

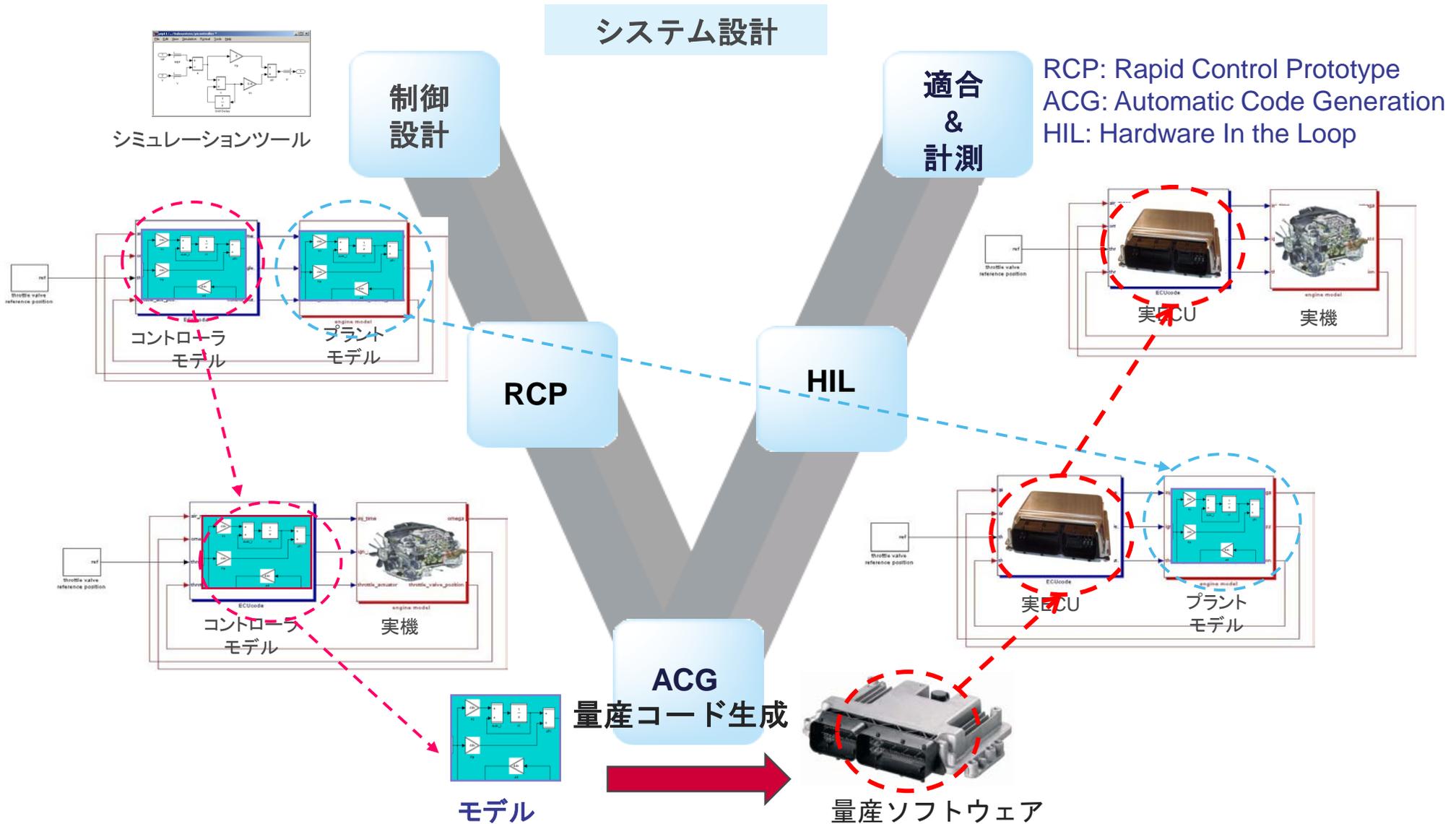
MBD適用時の車両適合の課題

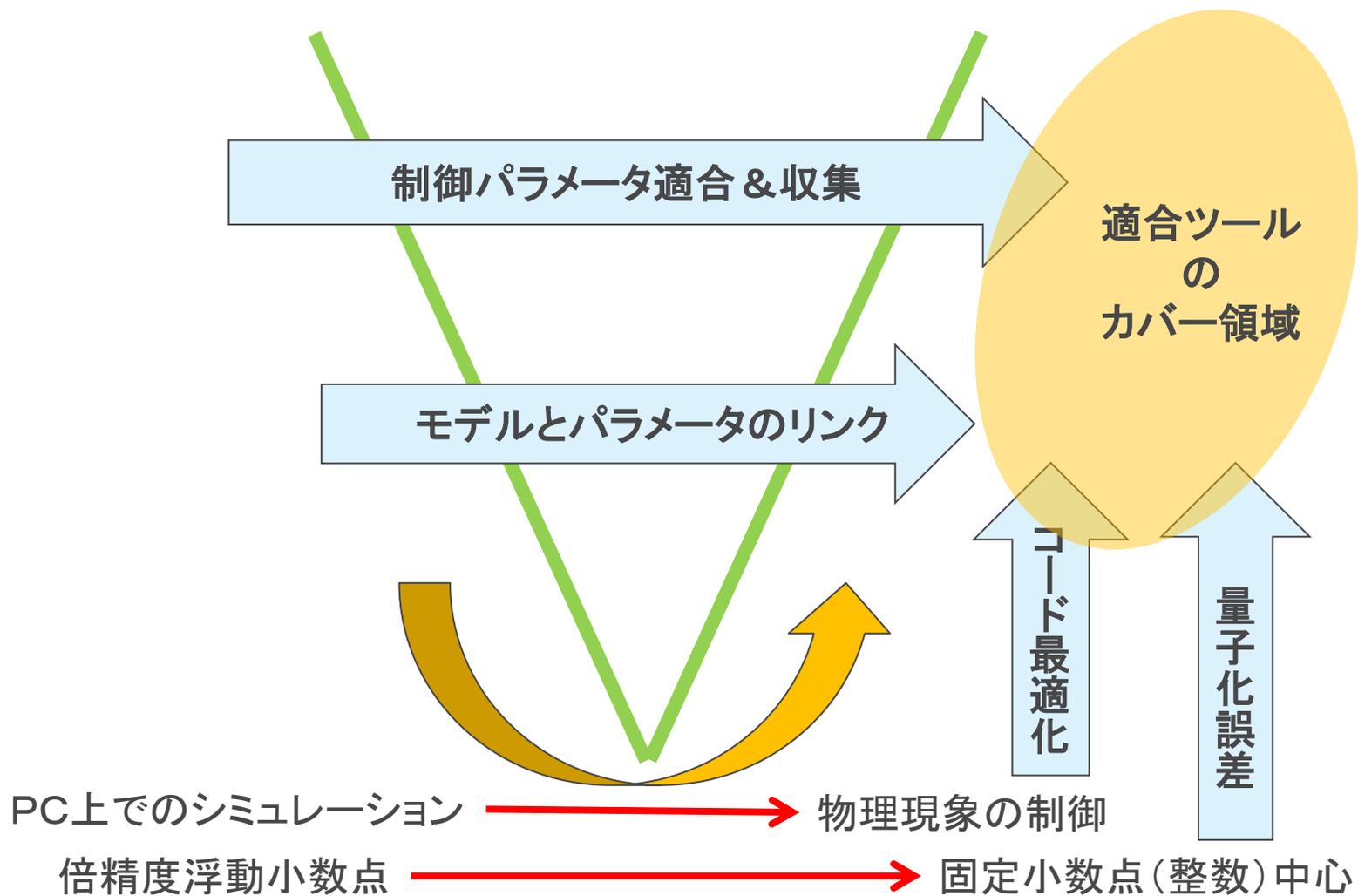


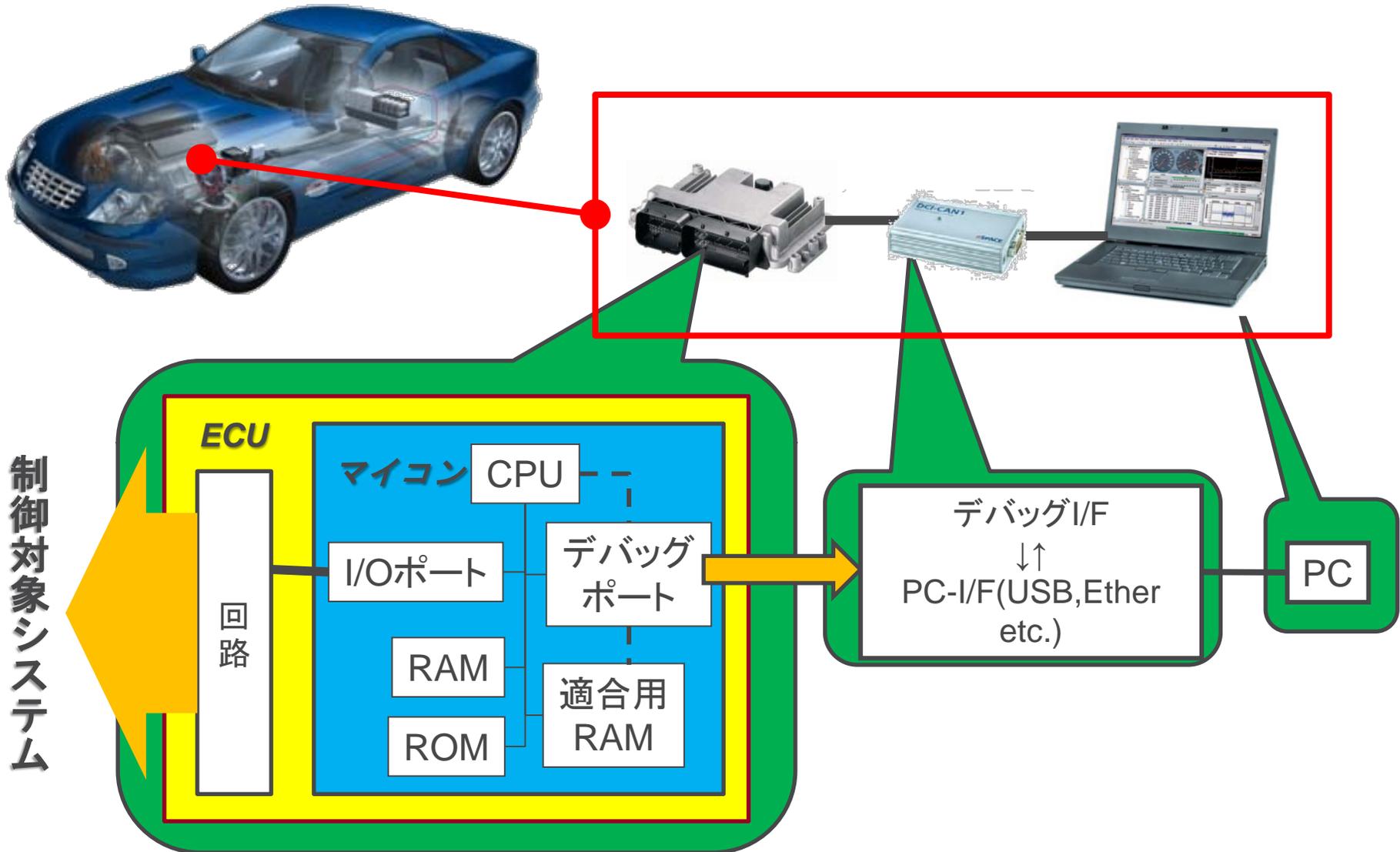
dSPACE Japan 株式会社

ソリューション技術部

佐々木 茂







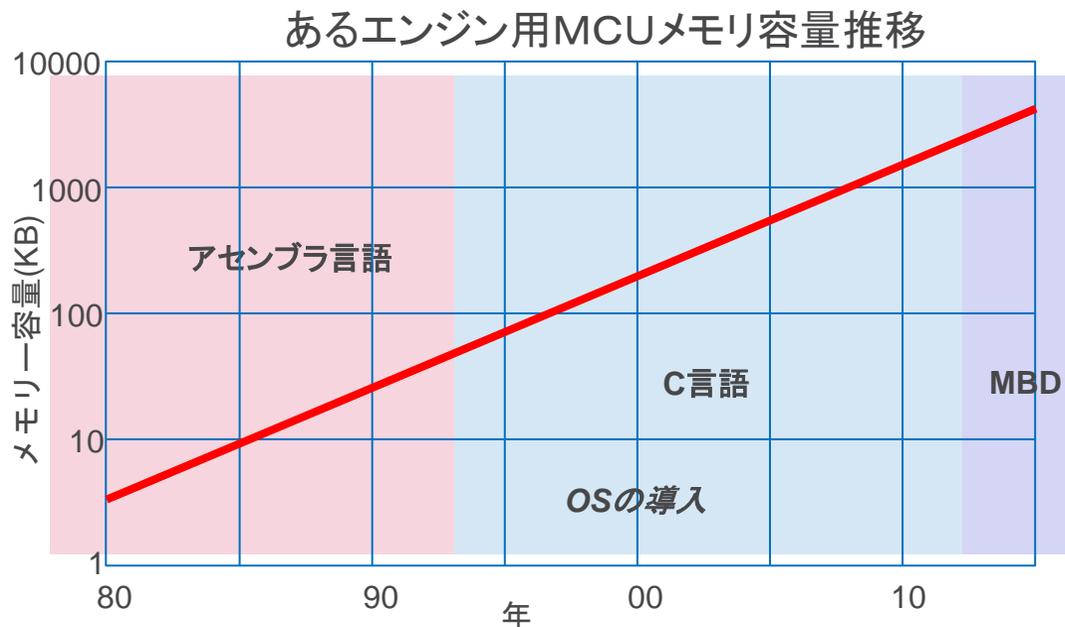
メモリの増加
30年間で1000倍

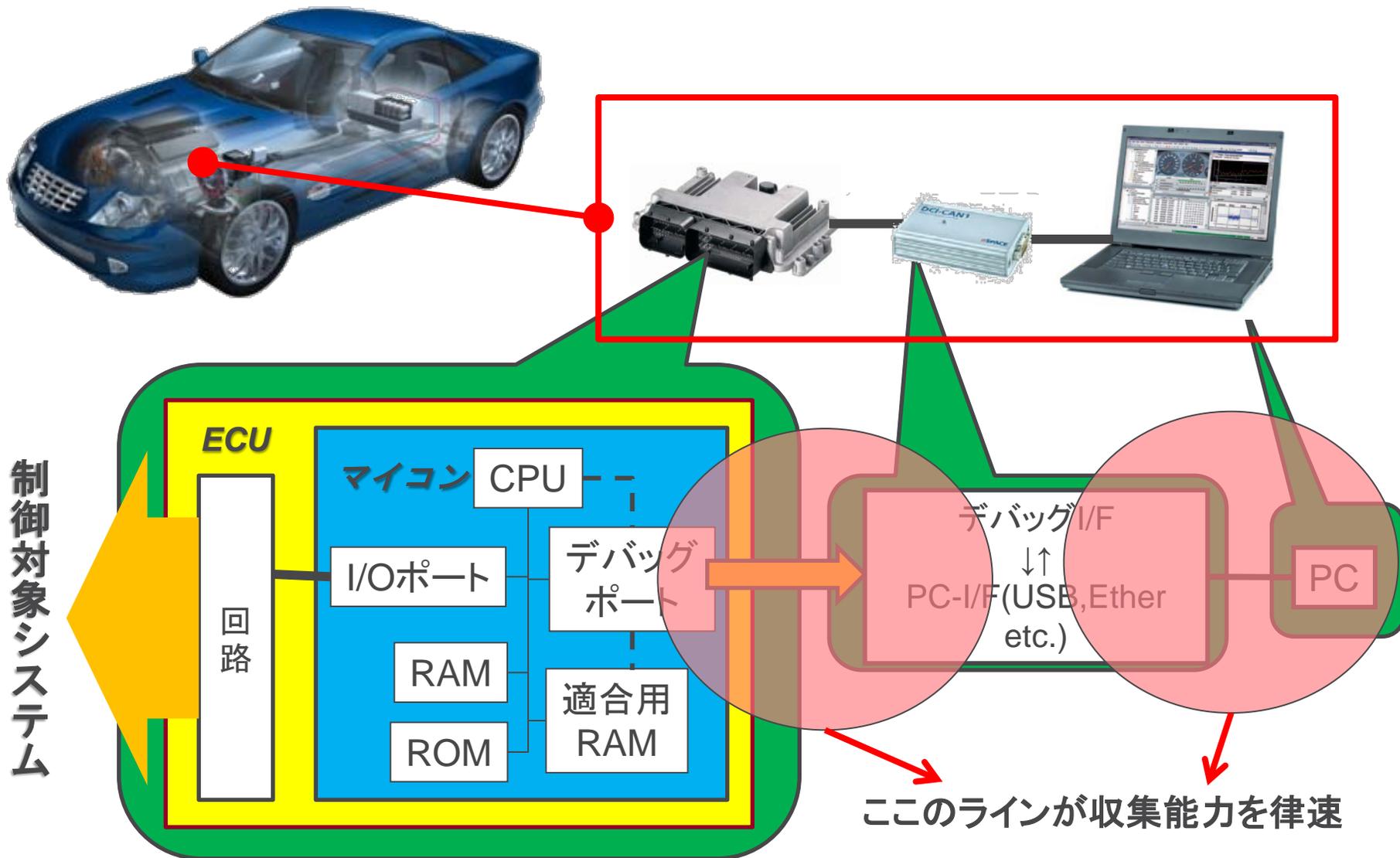


適合データの増加

<要因>

- レガシーコードは容量の半分以上がMapデータ
 - レガシーコードとMBD制御の混在環境が当面継続
- MBDによりより多くのパラメータが必要
 - 高度な制御手法をMap制御に置き換え
 - 高度な制御手法は安価なMCUでは動かない





制御対象システム

このラインが収集能力を律速

適合 = 制御量の計測



数百kHz～10MHzの
実行通信帯域要

車両挙動(物理量)の確認要 → 制御量の時系列サンプリング・解析の実施
上記帯域が確保できない場合:使用するラインの帯域を考慮した工夫要

適合に必要な通信帯域例

<コンベンショナル・ガソリン・エンジン割込み処理の対応>

Max. Rev. 6,000RPMと仮定 → 10ms/回転

通常10°C A～30°C A(280μs～830μs)毎約10Byteのデータを処理と仮定
概算で**1MHz/bit**(概算100kHz/Byte)の実効値が必要

<モータ制御・励磁処理の対応>

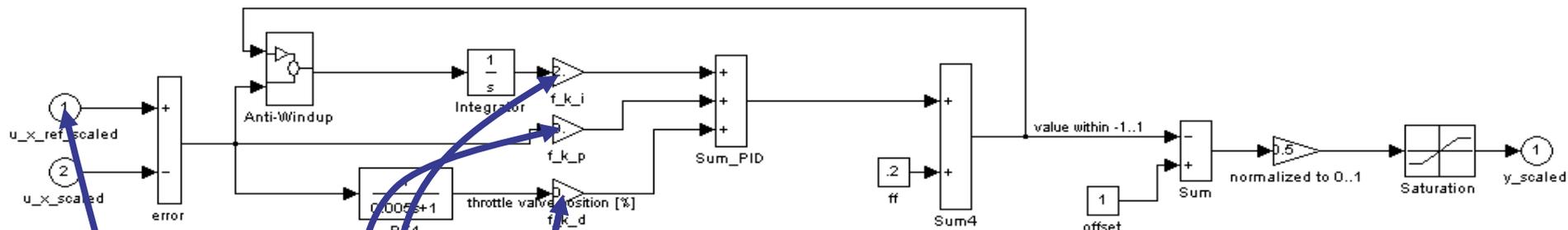
Max. Rev. 12,000RPMと仮定 → 5ms/回転

1°C A(14μs)毎約10Byteのデータを処理と仮定
概算で**10MHz/bit**(概算1MHz/Byte)の実効値が必要

<コンベンショナル・ガソリン・エンジン物理現象の対応>

100ms毎の物理現象200Byteのデータ収集要と仮定

0.5ms/Byteのデータを処理
概算で**200kHz/bit**(概算20kHz/Byte)の実効値が必要



Variable	Size	Type	Origin	Description
y_scaled	1x1	Float64	f_k_p	
Proportional Gain	1x1	Float64	f_k_i	
Integral Gain	1x1	Float64	f_k_d	
Differential Gain	1x1	Float64		

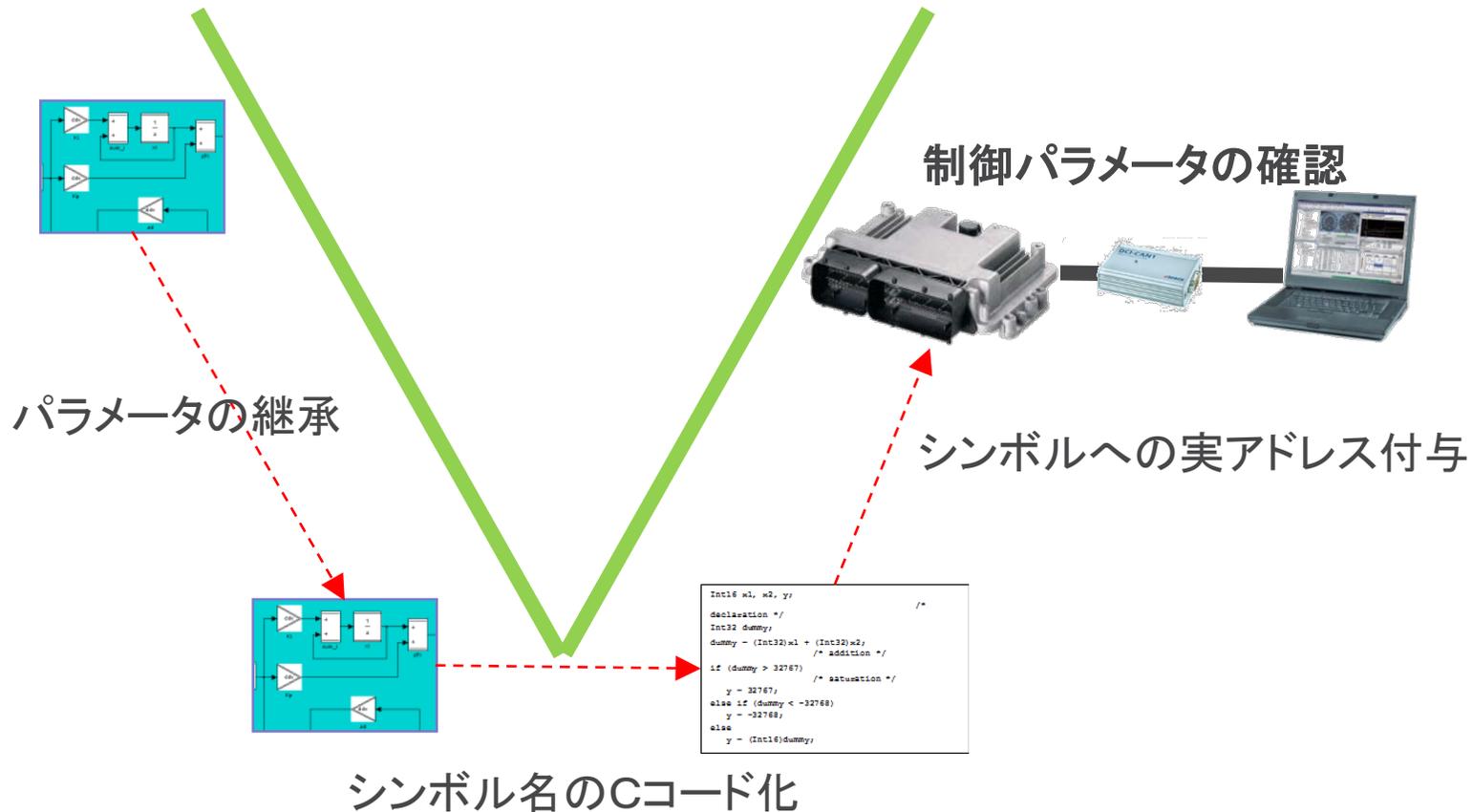
すべてのSimulinkブロック変数にアクセス可能



- モデル上のパラメータ
- Cコード化したシンボル名
- 適合ツールがアクセスするアドレス

これらの情報のリンク要

制御変数・定数をデータ格納領域から収集 → ローカル宣言不可



グローバル・スタティックを中心とした
制御変数・定数宣言



物理現象の制御状態確認

制御変数・定数の構造体宣言は なるべく避ける
(または、構造体メンバー抽出の仕組みを考慮する)



構造体メンバーの
アドレス取得困難

メンバーの出現順
メンバー配置オフセット量(パディング数)

} コンパイラの実装依存

メンバーオフセット抽出に工夫要
(コンパイラメーカーからの実装仕様入手困難)

メンバーのアクセス

構造体メンバー先頭位置から
オフセット量を直値展開

PC上でのシミュレーションと組込ソフトの違いに起因し、幾つかの課題が存在

<代表的な事例>

➤ 構造体宣言による定数・変数のハンドリング

- メンバーオフセット情報取得のための手段とリンク情報の設定対応要
- 構造体宣言メンバー数
→ コンパイラの操作上限を超えるケースが存在

➤ 割込み処理  割込み

```

a = 1;
if ( d > e ) {
    a = 2;
} else if ( d < e ) {
    a = 3;
}
    
```

組込ではNG

PCでは良くないが
組込ではGOOD

```

if ( d > e ) {
    a = 2;
} else if ( d < e ) {
    a = 3;
} else {
    a = 1;
}
    
```

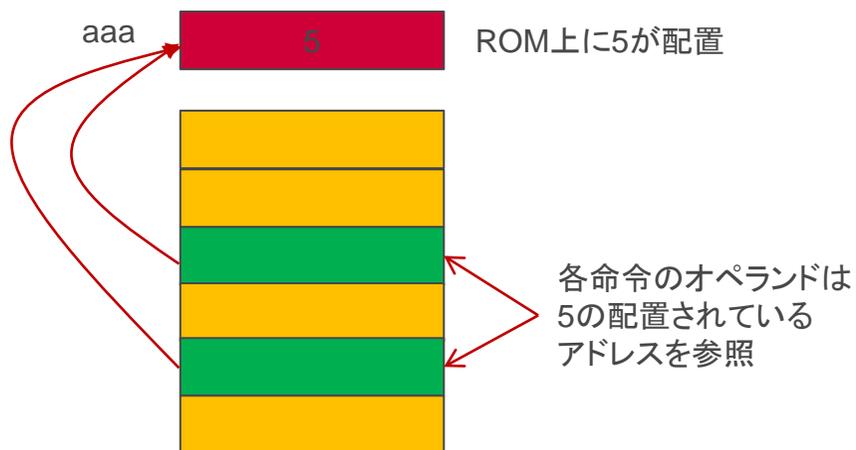
➤ 定数配置への配慮

制御定数のvolatile const 宣言



制御定数の適合効率向上

```
volatile const int aaa = 5;
```



```
const int aaa = 5;
```



aaa定数を適合する場合は、Volatile宣言をした方が効率的かつエラーが少ない

<参考>

制御変数・制御定数は数千におよぶ場合がある

→ Mapファイル等からa2lファイル等へ自動抽出する場合もある

モデル演算 ⇒ 組込ソフトの主たる誤差

- スケールの変換に起因する量子化誤差
- 実時間進行とシミュレーション計算の時間間隔に伴う積分誤差



上記誤差を念頭におかないと、モデルもしくは計測に間違いがあると思ひこむ

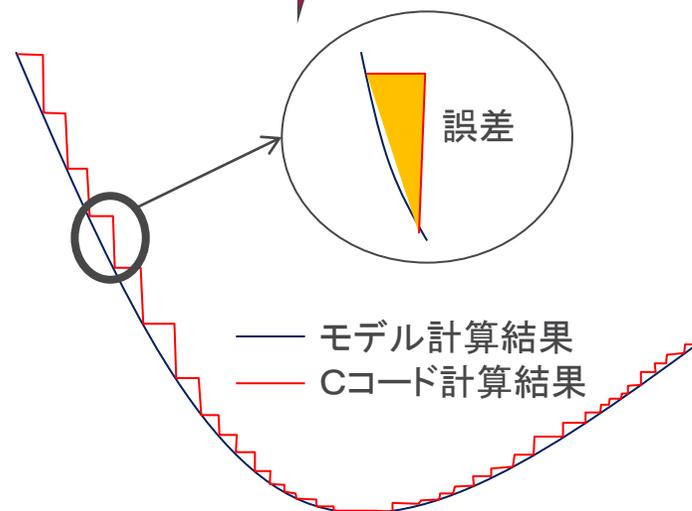
- 思ったとおりのサンプリング値とならない
- 思ったとおりの変化を示さない
- サンプリングデータから物理値への換算値が合わない



確認・追認試験で
無駄が発生

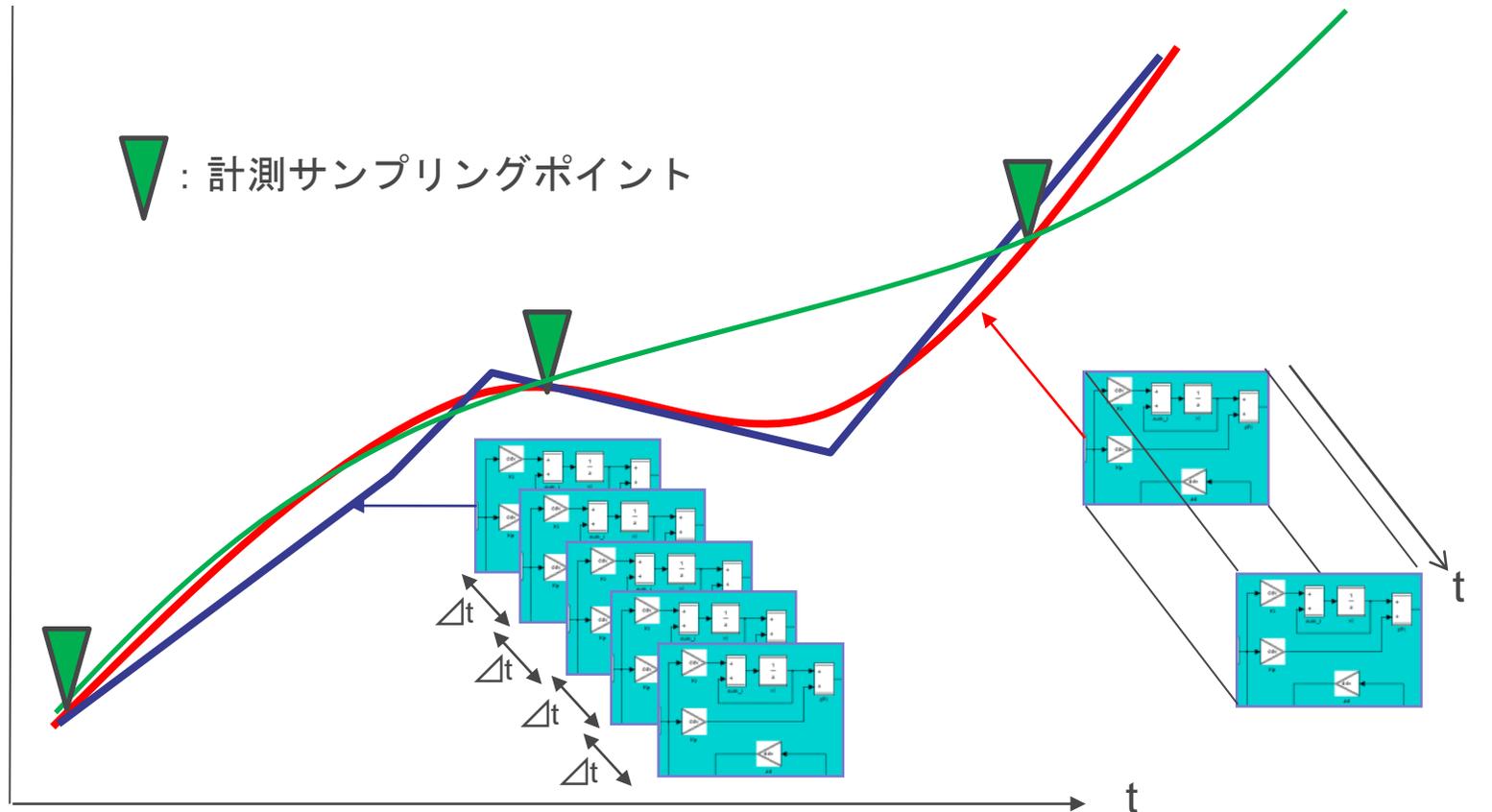
対応例

- モデルシミュレーション(MILS)結果と
Cコード・シミュレーション(SILS)結果 比較
⇒ 誤差を把握
- 誤差把握後、HILSおよび実車で挙動確認



<量子化誤差>

モデル演算値とCコード演算値との演算誤差)



- : 実際の動き
- : シミュレーションの動き
- : サンプリングデータからの推測線 (エリアシングノイズ発生事例)

<参考> $f_s > 2f$ f_s : サンプリング周波数 f : サンプリング対象周波数

まとめ

➤ 操作するパラメータ量に応じた、計測・適合環境の整備

➤ 後行程(適合)を考慮したモデル作成要

制御モデル設計時の情報が適合まで継承される

➤ モデルと現実の違い意識した適合・計測要

PCシミュレーションと組込ソフトの違い

量子化誤差

時間積分の誤差蓄積 等

Thank you



© Copyright 2014 by dSPACE.
ブランド名および製品名は、各社または各組織の商標または登録商標です。