

MATLAB[®] /Simulink[®]を活用した 電源システム設計フロー紹介

MathWorks Japan
アプリケーションエンジニアリング部
初井良治

内容

- MATLAB®/Simulink® アナログ・ミックスドシグナルシステム設計フロー
 - 適用事例(アナログ・ミックスドシグナル)
 - システム設計フロー
 - MATLAB®/Simulink®の利点
 - MATLAB®/Simulink®システム設計フロー

- システム設計ワークフロー(電源設計を中心とした適用例)
 - システム設計(モデリング)
 - システム設計(制御設計)
 - 回路設計・検証(アナログ)
 - 回路設計・検証(デジタル)

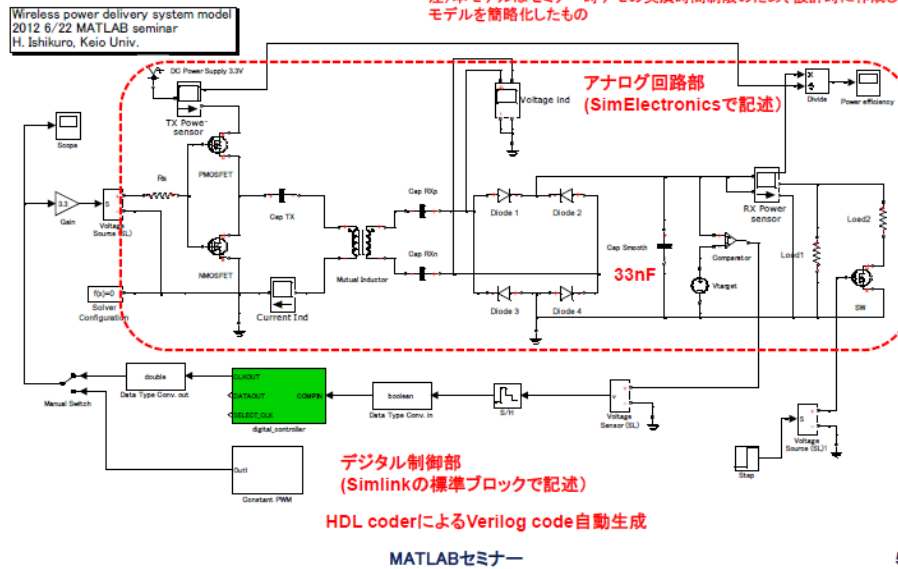
内容

- MATLAB®/Simulink® アナログ・ミックスドシグナルシステム設計フロー
 - 適用事例(アナログ・ミックスドシグナル)
 - システム設計フロー
 - MATLAB®/Simulink®の利点
 - MATLAB®/Simulink®システム設計フロー
- システム設計ワークフロー(電源設計を中心とした適用例)
 - システム設計(モデリング)
 - システム設計(制御設計)
 - 回路設計・検証(アナログ)
 - 回路設計・検証(デジタル)

事例紹介: 慶應義塾大学

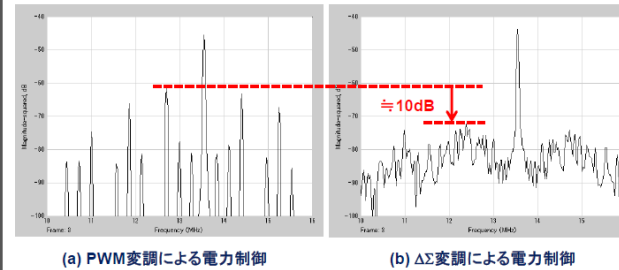
非接触給電用ミックスド・シグナルシステムのトップダウン設計

非接触給電システムモデル (MATLABデモ)



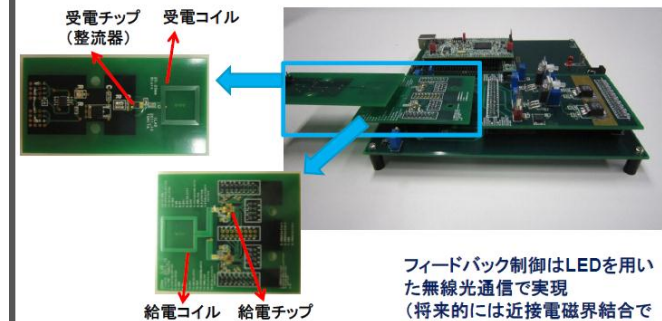
55

インダクタ電流のスペクトラム



変調信号のランダム化によりスプリアスレベルを10dB程度低減

非接触給電デモボード



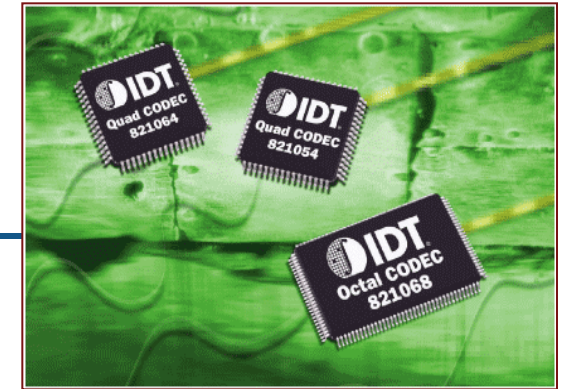
フィードバック制御はLEDを用いた無線光通信で実現 (将来的には近接電磁界結合で実装予定)

MATLABセミナー

59

事例紹介: IDT-Newave

数ヶ月単位での半導体設計期間短縮



音声帯域CODECチップ

チャレンジ

システム設計者と回路設計者の共同作業を確実にすることによる半導体設計プロセスの改善

シミュレーション

MathWorks製品を使用した、システムチームと回路チームへの統合開発環境の提供

結果

- シミュレーション時間短縮
- アルゴリズム欠陥の迅速な特定
- 次回製品リリース時のモデル再利用

“MathWorks製品を使用して、開発プロセスで3倍の効率を達成しました。”

Liu Xin
IDT-Newave

事例紹介: エプソントヨコム

ミックスドシグナル集積回路を2 か月で設計・検証

課題

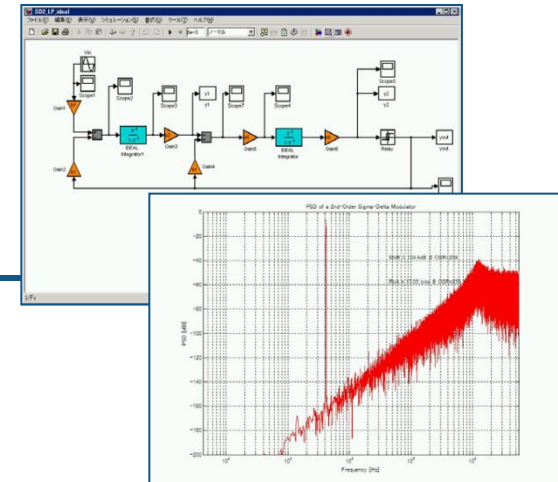
新しい16ビットADコンバータを含むミックスドシグナルICを2ヶ月で開発すること

ソリューション

システムレベル設計をモデリング、シミュレーション、検証するためにMathWorksのモデルベースデザインツールを採用

結果

- シミュレーション時間を数日から数分に短縮
- 開発期間を33%短縮
- 大幅な設計コスト削減



△ΣADCの Simulink モデルと、パワースペクトル密度のプロット。

「従来、回路レベルで3日、Verilog-Aで20分と非常に時間がかかっていたシステムシミュレーションにMATLABおよびSimulinkのシステムモデルを採用することにより、1分に短縮することができました。これにより短時間にかつシステムティックに各設計パラメータを決定し、システム仕様を決定することができました。」

上原 純 様
エプソントヨコム株式会社

MATLAB/SimulinkがSTARC認定ツールへ

- プレスリリース 2013年4月9日
STARCが MATLAB および Simulinkを新STARCAD-AMS
設計フローにおける 推奨設計ツールとして認定
 - モデルベースデザインワークフローおよびツールがミックスドシグナル
半導体の システムレベル設計と検証の効率化に有効と認められる

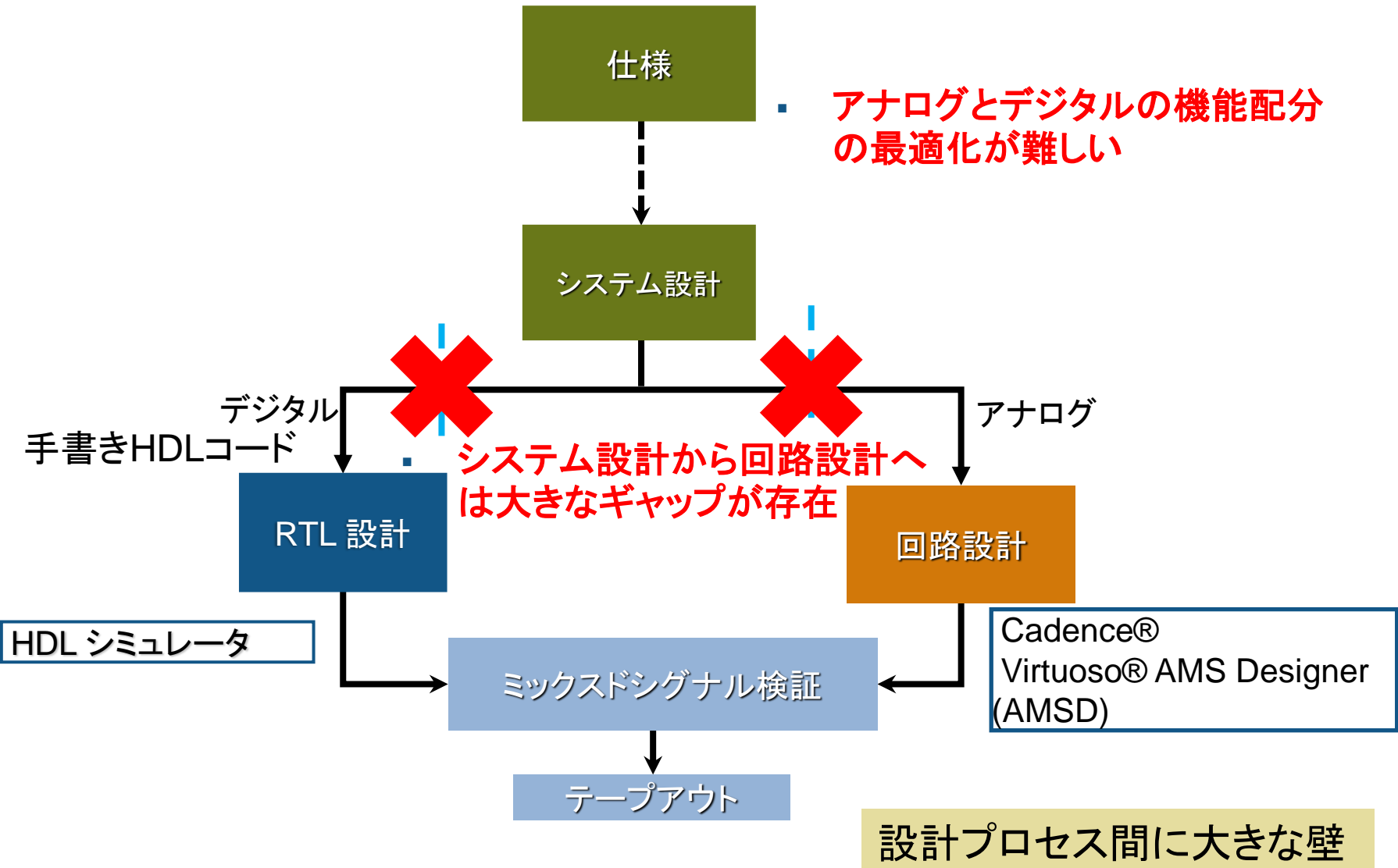
リンク先 <http://www.mathworks.co.jp/company/newsroom/article76584.html>

ニュースリリースからの抜粋:

『TAT (Turn Around Time) の半減 (およそ50%) を可能にし、
従来手法でのボトルネックとなっていた問題を
大幅に改善することが確認されました。』

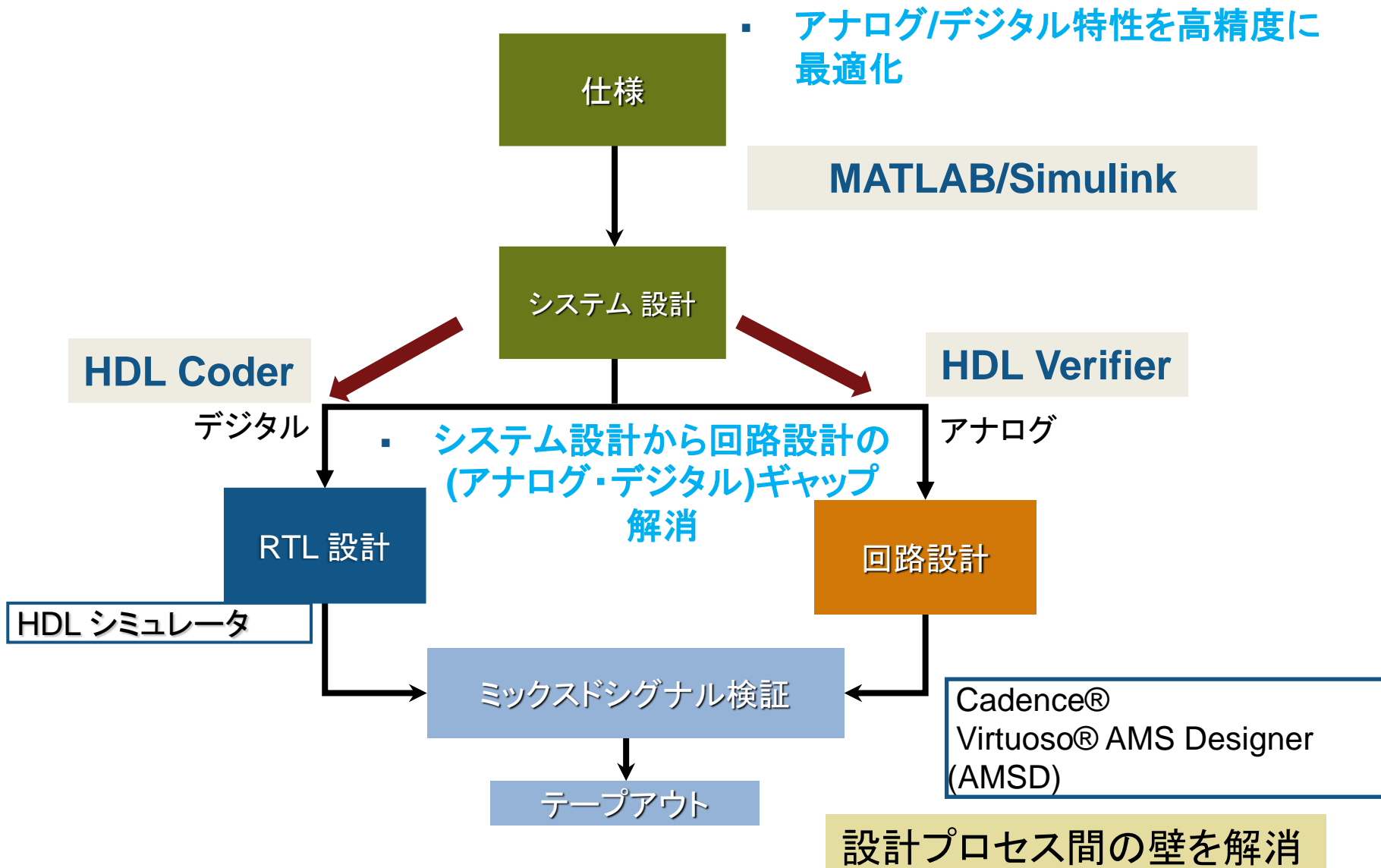
従来のアナログ・ミックスドシグナル設計フロー

各設計プロセスが分断



アナログ・ミックスドシグナル システム設計フロー

MATLAB/Simulinkフロー



MATLAB/Simulinkのシステム設計上の利点

概要

- シミュレーションプラットフォーム
 - 豊富な解析/可視化関数
 - 連続、離散ドメインに対応
 - 可変ソルバによる高速処理
- 各種オプションライブラリ
 - デジタル系ライブラリ
 - アナログ系ライブラリ
 - 豊富なオプションでモデルを迅速に具現化
- EDAツールとのリンク
 - HDLおよび回路シミュレータとの連携
 - 固定小数点化および自動コード生成
 - 包括的なトップダウンフローを実現

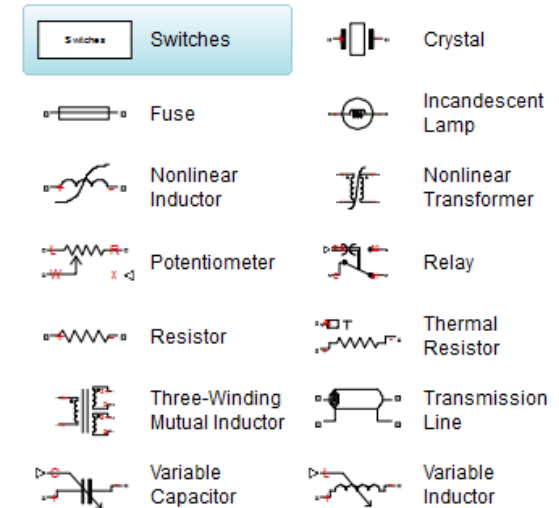
**MATLAB/
Simulink/
Stateflow**

**Control System Toolbox/
Signal Processing Toolbox/
DSP System Toolbox/
SimPowerSystems/
SimElectronics/ ...**

**Fixed Point Designer/
HDL Coder/
MATLAB Coder/
HDL Verifier/ ...**

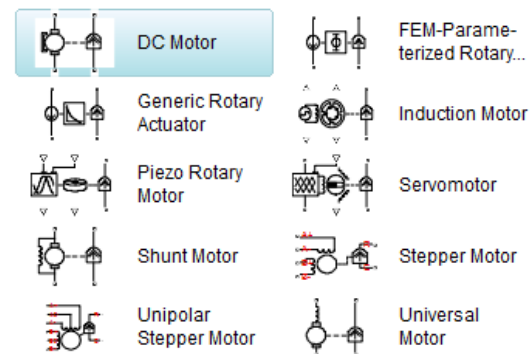
MATLAB/Simulinkのシステム設計上の利点 モデリング

- 回路ビヘイビアモデルでのモデリング
 - 理想スイッチ+RLCでのモデリング
 - トランジスタレベルでのモデリングにも対応
- 伝達関数でのモデリング
 - S領域での伝達関数表記
 - Z領域での伝達関数表記 (sからの変換)
 - 周波数特性データ (測定データ、回路シミュレータ結果) からの伝達関数の導出



- 物理モデリング
 - 制御対象プラントのモデリング (モータ等)

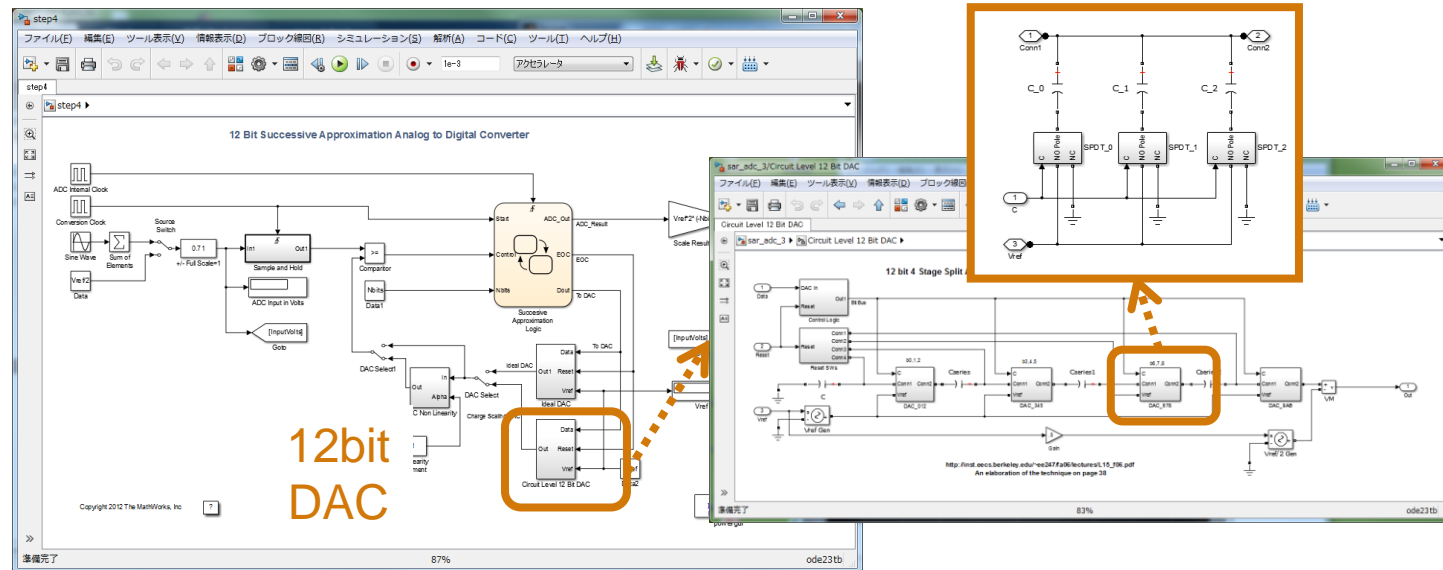
ライブラリ: Simscape/SimElectronics/Actuators & Drivers/Rotational Actuators



アナログ・ミックスドシグナル設計フロー システム設計



- Simulinkの豊富なモデルを用いて、伝達関数からRLC等回路素子まで各種抽象度でモデリングを可能にします。

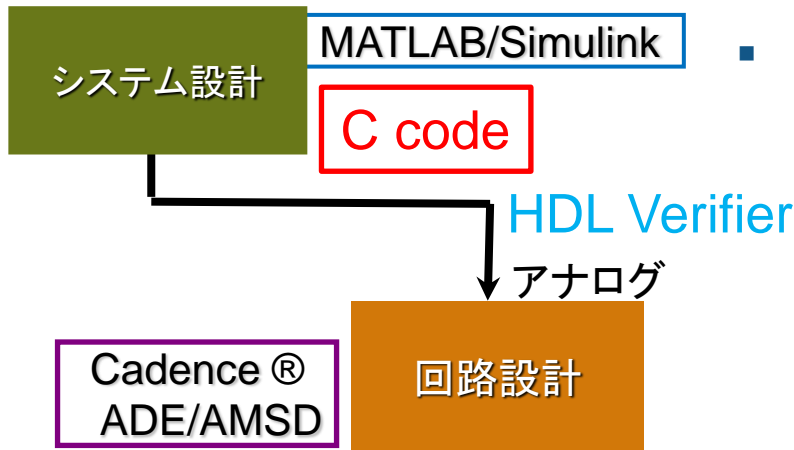


逐次比較型ADC (RLC素子モデル)

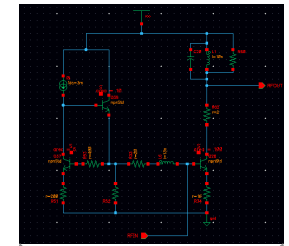
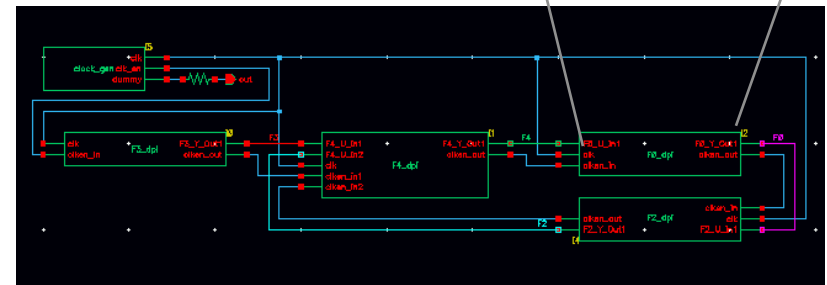
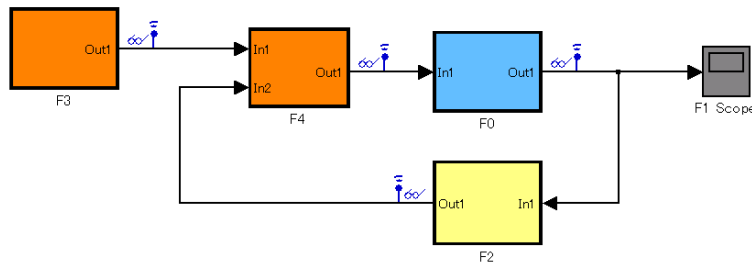
数値計算モデルから回路モデルまで、アナログ・デジタルをモデリング

アナログ・ミックスドシグナル設計フロー

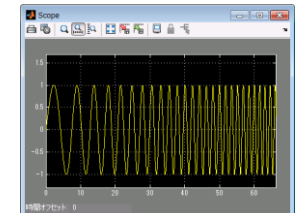
アナログ回路設計への適用:HDL Verifier



- Simulinkシステム設計モデルを回路シミュレータ上でビヘイビアモデルとしてシステム等価検証で活用

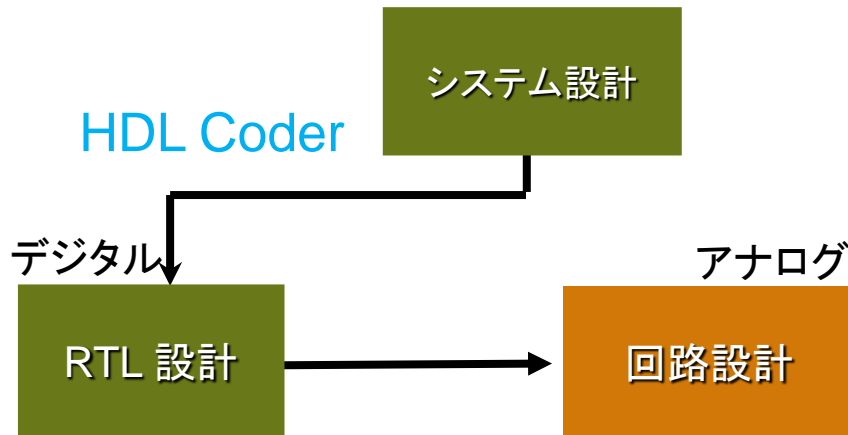


システム設計から回路設計へのシームレスなフロー

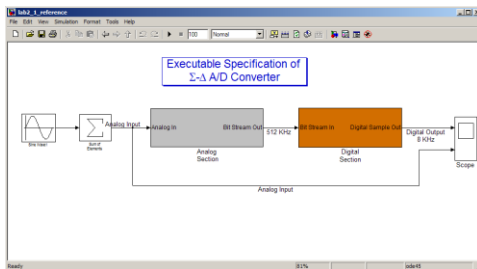


アナログ・ミックスドシグナル設計フロー

デジタル設計への適用: HDL Coder/HDL Verifier

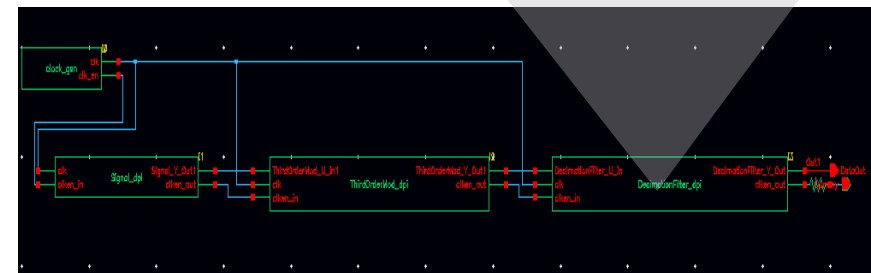


- Simulink上でデジタル部をHDL生成可能なブロックにより詳細設計。
- システム設計から詳細回路設計。



```

    /usr/local/work/hatsul/DPI_C/DLLTest/DTO/DTO/verlog.v
    input clk;
    input reset;
    input clk_enable;
    input signed [15:0] CLKIn; // sf16_Est3
    input signed [31:0] SIn; // int32
    output out;
    output Carry;
    output signed [31:0] TapSIn; // int32
    I
    wire Compare_To_Zero_out1;
    wire signed [31:0] alphaSIn_FF_out1; // int32
    wire TapCarryWide_out1; // int32
    wire signed [31:0] TapCarryWide_out2; // int32
    assign Compare_To_Zero_out1 = (CLKIn > 0 ? 1'b1 :
    1'b0);
    alphaSIn_FF u_SIn_FF (.clk(clk),
    .reset(reset),
  
```



システム設計 = RTL設計

AMSシミュレータ上でアナログ・デジタル協調検証

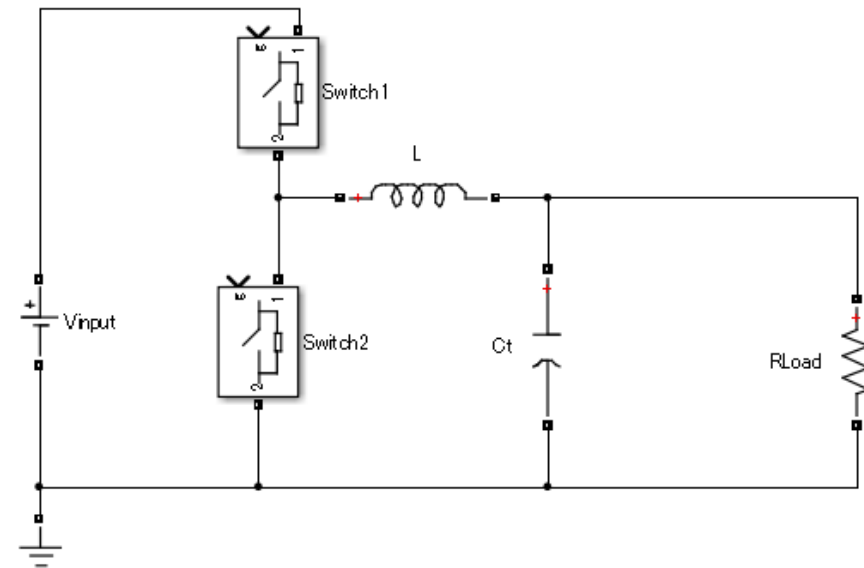
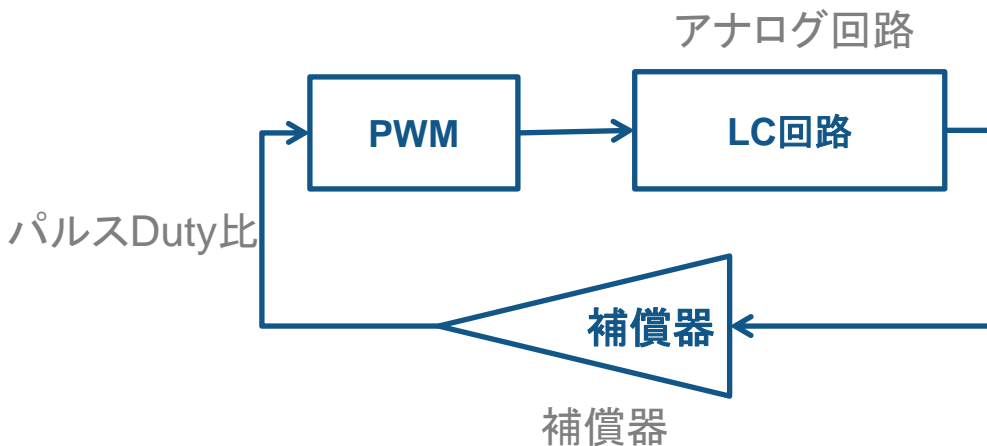
内容

- MATLAB®/Simulink® アナログ・ミックスドシグナルシステム設計フロー
 - 適用事例(アナログ・ミックスドシグナル)
 - システム設計フロー
 - MATLAB®/Simulink®の利点
 - MATLAB®/Simulink®システム設計フロー
- システム設計ワークフロー(電源設計を中心とした適用例)
 - システム設計(モデリング)
 - システム設計(制御設計)
 - 回路設計・検証(アナログ)
 - 回路設計・検証(デジタル)

MATLAB/Simulinkシステム設計デモ

降圧コンバータ

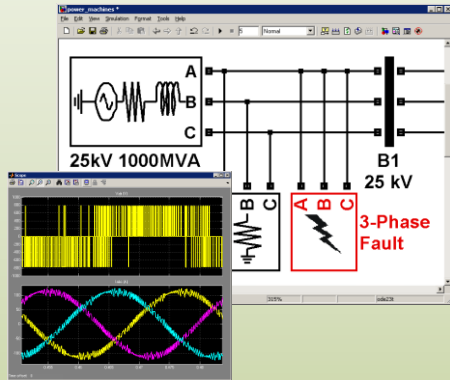
- DCDCコンバータ回路 12V→1.5V
 - DC 出力電圧(降圧コンバータ)モデリング
 - 制御設計
 - 回路設計への適用



物理モデリングツール

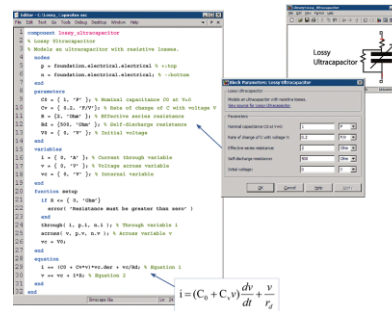
SimPowerSystems™

■ パワエレ・電力系統



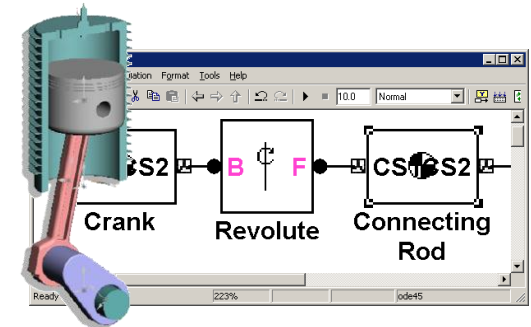
Simscape™

■ 物理モデリング基本環境



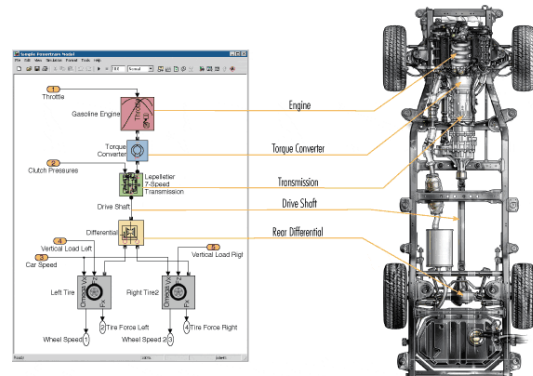
SimMechanics™

■ メカ(3D)



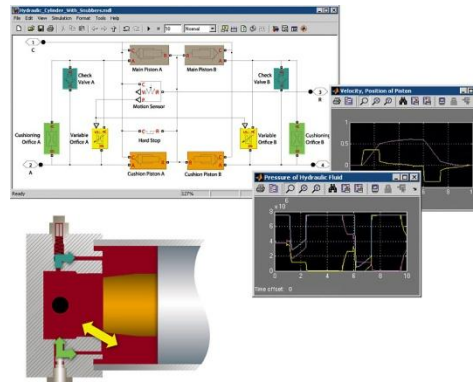
SimDriveline™

■ 車両駆動(1D)



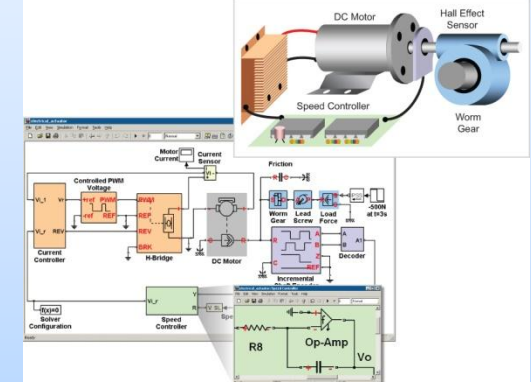
SimHydraulics®

■ 油圧(1D)

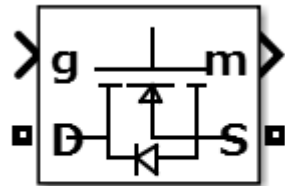


SimElectronics®

■ 電子回路(デジアナ)



半導体素子の詳細度の違い



Mosfet

- ✓ 理想スイッチ+RL要素
- ✓ ゲート信号
g=1→ON, g=0→OFF
- ✓ スイッチング損失を考慮しない
- ✓ 熱を考慮しない

SimPowerSystems™

SimElectronics®

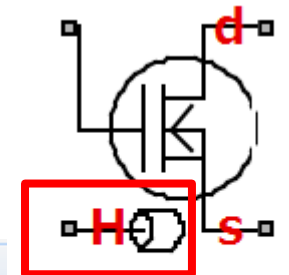
Simscape™

MATLAB®, Simulink®

Simscape block choices ▾

No thermal port

Show thermal port



N-Channel MOSFET

- ✓ 非線形なI-V特性
- ✓ ゲート駆動回路
- ✓ スイッチング損失を考慮
- ✓ 熱を考慮

半導体素子



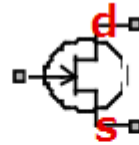
Diode

ダイオード



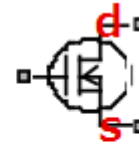
NPN Bipolar Transistor

NPN型 BJT



N-Channel JFET

Nチャンネル型 JFET



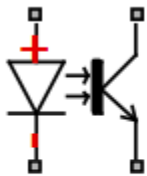
N-Channel MOSFET

Nチャンネル型 MOSFET



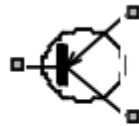
N-Channel IGBT

Nチャンネル型 IGBT



Optocoupler

フォトカプラ



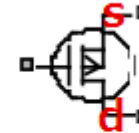
PNP Bipolar Transistor

PNP型 BJT



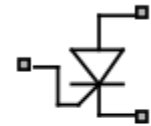
P-Channel JFET

Pチャンネル型 JFET



P-Channel MOSFET

Pチャンネル型 MOSFET



Thyristor

サイリスタ

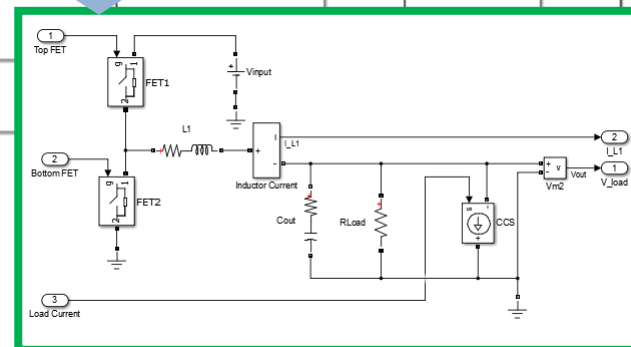
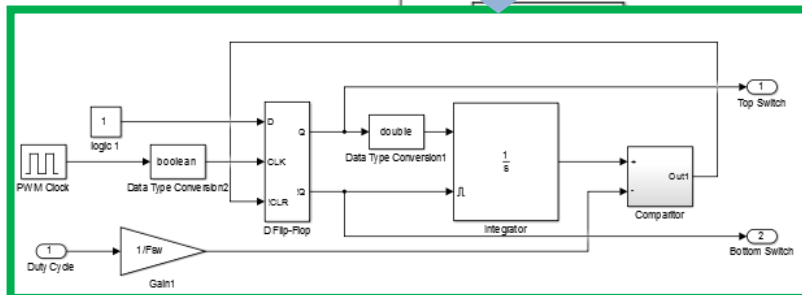
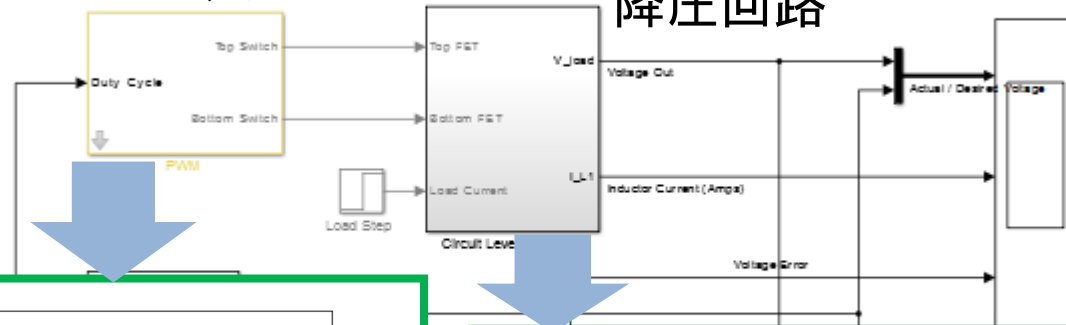
デモ1: 降圧コンバータ モデリング: デモ概要

- 伝達関数の様なモデルからRLC素子まで様々な抽象度でシステム設計のモデリングが可能
- 豊富なモデルを利用したデジタル・アナログ部でのシステムモデリングが可能

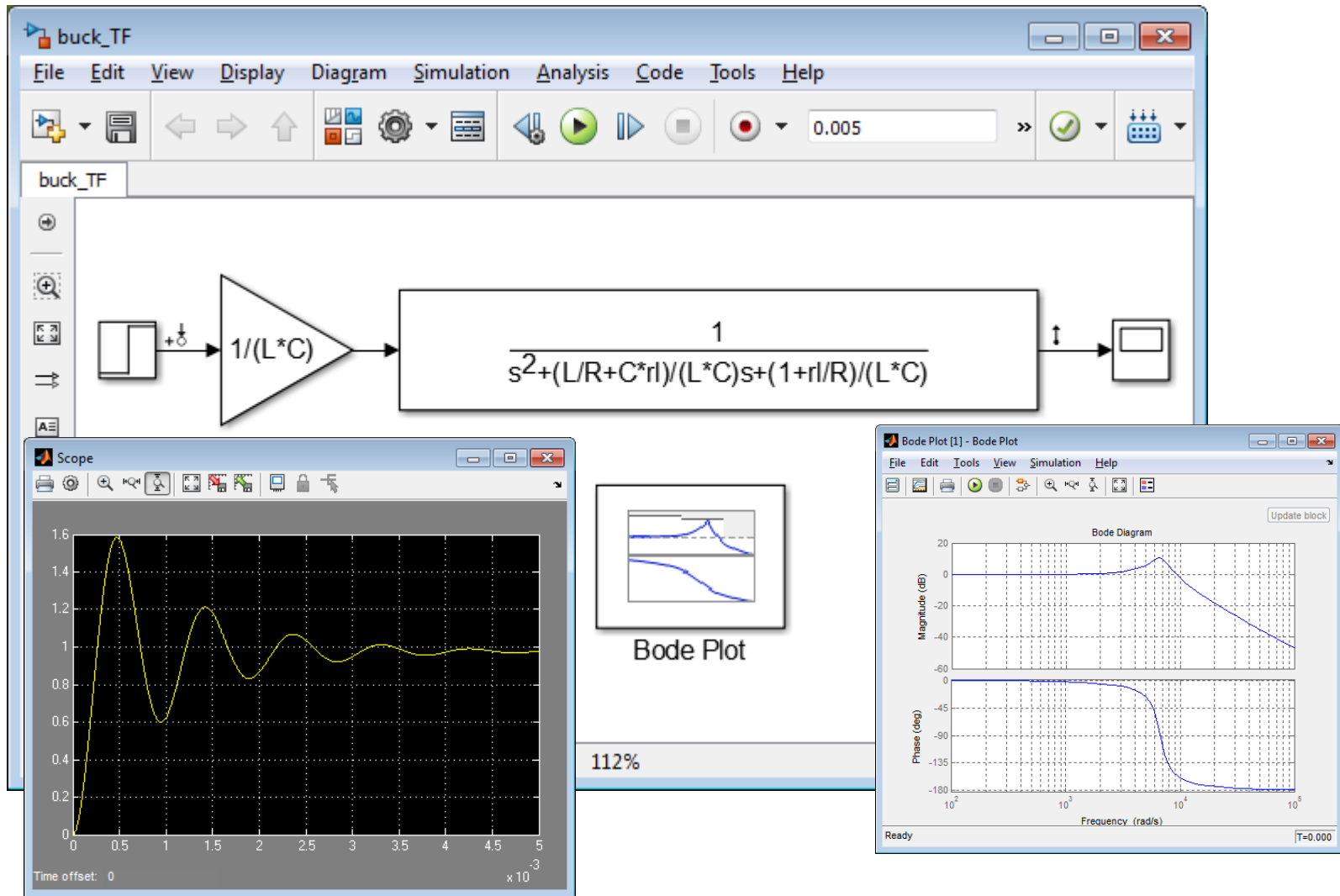
Switching Power Supply
with PID Controller and Plant Transfer Function Estimation

PWMモデル

降圧回路



デモ1: 降圧コンバータ モデリング(伝達関数)



抽象度の高い伝達関数レベルでのモデリングで高速シミュレーション
時間、周波数応答を容易に観測

デモ1: 降圧コンバータ モデリング: 詳細回路 MOSFET RLC 回路(SimElectronics)

The screenshot displays the SimElectronics interface for a buck converter model named 'buck_MOS'. The circuit includes a PWM generator (PWM 1), a MOSFET (JANTX2N6847), a diode (1N4600), and an inductor. A current monitor is placed on the MOSFET's drain. The MOSFET block parameters are shown in a dialog box:

ブロックパラメーター: JANTX2N6847 MOSFET

P-Channel MOSFET
 This block represents a P-channel MOSFET (or IGFET). The drain-source current I_{ds} for negative V_{ds} is given by:

$I_{ds} = 0$ if $-V_{gs} < -V_{th}$ (off)
 $I_{ds} = -K * [(V_{gs} - V_{th}) * V_{ds} - V_{ds}^2 / 2] * (1 + L * |V_{ds}|)$ if $0 < -V_{ds} < -V_{gs} + V_{th}$ (linear region)
 $I_{ds} = -(K/2) * (V_{gs} - V_{th})^2 * (1 + L * |V_{ds}|)$ if $0 < -V_{gs} + V_{th} < -V_{ds}$ (saturated region)

where K is a constant, V_{th} is the Threshold voltage, L is the channel modulation, V_{gs} is the gate-source voltage and V_{ds} is the drain-source voltage.

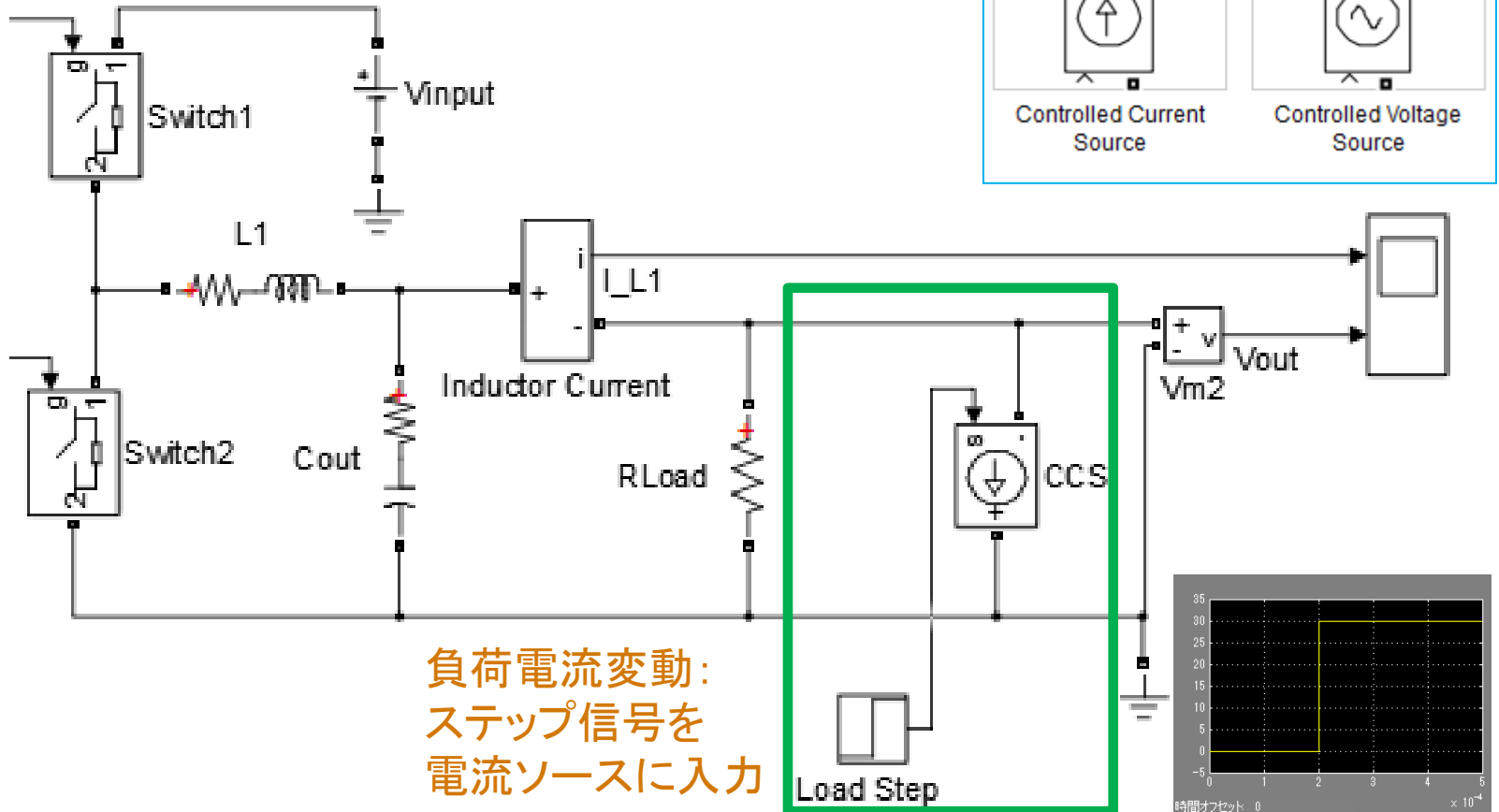
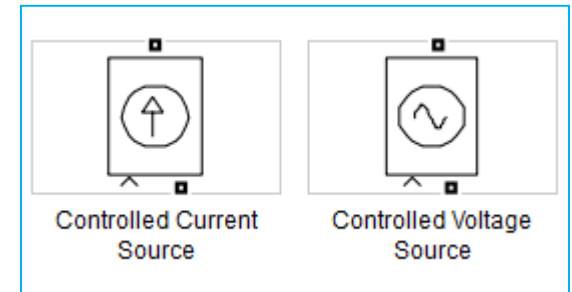
Parameters

Parameterization:	Specify from a datasheet	
Drain-source on resistance, $R_{DS(on)}$:	1.5	Ohm
Drain current, I_{ds} , for $R_{DS(on)}$:	-2.5	A
Gate-source voltage, V_{gs} , for $R_{DS(on)}$:	-10	V
Gate-source threshold voltage, V_{th} :	-2	V
Channel modulation, L :	0	1/V

トランジスターレベルでの詳細回路のモデリングにも対応

デモ1: 降圧コンバータ モデリング: 各種変動のモデリング(負荷電流等)

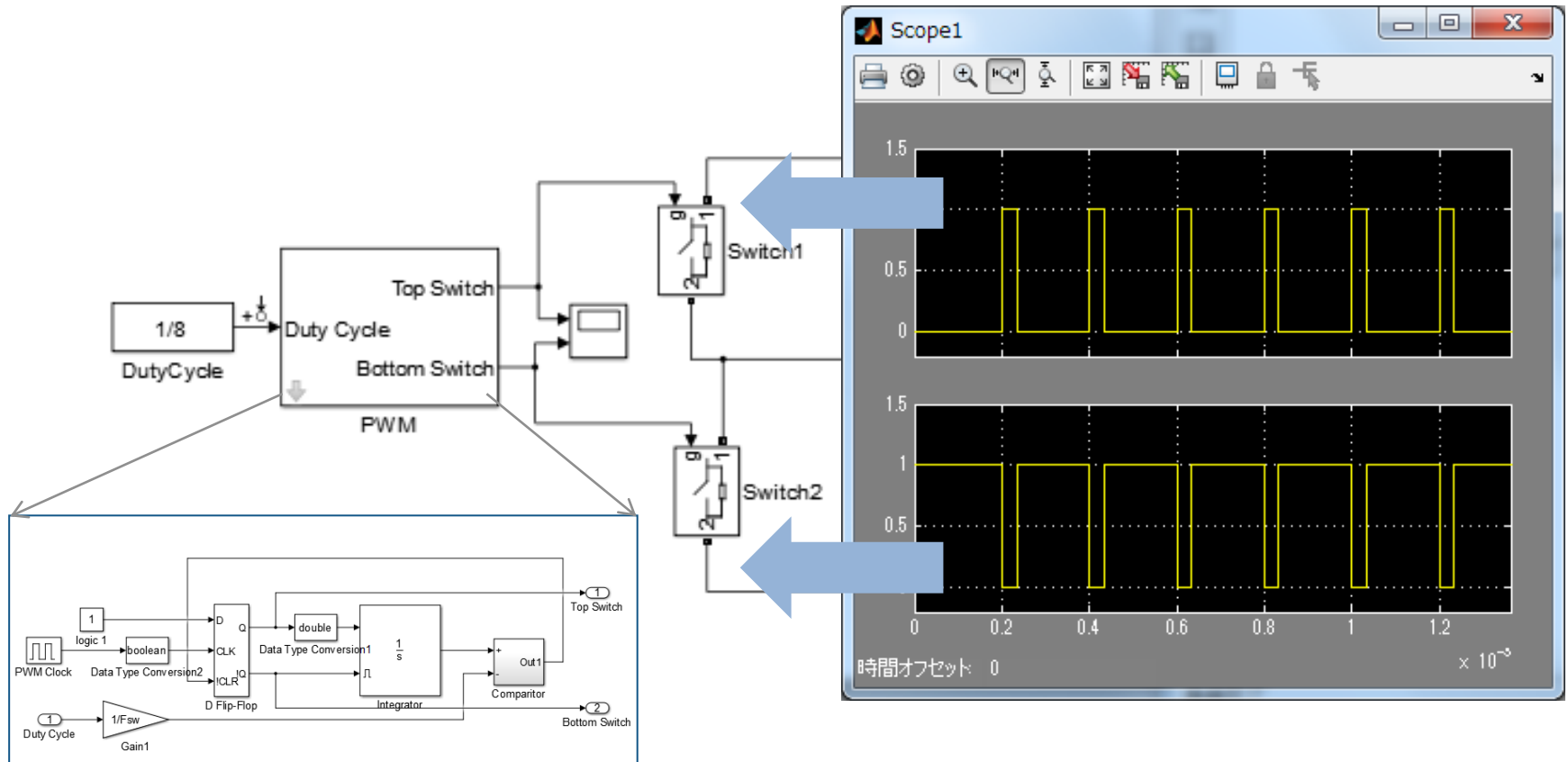
電圧源・電流源ブロック



負荷電流変動:
ステップ信号を
電流ソースに入力

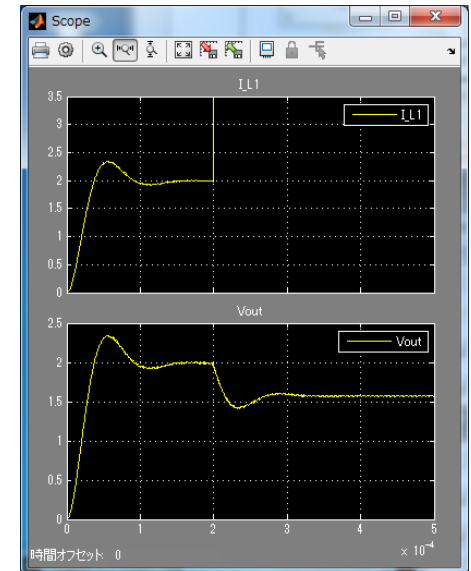
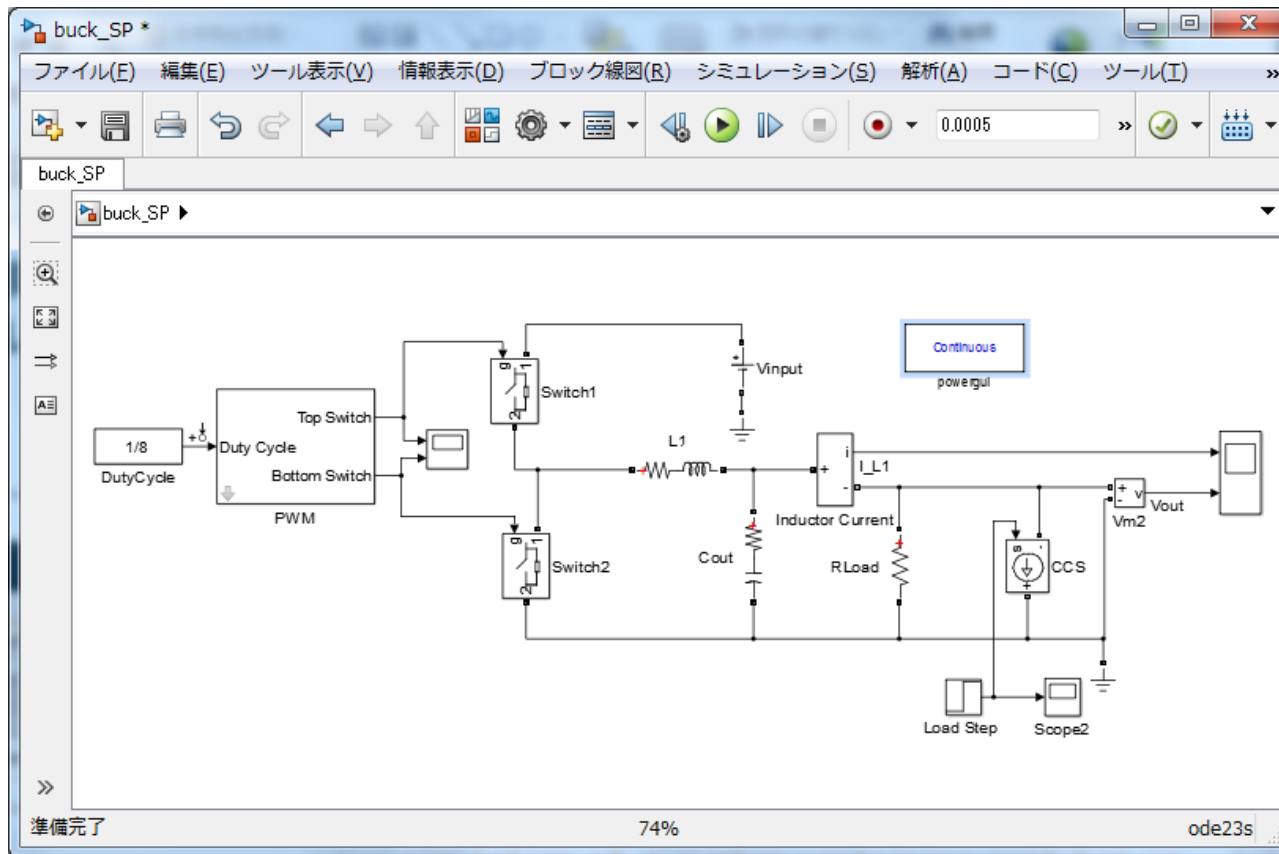
回路設計者が容易な回路素子レベルでのモデリングをサポート
各種変動もモデリング可能

デモ1: 降圧コンバータ モデリング: PWMのモデリング



デジタル制御部もSimulinkの豊富なブロックでモデル化

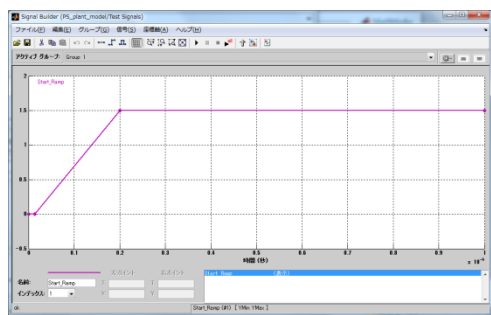
デモ1: 降圧型コンバータ モデリング: シミュレーション全体



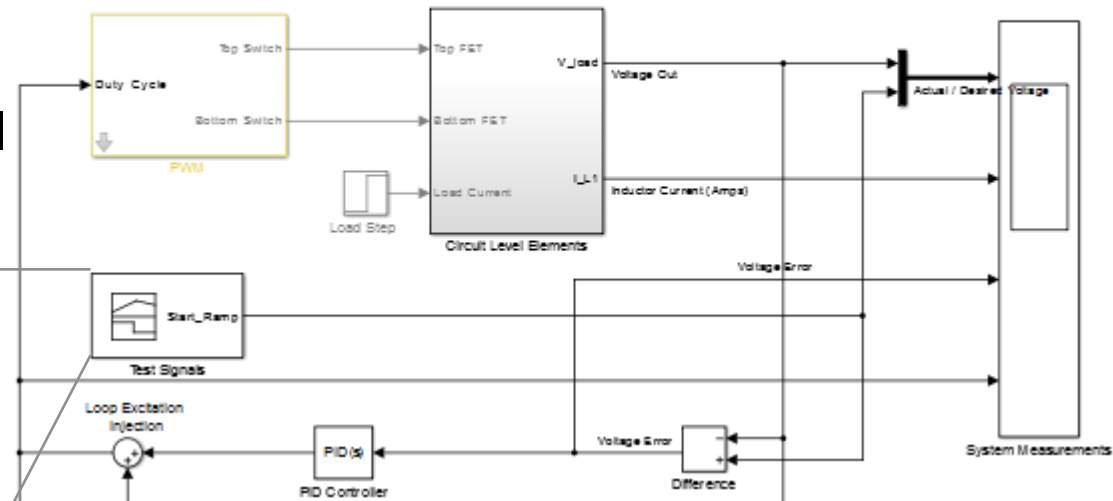
モデリングしたデジタル・アナログ部を可変ソルバで
高速シミュレーション

デモ1:降圧型コンバータ モデリング: サブキット化

基準電圧をTestSignal
ブロックで定義



Switching Power Supply
with PID Controller and Plant Transfer Function Estimation

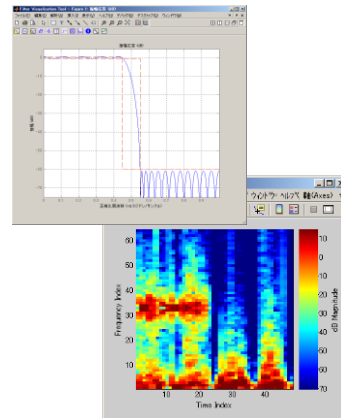


機能ブロック単位に容易にサブシステム化
サブシステムをライブラリ化して資源を再利用および共有

信号処理設計・解析オプション

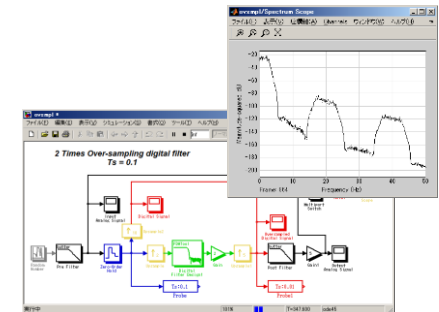
Signal Processing Toolbox™

- 信号生成
- フィルタ設計解析
 - アナログフィルタ
 - デジタルフィルタ
- スペクトル解析
- 線形予測



DSP System Toolbox™

- 高度なフィルタ設計
 - マルチレート、適応フィルタ、固定小数点化
- スペクトル推定
- デジタルフィルタ
- 行列、線形代数
- FFT/DCT/DWT

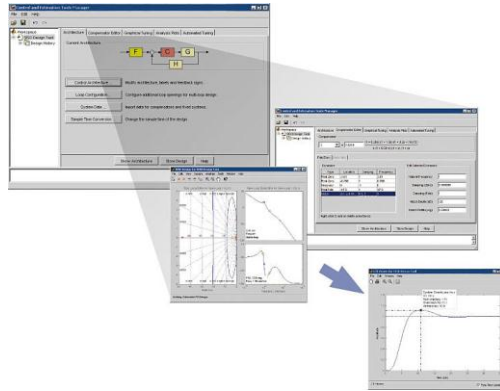


時間、周波数での信号処理及び解析

制御器設計・最適化ツール

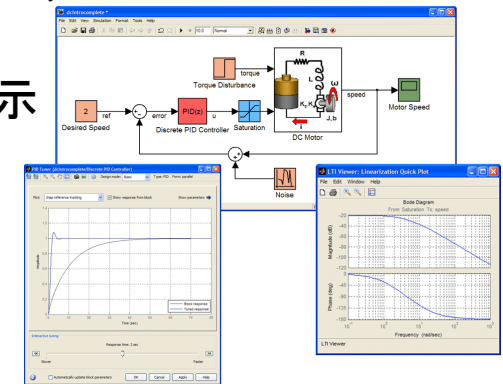
Control System Toolbox™

- 線形システム
定義・解析
- 古典制御設計
- 現代制御設計



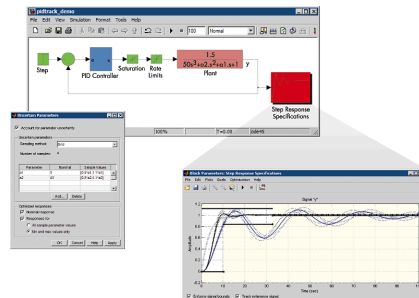
Simulink Control Design™

- Simulinkモデルの
線形解析・
周波数応答表示
- PIDゲインの
自動調節



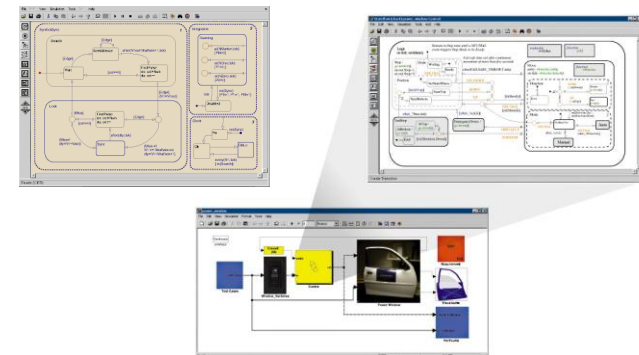
Simulink Design Optimization™

- 実験データを用いた
制御対象パラメータ推定
- 期待出力値
による制御
パラメータ調節



Stateflow®

- フローチャート・状態遷移図モデリング

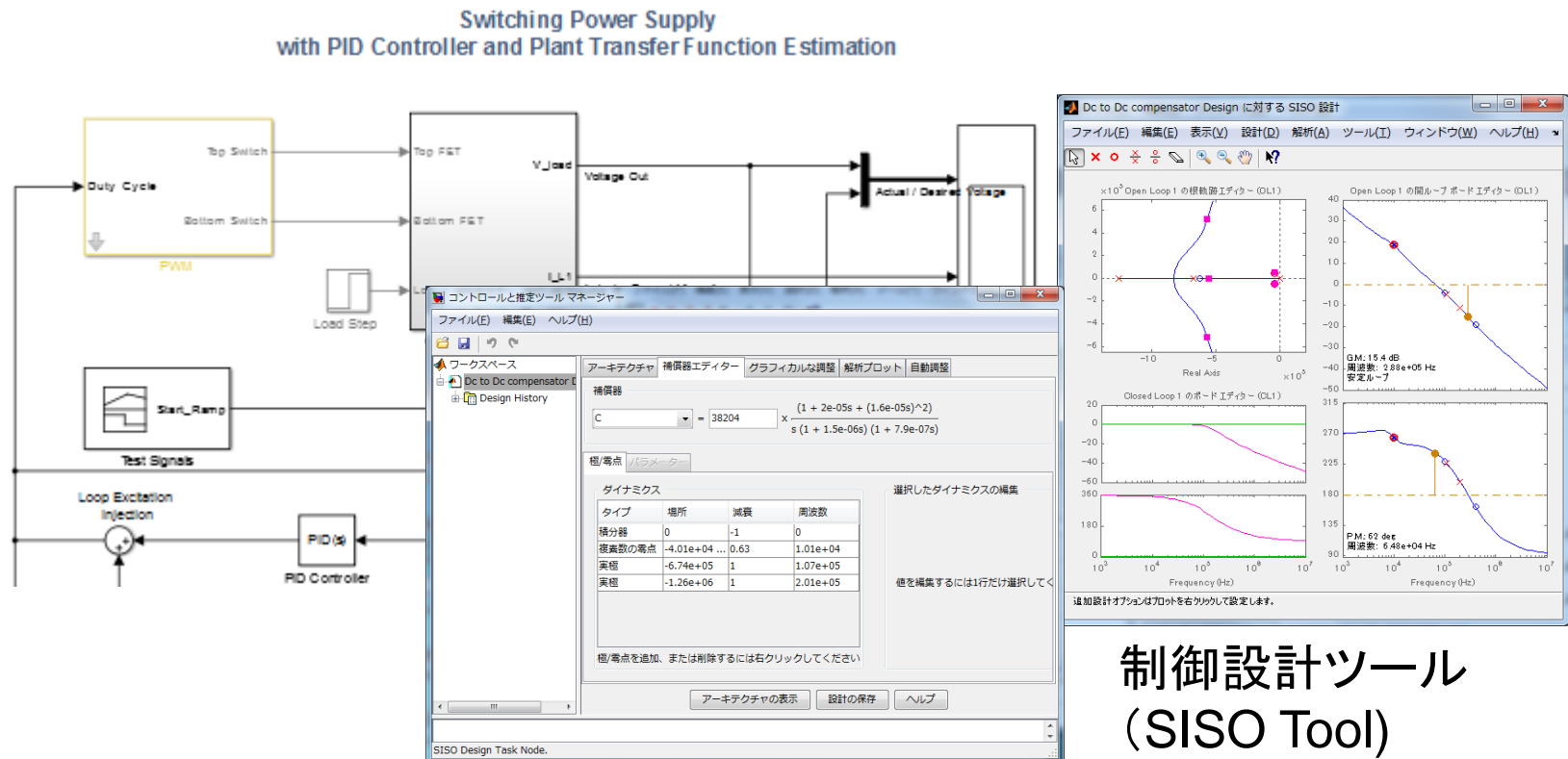


各種制御設計に活用

デモ2: 降圧コンバータ 制御設計

デモ概要

- 伝達関数の推定
- MATLAB/Simulinkの制御系ツールを使う事により様々な補償器の設計を応答波形を操作して可能

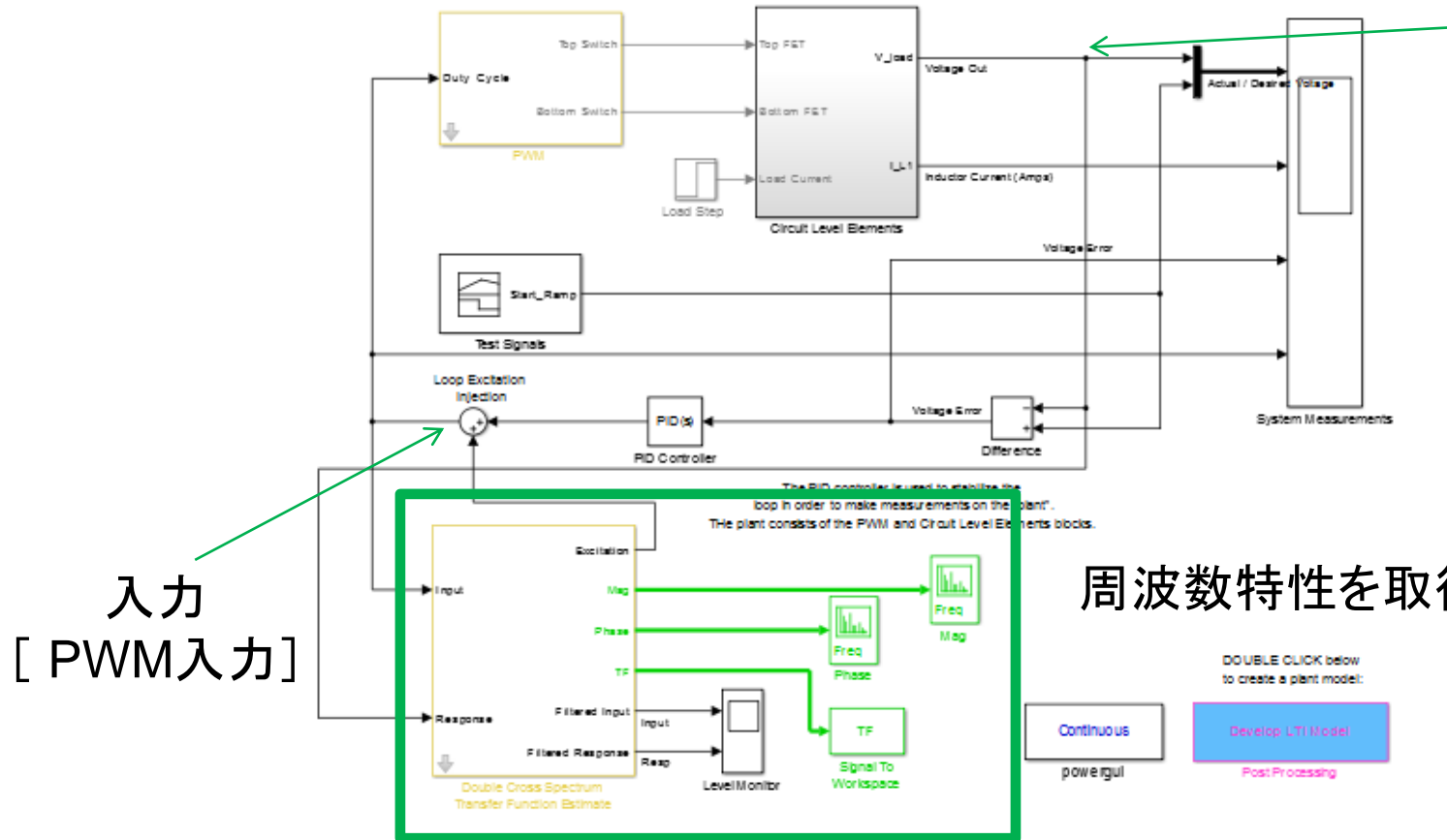


制御設計ツール (SISO Tool)

デモ2: 降圧コンバータ(制御設計) プラントモデル伝達関数の推定1

Switching Power Supply
with PID Controller and Plant Transfer Function Estimation

出力
[電圧出力]

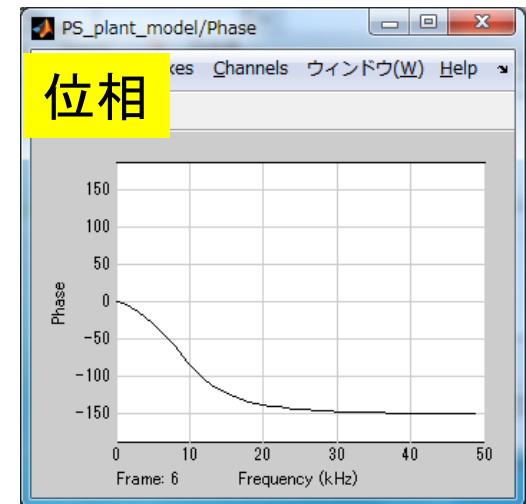
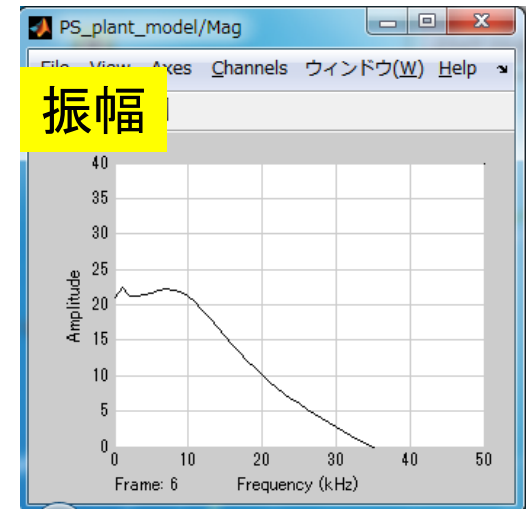
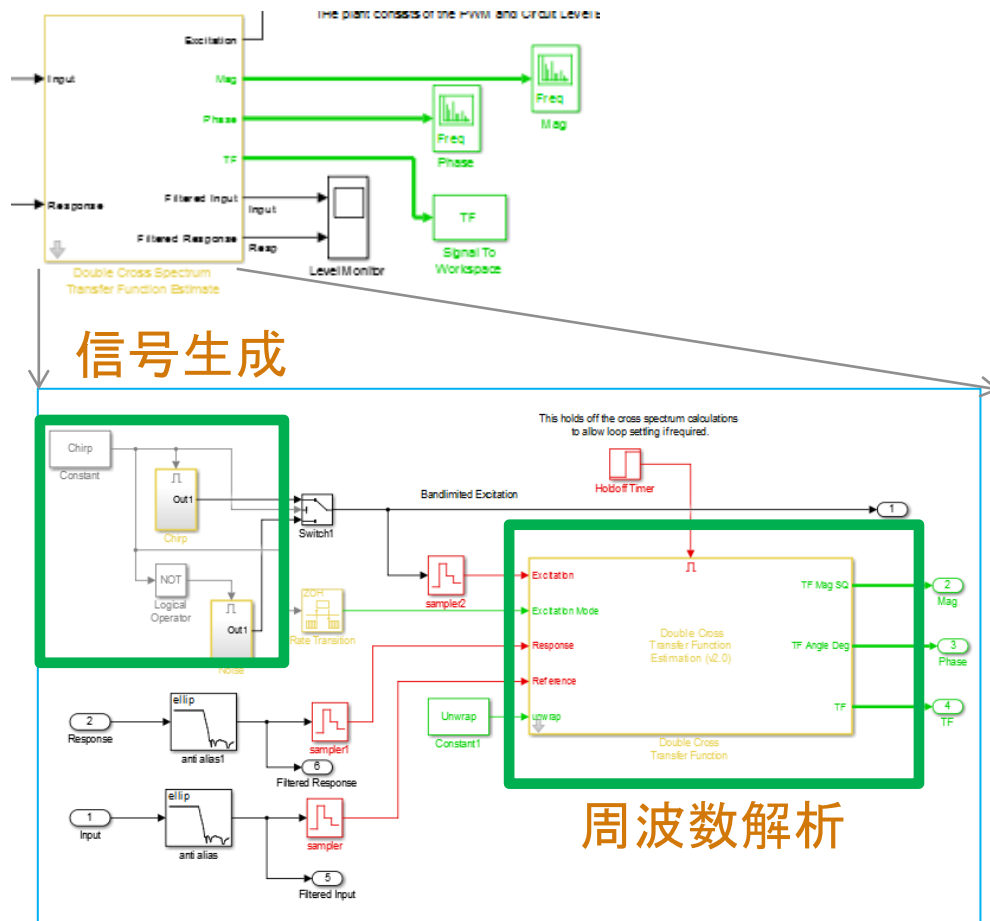


入力
[PWM入力]

周波数特性を取得

解析全帯域の信号を入力して、小さいレベルの信号を付加して、入力
と出力の周波数特性を取得する。アベラージュも可能

デモ2: 降圧コンバータ(制御設計) プラントモデル伝達関数の推定2



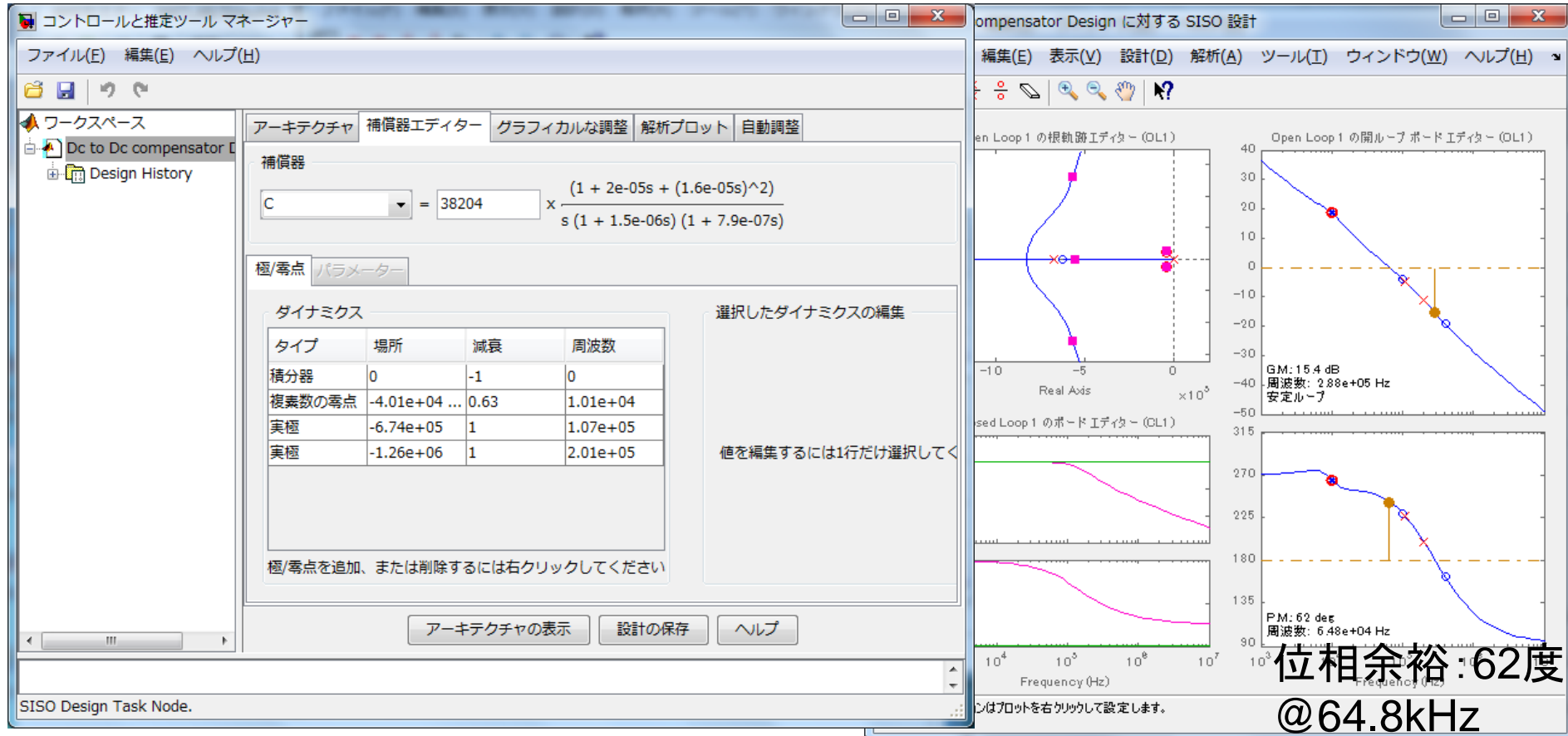
```
[num,den]=invfreqs(xfer,w,2,2,[],iter);
resp = freqs(num,den,w);
```

MATLAB: Signal Processing Toolboxの関数で振幅、位相データを伝達関数に変換

デモ2: 降圧コンバータ(制御設計) 補償器設計(SISOツール)

設計ツール

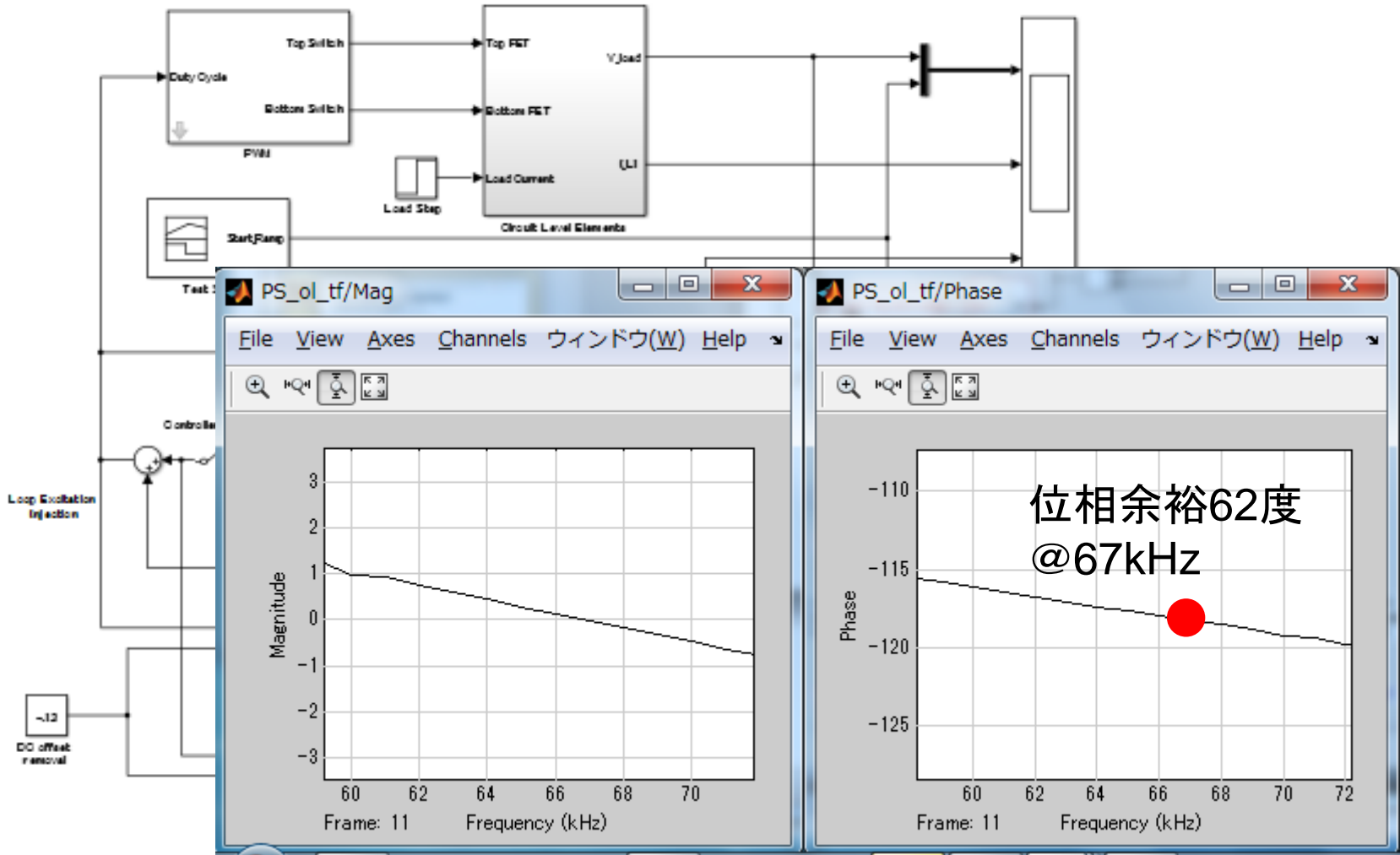
補償器設計後特性



各種の補償器設計も応答波形上から設計することが可能です。

デモ2: 降圧コンバータ(制御設計)

補償器設計: 安定解析

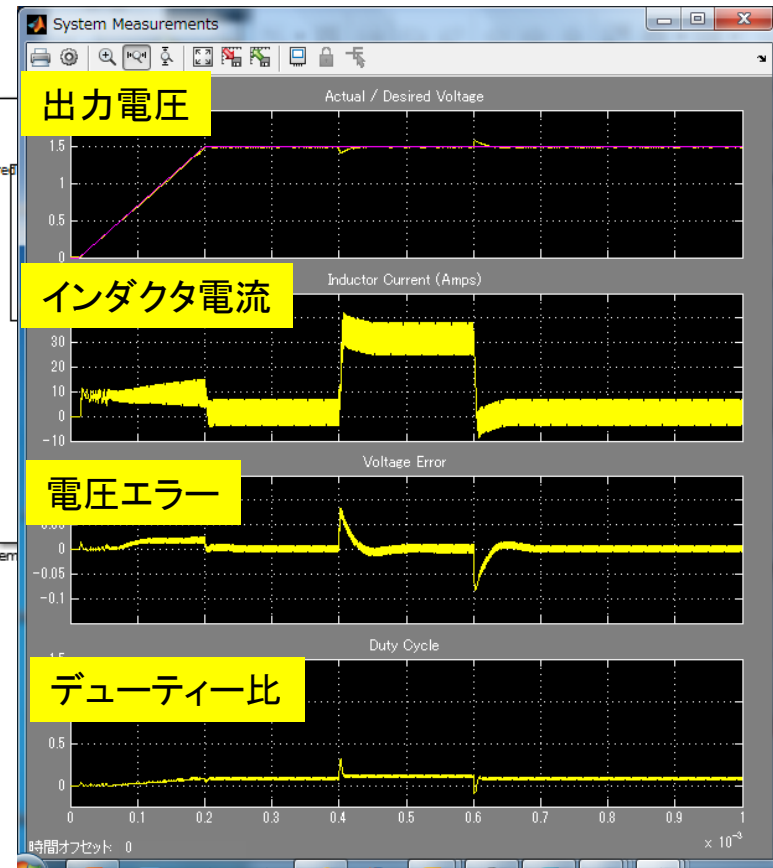
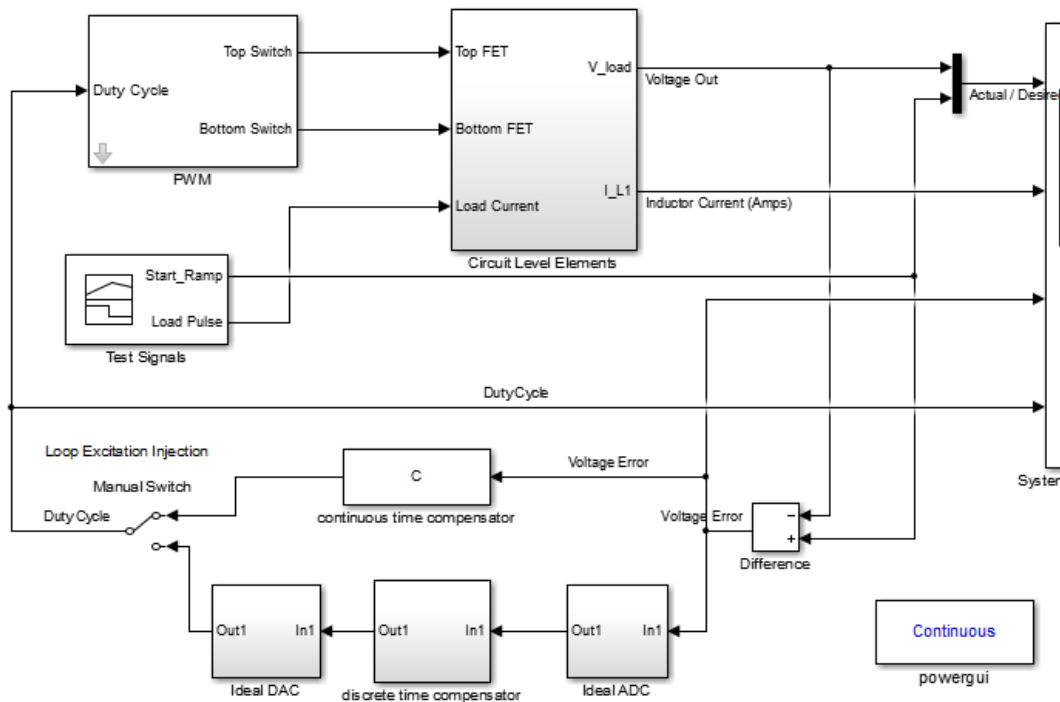


設計時の位相余裕と同等の結果が出ている事を確認

デモ2: 降圧コンバータ(制御設計)

補償器設計: 適用後のシミュレーション

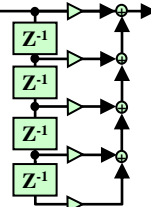
Switching Power Supply
Loop Compensator Designed by SISO Tool



HDLコード生成・検証オプション

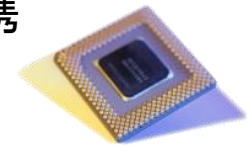
Filter Design HDL Coder

- MATLABで設計したIIR/FIR/マルチレート・フィルタからHDL生成
- ターゲット依存しないVHDL/Verilog
- テストベンチ(HDL, Model)生成



HDL Coder™

- Stateflow, MATLAB Functionを含むSimulinkモデルからHDL生成
- ターゲット依存しないVHDL/Verilog
- テストベンチ(HDL, Model)生成
- FPGAベンダツールと連携



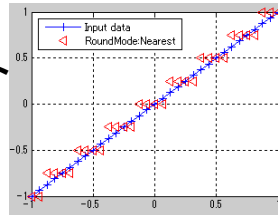
HDL Verifier™

- HDLシミュレータと協調シミュレーション
 - ModelSim / Incisive
- FPGA-in-the-Loop
- SystemC TLM 2.0生成
- System-Verilog DPI-Cリンク機能
(AMSシミュレータ、HDLシミュレータリンク機能)

固定小数点演算、シミュレーションオプション

Fixed-Point Designer™ (MATLAB)

- MATLABで固定小数点演算
- オーバーフロー発生箇所の検出
- 倍精度浮動小数点演算との比較
- 固定小数点フィルタのダイナミックレンジ、リミットサイクル発振の解析



Fixed Point Designer™ (Simulink)

- Simulink既存ブロックで固定小数点設定が可能
- 設定はC/HDLコード生成結果に反映
- 固定小数点アドバイザー
- スケーリングの最適化

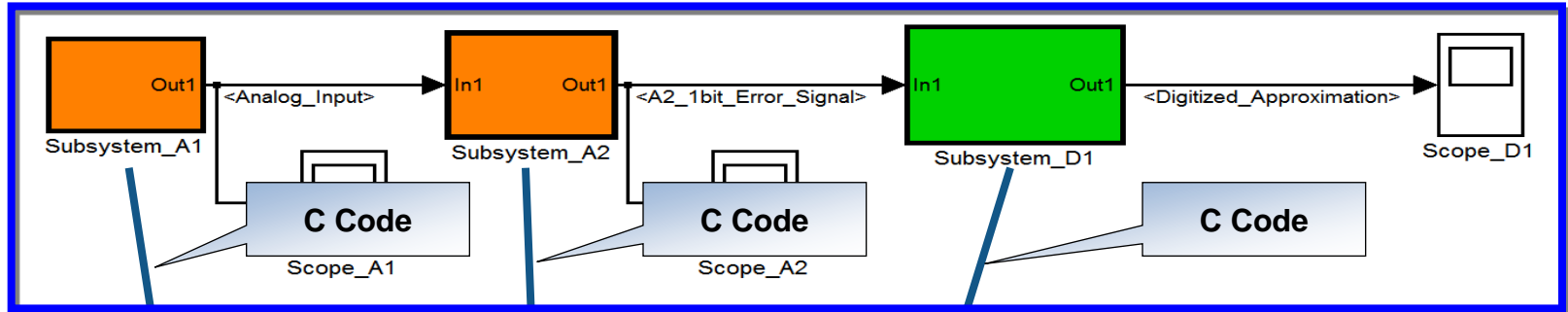
固定小数点化による精度検証が可能

Simulinkによるシステム設計からアナログ回路設計

HDL Verifier : DPI-C Link機能

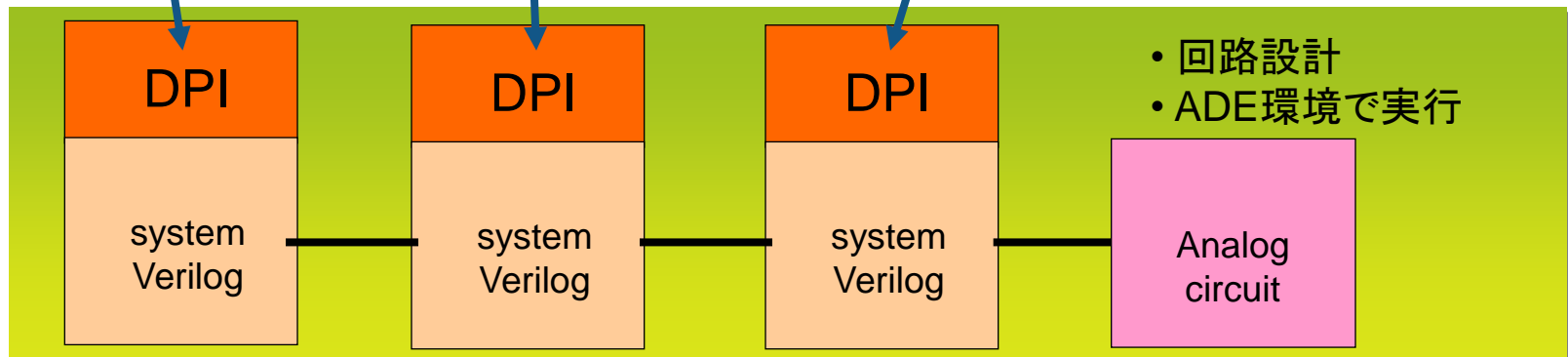
- HDL Verifier DPI-C Link の動きのイメージ図

Simulink



コンパイルした C コード

AMSD in ADE



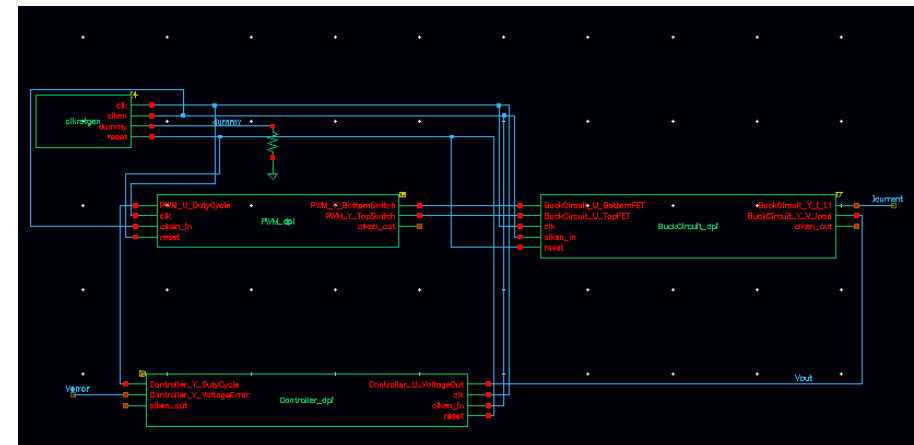
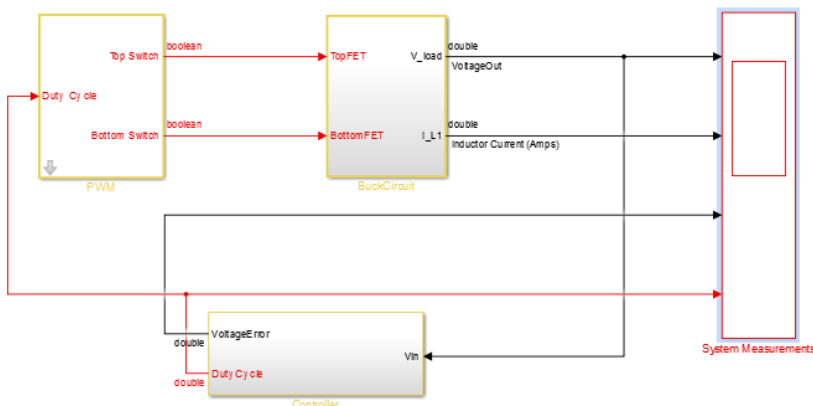
SimulinkのサブシステムをAMSシミュレータのビヘイビアモデルとして活用、アナログ・デジタル回路のシステム等価検証が可能

デモ3: 回路設計・検証での活用

デモ概要

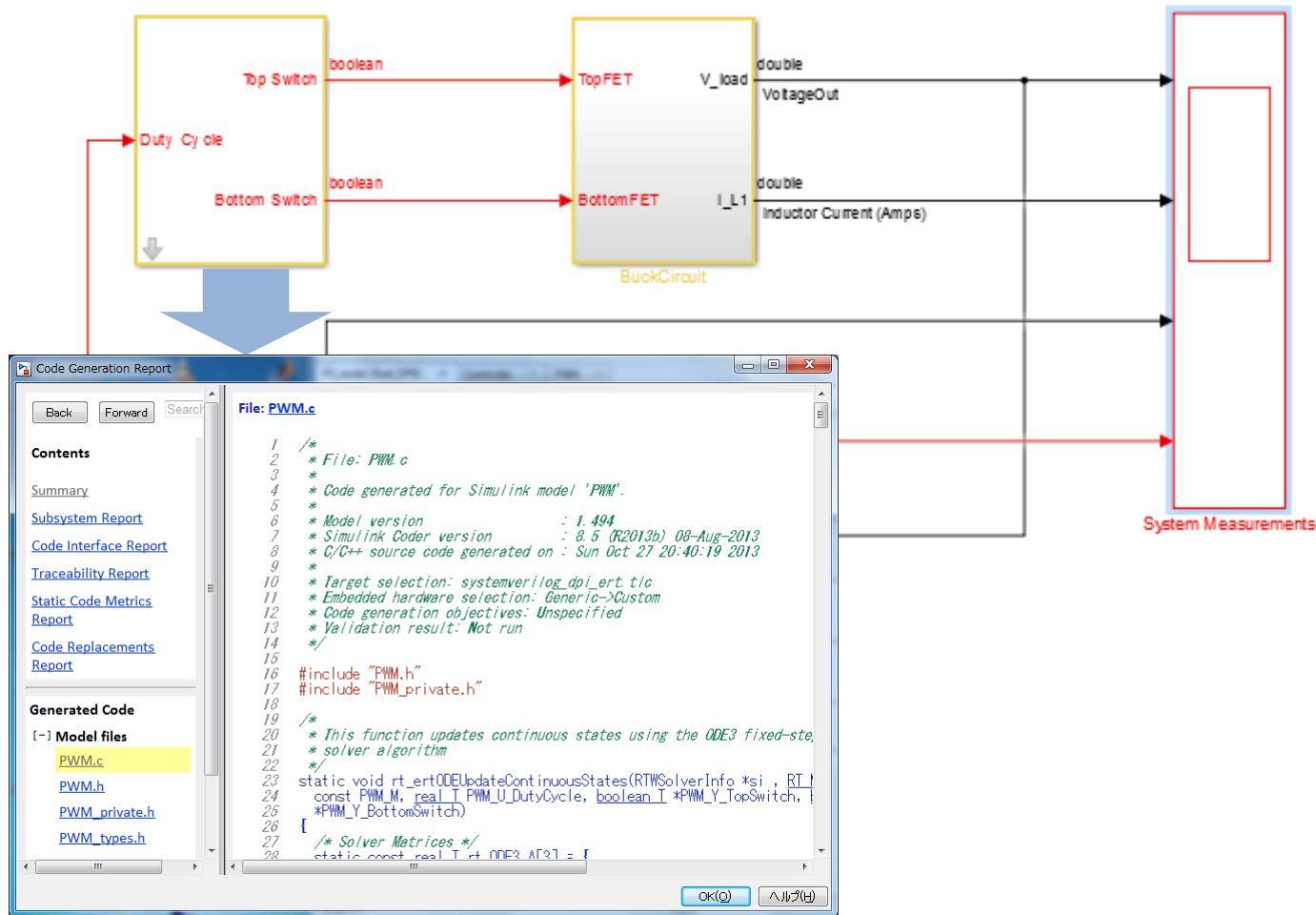
- Simulinkで行ったシステム設計モデルをAMSシミュレータ上でビヘイビアモデルとして活用可能
- システム設計の等価検証をAMSシミュレータ上で行えることを確認

AMSシミュレータ: Cadence ® Virtuoso ® AMS Designerを使用



デモ3: Simulink回路設計・検証

SimulinkからCコード生成



サブシステム毎にCコード、およびラッパーコード(C, SystemVerilog)生成

デモ3: Simulink回路設計・検証 AMSシミュレータ上でスキマティック作成

```

/usr/local/work/hatsul/DPI_C_13b/DCDC/DCDC/PWM_dpi/systemVerilog
/systemVerilog HDL for "DCDC", "PWM_dpi" "systemVerilog"

`timescale 1ps / 1ps

module PWM_dpi
(
  input clk,
  input reset,
  input clken_in,
  output clken_out,
  input real PWM_U_DutyCycle,
  output bit PWM_Y_TopSwitch,
  output bit PWM_Y_BottomSwitch
);

  parameter isLibContinuous = 0;

  import "DPI-C" function void DPI_PWM_initialize(inout bit PWM_Y_BottomSwitch);
  import "DPI-C" function void DPI_PWM_output(input real PWM_U_DutyCycle,inout bit PWM_Y_TopSwitch,inout bit PWM_Y_BottomSwitch);
  import "DPI-C" function void DPI_PWM_update(input real PWM_U_DutyCycle,inout bit PWM_Y_TopSwitch,inout bit PWM_Y_BottomSwitch);

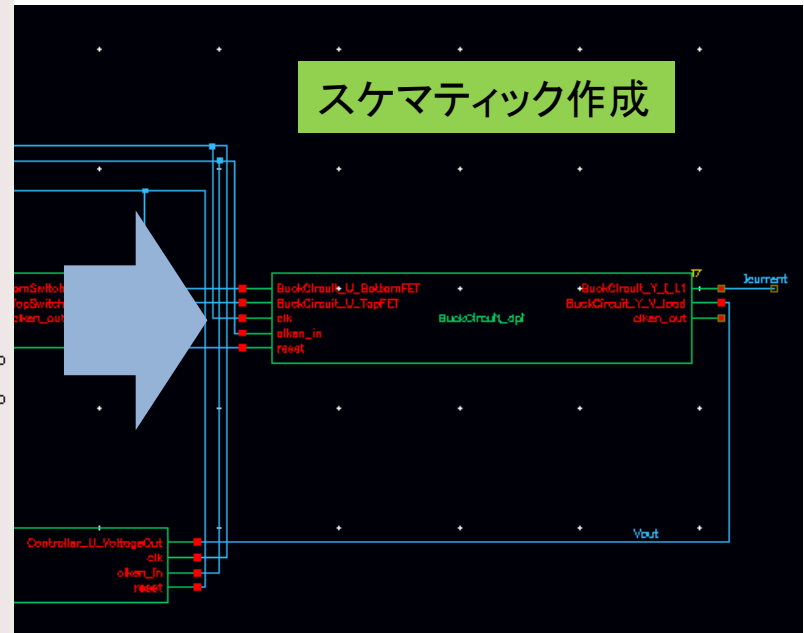
  always @(reset) begin
    DPI_PWM_initialize(PWM_Y_BottomSwitch);
  end

  always @(posedge clk) begin
    if(isLibContinuous == 1) begin
      DPI_PWM_output(PWM_U_DutyCycle,PWM_Y_TopSwitch,PWM_Y_BottomSwitch);
    end
    DPI_PWM_update(PWM_U_DutyCycle,PWM_Y_TopSwitch,PWM_Y_BottomSwitch);
  end

  always @(clken_in) begin
    if(isLibContinuous == 0) begin
      DPI_PWM_output(PWM_U_DutyCycle,PWM_Y_TopSwitch,PWM_Y_BottomSwitch);
    end
  end
end

```

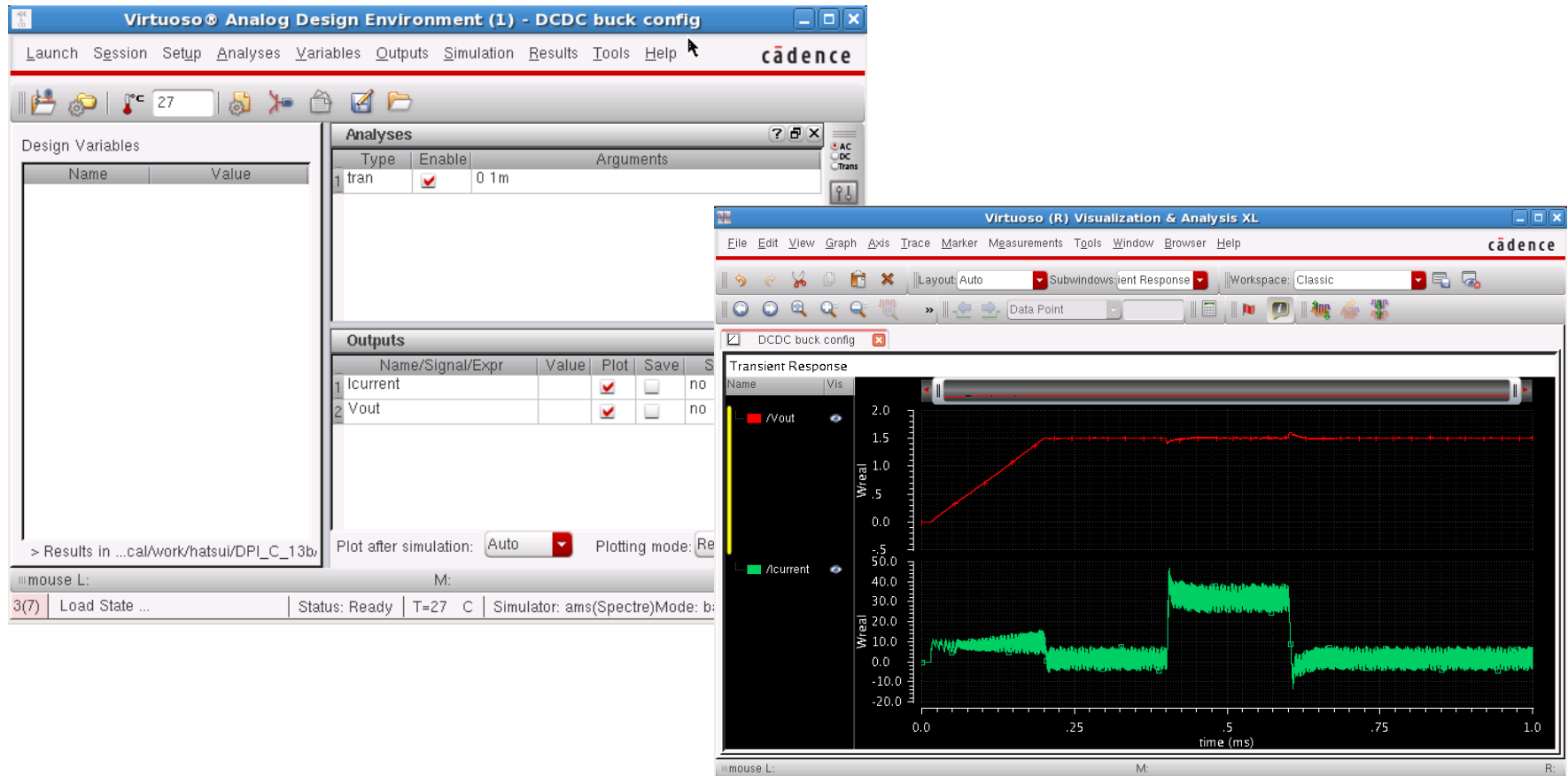
SystemVerilog View作成



スキマティック作成

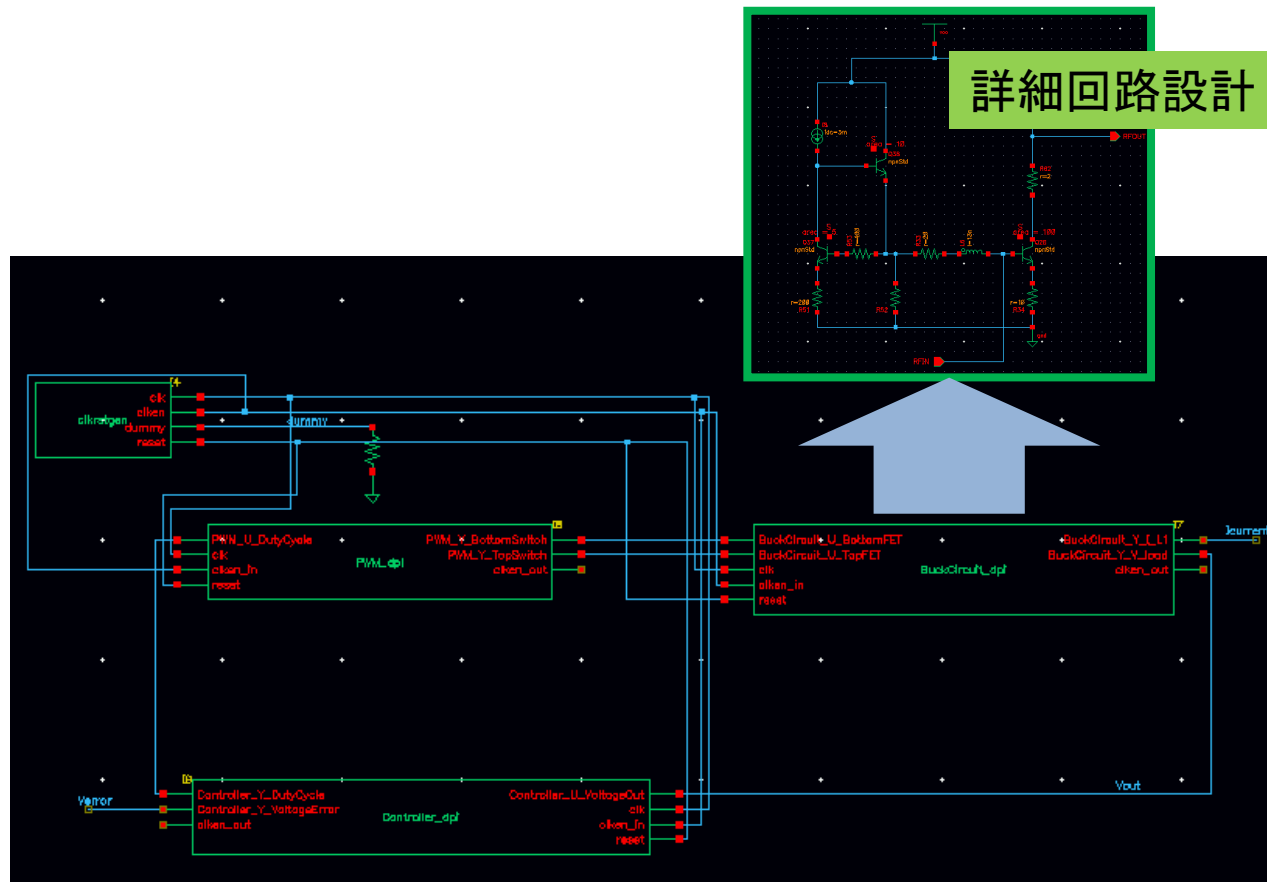
Simulinkから出力されたSystemVerilogラツパーファイルをView作成に利用

Simulinkシステム設計モデル回路設計への適用 AMSシミュレーション



SimulinkのサブシステムをAMSシミュレータのビヘイビアモデルとして活用、アナログ・デジタルの回路検証が可能

デモ3: Simulink回路設計・検証 AMSシミュレータ上で回路図作成



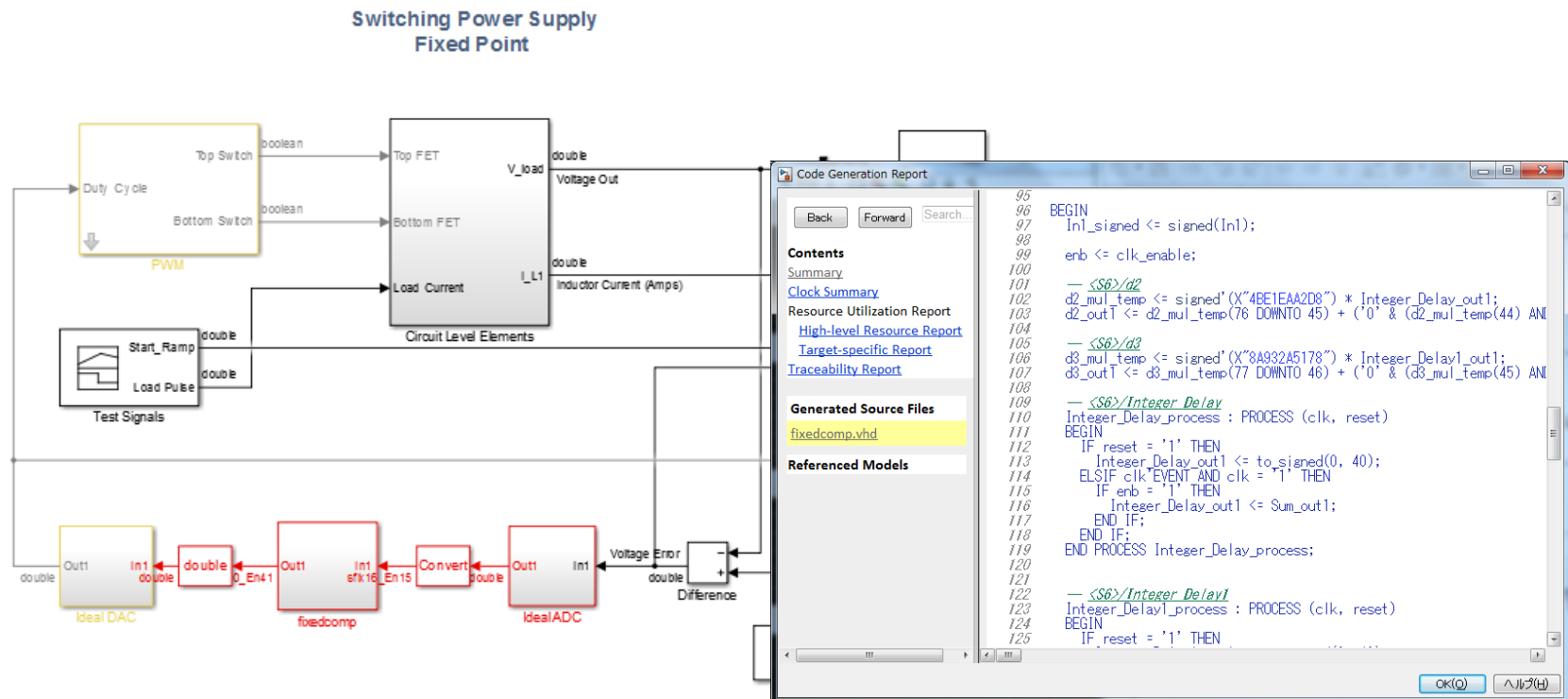
各部を詳細回路設計して、Simulinkビヘイビアモデルを詳細回路に置き換えて検証シミュレーション

注意: 今回のデモには含まれません。

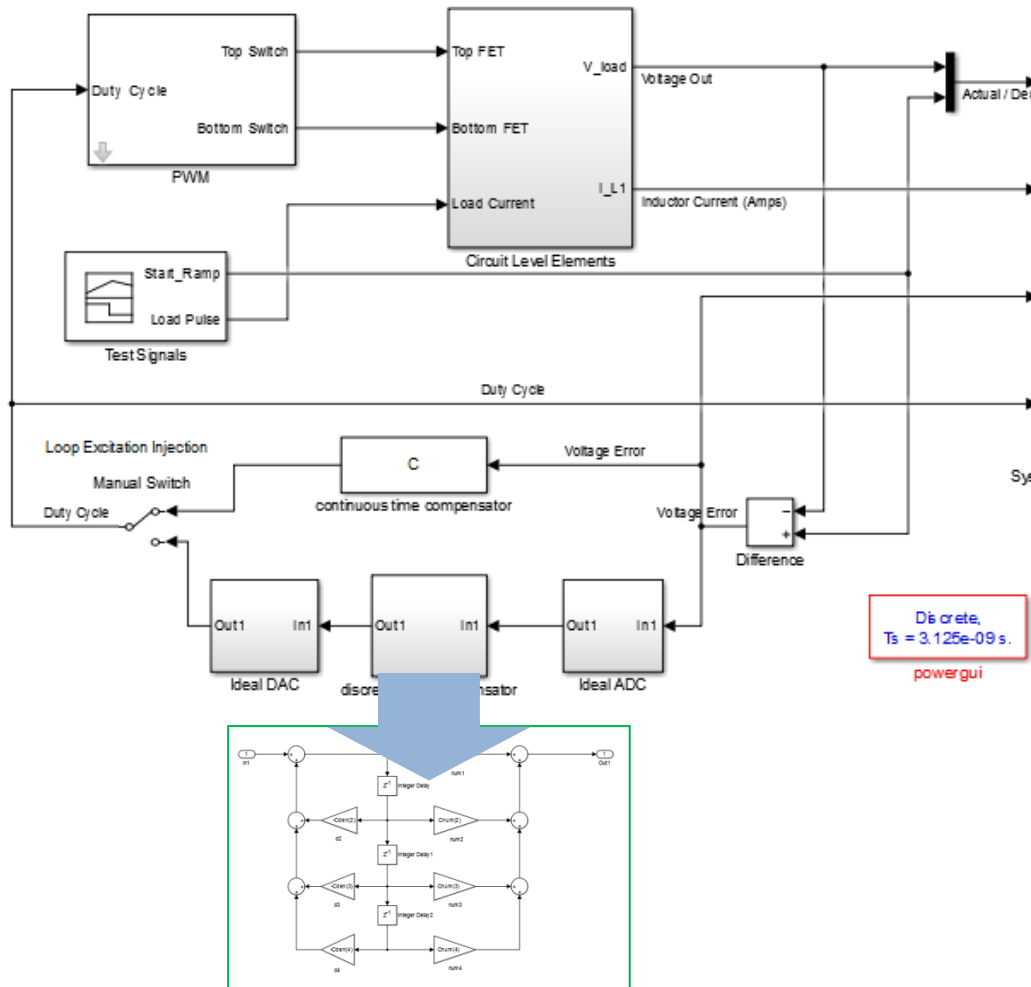
デモ4: デジタル回路設計・検証での活用

デモ概要

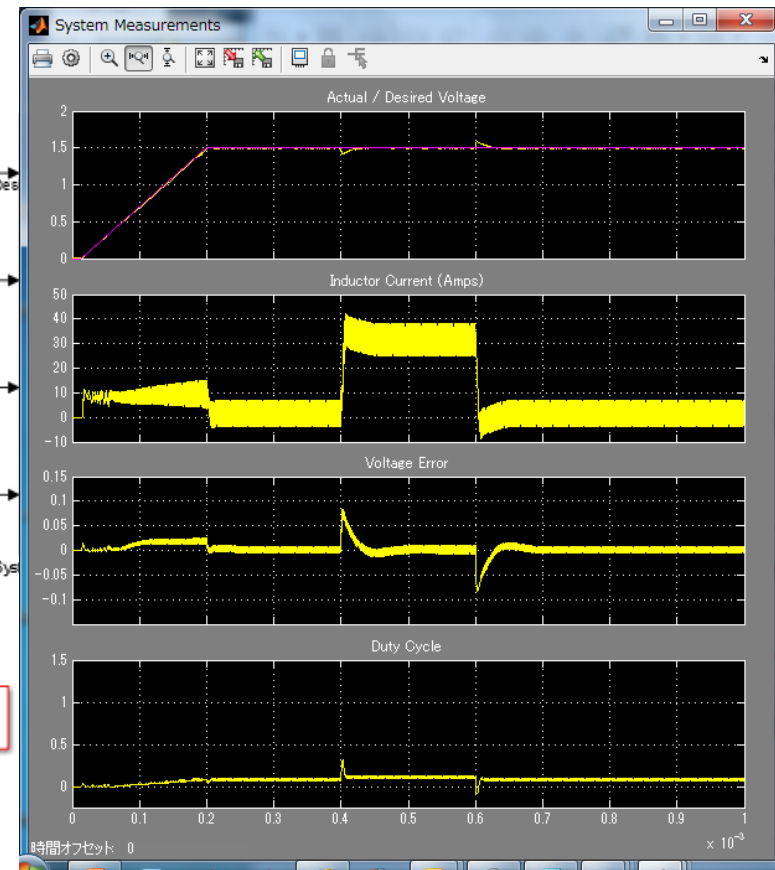
- デジタル部のモデルを離散化、固定小数点化を容易に実現
- ブロックモデルからHDLコード生成、システム設計時から高精度のモデリングが可能



デモ4: デジタル回路設計 補償器を離散化、モデル化

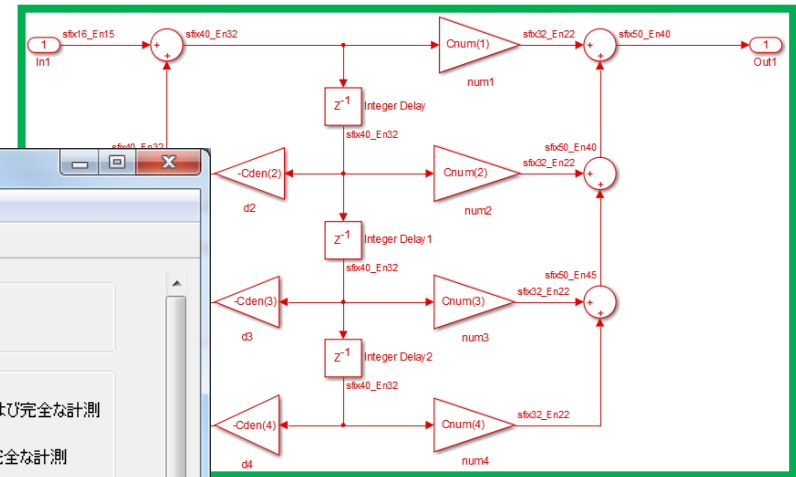
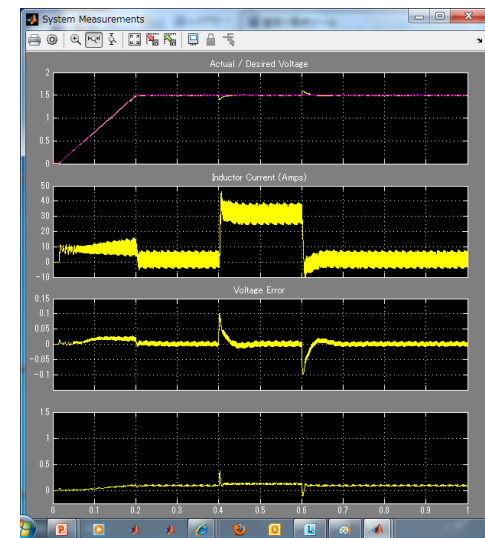
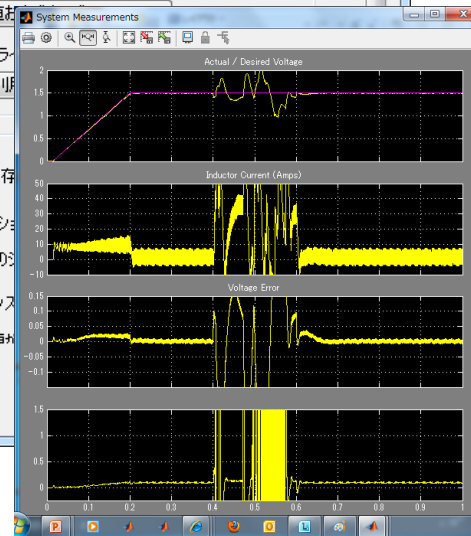


Discrete,
 $T_s = 3.125e-09$ s.
powergui



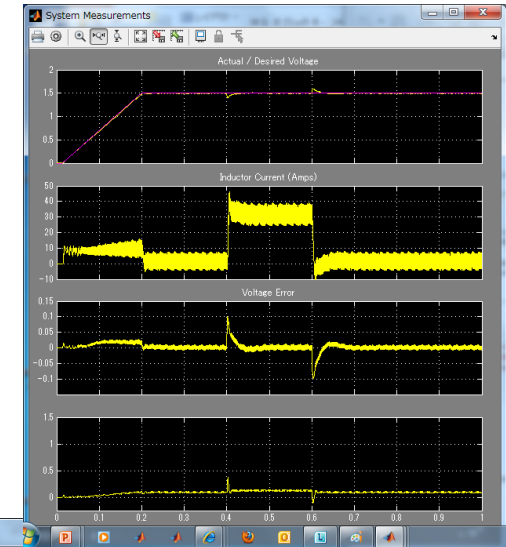
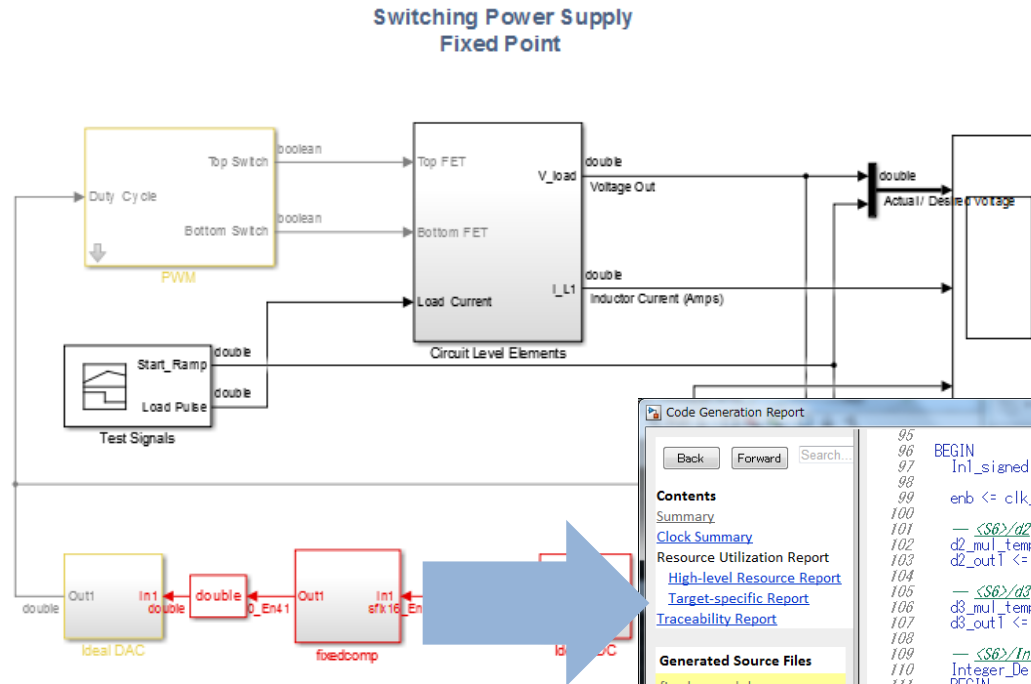
補償器を離散化してモデルに置き換える

デモ4: デジタル回路設計 固定小数点化

固定小数点ツールでオートスケーリング可能

デモ4: デジタル回路設計 HDLコード生成



Code Generation Report

Back Forward Search...

Contents

- Summary
- Clock Summary
- Resource Utilization Report
- High-level Resource Report
- Target-specific Report
- Traceability Report

Generated Source Files

- fixedcomp.vhd

Referenced Models

```

95 BEGIN
96   In1_signed <= signed(In1);
97
98   enb <= clk_enable;
99
100   -- <S6>/d2
101   d2_mul_temp <= signed('X'4BE1EAA2D8") * Integer_Delay_out1;
102   d2_out1 <= d2_mul_temp(76 DOWNTO 45) + ('0' & (d2_mul_temp(44) AND
103
104   -- <S6>/d3
105   d3_mul_temp <= signed('X'8A932A5178") * Integer_Delay1_out1;
106   d3_out1 <= d3_mul_temp(77 DOWNTO 46) + ('0' & (d3_mul_temp(45) AND
107
108   -- <S6>/Integer_Delay
109   Integer_Delay_process : PROCESS (clk, reset)
110   BEGIN
111     IF reset = '1' THEN
112       Integer_Delay_out1 <= to_signed(0, 40);
113     ELSIF clk'EVENT AND clk = '1' THEN
114       IF enb = '1' THEN
115         Integer_Delay_out1 <= Sum_out1;
116       END IF;
117     END IF;
118   END PROCESS Integer_Delay_process;
119
120
121
122   -- <S6>/Integer_Delay1
123   Integer_Delay1_process : PROCESS (clk, reset)
124   BEGIN
125     IF reset = '1' THEN

```

ブロックモデルからHDLコード生成、システム設計時から詳細設計可能

デモ4: デジタル回路設計 AMS検証

The screenshot displays a circuit diagram in an AMS simulation environment. The diagram includes several interconnected blocks: a clock generator (clkrogen), a PWM controller (PWL_dpl), a buck converter (BuckCircuit_U_Bottle), and a system controller (Controller_dpl). A large blue arrow points from the circuit diagram towards the Code Generation Report window on the right.

The Code Generation Report window shows the following content:

```

Code Generation Report
Back Forward Search...

Contents
Summary
Clock Summary
Resource Utilization Report
High-level Resource Report
Target-specific Report
Traceability Report

Generated Source Files
fixedcomp.vhd

Referenced Models

95 BEGIN
96   In1_signed <= signed(In1);
97
98   enb <= clk_enable;
99
100   -- <S6>/a2
101   d2_mul_temp <= signed('X"4BE1FAA2D8') * Integer_Delay_out1;
102   d2_out1 <= d2_mul_temp(76 DOWNT0 45) + ('0' & (d2_mul_temp(44) AND
103
104   -- <S6>/a3
105   d3_mul_temp <= signed('X"8A932A5178') * Integer_Delay1_out1;
106   d3_out1 <= d3_mul_temp(77 DOWNT0 46) + ('0' & (d3_mul_temp(45) AND
107
108
109   -- <S6>/Integer Delay
110   Integer_Delay_process : PROCESS (clk, reset)
111 BEGIN
112   IF reset = '1' THEN
113     Integer_Delay_out1 <= to_signed(0, 40);
114   ELSIF clk'EVENT AND clk = '1' THEN
115     IF enb = '1' THEN
116       Integer_Delay_out1 <= Sum_out1;
117     END IF;
118   END IF;
119 END PROCESS Integer_Delay_process;
120
121
122   -- <S6>/Integer Delay1
123   Integer_Delay1_process : PROCESS (clk, reset)
124 BEGIN
125   IF reset = '1' THEN

```

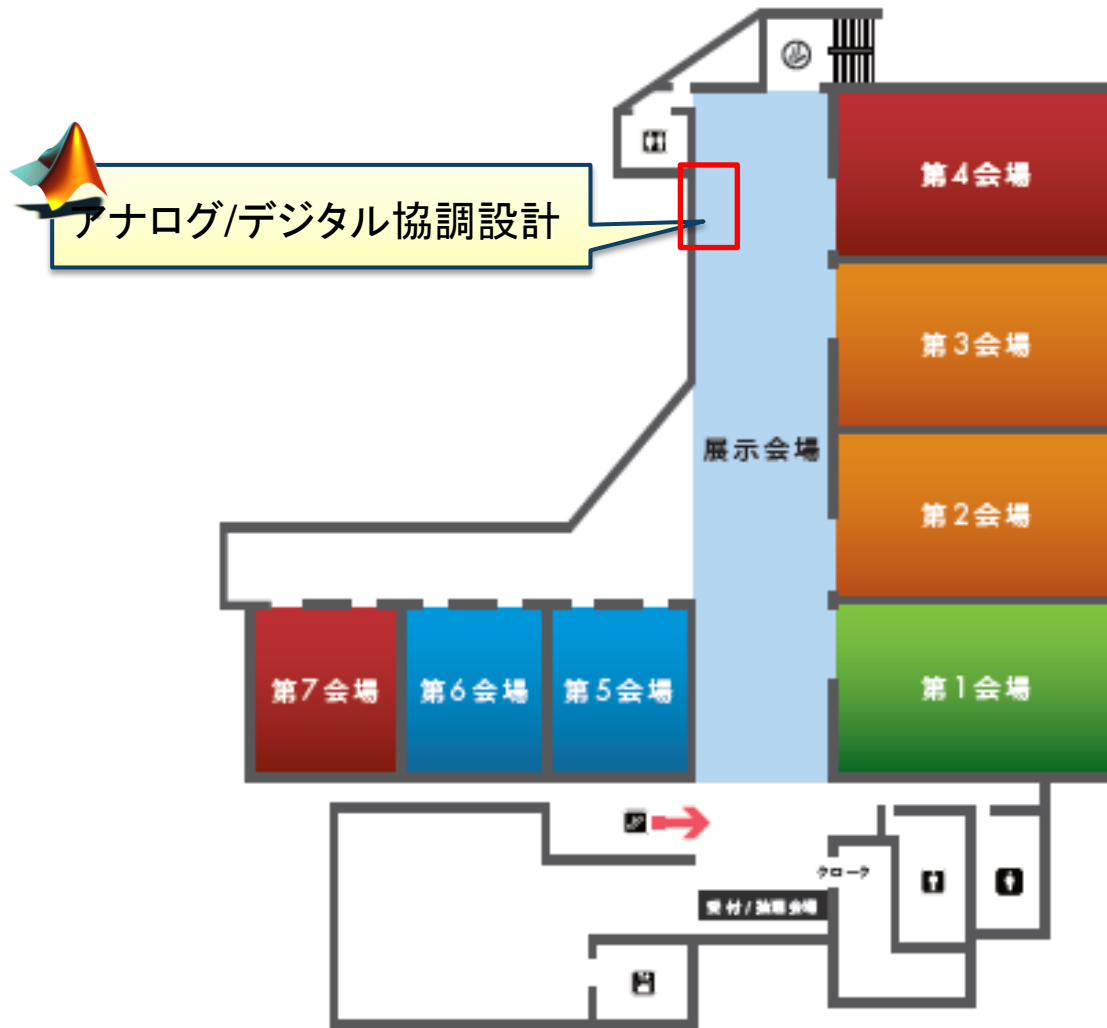
AMSシミュレータ上でデジタル、アナログ部を詳細回路に置き換えての検証

注意: 今回のデモには含まれません。

まとめ

- MATLAB/Simulinkはアナログミクスドシグナルシステム設計におけるSTARC推奨ツールです。
- システム設計において、抽象度が高いモデルから回路素子モデルまでモデリングが可能です。
- MATLAB/Simulinkでシステム設計を行って頂く事で、アナログ・デジタルの回路設計・検証にまで活用頂けます。
- 電源設計に置いてもシステム設計~回路設計でのフローで活用可能です。

デモブースのご案内



Appendix: AMS関連リンクページ

www.mathworks.co.jp/mixed-signal-systems/



Analyze, design, and verify mixed-signal systems

Mixed-signal engineers use MathWorks products to design and verify analog and digital components in the same model at the same time. They use MATLAB and Simulink and related products for behavioral modeling, rapid design exploration, predesign analysis, and verification of mixed-signal systems.

With MathWorks products you can:

- Create behavioral models of PLLs, DACs, ADCs, and other mixed-signal systems
- Efficiently simulate and improve mixed-signal designs
- Evaluate analog-digital design tradeoffs
- Verify designs before producing test chips

Learn more

[Analog and Digital Design – Combining Components Using Design Abstraction](#)
[Cosimulating and Verifying Mixed-Signal Systems](#)
[Mixed-Signal Analysis](#)
[ADC Design](#)
[PLL Design](#)
[Digital Predistortion](#)



無償ミックスドシグナル
ライブラリ

Contact Sales

Digital Predistortion

各種ユーザ事例

Digital predistortion is simple in theory, but difficult in practice. Engineers need to control sophisticated test equipment, analyze complex data, and build algorithms for DSPs or FPGAs. These individual tasks can require different tools, adding to the complexity. MATLAB provides a unifying environment for all these tasks. Engineers can use MATLAB to control power amplifier measurements, create digital predistortion algorithms, and generate C and HDL code for implementation.

Nujira built their digital predistortion and envelope tracking measurement system using MATLAB. In addition, MATLAB controls their Tektronix, Rohde & Schwarz, and Agilent test equipment via the SCPI and GPIB interfaces and automates their testing process. Commscope uses MATLAB to analyze their large power amplifier data sets to create their digital predistortion algorithms. Alcatel-Lucent employed a similar process to develop their algorithms, and used Xilinx® System Generator to develop digital predistortion code for their Xilinx FPGAs: “The quality of the generated code was so high that we didn’t make a single change in the production system,” said Dr. Rudolf Wessel, engineering manager at Alcatel-Lucent.



[Digital Predistortion: Seeing Clearly in the House of Mirrors](#)



[Digital Predistortion for Power Amplifiers](#) 3:04