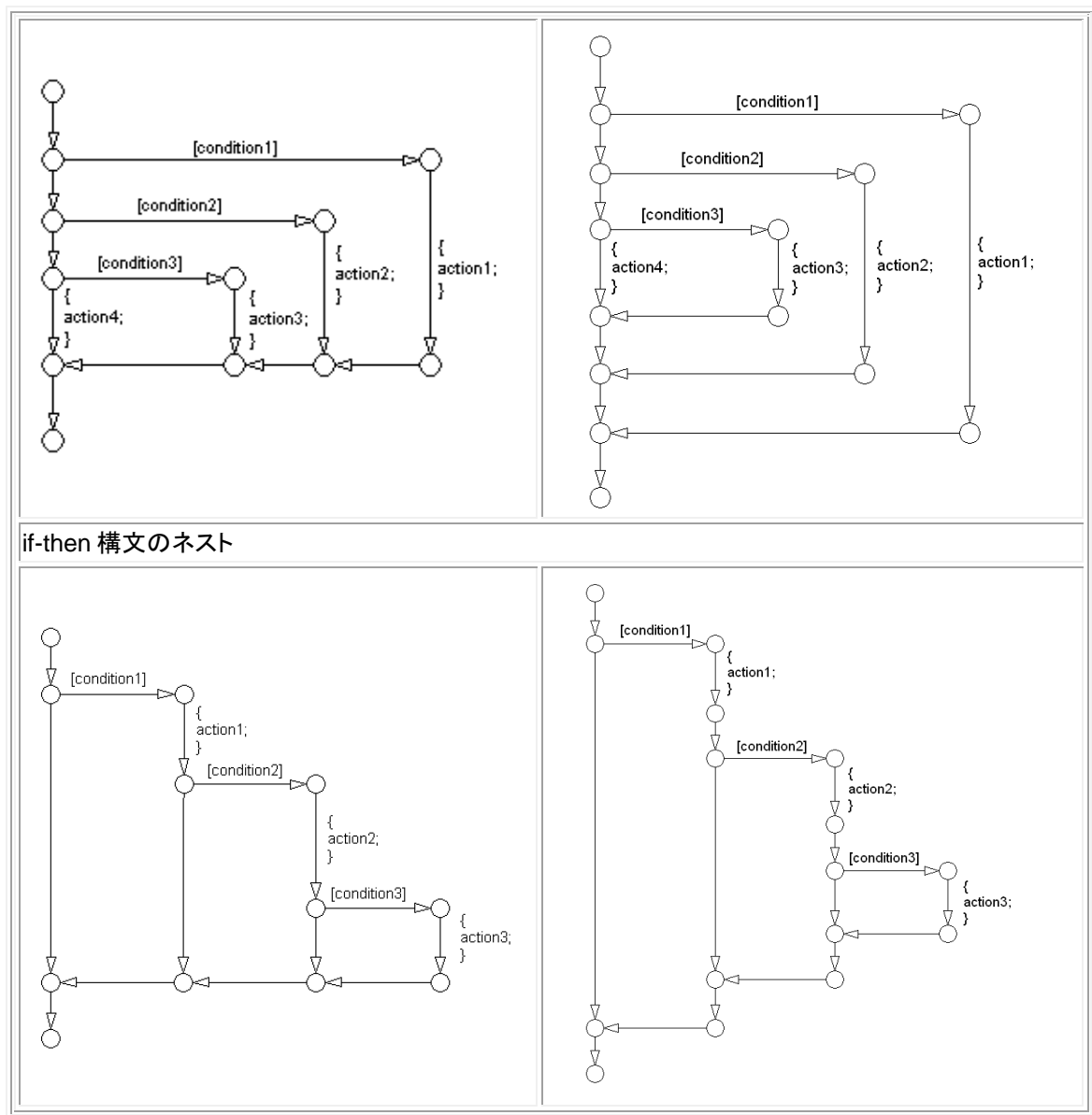


if 構文は以下のパターンで代用できる。

直線で記述するフローチャートパターン

代用の直線で記述するフローチャートパターン

if-then-else-if 構文



## 15. 付録 D: Stateflow オブジェクトクイックリファレンス

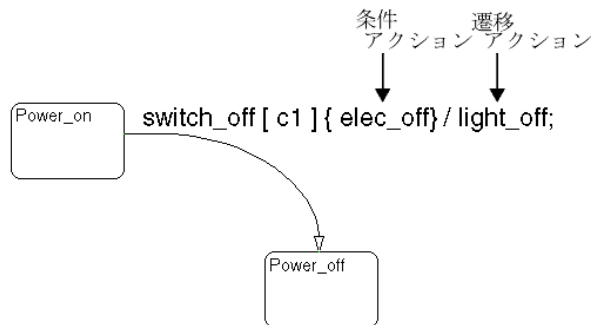
MATLAB v6.0 オンラインヘルプドキュメント“Stateflow.html”を参照。Stateflow ダイアグラムオブジェクトと専門用語に関する優れたクイックリファレンスである。

## 16. 付録 E: 参考資料

注意: 本章は、MATLAB R14SP3 の日本語版 Stateflow マニュアルから転記を行った。詳細は、最新 MATLAB の Stateflow マニュアルを参照すること。

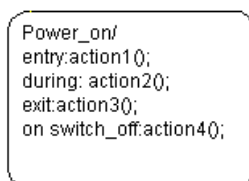
### アクション

アクションは、Stateflow ダイアグラムの実行の一部として発生します。アクションは、1 つの状態から別の状態への遷移の一部として、または状態のアクティビティに応じて実行されます。遷移は、条件アクションと遷移アクションをもつことができます。たとえば、つぎのようになります。



### アクション言語

アクション言語は、指定可能なアクションおよびそれらに関連する表記法のカテゴリを定義します。たとえば、状態は以下に示すように entry, during, exit, on event\_name アクションをもつことができます。



アクションでは、ファンクションコール、イベントのブロードキャスト、変数割り当て等を行うことができます。アクションとアクション言語に関する詳細は、"Stateflow でのアクションの利用"を参照してください。

### API (application programming interface)

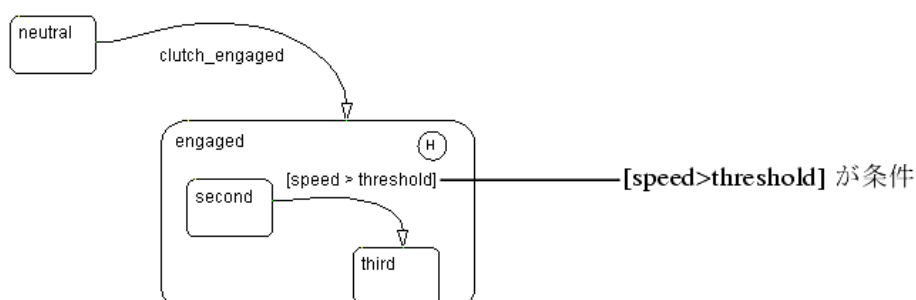
プログラミングまたはスクリプト環境からアプリケーションプログラムを使って、アクセスおよび通信を行うためのフォーマット

### チャートインスタンス

Stateflow モデルから、Simulink ライブラリにストアされているチャートへのリンクです。ライブラリにあるチャートは、多数のチャートインスタンスを持つことができます。ライブラリのチャートを更新すると、そのチャートのすべてのインスタンスが自動的に更新されます。

### 条件

条件は、ブーリアン表現であり、指定した表現が真であれば遷移が発生するように指定します。例えば、つぎのようになります。

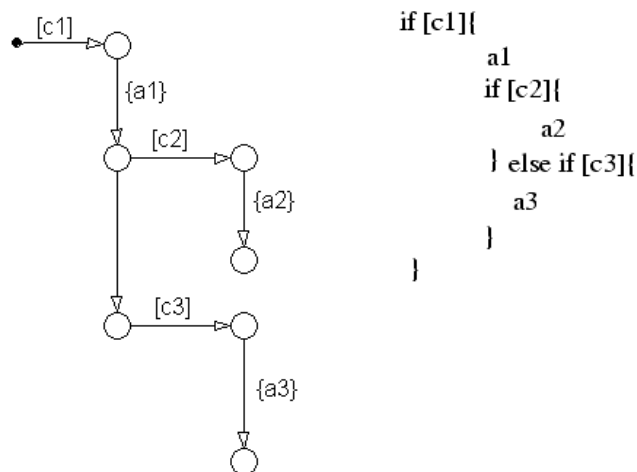


上の例で、状態 second がアクティブと仮定します。イベントが発生し、データスピードの値がデータスレッショルドの値より大きい場合は、状態 second と third 間の遷移が発生し、状態 third がアクティブになります。

### コネクティブジャンクション

コネクティブジャンクションはシステムの判断ポイントです。コネクティブジャンクションはグラフィカルオブジェクトであり、Stateflow ダイアグラムの表現をシンプルにし、効率的なコード生成を容易にします。コネクティブジャンクションは、求められるシステムの挙動を表現するために、補助的な方法を提供します。

この例題は、if コード構造の流れを表現するために、コネクティブジャンクション(小さい円)をどのように使うかを示しています。



名前	ボタンアイコン	説明
Connective junction		2 つ以上の状態のうち 1 つの状態から出ている遷移が、同じイベントを基に行われるが違う条件によって保護される状況を扱うために、コネクティブジャンクションが使われます。

詳細は、"コネクティブジャンクション"を参照してください。

### データ

データオブジェクトは、Stateflow ダイアグラムで参照する数値をストアします。データオブジェクトの表現についての詳細情報は、"データの追加"を参照してください。

### データディクショナリ

データディクショナリは、Stateflow ダイアグラム情報がストアされるデータベースです。Stateflow ダイアグラムオブジェクトを作成した場合、一度 Stateflow ダイアグラムを保存すると、オブジェクトに関する情報がデータディクショナリにストアされます。

### デバッグ

"Stateflow Debugger"を参照してください。

### 構造


状態が 1 つ以上のサブステートで構成される場合、**構造**をもちます。少なくとも 1 つの状態を含む Stateflow ダイアグラムも、構造を持ちます。階層を表現することは、状態がその階層でグループ化される方法に関して、いくつかのルールを必要とします。スーパーステートは、並列(AND)か排他的(OR)の構造のいずれかをもちます。階層の特定のレベルでのすべてのサブステートは、同じ構造である必要があります。

**並列状態の構造:** 並列(AND)状態構造は、状態の境界線が破線で表現されます。階層が同じレベルの全状態が同時にアクティブである場合、この表現は適切です。並列状態でのアクティビティは、本質的に独立です。

**排他的状態の構造:** 排他的(OR) 状態構造は、状態の境界線が実線で表現されます。排他的(OR)構造は、お互いに排他的であるシステムモードを記述するのに使われます。同じ階層レベルでの 1 つの状態のみが、一度にアクティブとなりえます。

### デフォルト遷移

デフォルト遷移は、2 つ以上の隣り合った排他的(OR)状態にあいまいさがある場合、先に始まる排他的(OR)状態を指定するために使われます。たとえば、デフォルト遷移は、他の情報がない場合に、排他的(OR)構造をもつスーパーステートのどのサブステートにシステムがデフォルトで入るかを指定します。デフォルト遷移は、ジャンクションをデフォルトと指定することもできます。デフォルト遷移を表現するには、ツールバーからデフォルト遷移オブジェクトを選択し、ドロップして遷移先のオブジェクトに付け加えます。デフォルト遷移オブジェクトは、遷移元オブジェクトはありませんが、遷移先を持つ遷移です。

名前	ボタンアイコン	詳細
デフォルト遷移		階層のこのレベルが始まるとき、デフォルトでアクティブになる状態を示すために、デフォルト遷移を使います。

詳細は、"デフォルト遷移"を参照してください。

### イベント

イベントは、Stateflow ダイアグラムの実行を駆動します。Stateflow ダイアグラムに影響するすべてのイベントが定義される必要があります。イベントの発生は、Stateflow ダイアグラムにある状態が評価される状況を引き起こします。イベントのブロードキャストは、遷移を引き起こすトリガとなることができ、かつ/あるいは、アクションを実行するトリガとなることができます。イベントは、この階層でのイベントの親から始まり、トップダウン則でブロードキャストされます。

イベントは、Stateflow Explorer で追加、削除、編集されます。詳細は、"イベントの追加"を参照してください。

### エクスプローラ

Stateflow の任意の親オブジェクトに対してデータおよびイベントオブジェクトを表示、修正、作成するためのツール。エクスプローラは、Stateflow マシンに対してもターゲットの表示、修正、作成を行います。"Stateflow Explorer"を参照してください。

### Finder

Simulink Find ツールをサポートしないプラットフォーム上で Stateflow ダイアグラムでのオブジェクトの検索を行うツール "Stateflow Finder"を参照してください。

### 有限ステートマシン (FSM)

有限ステートマシン (Finite State Machine: FSM) は、イベントドリブンシステムの表現です。また、FSM は、リアクティブシステムを記述するためにも使われます。イベントドリブンまたはリアクティブシステムでは、システムは、あるモードまたは状態から他の指示されたモードまたは状態へ、変更を定義している条件が真である条件下で、遷移します。

### 流れ図

流れ図は、状態、またはデフォルト遷移セグメントからスタートする遷移セグメントではじまる一連の判定点を与える流れパスです。

### 流れパス

流れパスは、遷移セグメントとジャンクションの順番付けされたシーケンスであり、各継承セグメントが先のセグメントの終端となるジャンクションから始まります。

### 流れサブグラフ

流れサブグラフは、同じ遷移セグメントから始まる一連の流れパスです。

### グラフィカル関数


グラフィカル関数は、そのロジックが流れ図によって定義される関数です。"関数を利用したアクション"を参照してください。

## 階層

階層は、階層は、他のより高いレベルにある状態内部に状態をおくことにより、複雑なシステムを組織化することを可能にします。階層化設計は、通常遷移の数が減少し、きちんとした、より扱いやすいダイアグラムを実現します。詳細は、"Stateflow オブジェクトの階層"を参照してください。

## ヒストリジャンクション

ヒストリジャンクションは、履歴情報に基づいて、遷移の遷移先のサブステートを指定する方法を与えます。スーパーステートにヒストリジャンクションがある場合、遷移先のサブステートへの遷移は、最後にアクセスしたサブステートであると定義されます。ヒストリジャンクションは、置いてある階層のレベルに適用されます。

名前	ボタンアイコン	詳細
ヒストリジャンクション		ヒストリジャンクションは、置いてある階層レベルに入った場合に、最後にアクティブだった状態が次回アクティブになるように使います。

詳細は、つぎの節を参照してください。

- "ヒストリジャンクション"
- "デフォルト遷移とヒストリジャンクションの例"
- "デフォルト遷移のラベル付けの例"
- "ヒストリジャンクションの内部遷移の例"

## 内部遷移

内部遷移は、遷移元の状態を終了しない遷移です。内部遷移は、XOR 構造を持つスーパーステートに対して定義する場合、最も効果的です。内部遷移を使うと、Stateflow ダイアグラムを大変シンプルにできます。詳細は、"内部遷移" と"ヒストリジャンクションの内部遷移の例"を参照してください。

## ライブラリリンク

ライブラリリンクは、Simulink ブロックライブラリのライブラリモデルにストアされているチャートへのリンクです。

## ライブラリモデル

Simulink ライブラリにストアされた Stateflow モデルです。チャートをコピーすることによって、モデルにライブラリからのチャートを含ませることができます。ライブラリからモデルにチャートをコピーする場合、Stateflow は物理的にはモデルにチャートを含まず、ライブラリチャートへのリンクを作成します。1 つのチャートに複数のリンクを作成できます。各リンクは、チャートインスタンスと呼ばれます。モデルにライブラリからのチャートを含む場合、Stateflow マシンもまた含みます。従って、ライブラリチャートへのリンクを含む Stateflow モデルは、複数の Stateflow マシンを持ちます。Stateflow がライブラリモデルからのチャートを含むモデルをシミュレーションする場合、たとえそのライブラリモデルのうちいくつかだけへのリンクがあるとしても、そのモデルはライブラリモデルからの全チャートを含みます。しかしながら Stateflow がスタンドアロンか RTW ターゲットを生成する場合、リンクのあるチャートのみを含みます。ライブラリモデルへのリンクを含むモデルは、ライブラリモデルの全チャートが文法チェックを通り、コンパイルエラーが起きない場合のみシミュレーションできます。

## マシン



ライブラリリンクからのチャートインスタンスを除いた Simulink モデルによって定義されたすべての Stateflow ブロックの集合です。モデルが任意のライブラリリンクを含む場合、リンクの元になるモデルによって定義されている Stateflow マシンも含まれます。

### **表記法**

オブジェクトと、それらのオブジェクトの関係を支配するルールを定義します。Stateflow の表記法は、Stateflow ダイアグラムによって伝えられる設計情報を伝達する、共通の言語を与えます。

Stateflow は、つぎに示すもので構成されます。

- グラフィカルオブジェクトの集合
- 非グラフィカルな、テキストベースオブジェクトの集合
- これらのオブジェクト間で定義された関係

### **並列処理**

並列処理のあるシステムは、同時にアクティブとなりうる 2 つ以上の状態をもつことができます。並列状態のアクティビティは、基本的に独立です。並列処理は、並列(AND)状態構造で表現されます。

詳細は、"状態の構造"を参照してください。

### **Real-Time Workshop®**

Simulink についての自動 C 言語コードジェネレータです。Simulink ブロックダイアグラムモデルから C コードを直接生成し、様々な環境でリアルタイムに実行できるプログラムを自動的にビルドします。詳細は、Real-Time Workshop のドキュメントを参照してください。

### **rtw ターゲット**

Real-Time Workshop によって生成されたコードからビルドされた、実行可能プログラムです。詳細は、"ターゲットのビルド"を参照してください。

### **S-Function**

Simulink を Stateflow と一緒にシミュレーションに使用する場合、Stateflow は各 Stateflow マシンについて S-function(MEX- ファイル)を生成し、モデルのシミュレーションをサポートします。この生成コードはシミュレーションターゲットであり、Stateflow の中で sfun ターゲットと呼ばれます。

詳細は、Using Simulink を参照してください。

### **意味論**

意味論は、表記法がどのように解釈され、裏でどのように実行されるのかを説明します。完全な Stateflow ダイアグラムは、どのようにシステムが振る舞うのかを伝えます。Stateflow ダイアグラムは、遷移と状態に関連するアクションを含みます。意味論は、Stateflow ダイアグラムの実行中にアクションの生じるシーケンスを説明します。

### **Simulink**

Simulink は、動的システムのモデリング、シミュレーション、解析のためのソフトウェアパッケージです。Simulink は線形、非線形システムをサポートしており、連続時間、サンプルされた時間(離散時間)、または両方のハイブリッドをサポートしています。システムはマルチレート、すなわち、違うレートでサンプリングされる、あるいは更新される部分を持っているシステムも実現できます。

Simulink では、マウスを使ってブロックを接続し、キーボードでブロックパラメータを編集して、ブロックダイアグラムとしてシステムを表現することができます。Stateflow はこの環境の一部で、Stateflow ブロックはマウスされた Simulink モデルです。Stateflow block は Simulink モデルをマスクします。Stateflow は各 Stateflow マシンに対応する S-function をビルドします。この S-function は、Simulink がシミュレーションと解析を行う際の媒体となります。


Stateflow がモデリングする制御動作は、Simulink ブロックダイアグラムでモデリングされたアルゴリズムな動作を補足します。Stateflow ダイアグラムブロックを Simulink モデルに含めることにより、Simulink シミュレーションにイベントドリブンの動作を追加できます。通常の Simulink ブロックセットと Stateflow blocks を組み合わせて、データフロー、コンとロールフローの両方を表現するモデルを作成します。組み合わされたモデルは Simulink を使ってシミュレーションされます。

Using Simulink ドキュメントには、Simulink の使用法が説明されており、Simulink ブロックの操作方法、ブロックパラメータへのアクセス方法、モデルを構築するためにブロックを接続する方法が説明されています。また、標準の Simulink ライブラリの各ブロックリファレンスも記載されています。

### 状態

リアクティブシステムのモードのことを **状態**といいます。多くの状態が、リアクティブシステムでは起こります。Stateflow ダイアグラムの状態は、それらのモードを表現します。状態のアクティビティまたは非アクティビティは、イベントと条件の間の遷移に基づいて動的に変化します。

全状態は階層構造になっています。単一の状態で構成される Stateflow ダイアグラムでは、その状態の親は Stateflow ダイアグラムそれ自身です。また状態には履歴があり、履歴は Stateflow ダイアグラム内の同じ階層レベルに適用されます。状態はアクションを持つことができます。アクションはアクションタイプに基づいて順次実行されます。アクションタイプには、entry, during, exit, on event\_name があります。

名前	ボタンアイコン	詳細
状態		状態は、システムのモードを表現するために使われます。

### Stateflow ブロック

Stateflow ブロックはマスクされた Simulink モデルであり、空でタイトルのない Stateflow ダイアグラムと同じです。Stateflow block ブロックを使って、Stateflow ダイアグラムを Simulink モデル内に置きます。

Stateflow がモデリングする制御動作は、Simulink ブロックダイアグラムでモデリングされたアルゴリズムな動作を補足します。Stateflow blocks を Simulink モデルに含むことにより、Simulink シミュレーションに複雑なイベントドリブンの動作を追加できます。通常の Simulink とツールボックス、ブロックセットと Stateflow blocks を組み合わせて、データフロー、コントロールフローの両方を表現するモデルを生成します。組み合わされたモデルは Simulink を使ってシミュレーションされます。

### Stateflow Debugger

Stateflow デバッガを使って、Stateflow ダイアグラムをデバッグし、アニメーションを表示することができます。Stateflow ダイアグラムのシミュレーションにおいて、各状態は全コード適用範囲で評価されます。この適用範囲の解析は、ターゲットがデバッグオプションを付けてコンパイル、ビルドされると、自動的に実行されます。デバッガはまた、動的なチェックを行うために使うことができます。デバッガは Stateflow マシン上で動作します。

### Stateflow ダイアグラム

Stateflow を使って、Stateflow ダイアグラムを作成します。Stateflow ダイアグラムは、有限ステートマシンのグラフィカルな表現でもあり、**状態と遷移**がシステムの基本的要素です。Stateflow ダイアグラムに関する詳細は、"Stateflow と Simulink"を参照してください。

### Stateflow Explorer

エクスプローラを使用して、データ、イベント、ターゲットオブジェクトを、追加、削除、編集できる。

### Stateflow Finder

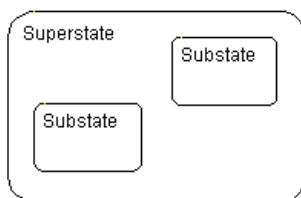
ユーザの指定した検索基準を基にオブジェクトのリストを示します。検索結果の出力表示画面で、オブジェクトをクリックすると、そのオブジェクトのプロパティダイアログボックスに直接アクセスすることができます。詳細は、"Stateflow Finder の使用方法"を参照してください。

### サブチャート

サブチャートは、他のチャートに含まれるチャートです。"サブチャートを利用したチャート"を参照してください。

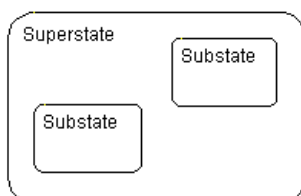
### サブステート

状態が、スーパーステートに含まれる場合は、サブステートです。



### スーパーステート

状態が、サブステートと呼ばれる他の状態を含む場合は、スーパーステートです。



### スーパートランジション

スーパートランジションは、違うサブチャートにあるオブジェクト間での遷移です。詳細は、"スーパートランジションを利用した遷移"を参照してください。

### ターゲット

モデル内の Stateflow ダイアグラムから生成されるコードに対するコンテナオブジェクト。Stateflow は、モデルに対するすべての Stateflow ダイアグラムを Stateflow マシンとして表わします。これは、ターゲットオブジェクトが Stateflow マシンに属することを意味します。

Stateflow は、シミュレーション、Real-Time Workshop およびカスタムターゲットを含むすべてのターゲットタイプについてコードを生成します。詳細は、第 12 章, "ターゲットのビルド"を参照してください。

### トップダウン処理

トップダウン処理は、Stateflow が状態とイベントを処理する方法です。特に、Stateflow は状態の前にスーパーステートを処理します。Stateflow は、最初にスーパーステートがアクティブである場合のみ、状態を処理します。

### 遷移

遷移は、システムがある状態から別の状態へ移る状況を言います。遷移の各端は、遷移元、遷移先オブジェクトに接続されます。遷移元は遷移の始まりで、遷移先は遷移の終わりのところ。遷移が起こると、多くの場合、イベントも発生します。

### 遷移パス

遷移パスは、状態で始まり、状態で終了する流れパスです。

### 遷移セグメント

完全な状態間遷移の一部である、状態からジャンクション(state-to-junction)、ジャンクションからジャンクション(junction-to-junction)、ジャンクションから状態(junction-to-state)です。遷移セグメントは、広い意味で遷移として解釈される場合があります。

### バーチャルスクロールバー

バーチャルスクロールバーは、選択リストをスクロールすることによって、変数を設定できるようにします。バーチャルスクロールバーを使ってメニュー項目の上でマウスを動かすと、カーソルが 2 重の矢印に変わります。バーチャルスクロールバーは、垂直方向あるいは水平方向に動きます。動く方向は、矢印の示す方向となります。値を変更するには、水平方向あるいは垂直方向に、マウスをドラッグしてください。