

# Mプログラム標準化への取り組み



三菱電機コントロールソフトウェア(株)  
姫路事業所 技術第3部 技術第3課  
谷 正一

三菱電機(株)姫路製作所  
制御機器第一製造部  
制御技術設計第6G  
山田 元美

2014年 12月18日

1. 標準化の背景
2. 標準化の目的
3. 標準化の内容
4. 今後の課題

1. 標準化の背景
2. 標準化の目的
3. 標準化の内容
4. 今後の課題

## ■ Mプログラム = MATLABのプログラム言語

### ■ 特徴

#### 長所

MATLABとの親和性が高い

豊富な関数

Simulinkモデル  
の扱いが容易

高速な行列演算

#### 短所

特殊な言語仕様、支援環境や  
コーディングルールが未整備

CやVBとは  
異なる仕様

MISRA等の  
基準が無い

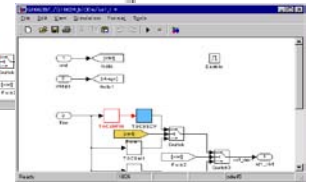
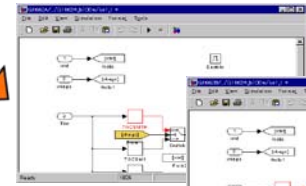
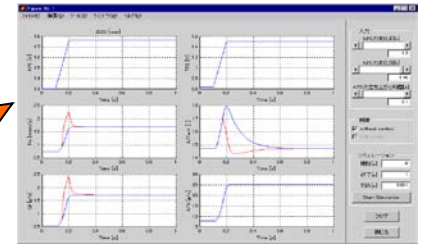
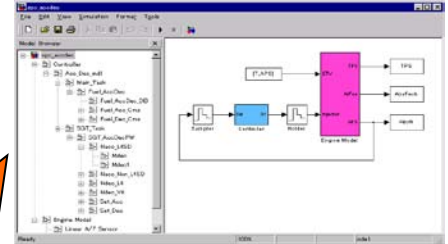
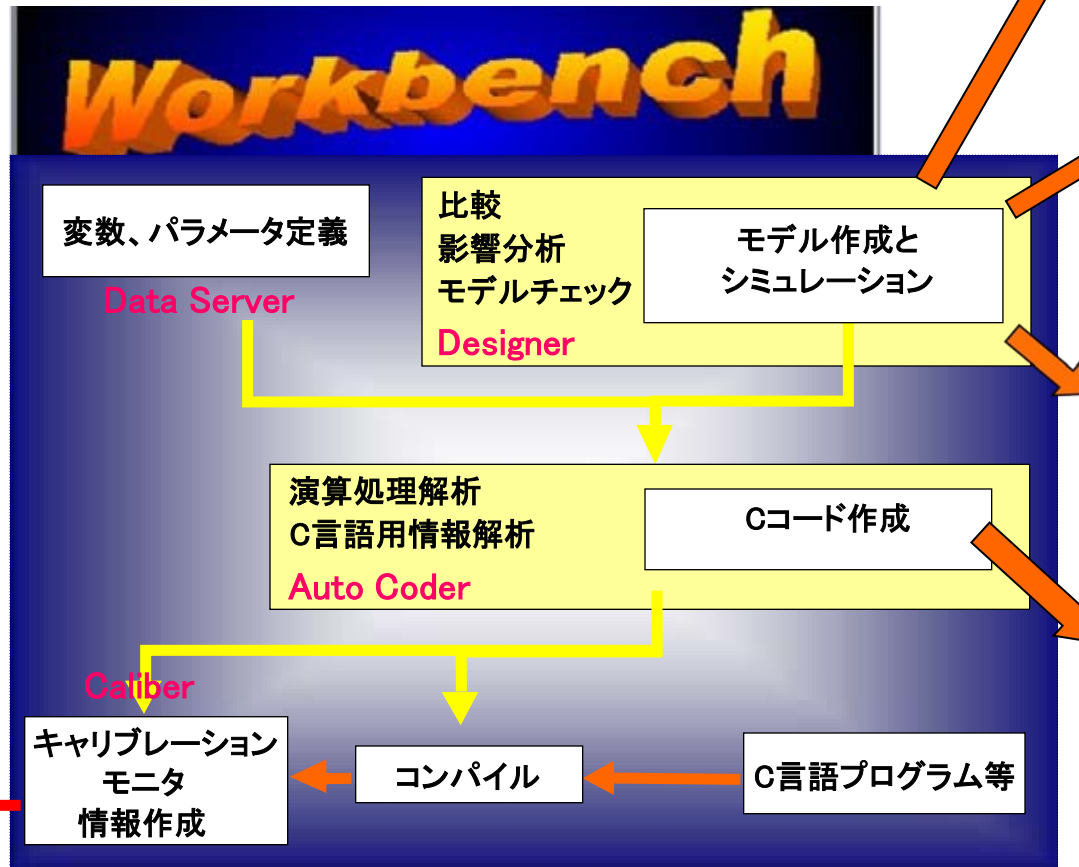
市販のユーティリティ  
ツールが無い

# 標準化の背景: 統合環境Workbench

自動車機器事業本部

## Workbench: 内製のMBD統合開発環境

- 1999年頃より開発をスタート
- 設計者が一貫した操作で使用できる



```

// Example C code snippet from the screenshot
#include <stdio.h>
#include <math.h>

int main() {
    double x = 1.0;
    double y = 2.0;
    double z = x * y;
    printf("z = %f\n", z);
    return 0;
}
    
```



実機試験

# 標準化の背景:Mプログラムの量

自動車機器事業本部

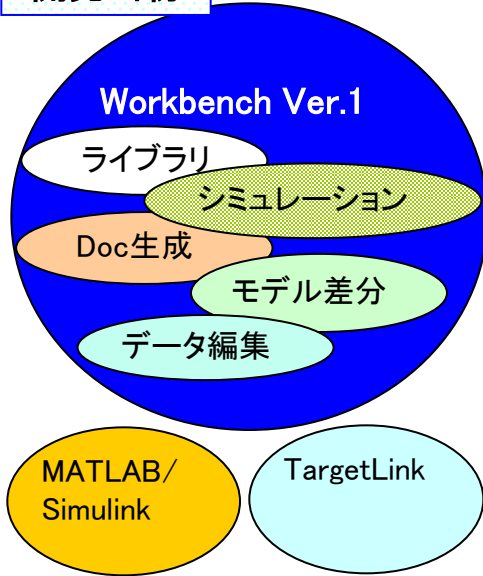
時間

2000年頃

2000年代半ば

~現在

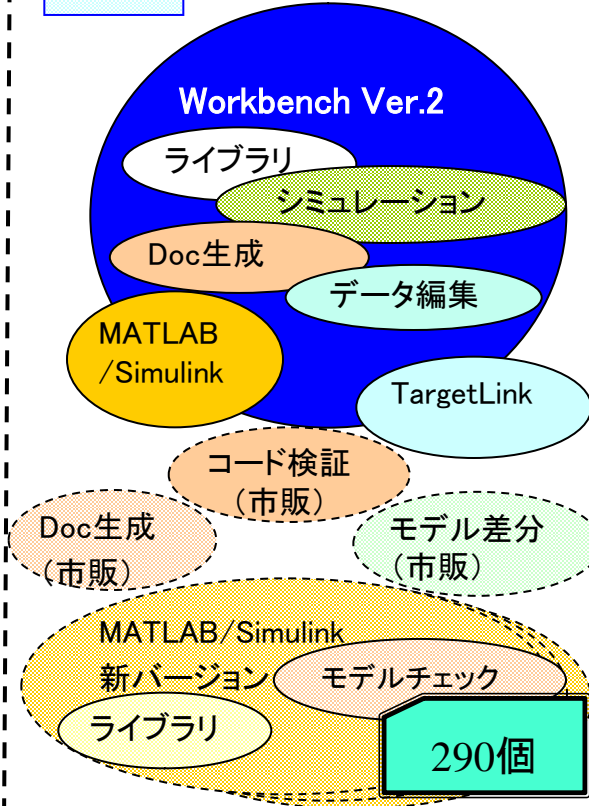
開発当初



Mプログラム数: 242個

- 当時存在しなかった機能を内製し、モデルベース開発をいち早く実現

Ver.2

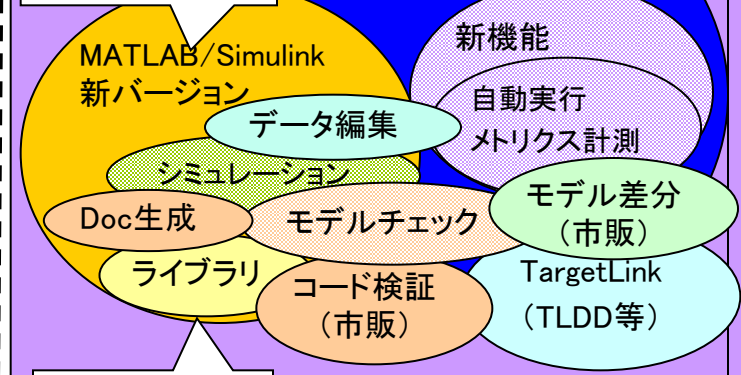


290個

- MATLAB/Simulink新バージョン登場
- 市販ツールの充実

新バージョン

新バージョンへの対応



有効なBlockを利用

複数Verへの対応

新機能追加

368個

開発目的

- 市販ツールを上手に活用して、新バージョンに迅速に対応する
- ユーザビリティの向上

1. 標準化の背景
2. 標準化の目的
3. 標準化の内容
4. 今後の課題

## Mプログラム

- プログラム記述の自由度が高い
- プログラムの可読性はあまり良くない
- メンテナンス性が悪くなりやすい

## 標準化のねらい

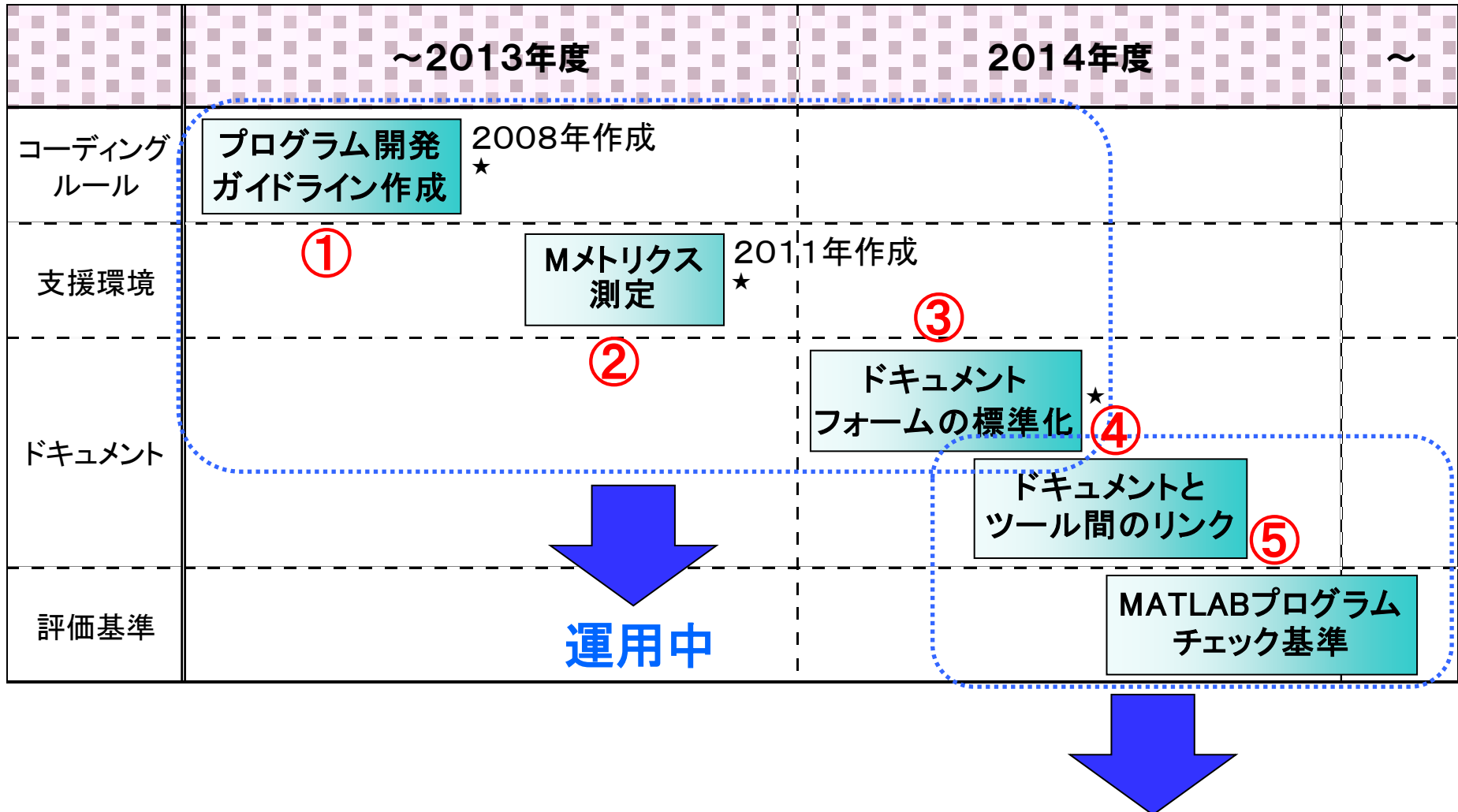
プログラム品質(開発効率・実行効率・保守性)の向上



1. 標準化の背景
2. 標準化の目的
3. 標準化の内容
4. 今後の課題

# 標準化の内容: 取り組み内容

自動車機器事業本部



構築／作成中

# 標準化の内容: ①Mプログラム開発ガイドライン

自動車機器事業本部

設計メモ	機種名	I SDP	作成日付	2008- 04- 14	取扱	B
			No.	XX - XXX - XXXX		
表題	MATLAB言語によるプログラム開発ガイドライン					
目的	MATLAB					
内容 (方法)	<ol style="list-style-type: none"> <li>1. エディタ 使用エディタ、エディタ設定、フォント設定など</li> <li>2. Mファイル ファイル名、プログラムの履歴の記述方法など</li> <li>3. 関数 関数名命名規則、関数間の区切り、履歴記述など</li> <li>4. 変数 変数名命名規則など</li> <li>5. コメント 書式、記述位置など</li> <li>6. プログラム記述方法 文法上の注意など</li> </ol>					
結論	7. Ex					

**(全18P:35個のルール)** 以上

# 標準化の内容:ガイドライン実例1(関数)

タイトル	入れ子関数の使用
内容	<p>入れ子関数は使用しない。</p> <p><u>作成例(関数Aの内部に関数Bを入れ子にしている例)</u></p> <pre>function x = A(p1)     p2 = 10 ;     x = B(p1);         function y = B(p1)             y = p1 * p2 ;         end     end end</pre> <div data-bbox="1166 736 1760 886" style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block;"> <p>引数で渡さなくても、 関数Aの変数を参照可能</p> </div>
根拠	プログラムの保守性向上のため

# 標準化の内容:ガイドライン実例2(Prg記法)

自動車機器事業本部

タイトル	exist 関数の使用
内容	<p>exist 関数を使用する場合は、検索対象の種別を明確にして実行すること。</p> <div data-bbox="1064 429 1792 558" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p><u>exist関数</u>: 変数・関数・フォルダー等の存在をチェック</p> </div> <p><u>具体例</u></p> <p>○ if (exist('test','file') ~= 0)            ...            end</p> <div data-bbox="821 658 1184 765" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>検索対象の種別</p> </div> <p>× if (exist('test') == 4)            ...            end</p> <div data-bbox="1238 768 1756 875" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>4の意味がわからない</p> </div> <p>× if (exist('test') ~= 0)            ...            end</p> <div data-bbox="1203 1015 1792 1122" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>検索対象の種別が不明確</p> </div>
根拠	プログラムの保守性向上のため

# 標準化の内容:ガイドライン実例3 (Prg記法)

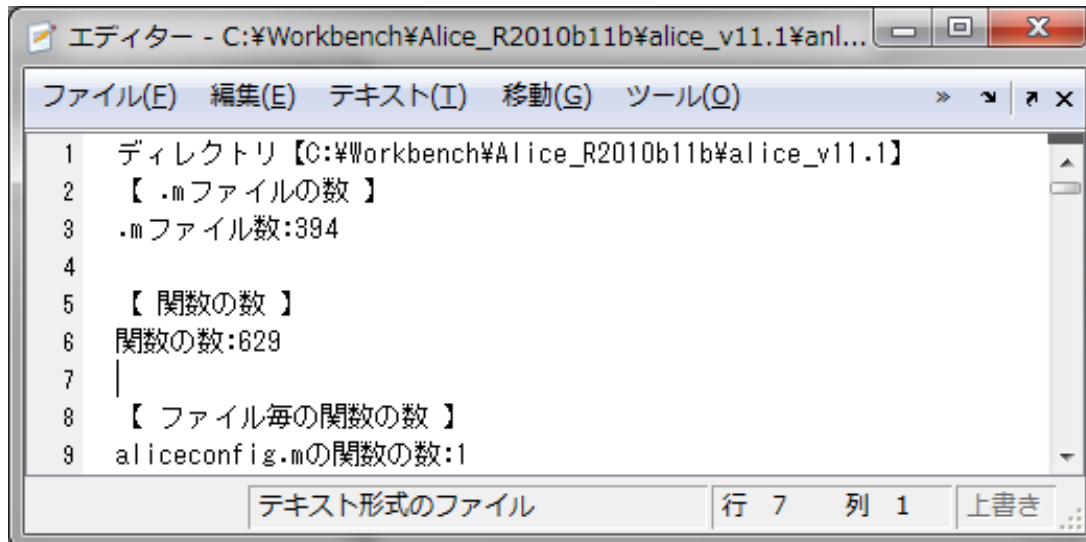
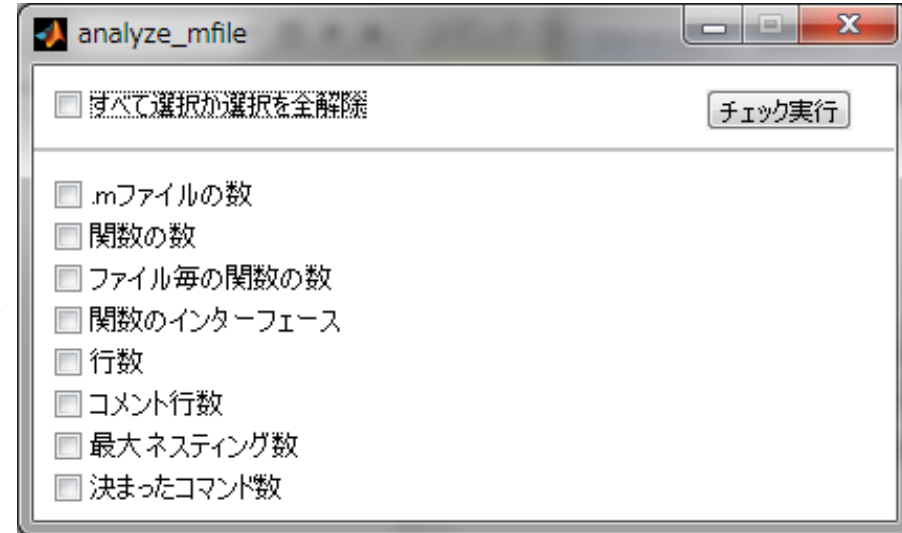
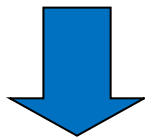
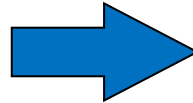
自動車機器事業本部

タイトル	IF 文の使用方法
内容	<p>(1) IF の判断文では True/False で判定すること。</p> <p><u>具体例</u></p> <ul style="list-style-type: none"> <li>○ if strcmp(str,'test') == true</li> <li>× if strcmp(str,'test')</li> </ul> <p>(2) IF 文ではカンマ区切りの書式は使用しない。</p> <p><u>具体例</u></p> <ul style="list-style-type: none"> <li>○ if a&gt;0     b=10;     end</li> <li>× if a&gt;0,b=10,end</li> </ul>
根拠	プログラムの可読性向上のため

# 標準化の内容: ②Mメトリクスツール

## Mメトリクスツール

- ◆ Mプログラムのメトリクスを計測  
(品質要素を数値化)
- MATLABコマンドウィンドウから起動
- 計測項目は選択式(右図)
- 項目毎にレポート出力  
(下図)



# 標準化の内容: ③④ドキュメントフォーマット

自動車機器事業本部

項目	内容
関数名	option_einfo ... ① mファイル名
書式	op_err_msg = option_einfo(error_code,varargin) ... ②
説明	エラーオプション情報の取得 ... ③
引数1	error_code エラーオプション番号 (error_codeが定義されていない場合は、MATLABで実行時エラーになる) ... ④
引数2	varargin エラーオプションメッセージ付加情報 (対象ブロック名や対象ファイル名など)
戻り値1	op_err_msg エラーオプションメッセージ ... ⑤
備考	オプションメッセージ情報の登録は本ソースコードの以下の代入文を追加することにより行う。 ... ⑥

```

1  function op_err_msg = option_einfo(error_code,varargin)
2  % エラーオプション情報の取得
3  % (入力)
4  % error_code : エラーオプション番号
5  % (error_codeが定義されていない場合は、MATLABで実行時エラーになる)
6  % varargin : エラーオプションメッセージ付加情報 (対象ブロック名や対象ファイル名など)
7  % (出力)
8  % op_err_msg : エラーオプションメッセージ
9  %
10 % オプションメッセージ情報の登録は本ソースコードの以下の代入文を追加することにより
11 % 行う。
12
13 % 変更履歴
14 % -----
15 % 2009/10/29 コード生成エラーのオプションメッセージ対応 Ver1.00 橋本
16 % -----
17

```

ツールでの  
自動更新

逆変換にも  
対応



三菱電機株式会社

## MATLABプログラムチェック基準

1. 概要
2. 運用方法
3. ソフトウェアチェックシート
  - 1 要件定義
  - 2 仕様書作成

## MATLABプログラムチェック基準

MATLABプログラムの品質を確保することを目的として、プログラム作成に当たり遵守すべき作法(指針)、及び最低限チェックすべき項目を規定したもの(チェックリスト)

本基準に拠るチェックは、MATLAB上で作成した機能確認を行う段階において設計者自身により行うものである。

### 2)適用方法

本基準に拠ってチェックを行い、ランクに応じて適切な処理を行わねばならない。処理後、採番して報告しなければならない。

Aランク:ソフトウェア品質を確保する上で、必ず遵守すべき項目である。

NG判定となる場合は、必ず理由を明記の上、上長承認が必要である。

Bランク:OK判定が望ましいが、必須では無い。NGの場合は理由を明記する。

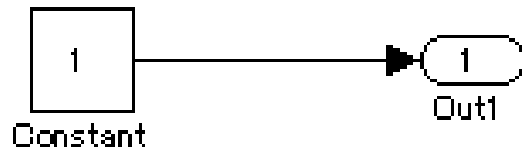
MATLABプログラムチェック基準

ED-XXX-X

## チェック項目 例1

「配列データを扱う場合は、データ数が1と2以上の場合を考慮すること。」

### データ数が1の場合



```
>> a = find_system(bdroot(gcbh), 'BlockType', 'Constant')
```

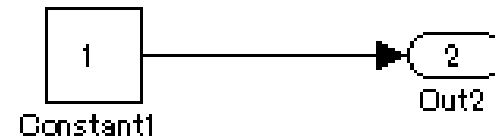
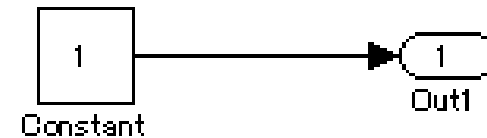
```
a =  
1.9320e+003
```



```
>> get_param(a, 'Value')
```

```
ans =  
1
```

### データ数が2の場合



```
>> a = find_system(bdroot(gcbh), 'BlockType', 'Constant')
```

```
a =|  
1.0e+003 *  
1.9320  
1.9430
```



```
>> get_param(a, 'Value')
```

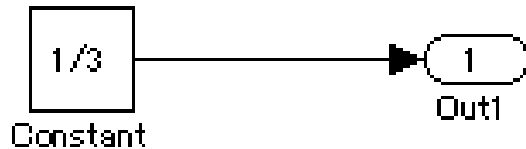
```
ans =  
'1'  
'1'
```

**データの形式が違う！**

## チェック項目 例2

### 「str2numとstr2doubleの使用方法(使い分け)は適切か」

分数値を持ったConstantブロック



```
>> a = get_param(gcb, 'Value')
```

```
a =
```

```
1/3
```

str2numで実行した場合

```
>> str2num(a)
```

```
ans =
```

```
0.3333
```

str2doubleで実行した場合

```
>> str2double(a)
```

```
ans =
```

```
NaN
```

数値文字の  
扱い方が違う!

1. 標準化の背景
2. 標準化の目的
3. 標準化の内容
4. 今後の課題

- 作成したガイドラインは抽象度が高く具現化が必要
  - 具体的な規程・標準類を定義し、運用する。
- ガイドライン違反箇所の抽出は工数大
  - 適合性を自動検出(修正)するチェッカーを開発
- Mプログラム品質を図る尺度(基準)が無い
  - Mメトリクス計測データを蓄積・分析し、Mプログラムの品質分析(定量化)ツールを開発
  - Mプログラムの依存関係解析機能の開発
- Mプログラム開発における一般的なガイドラインが無い
  - Mプログラミングにおける業界標準の検討

# 標準化の内容:今後の取り組み

自動車機器事業本部

