

ムラタ流MBD

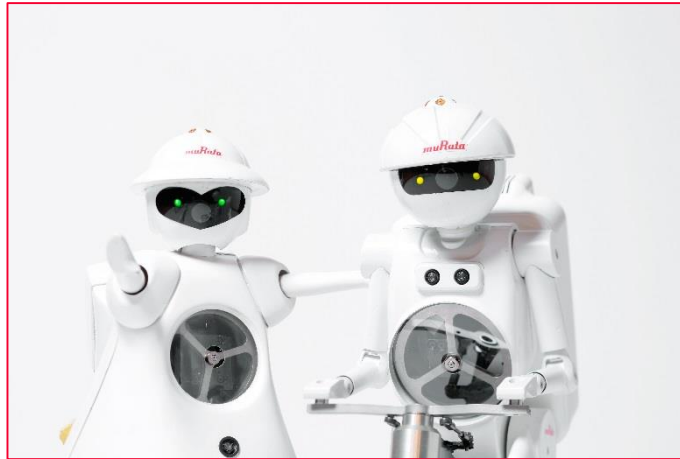
エネルギーマネジメントシステム(EMS)
向け組み込み開発の事例

株式会社村田製作所

馬 躍



村田製作所について



村田製作所は、最先端の技術、部品を創出する総合電子部品メーカーです。Innovator in Electronicsをスローガンに掲げ、豊かな社会の実現をめざします。

ムラタの強み

- 最先端の材料を研究開発
- 広範囲な製品ラインナップ
- グローバルな生産、販売ネットワーク

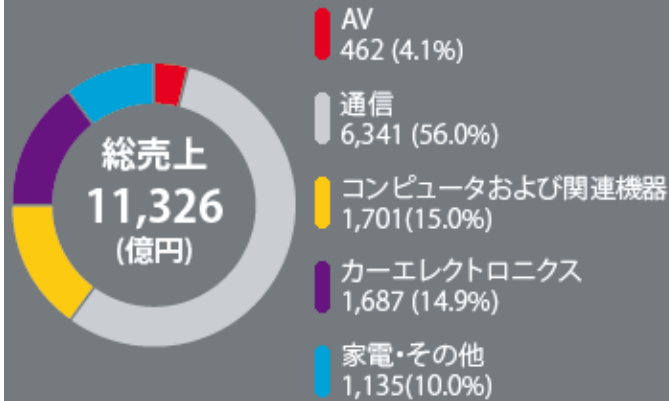
ムラタのプロフィール

- 売上高： 1兆1千355億2千4百万円
- 従業員数： 59,978名
- 企業数： 97社（国内31社、海外66社）
- 創業： 1944年

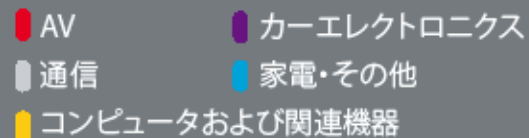
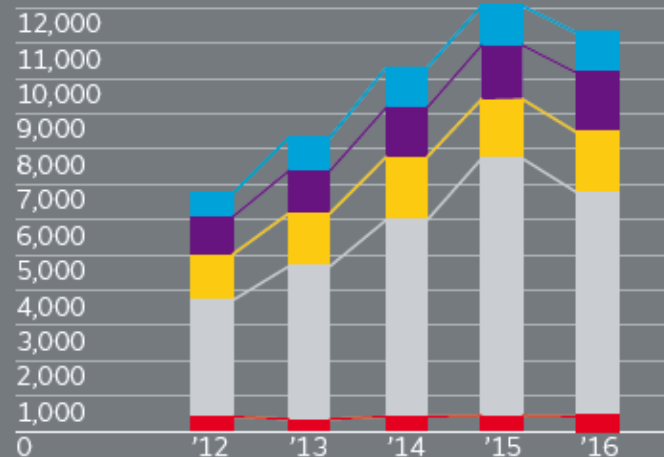
※売上高は、2017年3月期決算。
※従業員数は2017年3月31日時点のものです。
※グループ企業数は2017年3月31日時点のものです。
※村田製作所はグループ企業数に含まれておりません。

村田製作所の事業領域

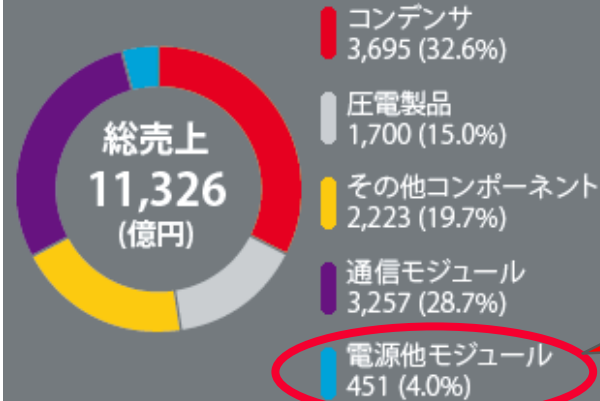
用途別売上高



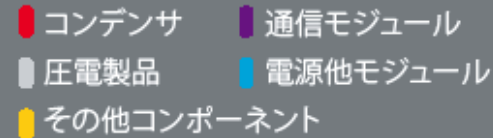
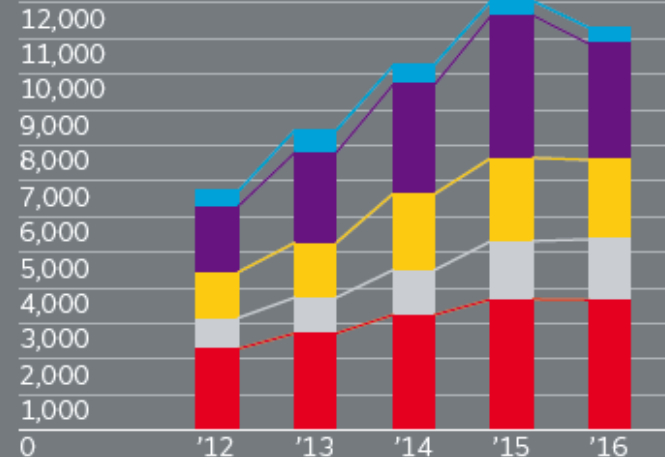
用途別売上高
(億円)



製品別売上高



製品別売上高
(億円)



パワエレ技術を活
かしてEMSの開発
を目指して、ここ
からスタート
(六年前)

#EMS = Energy
Management
System

エネエース：住宅用小型創蓄システム

長瀬産業、村田製作所 軽量・コンパクトな太陽光発電・リチウムイオンバッテリーハイブリッドシステムを共同開発

2017/04/13

株式会社村田製作所
代表取締役社長 村田 恒夫



※2 電力会社からの出力制御要請に対応します。インターネット回線への接続およびソフトウェアのアップデートが必要です。

※3 端末はセットに含まれません。見える化ユニットの機能をご利用いただくには、お客様にて端末をご用意して頂く必要があります。

<https://eneease.com/>

家庭用コンパクトHEMSセット



ソニーエナジーデバイス⇒村田製作所

ソニー株式会社からの電池事業の取得の完了に関するお知らせ

2017年9月1日

当社は、2016年10月31日付「ソニー株式会社からの電池事業の取得に関するお知らせ」および2017年7月19日付「ソニーから村田製作所への電池事業の譲渡完了の見通しに関するお知らせ」において公表いたしましたとおり、本日、村田製作所グループによるソニーグループの電池事業（対象事業）の譲受けが完了いたしましたので、お知らせいたします。

これに伴い、本譲渡のためにソニー株式会社が2016年12月9日に設立し、本日を効力発生日とするソニーグループによる会社分割により対象事業を承継した株式会社東北村田製作所を当社が取得いたしました。

当社は、ソニーグループの培ってきた電池事業の技術力と事業経験を承継し、注力市場であるエネルギー分野の中核事業にすえ、「中期構想2018」における経営目標の実現に向けた取組みを実行してまいります。

なお、2018年3月期の当社連結業績に与える影響につきましては現在精査中であり、確定次第お知らせいたします。

成長戦略

1. 通信市場での競争優位の追求
2. 注力市場での事業拡大
3. 更なる長期を見据えた市場開拓

本題に戻ります

以上は会社 P R、
ありがとうございました！



- ▀ **M** Murata's
- ▀ **B** Better (maybe not the best)
- ▀ **D** Decision

開発品や組織の現状と目標を分析して
村田製作所（にある我々の部門）にとって
最適だと思われるモデルベース開発の進め方

V字型MBDは素晴らしいですが



処理速度の差、離散と連続の差、アナログとデジタルの差、ノイズの影響、など、実機とモデルの差が存在

！ RCP≠マイコン！ HIL≠実回路！

コスト（金銭）：
RCPとHIL機器は高価

コスト（時間）：
RCPとHILの実験に人員と時間を投入

出典：MathWorks®社セミナー資料

業界別コスト対効果の差

	Non-MBD Cost	MBD Cost	MBD Advantage
Telecom/Datacom	\$6,279,861	\$3,224,478	94.9%
Auto/Transportation	\$3,151,078	\$2,270,597	38.8%
Ind Automation	\$1,593,047	\$1,605,783	-0.8%
Medical	\$2,269,310	\$1,265,059	79.4%
Mil-Aero			Special analysis

出典：
Comparing Embedded Design Outcomes
With and Without Model-Based Design.
2010,
American Technology International

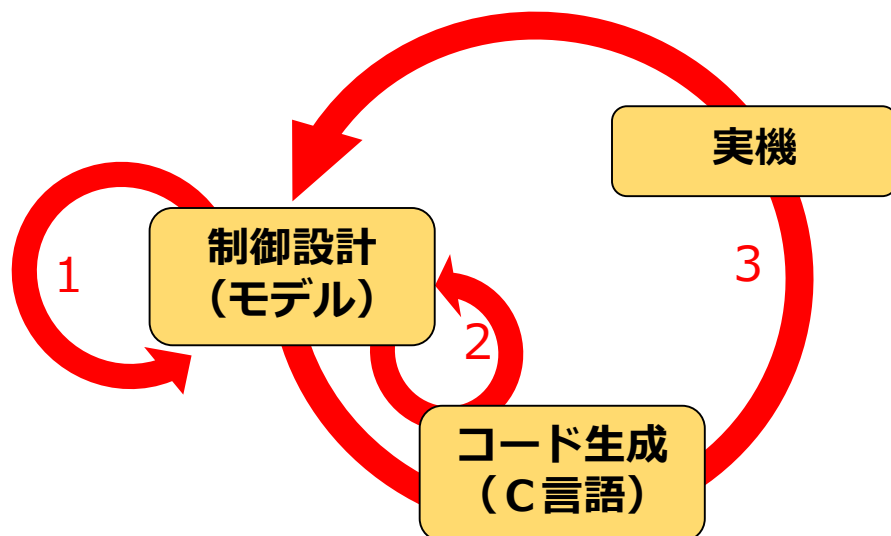
Table IV-1: Total Development Costs and ROI Advantage by Market

The industrial automation market segment is quite diverse and design upgrades are typical of this segment. Our experience indicates that industrial automation still relies heavily on physical prototyping, and the use of modeling is still relatively new. So, this data might reflect the need to develop stronger modeling skills in that workforce. Also, there are applications in industrial automation that are so simple that MBD isn't warranted.

- ① ソフトウェアの割合小
- ② 参入遅れ
- ③ スキル不足
- ④ 案件が簡単の場合がある
- ⑤ ハードウェアが安い

電源分野の場合、自動車・宇宙開発・医療機器と同じ手法で進むと、Advantageが出ない場合がある。必要なのは、それぞれのアプリケーションに相応しいコスト対効果の高いモデルベース開発手法である

3 ループ型簡易MBD



3 L o o p s

1. シミュレーションベースのモデル設計（マイコンの動作を意識する制御モデル作成 & 実回路を意識するプラントモデル作成）
2. コード生成、**コード確認**と最適化のためのモデル変更
3. 実機デバッグによるモデル修正 & チューニング

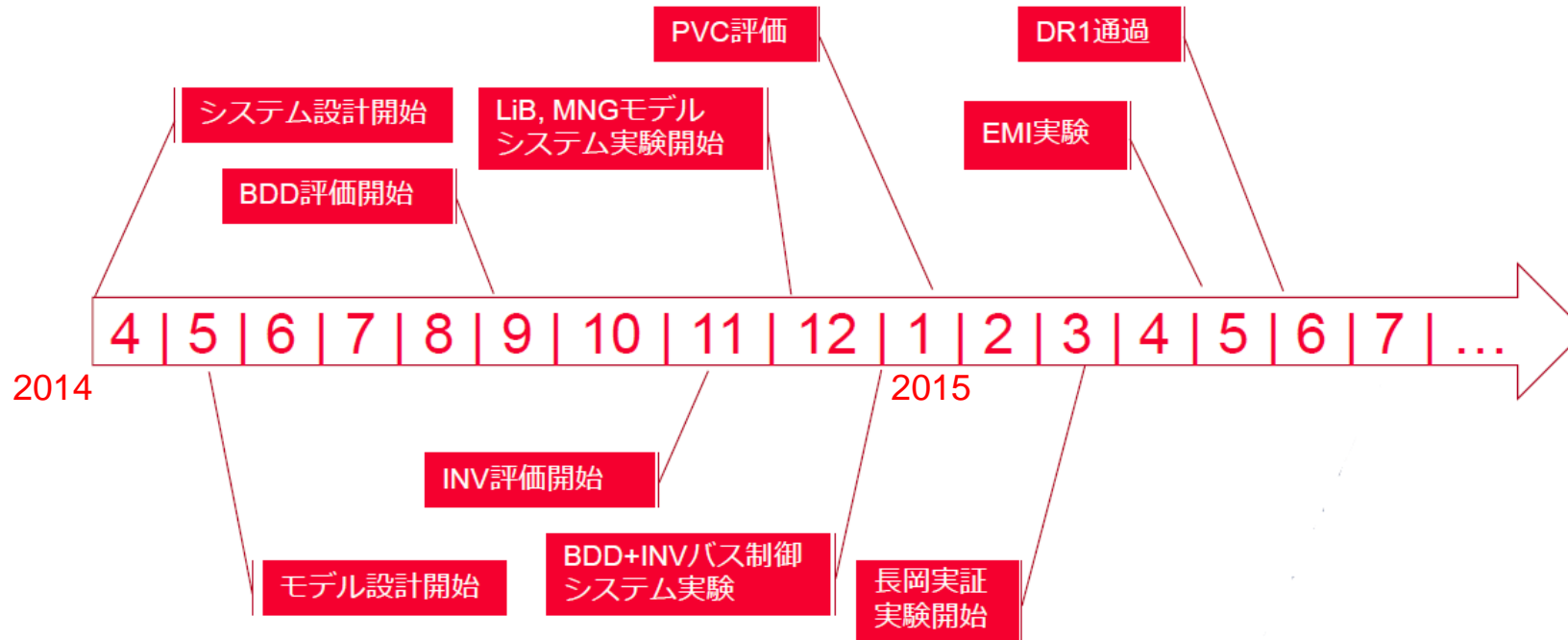
高速・低コスト・高信頼性のモデルベース開発手法の確立を目指す

結果（例 1）

試作品：ラック型5kW三相EMS



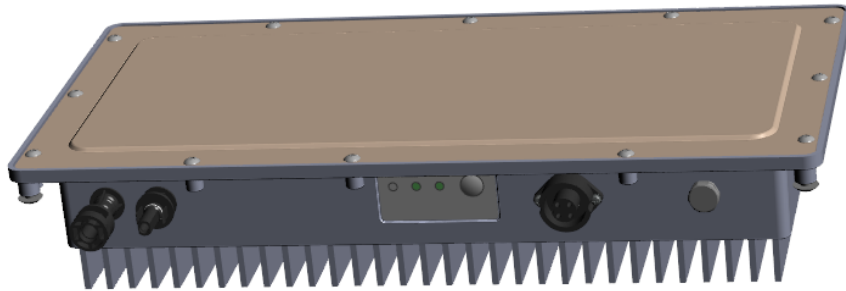
開発時間が大幅低減



○ 工数**二人で半年以内**（専任ではない）（一人MBD初心者）

○ ペリフェラル設定、制御、通信、LCD表示、EEPROM操作を含むすべてのソフト
100%自動コード生成 ⇒ ハンドコード 0 なので信頼性向上

試作品：太陽光発電ミニインバータ



- ALL in ONE構成
- マイコン一枚
- MPPT、系統連系動作、系統連系保護機能、通信、など含むため、処理時間が厳しく、モデルの最適化とタスクの分散設計に苦労

- **時間短縮**：若者二人中心で半年（完成形じゃない）
- **入門しやすい**：ソフト初心者の一人もOJTで成長

少し中身を覗いてみよう

ほんとうに少しだけね



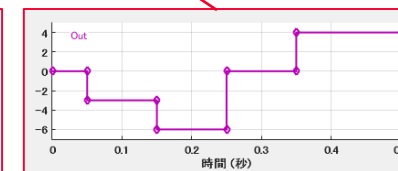
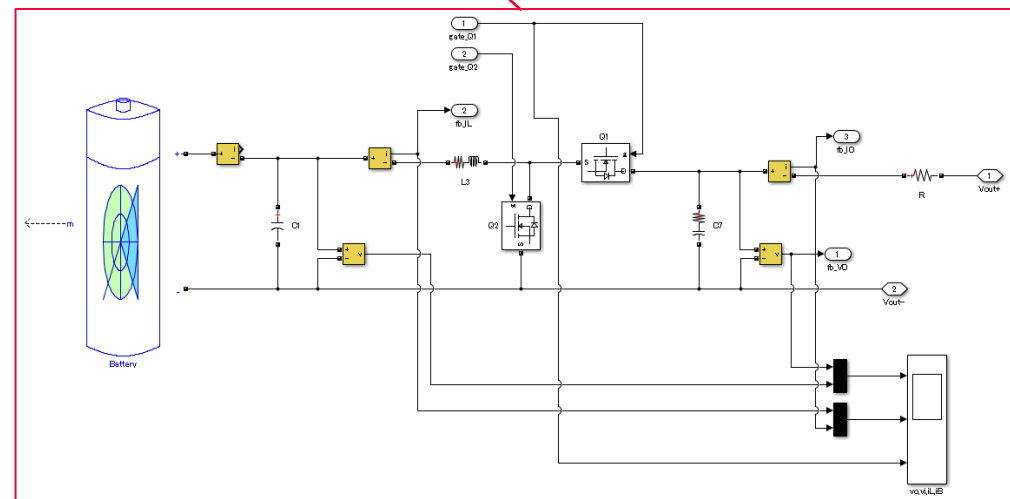
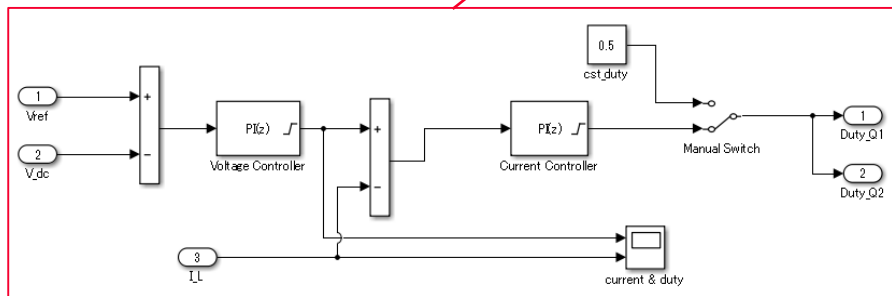
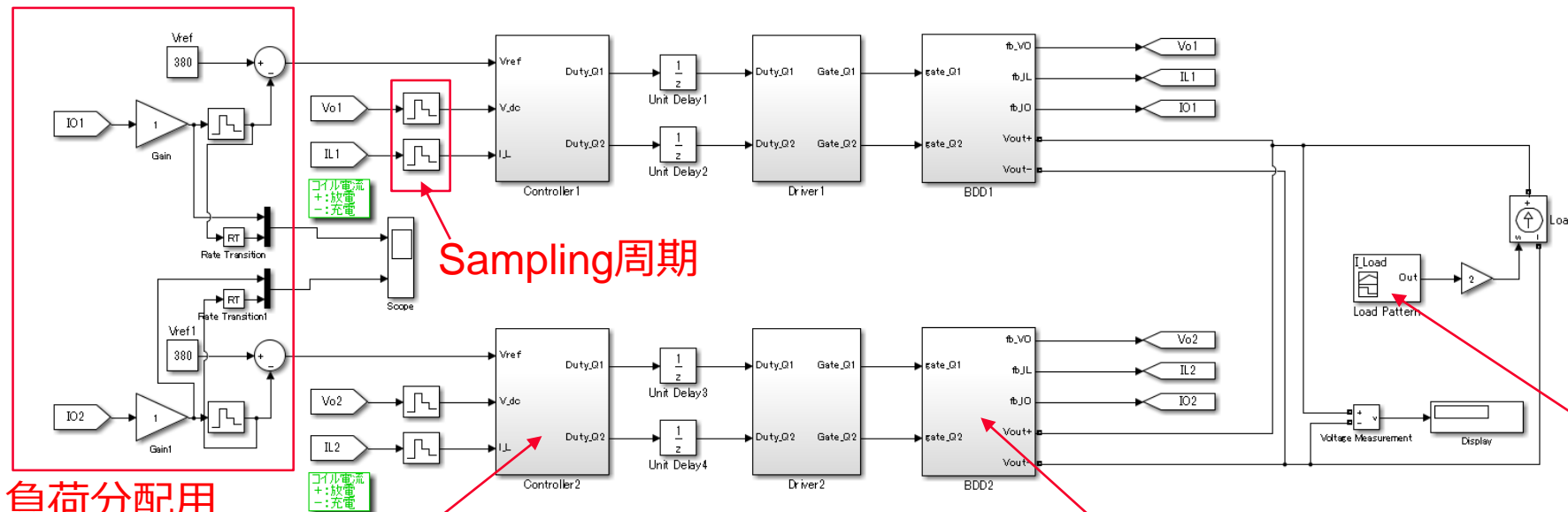
目的

- 主に初期検討用
- 回路方式や常数決め、制御器構成、原理検証、パラメータ決め
- 制御器の検討はTF（伝達関数）ベースが多いが、線形化し難い場合は実回路モデル

SimPowerSystems™を使用

- SimElectronics® より速い
- 3rd Partyツールより親和性がよい
- ライブラリが豊富、THDやRMSの計算機能も便利
- R2016a以降で、「Simscape™ Power Systems™」に名称変更されて、機能改善もされたようですが、ムラタはまだ従来のSimPowerSystemsを使用。（いずれ替わるでしょうが）

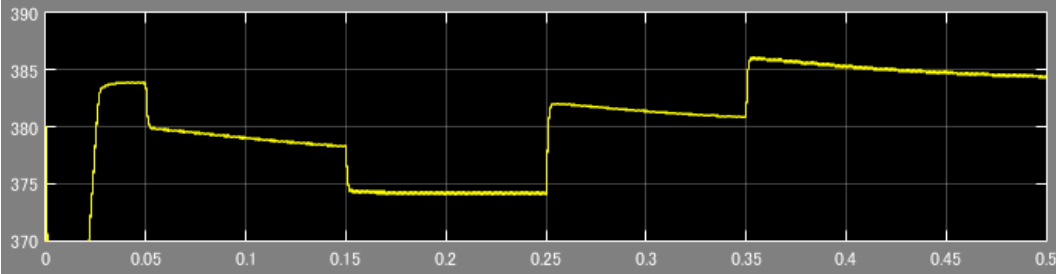
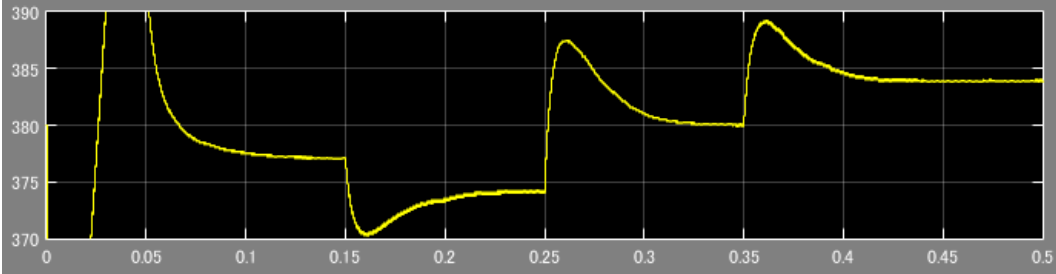
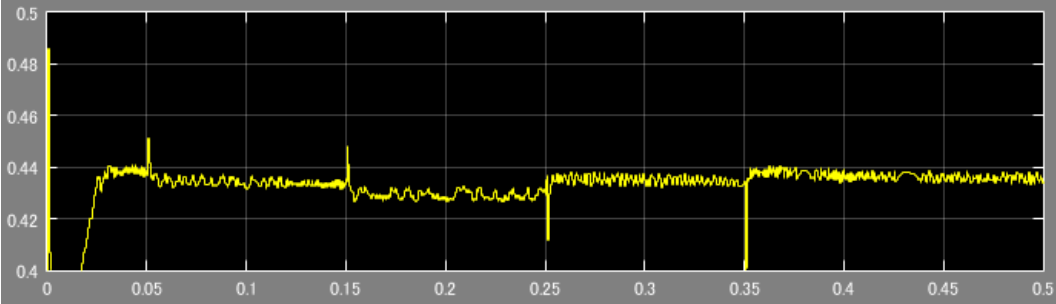
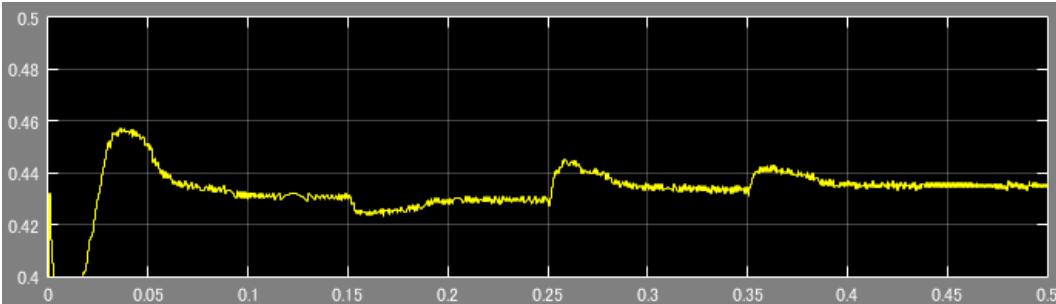
例：インタリーブ双方向DC/DCコンバータ



双方向boost回路

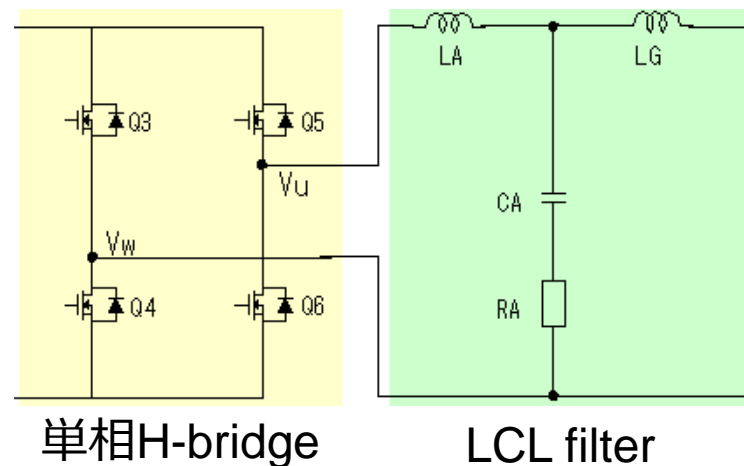
Simulation例



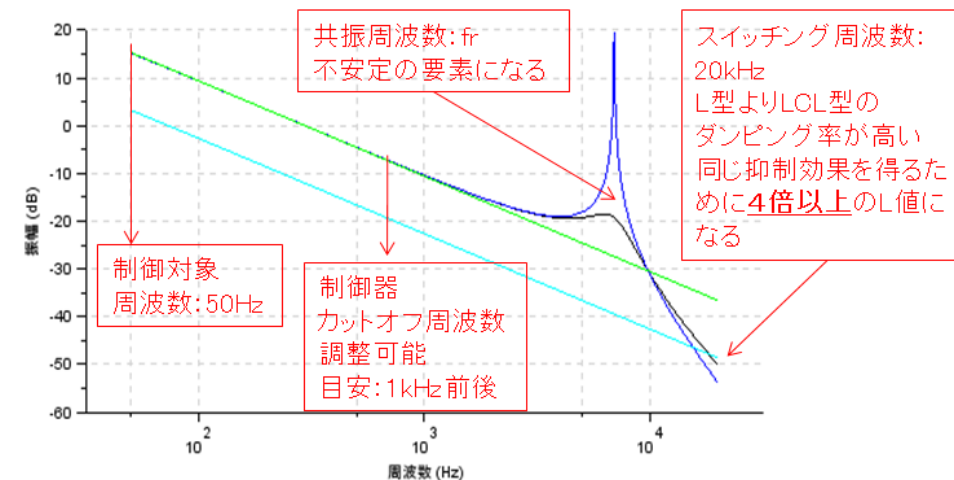
	電圧制御器比例ゲイン = 5	電圧制御器比例ゲイン = 1
バス 電 圧		
D U T Y		
比 較	電圧overが小さいが、収束に時間が掛る Duty値の揺れが大きい	電圧overが大きいが、収束が速く、 Duty値が安定

回路常数、サンプリング周波数、離散化、等の設定は実回路と一致すれば、制御器構成&回路Topology検討、制御&回路パラメータのチューニングなど、事前確認ができる（実回路で見えないものも可視化）

伝達関数ベースの例：LCLフィルター



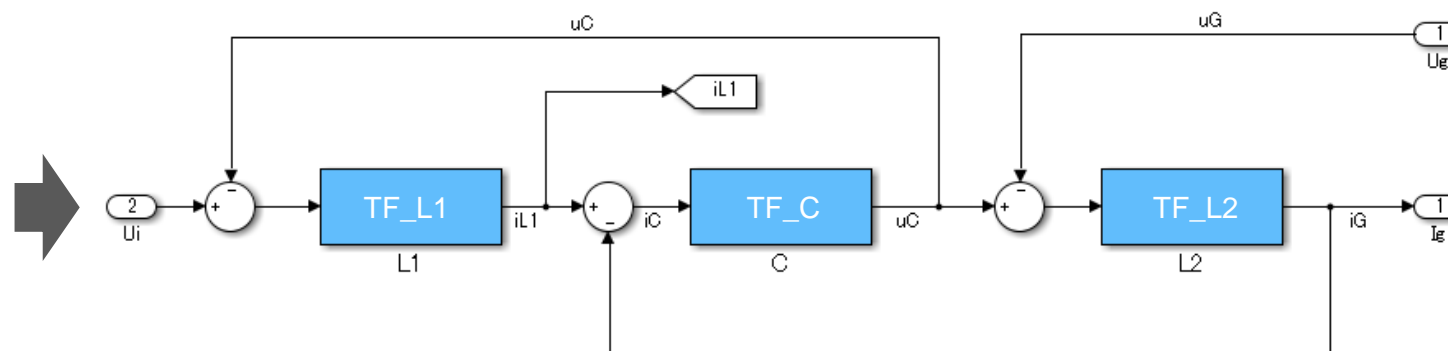
- LCL型フィルター
- LCL+R型フィルター
- L型フィルター (同等インダクタンス)
- L型フィルター (4倍インダクタンス)



Pro: インダクタの小型化; Con: 不安定要素; 高度な制御にセンサーが増える
目標：一つのPI制御器で高精度な電流制御

【伝達関数】

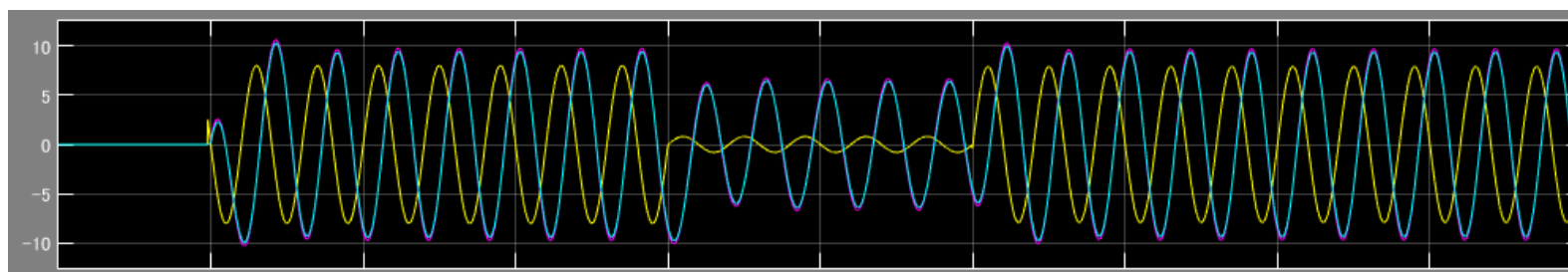
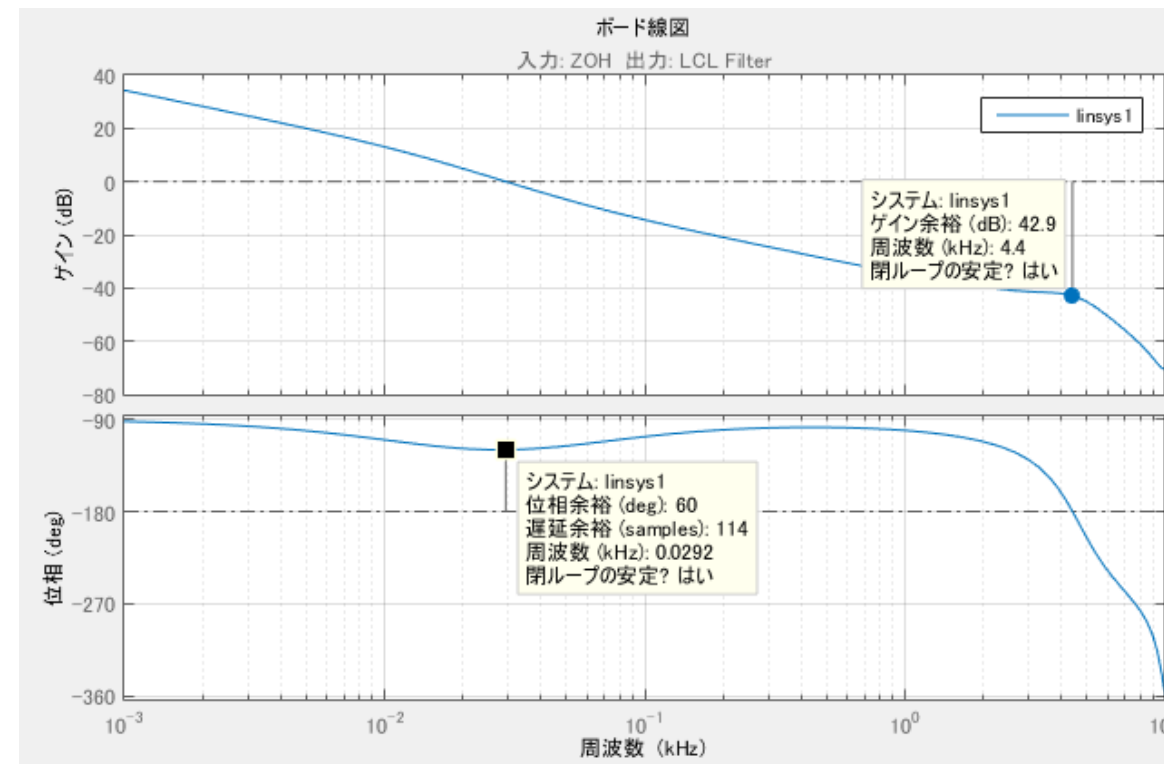
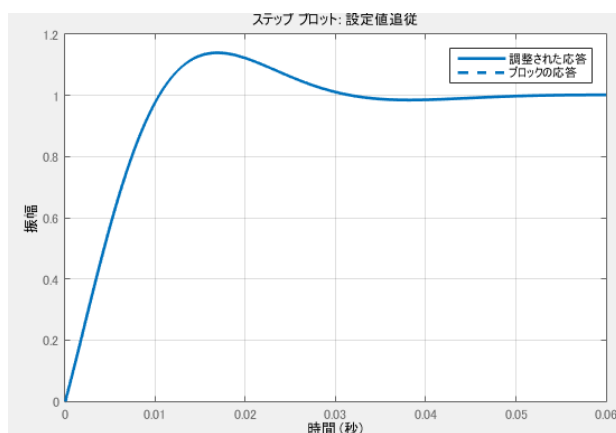
$$G(s) = \frac{i_{L_g}}{V_i} = \frac{R_d Cs + 1}{LL_g Cs^3 + (L + L_g)R_d Cs^2 + (L + L_g)s}$$



Control System Toolbox™を用いて自動調整

ワンクリックでControl System Toolboxから
PI制御器を自動設計してくれた

右 : open loop Bode diagram
下 : unit step response



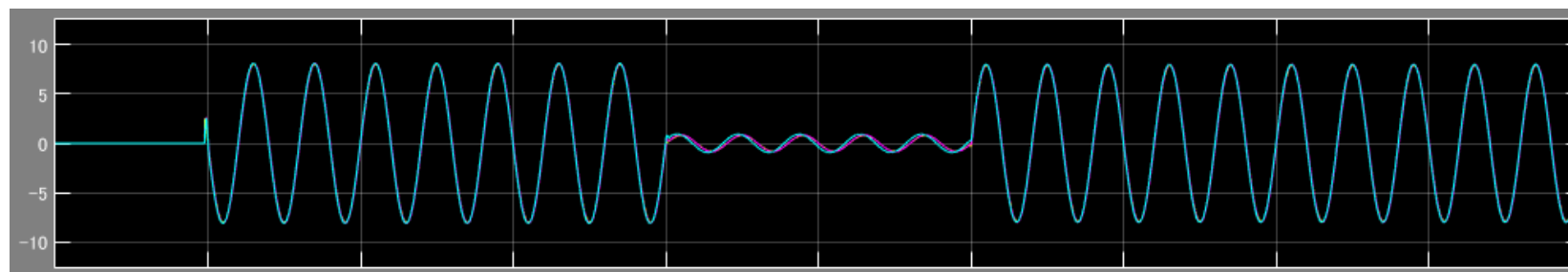
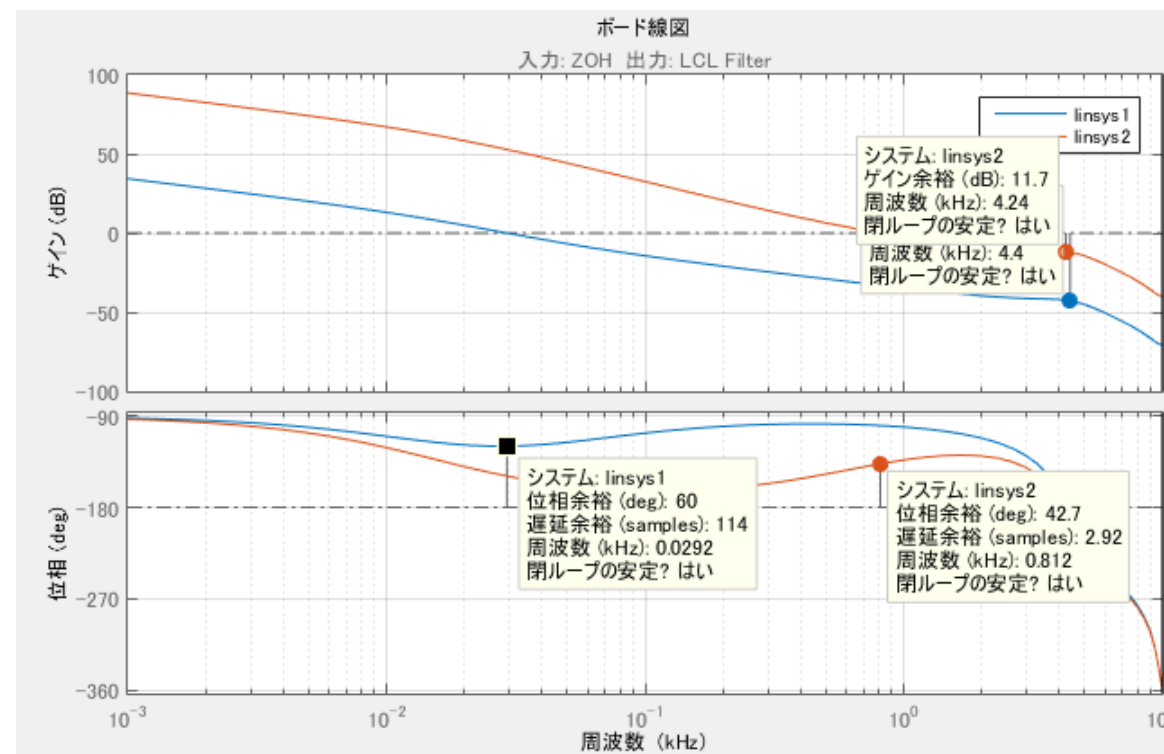
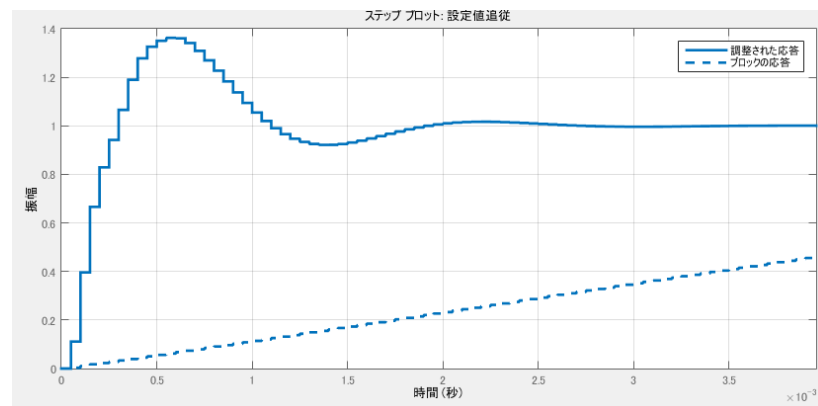
結果：系統連系の応答が遅い
(指令値黄色)

！指令値は50Hzのサイン波

Control System Toolboxで再調整

ゲインを調整し、安定性を確保しながら、
応答スピードを上げ

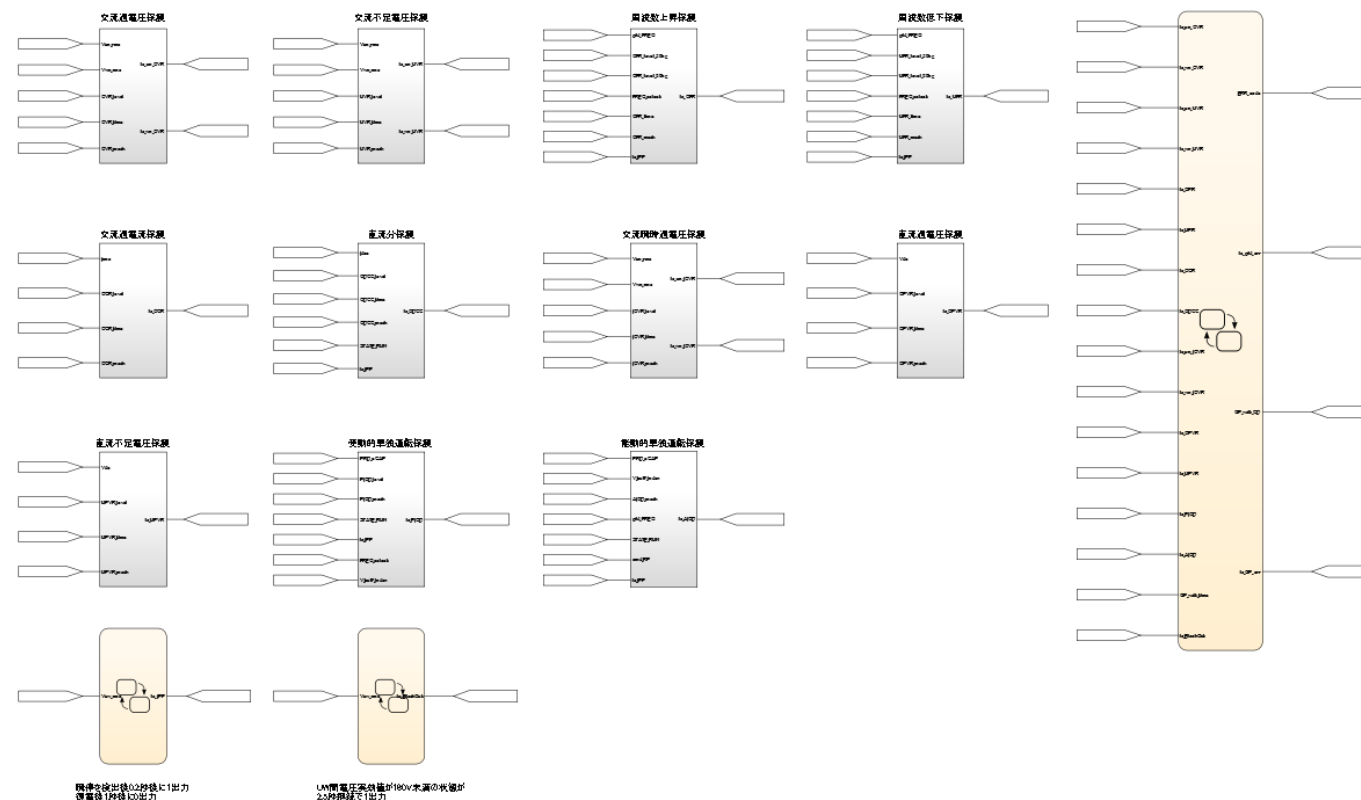
右 : open loop Bode diagram
下 : unit step response



Simulation結果OK

同じゲインで実機もTHD<5%
の電流制御品質確認

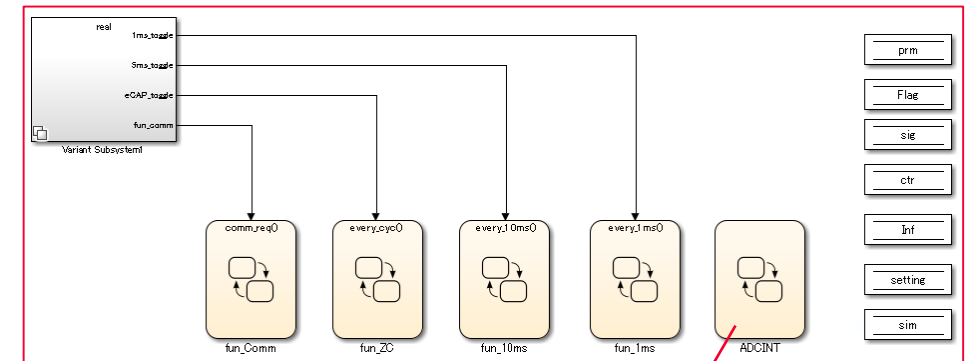
複雑な系統連系保護も項目ごとモデル



系統連系試験の実機テストは煩雑であり、
シミュレーションで系統のアブノーマルを簡単に再現することができ、
保護試験のロジック、タイマー、相互影響などを事前確認し、
開発期間を短縮することができた

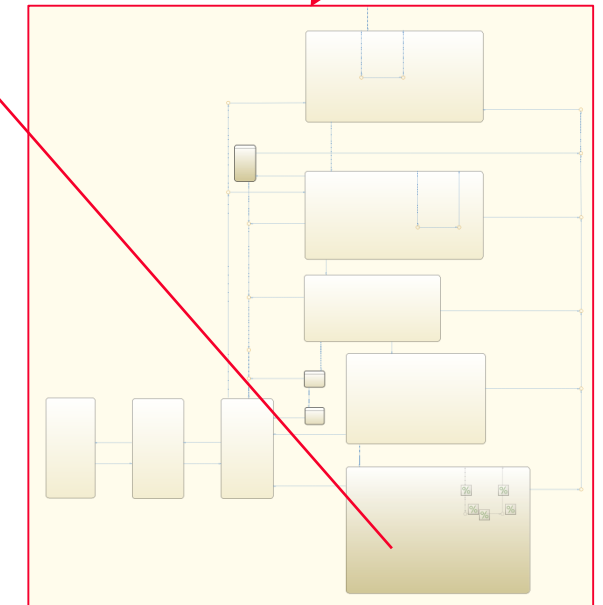
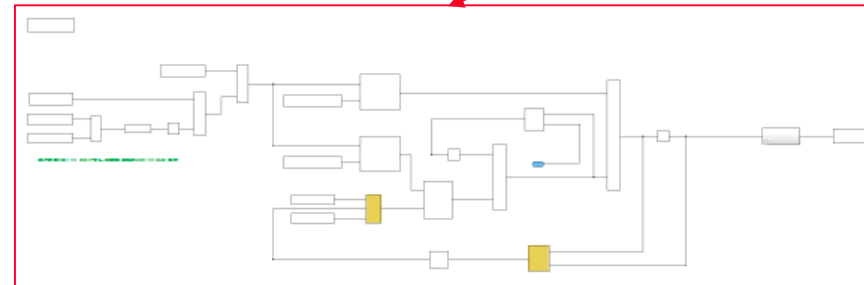
Embedded Coder[®]による100%自動コード生成

- Top layer : 割り込み関数定義、タスク配分
- 2nd layer : Stateflow[®] による状態遷移設計
- 3rd layer : Simulink[®] による制御設計、マイコンペリフェラル設定
- Data Structure: Global, variable
- ...



```
pwm_boost(ctr_VPV_ISO(ctr_VDC2IAC(NO_RESET);  
//pwm_bridge_halfcopper
```

Simulink Function ctr_VDC2IAC(rst)



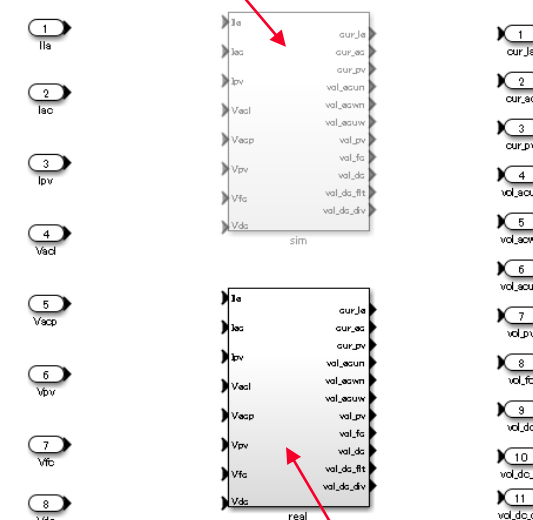
100%自動コード生成のモデル作り

- 割り込み, ADC, PWM, GPIO, 通信, などマイコンのブロックセットを徹底的に使用
 - もちろんマイコンのマニュアルを参照（機能を把握, REGを覚える必要ない）
 - 検証済み：TI (C2000), Microchip (dsPIC), Renesas(RXファミリ), Raspberry PI
- Stateflow[®] は骨、Simulink[®] は肉（小規模な実装はSFなくても良い）
- JMAABのガイドラインを意識（こまめな点検が必要、可読性向上）
- **必ず生成したコードを確認！**
 - 多くの場合はモデル変更でダイエット可能
 - Stateflow[®] に書いたロジックとタイミングはコード上で一致するか
 - 自動生成した変数名は分かりにくいことが多い
- **モデル作る前にデータ構造設計**

Variant SubsystemでSimとCGモデルを統合

- シミュレーションで検証したモデルをそのままコード生成したいが、入出力は物理的に別
- 以前、二つのモデル間コピペ、コメントアウト、予備ポート用意などの方法
- MathWorks 社と相談し、教えていただいた
- Variant Subsystemを使用すると、シミュレーション用とコード生成用は同じモデルで設計することができ、制御変数で用途を切り替えできる
- 管理性、作業性向上

シミュレーション用



コード生成用

So far so good, so what's next?

Betterへ常に変化もムラタ流



■ Murata's Better Decision

- 枠組みを超えて、問題解決にどんな手段がよいかを考えること

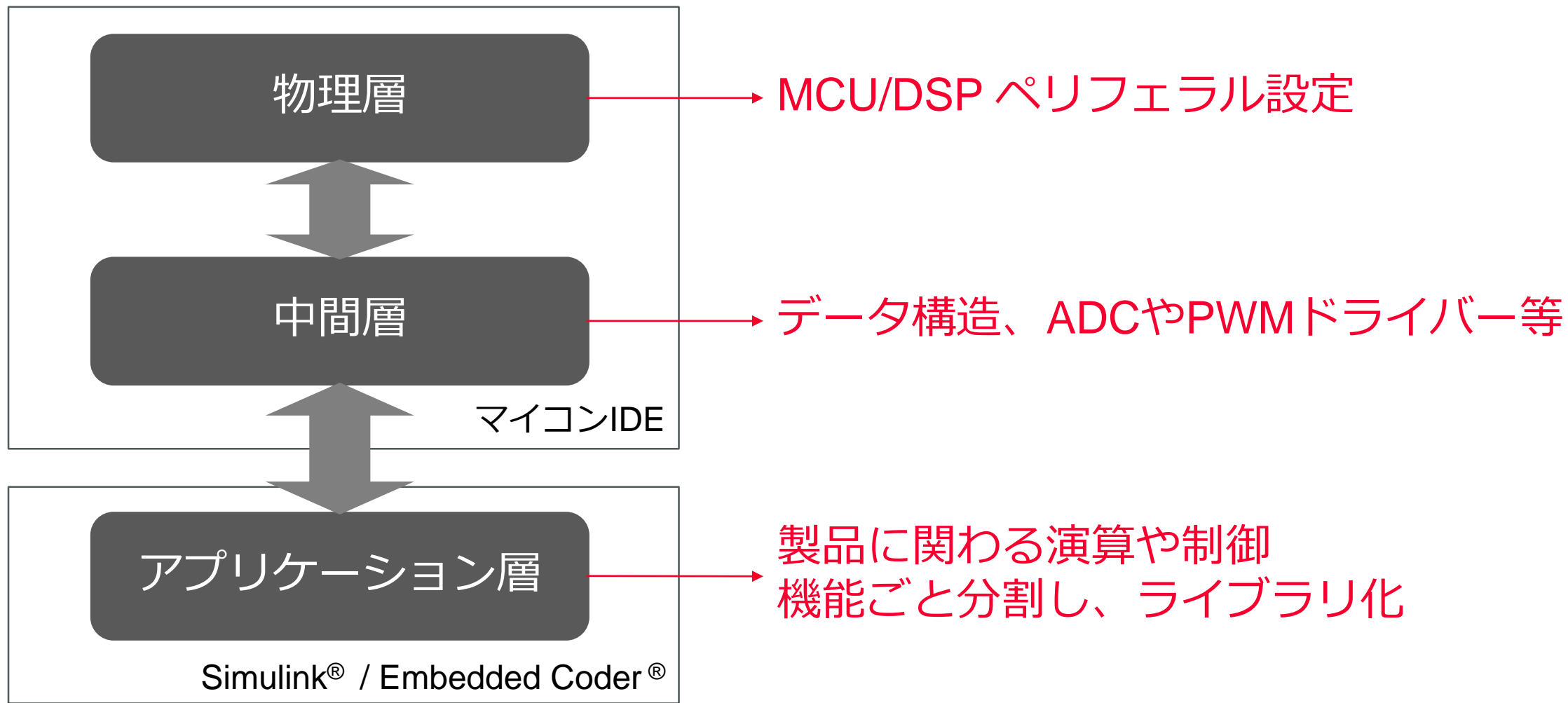
■ 環境変化

- 組み込み技術者の増員
- 開発品ラインナップの増加
- ソフトウェア開発の組織作りは縦型⇒横型
- モデルは巨大化
- 機器に情報処理系のタスクが増加

■ Doing

- C言語の開発リソースと融合するモデル作り
- モデルのモジュール化、ライブラリ化
- 上流設計にシステムエンジニアリング導入

モデル作りはアプリケーション層に限定



まとめ？

- ❖ エネルギーマネージメントシステムの開発にMathWorks 社製品群を活用し、開発時間の短縮と信頼性の向上に著しく効果がありました
- ❖ 人材育成にも注力して、輪を広げています
- ❖ 社内に「MATLAB 研究会」の活動があり、幅広く情報交換して、共にスキルアップしていきたい
- ❖ **一番重要なことは「困ったら、MathWorks社と相談！」**



Thank You Very Much

Ma, Yue 2017/10/31

@ MATLAB EXPO 2017 Japan

