

具体例で学ぶ信号処理ワークフロー ～すぐに使えるテクニック:実践編～

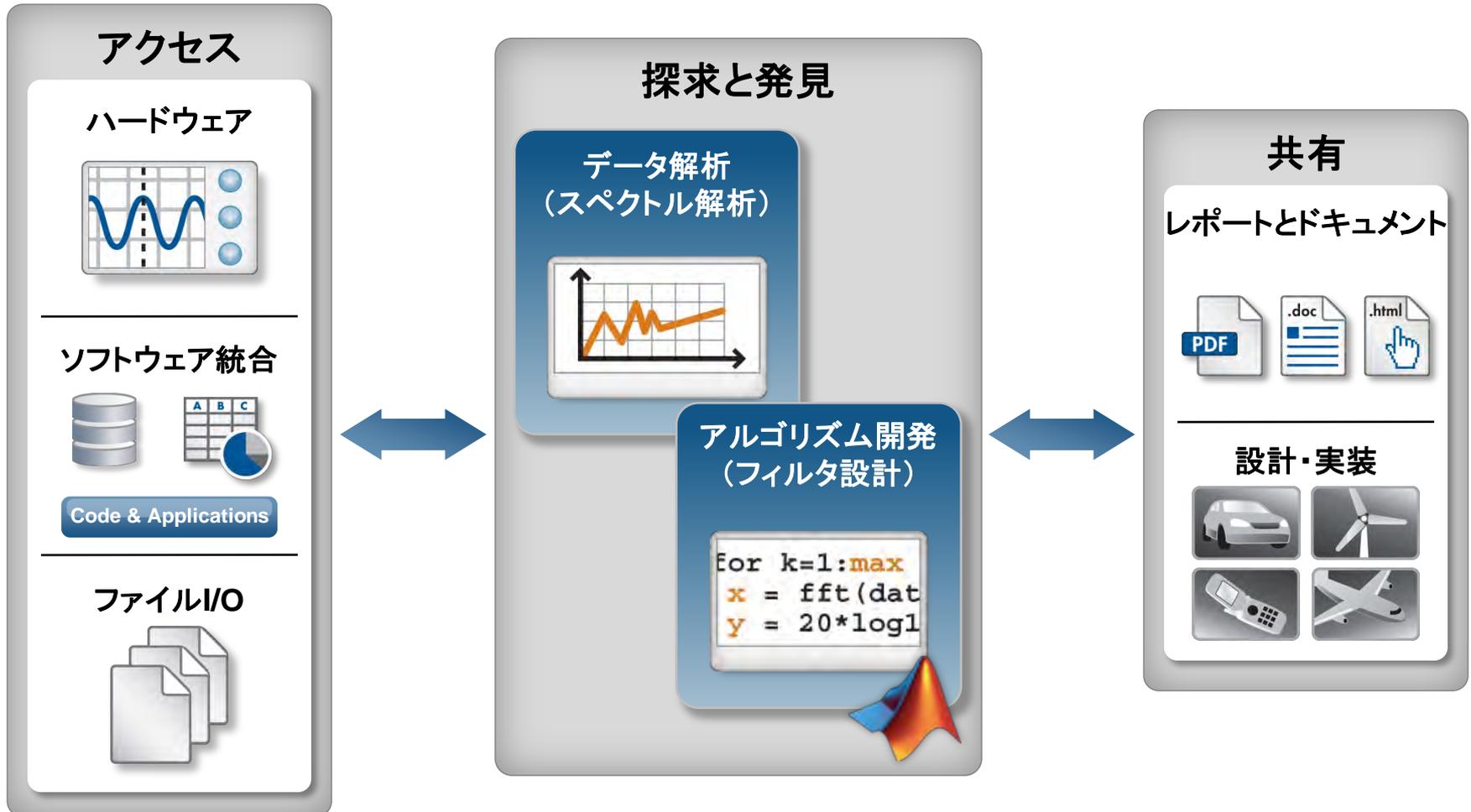
MathWorks Japan

アプリケーションエンジニアリング部

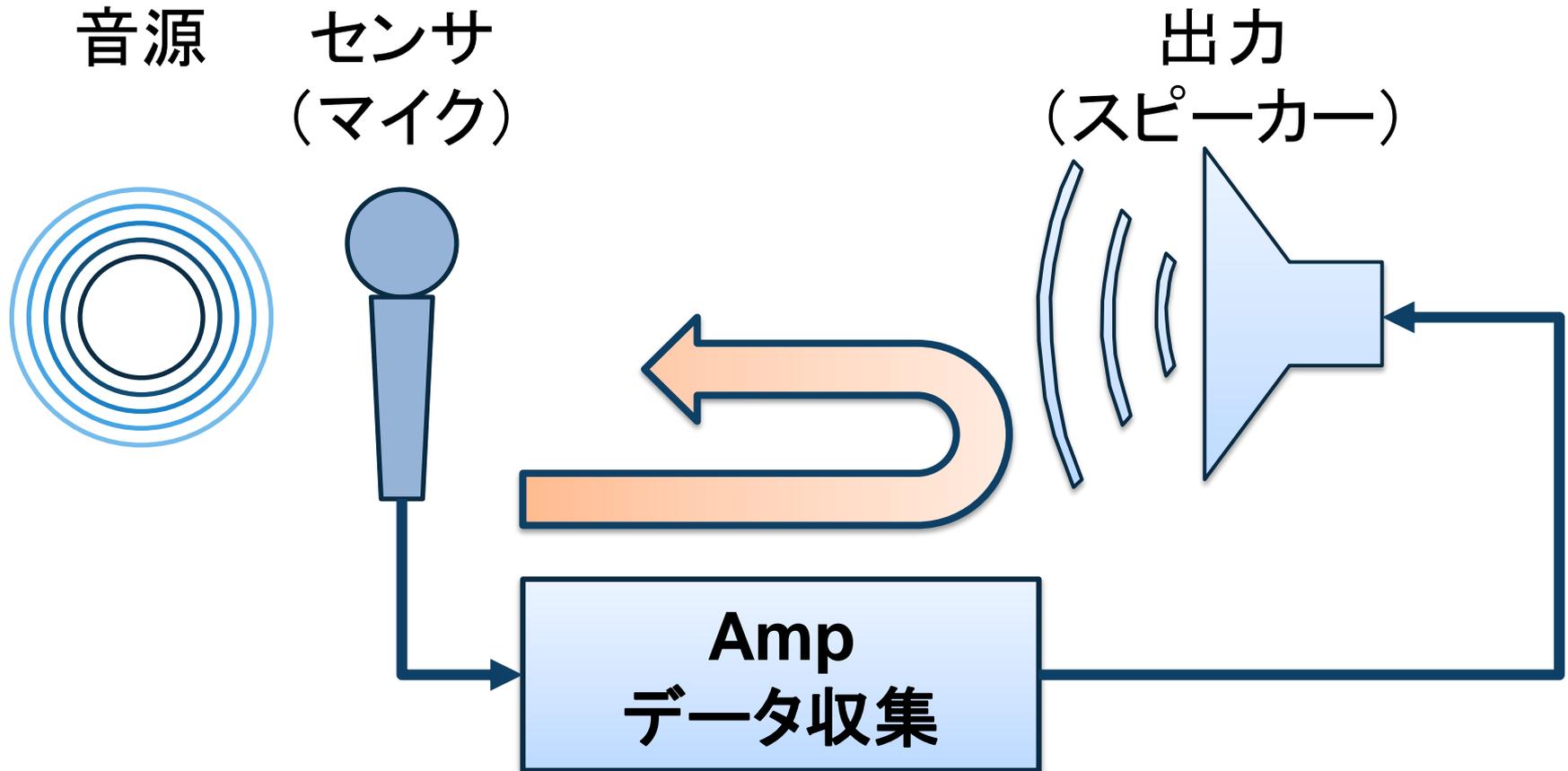
シニアアプリケーションエンジニア

松本 充史

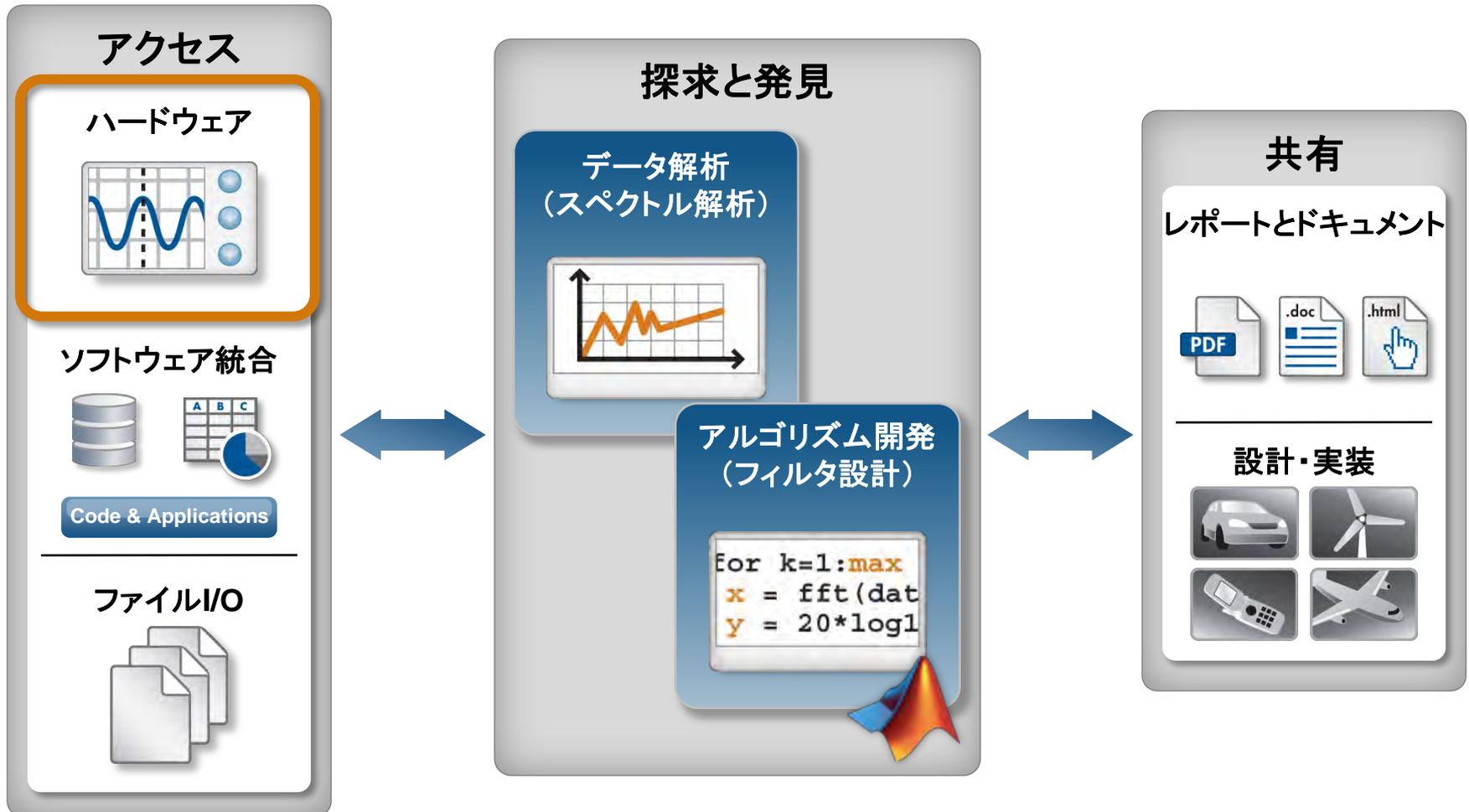
アジェンダ: エンジニアリング・ワークフロー



例題：信号のハウリングとその除去



アジェンダ: エンジニアリング・ワークフロー



ハードウェアデバイスからデータ取得用オプション

DSP System Toolbox

- 信号処理オプション
 - フィルタ設計、解析、スペクトル推定など
- サウンドカードへのアクセス機能を提供
 - マルチチャンネル対応
 - Kernel Streamingドライバ対応



Data Acquisition Toolbox

- データ収集ボードからデータ取得
- アナログI/O、デジタルI/O
- ハード/ソフトトリガ



Instrument Control Toolbox

- 測定器からのデータ読み込み
- 測定器制御
- 各種通信プロトコル対応



Image Acquisition Toolbox

- 画像、映像を直接取り込み
 - フレームグラバ、カメラ
 - DCAM, Camera Link, GigE
 - Web Camera(VFM, WDM)



オンラインデータの解析やシミュレーションでの利用に有効

ハードウェアからのデータ収集： オーディオデータ

- 要求：
 - チャンネル数 (Mono/Stereo, Multi Ch)
 - トリガ/イベント
 - 低レイテンシ



ハードウェアからのデータ収集: オーディオデータ

要求	関数/ブロック	オプション	ドライバ/ ハードウェア
モノラル/ステレオ (2chまで)	audioplayer audiorecorder	MATLAB	Windows Direct Sound/ サウンドカード
トリガ、 コールバック	analoginput analogoutput	Data Acquisition Toolbox	Windows Direct Sound/ サウンドカード、 データ収集ボード
マルチ チャンネル データ (3ch以上)	dsp.AudioRecorder, dsp.AudioPlayer/ From Audio Device, To Audio Device	DSP System Toolbox	Windows Direct Sound/ サウンドカード、 オーディオインターフェース
低レイテンシ	同上	同上	WDM-KS/ オーディオインターフェース
低レイテンシ	同上(要PortAudio ASIOドライバの統合)	同上	ASIO/ オーディオインターフェース

ハードウェアからのデータ収集： オーディオデータ以外

要求	関数/ブロック	オプション	ハードウェア/ ドライバ
デジタルデータ	digitalio	Data Acquisition Toolbox	データ収集ボード
UDP	dsp.UDPSender, dsp.UDPReceiver/ UDP Send, Receive	DSP System Toolbox	Ethernetポート
Serial Data (Bluetooth) TCP/IP, UDP	serial(bluetooth), tcpip	Instrument Control Toolbox	シリアルポート (Bluetooth デバイス) Ethernetポート
ビデオカメラ	videoinput	Image Acquisition Toolbox	フレームグラバ, GigE, Camera Link Video for Windows (Webカメラ)

ハードウェアからのデータ収集： サウンドカードからオーディオデータ収集

%% Objectの定義

```
hain = dsp.AudioRecorder( 'SampleRate', 22050,  
'NumChannels',1);
```

```
hmfw = dsp.AudioFileWriter( 'wv_matlab.wav',  
'SampleRate', 22050);
```

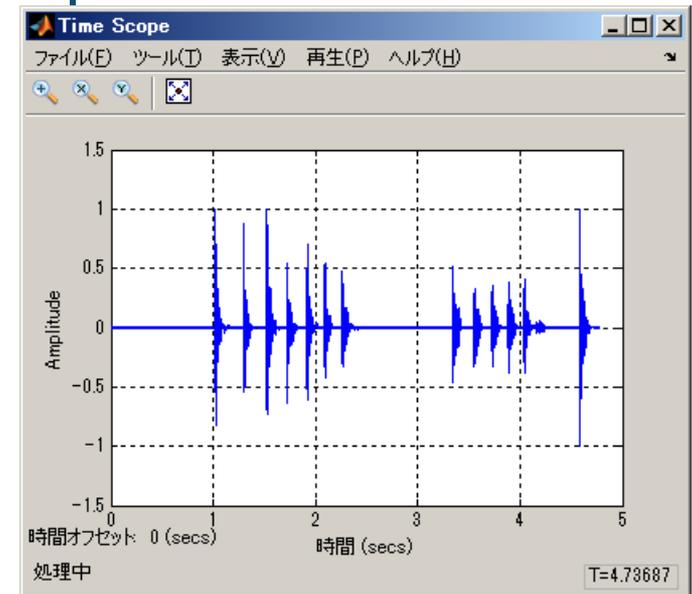
```
hts = dsp.TimeScope( 'SampleRate',22050,  
'YLimits',[-1.5 1.5], 'TimeSpan', 5, 'Grid', true);
```

tic %% ストリーミング処理

```
while toc < 5
```

```
    out = step(hain);           % Audio In  
    step(hmfw,out);           % File Write  
    step(hts,out);            % Plot
```

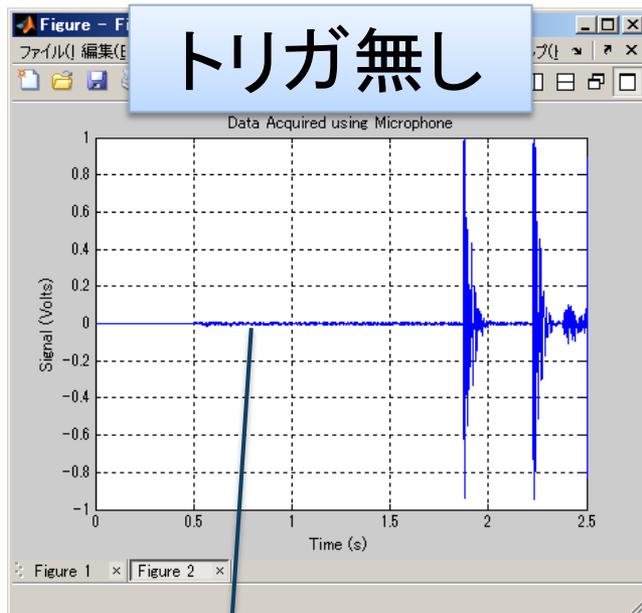
```
end
```



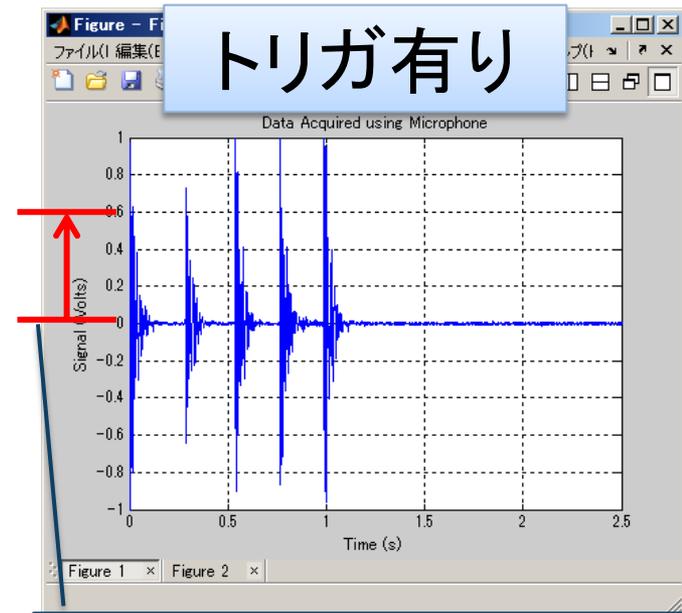
Demo: *hardware_dspsyso.m*

トリガ信号によるデータ収集:

- ソフトウェアトリガ機能により必要箇所だけの収録
 - Threshold, \uparrow/\downarrow , トリガCh, 取得時間などの設定



録音開始時の無音時間により
必要部分の取り逃し



スレッシュホールドを超えたら
録音開始

Demo: [hardware_daq.mdl](#)

リアルタイムシミュレーションに対する要求： 低レイテンシ

- Kernel Streaming技術
 - Windows OSのKernel ミキサーを含むレイヤー層をバイパス、ハードウェア・ドライバの直接アクセスにより低レイテンシを実現
- Windows OS対応Kernel Streamingドライバ
 - WDM-KS:DSP System Toolboxで対応
 - ASIO:MATLAB非対応⇒拡張可能

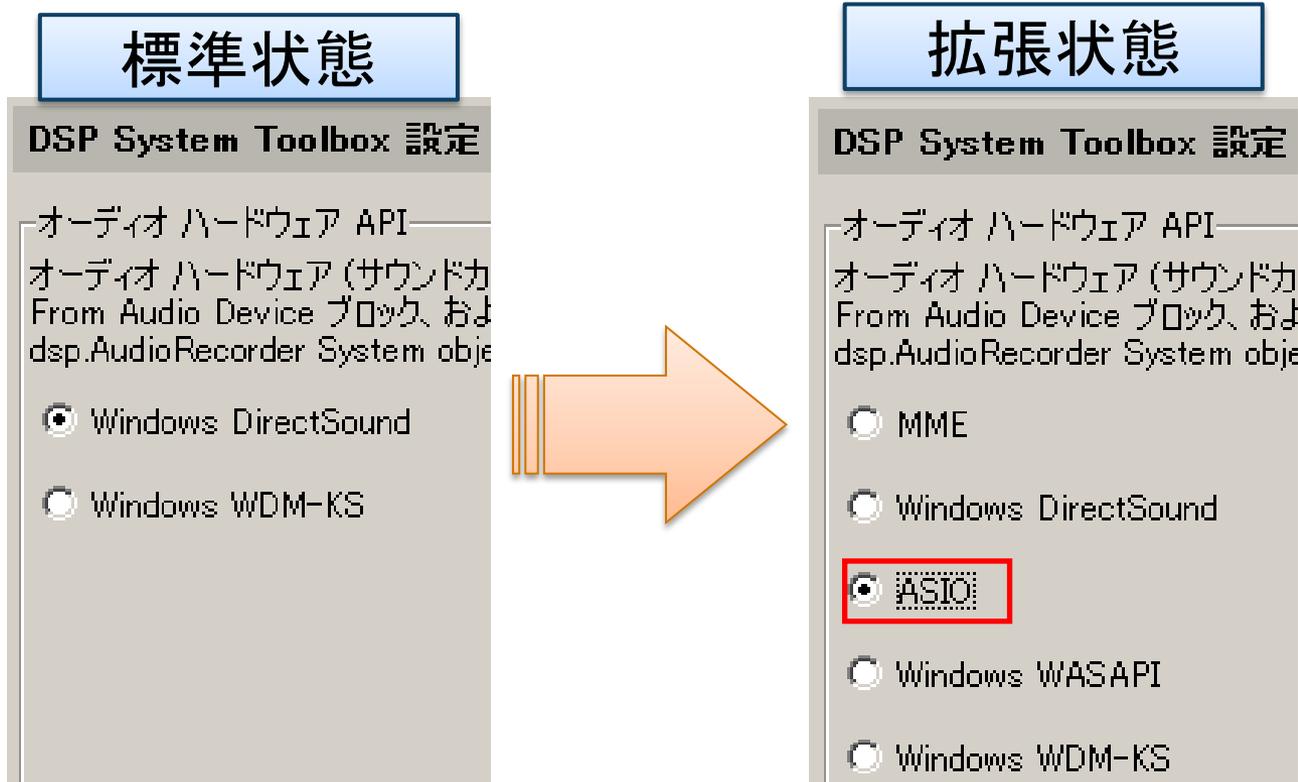
ドライバの違いによるレイテンシ測定結果
(出力+入力レイテンシの総和)



ドライバ	Buffer Size	Fs (kHz)	Queue Duration	レイテンシ	
				Samples	msec
Windows DirectSound	16	48	0.1	16962	353.4
WDM-KS	32	48	0.12	14445	300.7
ASIO	16	48	0.03	2603	54.2

ハードウェアからのデータ収集： ASIOドライバのインポート

- DSP System Toolboxで選択可能なドライバを拡張
 - オープンソースライブラリ、DirectX SDK、ASIO SDKなどを利用
 - [ファイル]メニュー⇒[設定]⇒DSP System Toolbox



ハードウェアからのデータ収集： ASIOドライバのインポートに必要なソフトウェア

- ASIOドライバのインポートに必要なソフトウェア
 - Microsoft DirectX SDK
 - Microsoft Visual Studio (MSVC)
 - Steinberg ASIO SDK (ASIOドライバ開発元)：
<http://www.steinberg.net/en/company/developer.html>
 - PortAudio library (オープンソースAudio I/Oライブラリ)：
<http://www.portaudio.com/>

ハードウェアからのデータ収集： ASIOドライバの利用手順

1. ¥portaudio¥src¥hostapi¥asioにASIOSDK2をコピーして
asiosdkにフォルダ名を変更
2. C:¥Program Files¥<Microsoft DirectX SDK Lib>¥Include
からdsound.hとdsconf.hを¥portaudio¥includeにコピー
3. ¥portaudio¥build¥msvc¥にあるportaudio.slnをMSVCで開
いて C:¥Program Files¥<Microsoft DirectX SDK
Lib>¥Lib¥x86にあるdsound.libをMSVCに追加
MSVCプロジェクトのプロパティで、構成:[すべての構成]を選
択、C/C++⇒最適化⇒フレームポインタなし=はい
C/C++ ⇒コード生成⇒構造体メンバのアライメント=4バイト、
C/C++ ⇒コード生成⇒ランタイムライブラリ=マルチスレッド
C/C++ ⇒コード生成⇒浮動小数点モデル=Fast

ハードウェアからのデータ収集： ASIOドライバの利用手順(つづき)

4. 構成:[Release]に設定して、C/C++ ⇒プリプロセッサ⇒
PA_ENABLE_DEBUG_OUTPUTを削除、
PA_USE_ASIO=1に設定
5. メインウィンドウにてRelease設定にしてビルド
¥portaudio¥build¥msvc¥Win32¥Releaseに生成される
portaudio_x86.dllをlibportaudio.dllにファイル名変更して
\$matlabroot¥bin¥win32に移動
6. MATLABを再起動
ファイルメニューの設定⇒DSP System Toolbox設定=ASIO

ハードウェアからのデータ収集： ASIOドライバの利用

- ⊕ 一般
- ⊕ キーボード
- ⊕ フォント
- ⊕ 色
 - コード アナライザ
 - ツール バー
 - コマンド ウィンドウ
 - コマンド履歴
- ⊕ エディター/デバッガ
- ヘルプ
- Web
- 現在のフォルダ
- 変数エディタ
- ワークスペース
- GUIDE

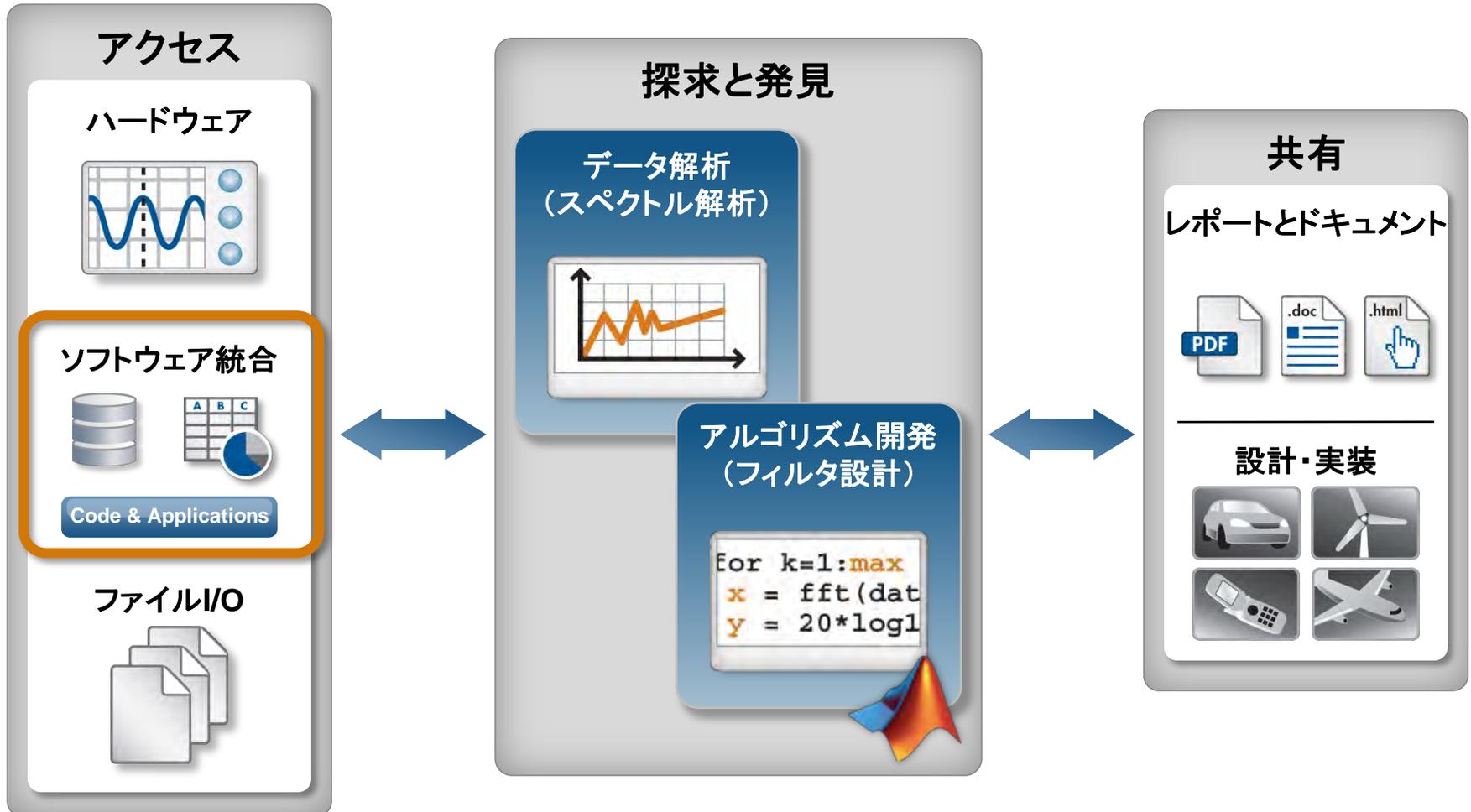
DSP System Toolbox 設定

オーディオ ハードウェア API

オーディオ ハードウェア (サウンドカード) との通信に使用する API を To Audio Device ブロックと From Audio Device ブロック、および dsp.AudioPlayer System object と dsp.AudioRecorder System object に対して選択してください。

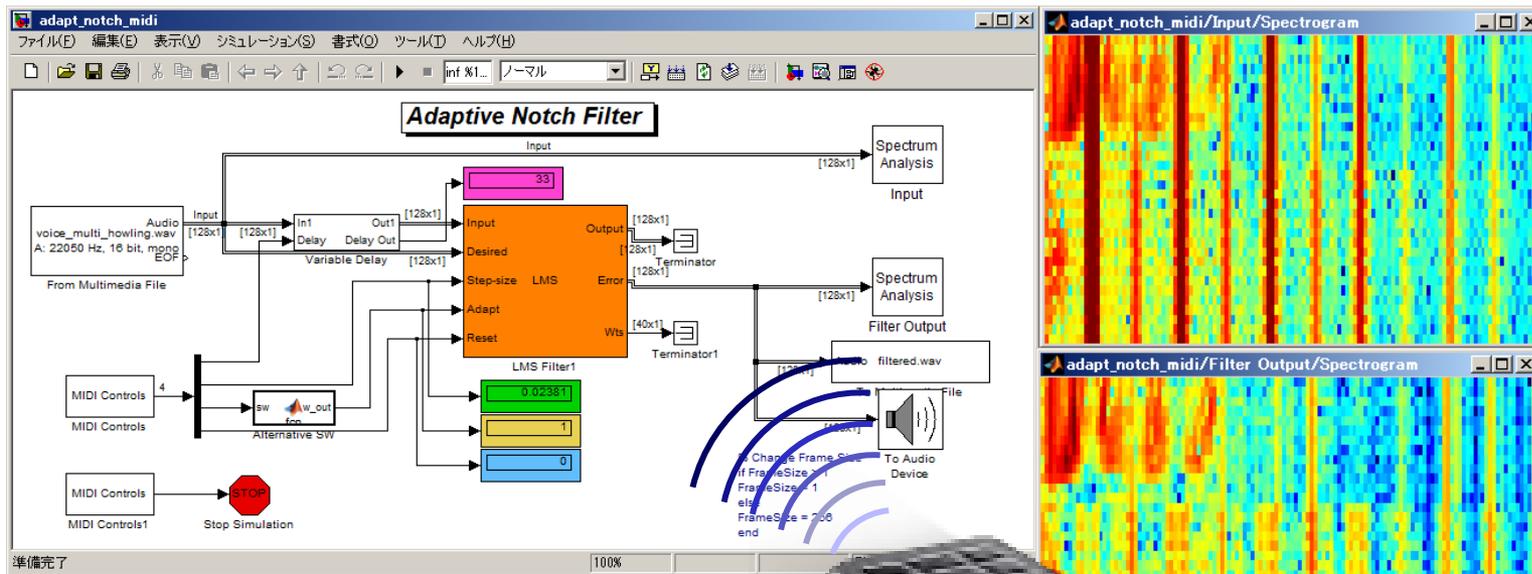
- MME
- Windows DirectSound
- ASIO**
- Windows WASAPI
- Windows WDM-KS

アジェンダ: エンジニアリング・ワークフロー



ソフトウェア資産の統合:

- 例: Simulinkのパラメータチューニングを簡単に!
⇒外部コントローラの利用



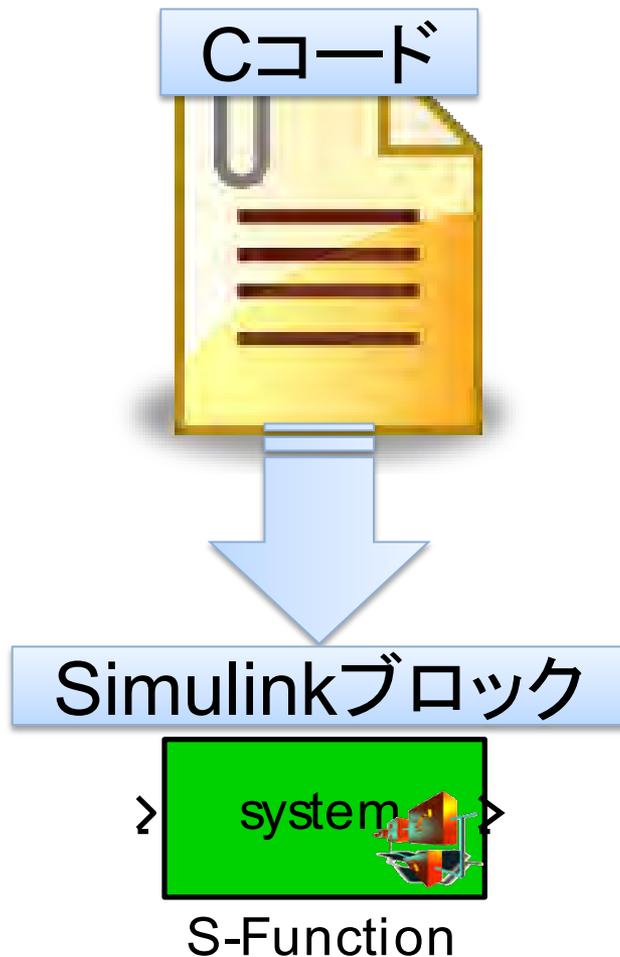
Demo: `dsp_anc_audio_midi.mdl`

ソフトウェア資産の統合：

- 汎用のコントローラをSimulinkのコントローラとして流用
 - ボリュームノブやフェーダーの操作量
⇒例：MIDI-USB機器のControl Change信号
 - 加速度や姿勢情報
⇒例：Bluetooth対応デバイス
- Windows APIを利用する場合はC言語での開発が便利



S-Function機能によりCコードを Simulinkブロック化 (Simulink標準機能)

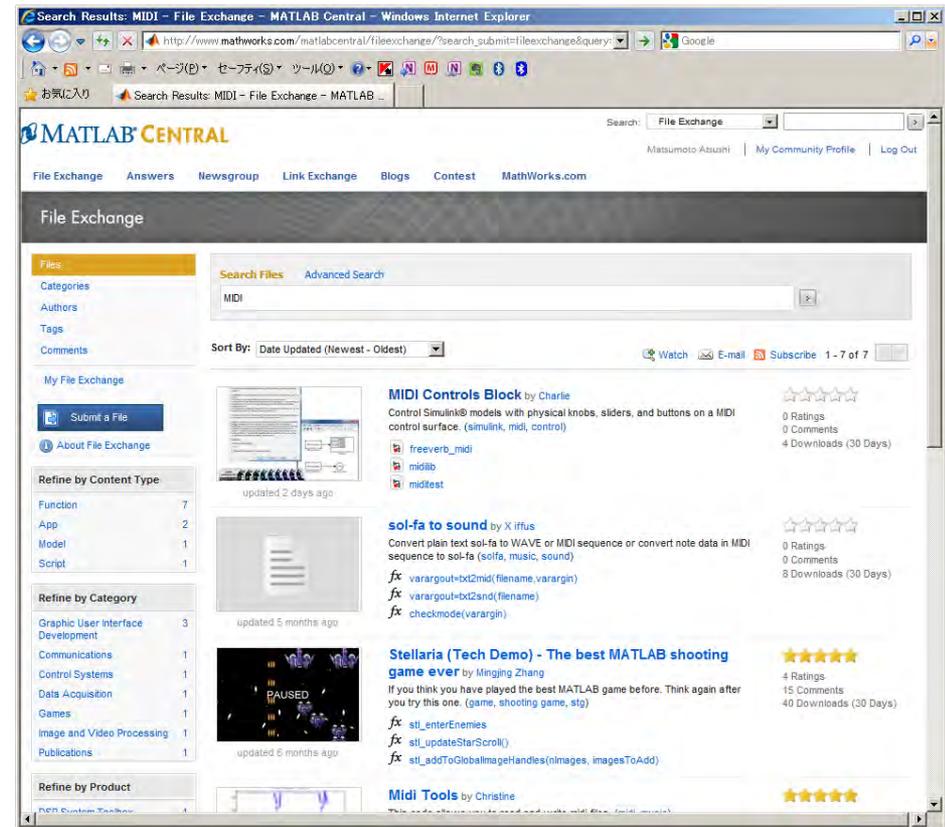


- S-functionの利用用途
 - C設計資産の再利用
 - Windows APIなどの利用
- 作成手段
 - 手書き: ユーザCコードにSimulink APIを追加
 - Legacy Code Tool機能: MATLAB コマンドでAPIを自動追加
 - S-Function Builder: GUIでAPIを自動追加
- C⇒MATLABはMEX機能

MATLAB/Simulinkユーザコミュニティ

MATLAB® CENTRAL

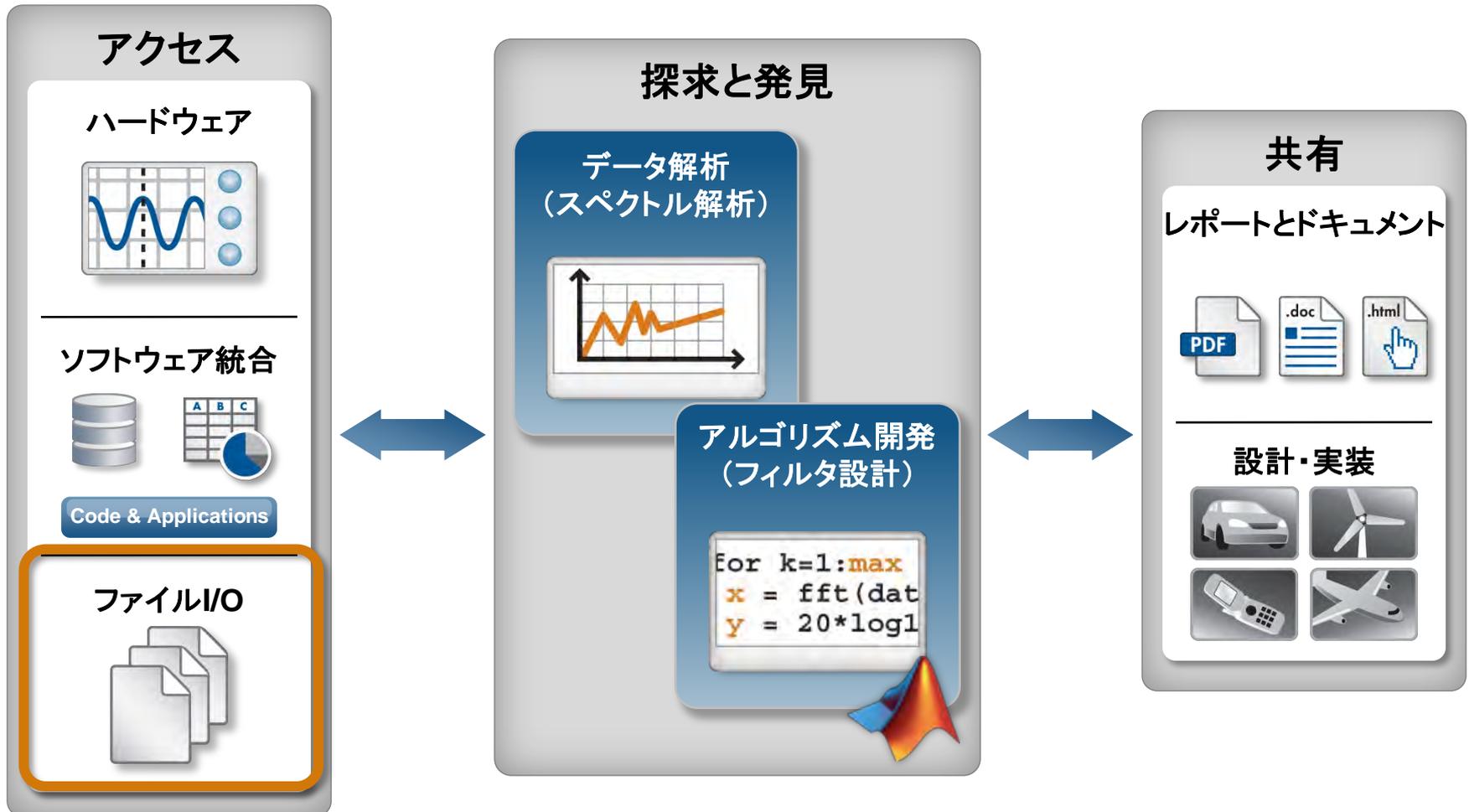
- File Exchange
 - ユーザが投稿したMATLABコード、Simulinkモデル等のファイルをダウンロード
 - 製品にない機能を利用可能
- ニュースグループ
- ブログ
 - MathWorks開発者の投稿



MIDI Controlブロックのダウンロード

<http://www.mathworks.com/matlabcentral/fileexchange/31537>

アジェンダ: エンジニアリング・ワークフロー



ファイルI/O: Text data Read/Write関数一覧

関数	非数値データ	範囲指定	
csvread csvwrite	No	Yes	カンマ区切りのみ
dlmread dlmwrite	No	Yes	区切り文字指定可能
load/save	No	No	Textは数値データのみ MATファイルにも対応
textscan	Yes	Yes	データフォーマットを指定してRead可能 fopen/fcloseが必要
importdata	ヘッダのみ	No	Text, Excel, MAT, WAV, イメージ
ファイルメニュー/ ダブルクリック	Yes (csv以外はヘッ ダのみ)	Yes (csvのみ)	同上

ファイルI/O: MATファイル

- MATファイル: MATLAB形式のバイナリファイル
- MATLABバージョンによってフォーマットが異なる

MAT File Version	MATLAB Version	特徴
V4	1 – 4	非圧縮, 2D double, 文字, スパース行列
V6	5 – 6	非圧縮、N次元配列、構造体、セルのサポート
V7	7.0 – 7.2 (R14 - R2006a)	データ圧縮、Unicode 文字エンコーディングのサポート、現在バージョンのデフォルト設定
V7.3	7.3 – (R2006b -	HDF5フォーマットを使用 2GB 以上のデータ(64bit OS)対応 部分的なRead/Write オーバーヘッドがあるためV7より大容量になりがち

ファイルI/O: マルチメディアファイル (Windows)

ファイルの種類	関数/ブロック	オプション	備考
wav	wavread, wavwrite	不要	
au	auread, auwrite	不要	
avi, mpg, wmv など	VideoReader, VideoWriter	不要	Microsoft DirectShowがサポートする任意形式に対応
wav, wma, aif, mp3, au, snd	dsp.AudioFileReader dsp.AudioFileWriter/ From Multimedia File, To Multimedia File	DSP System Toolbox	Simulink/System Object のストリーミング・シミュレーションで高速に利用可
qt, mov, avi, asf, asx, wmv, mpg, mpeg, mp2など	vision.VideoFileReader, vision.VideoFileWriter/ From Multimedia File, To Multimedia File	Computer Vision System Toolbox	同上 Microsoft DirectShowがサポートする任意形式に対応

ファイルI/O: 低水準関数の利用~HEXデータの読み書き

課題: 16進数データ列のRead/Write (例: 0x7abc)

- Write: `fprintf(fid, '0x%+04x¥n', data)`
- Read: `fscanf(fid, '%x¥n')`

符号付きデータの場合: オフセット加える

- Write: `fprintf(fid, '0x%+04x¥n', s1 + (s1 < 0) * 2^16);`
- Read: `s2 = s2 - (s2 > 2^15 - 1) * 2^16;`

※ 16進数変換コマンドdec2hexは負のデータ非対応

Demo: `readwrite_hex.m`

ファイルI/O: MATファイルの使い方

- バージョンの切り替え方 (Write時)

```
>> save('filename','A','-v7.3')
```

```
>> load('filename')
```

– またはデフォルト設定の変更 ([ファイル]->[設定]->[一般]->[MAT
ファイル])

- 部分的にMATファイル内のデータを修正

```
>> io_exp = matfile('data_v73_partial.mat','Writable',true);
```

```
>> io_exp.A2(1,500000:end) = 0.5*A2(1,500000:end);
```

- 部分的にMATファイルのデータを読み込み

```
>> io_exp = matfile('data_v73_partial.mat');
```

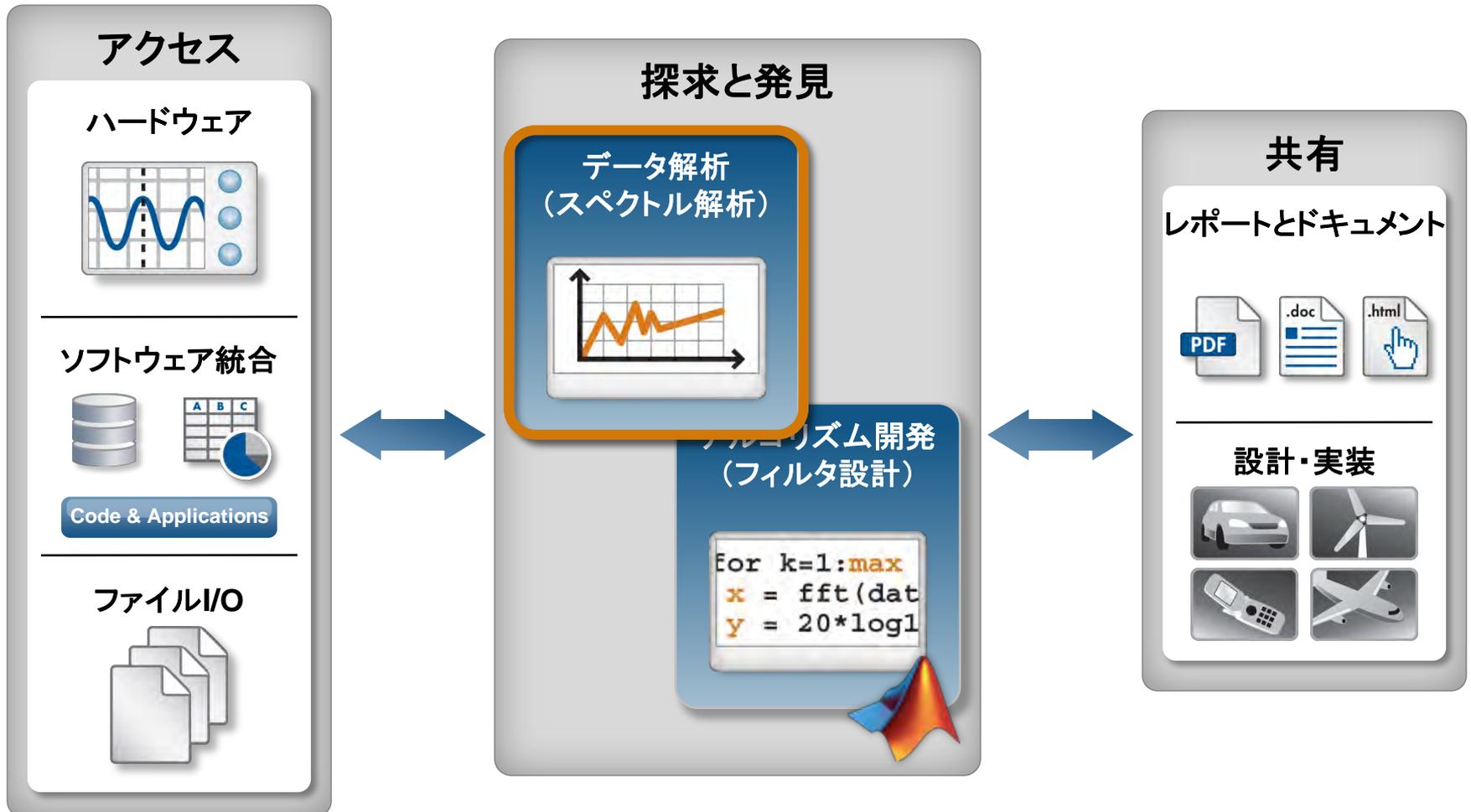
```
>> A2_partial = io_exp.A2(1,1:100000);
```

Demo: readwrite_mat.m

ファイルI/O: MATファイル

- Version 5 MATファイル (MATLAB Version5以降で対応) はフォーマットを公開
 - http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/matfile_format.pdf
- Version 7.3はHDF5形式 (オープンソースのファイルフォーマット)
 - <http://www.hdfgroup.org/>

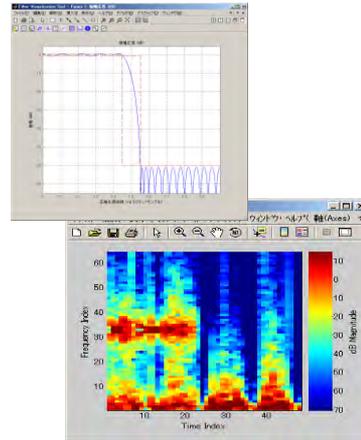
アジェンダ: エンジニアリング・ワークフロー



信号処理設計・解析・シミュレーション用オプション

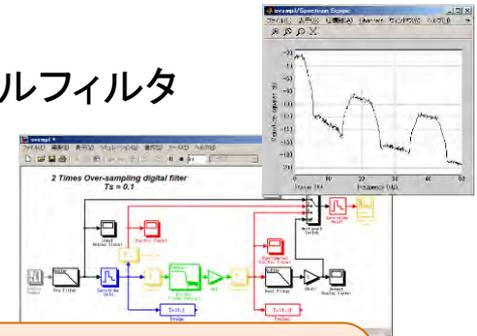
Signal Processing Toolbox

- 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
- 窓関数
- 線形予測
- GUI
 - フィルタ設計、信号解析、窓関数解析



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
- ストリーミングでのシミュレーション
- スペクトル推定
- アナログ・デジタルフィルタ
- 行列、線形代数
- FFT/DCT/DWT



信号処理システム開発・解析およびシミュレーションを
短期間で実現

~R2010b

R2011a~

Filter Design
Toolbox



Signal Processing
Blockset



DSP
System Toolbox

窓関数

窓関数の効果

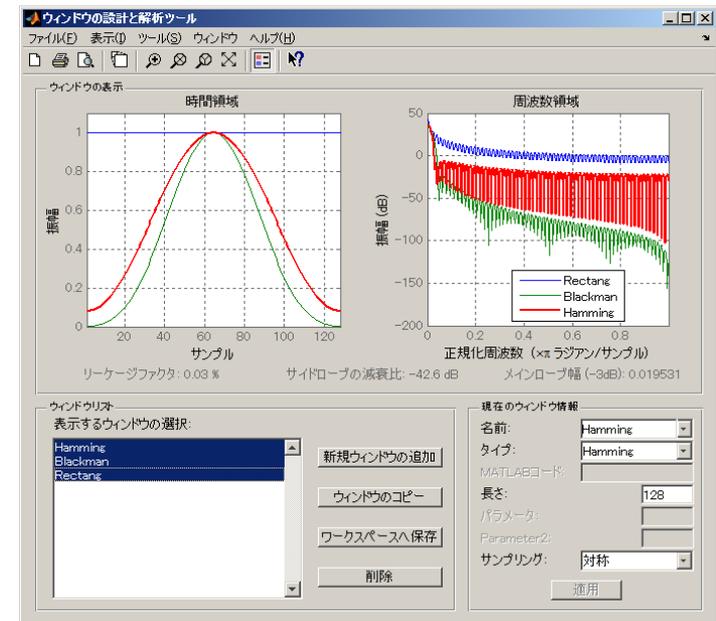
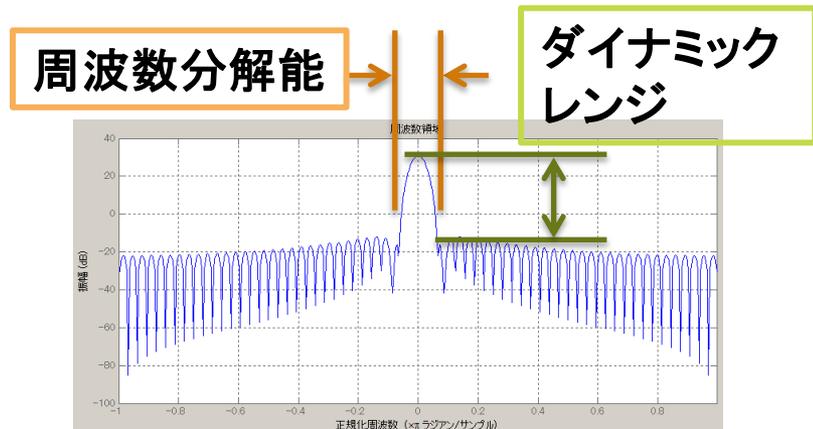
- フーリエ変換の対象となる信号両端の不連続点を減少

トレードオフ

- メインローブが狭い(周波数分解能)
- サイドローブが低い(ダイナミックレンジ)

提供されている窓関数

- barthannwin, bartlett, blackman, blackmanharris, bohmanwin, chebwin, flattopwin, gausswin, hamming, hann, kaiser, nuttallwin, parzenwin, rectwin, tukeywin, triang

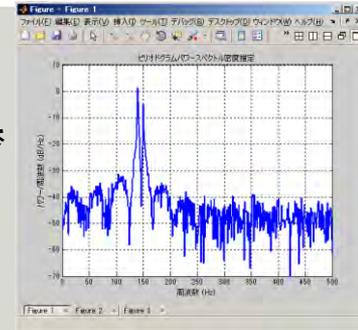


`>> wintool`

スペクトル推定方法

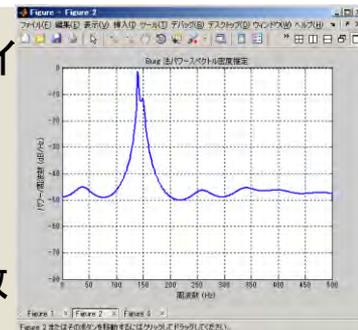
ノン パラメトリック法

- 信号から直接PSD(パワースペクトル密度)推定
- 基本形Periodogramに対し、WelchやMultitaperでは分散を減少するための処理
- 手法: Periodogram, Welch, Multitaper



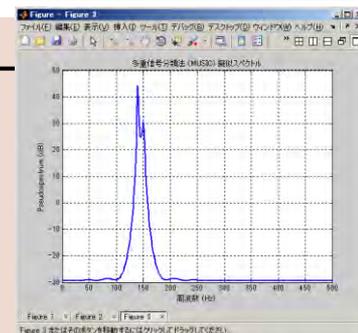
パラメトリック法

- 推定PSDはパラメータを持つ線形システムにホワイトノイズを入力した結果と仮定
- 滑らかだが誤差が発生しやすい
- 短い信号で良好な傾向
- 手法: Yule-Walker AR, Burg, 共分散, 修正共分散



部分空間法

- 自己相関行列の固有解析または固有値分解をベースに、信号に対する周波数成分の推定
- 周波数解像度が高いが振幅は誤差大
- ノイズが混入した正弦波などで有効
- 手法: MUSIC, 固有ベクトル



ノンパラメトリック法による関数/オブジェクト

手法	関数/ オブジェクト	処理内容	特徴
Periodogram	periodogram/ periodogram	FFTした結果を信号点数で平均化	データ長が長く、SNRが高い場合に有効。窓関数を使用するのは修正 Periodogram と呼ばれる
Welch	pwelch/ welch	信号をセグメントで分割して窓関数付でFFT	分散は減少するがオーバーラップによる冗長な情報が付加される。周波数分解能は低い。
Multitaper	pmtm/ mtm	許容値を満たす最適な窓関数のバンクを選択して処理	分散を減少させつつ解像度は向上。データ長が短いでも有効。計算負荷は大きい。

パラメトリック法による関数/オブジェクト

手法	関数/ オブジェクト	処理内容	特徴
Yule-Walker AR	pyulear/ yulear	Yule-Walker方程式(行列式)を解くことで前方予測誤差を最小にするようARモデルを求める。	長いデータに対して有効だが、短いデータに対しては精度低い。
Burg	pburg/ burg	Levinson Durbin再帰法を満たす前方/後方予測誤差を最小にするようARモデルを求める。	短いデータに対して高解像度。ピーク位置が初期位相に非常に影響を受ける。
共分散	pcov/ cov	前方予測誤差を最小にするようARモデルを求める。	短いデータに対してYule-Walkerより良好な解像度。不安定なモデルを作成する可能性がある
修正共分散	pmcov/ mcov	前方/後方予測誤差を最小にするようARモデルを求める。	短いデータに対して高解像度。不安定なモデルを作成する可能性がある。 ピーク位置が初期位相に若干影響を受ける。

部分空間法などの関数/オブジェクト

手法	関数/ オブジェクト	処理内容	特徴
MUSIC (Multiple Signal Classification)	pmusic/ music	自己相関行列の固有値解析を行って求める	周波数情報は高解像度。 正弦波+ホワイトガウスノイズ信号に適す。振幅情報に誤差が大きい レーダー、通信などに利用
固有ベクトル	peig/ eigenvector	MUSIC法の重み付バージョン	同上

手法	関数/ オブジェクト	処理内容	特徴
Spectrogram	spectrogram	短時間FFTを繰り返し実行し時間軸方向に並べる	周波数スペクトルを時系列に並べてダイナミックな周波数特性を把握可能 声紋分析などに利用

スペクトル解析関数の使い方

スペクトルを
表示

$function(xn, window, NFFT)$

解析手法を指定
(例: *pwelch*, *pmcov*)

パラメータ: 窓関数、FFT点数
など(手法によって異なる)

データを
出力

$[P_{xx}, F] = function(xn, window, NFFT)$

パワース
ペクトル

周波数軸
ベクトル

スペクトル解析オブジェクトの使い方

オブジェクト作成とPSD表示

解析手法を指定
(例: *welch*, *cov*)

```
Hs = spectrum.function( param );  
psd(Hs,x,'NFFT',512);
```

パラメータ: 窓関数、信号など
(手法によって異なる)

オブジェクト作成とPSDオブジェクト作成

```
Hs = spectrum.function( param );  
Hp = psd(Hs,x,'NFFT',512);
```

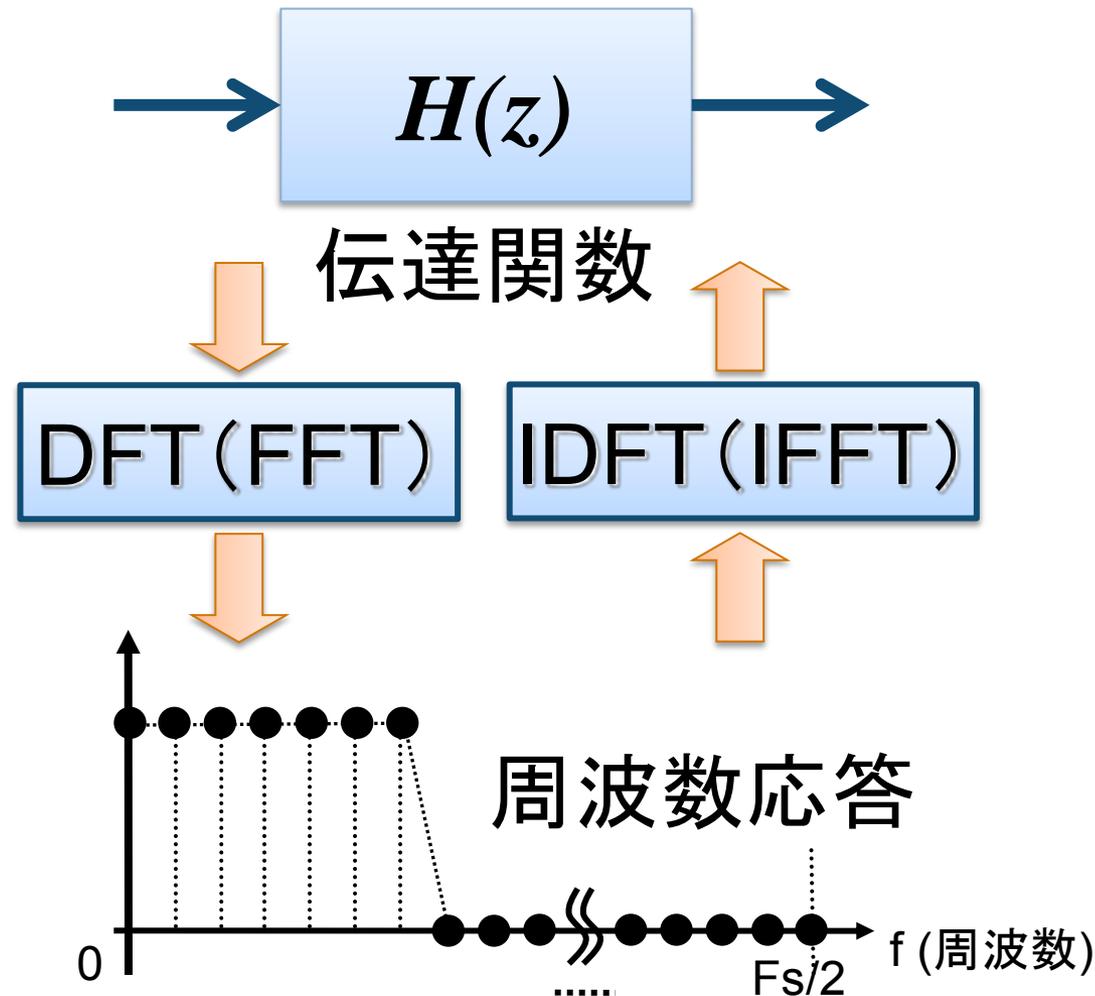
モデルや時系列データからの周波数応答/ 伝達関数推定

- どのようにして推定する？
 - 線形/非線形モデル
 - モデル、時系列データ

手法	対象	使用用途
linmod	Simulink線形モデル	線形なSimulinkモデルの伝達関数
tfestimate	線形モデルに対する時系列入出力データ	時系列入出力データから伝達関数
SIN波入力 (プログラム)	非線形モデル (固定小数点含む)	非線形なモデルに対して適用可能 SIN波入力⇒入出力FFT⇒応答推定 Simシリーズなどの物理モデルに有効
System Identification Toolbox(製品)	線形/非線形モデルの時系列データ	実測データなどからシステム同定を行い、 数式モデルを推定

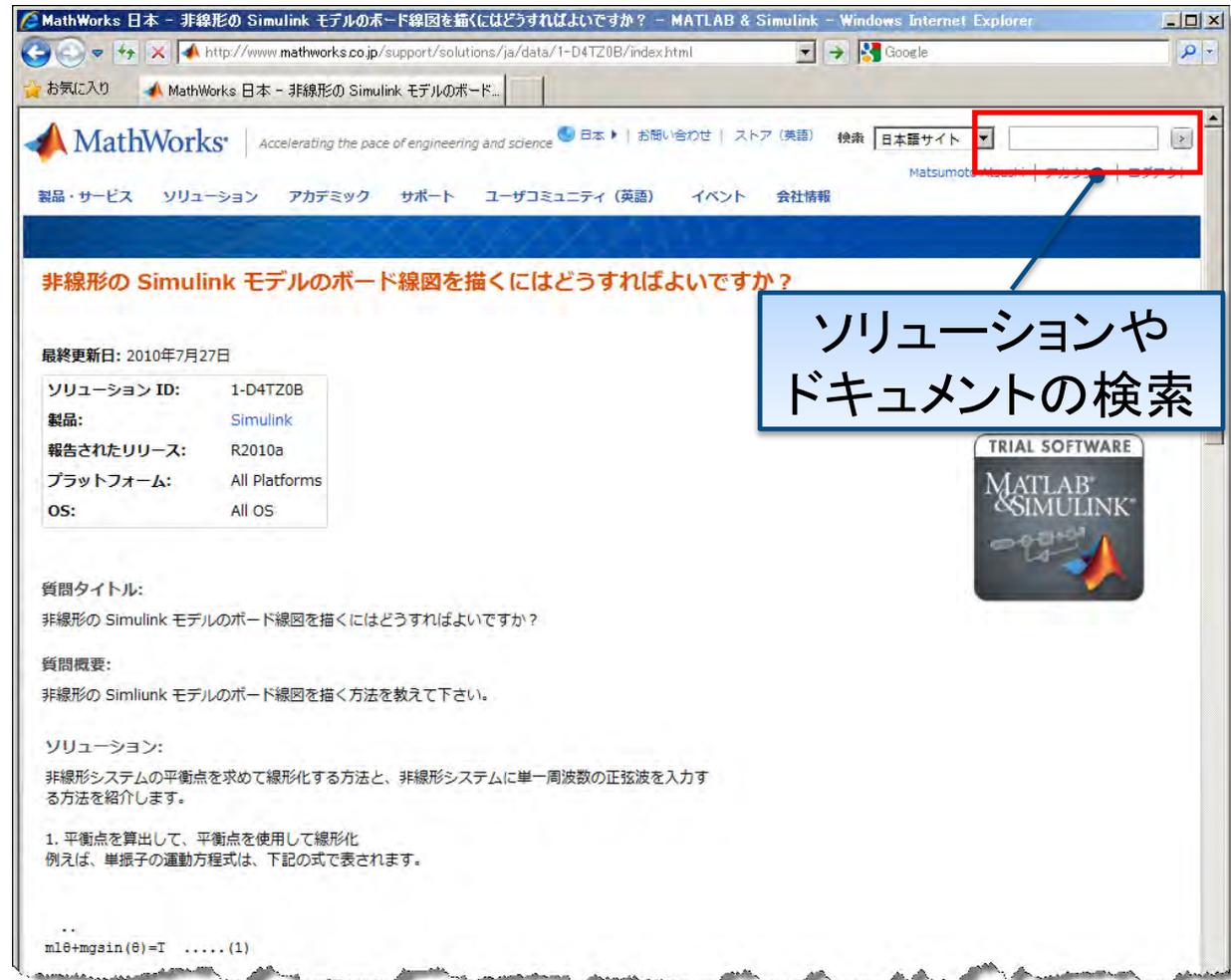
Demo: *fresp_sl.m*

周波数応答と伝達関数の関係



MathWorks Webサイトのソリューションページ <http://www.mathworks.co.jp/support>

Webサイトで有用な
 ソリューションを提供
 ID: 1-D4TZ0B
 「非線形のSimulink
 モデルのボード線図
 を描くにはどうすれ
 ばよいですか？」



MathWorks 日本 - 非線形の Simulink モデルのボード線図を描くにはどうすればよいですか? - MATLAB & Simulink - Windows Internet Explorer

http://www.mathworks.co.jp/support/solutions/ja/data/1-D4TZ0B/index.html

MathWorks Accelerating the pace of engineering and science 日本 | お問い合わせ | ストア (英語) 検索 日本語サイト

製品・サービス ソリューション アカデミック サポート ユーザコミュニティ (英語) イベント 会社情報

非線形の Simulink モデルのボード線図を描くにはどうすればよいですか?

最終更新日: 2010年7月27日

ソリューション ID:	1-D4TZ0B
製品:	Simulink
報告されたリリース:	R2010a
プラットフォーム:	All Platforms
OS:	All OS

質問タイトル:
 非線形の Simulink モデルのボード線図を描くにはどうすればよいですか?

質問概要:
 非線形の Simulink モデルのボード線図を描く方法を教えてください。

ソリューション:
 非線形システムの平衡点を求めて線形化する方法と、非線形システムに単一周波数の正弦波を入力する方法を紹介します。

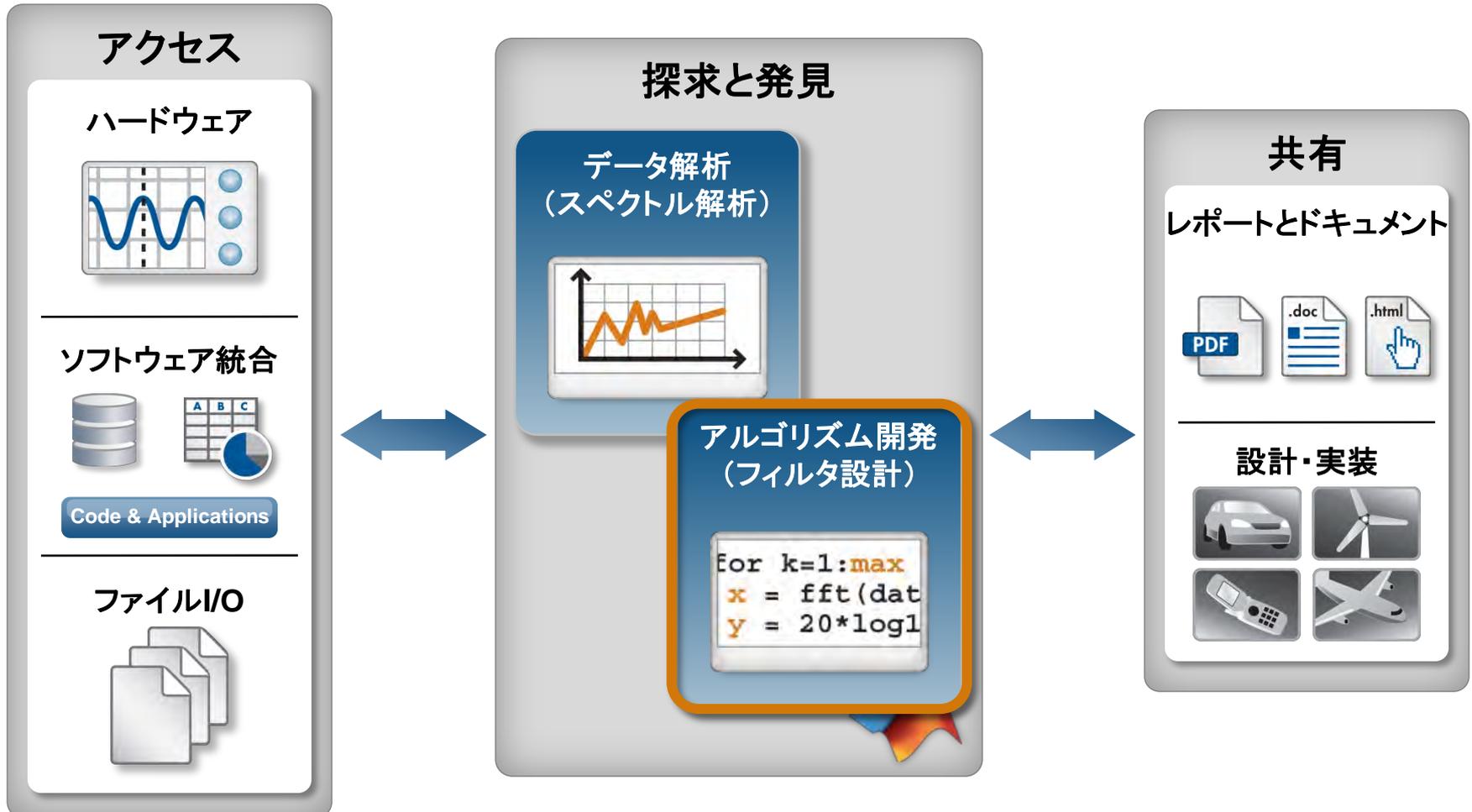
1. 平衡点を算出して、平衡点を使用して線形化
 例えば、単振子の運動方程式は、下記の式で表されます。

..... (1)

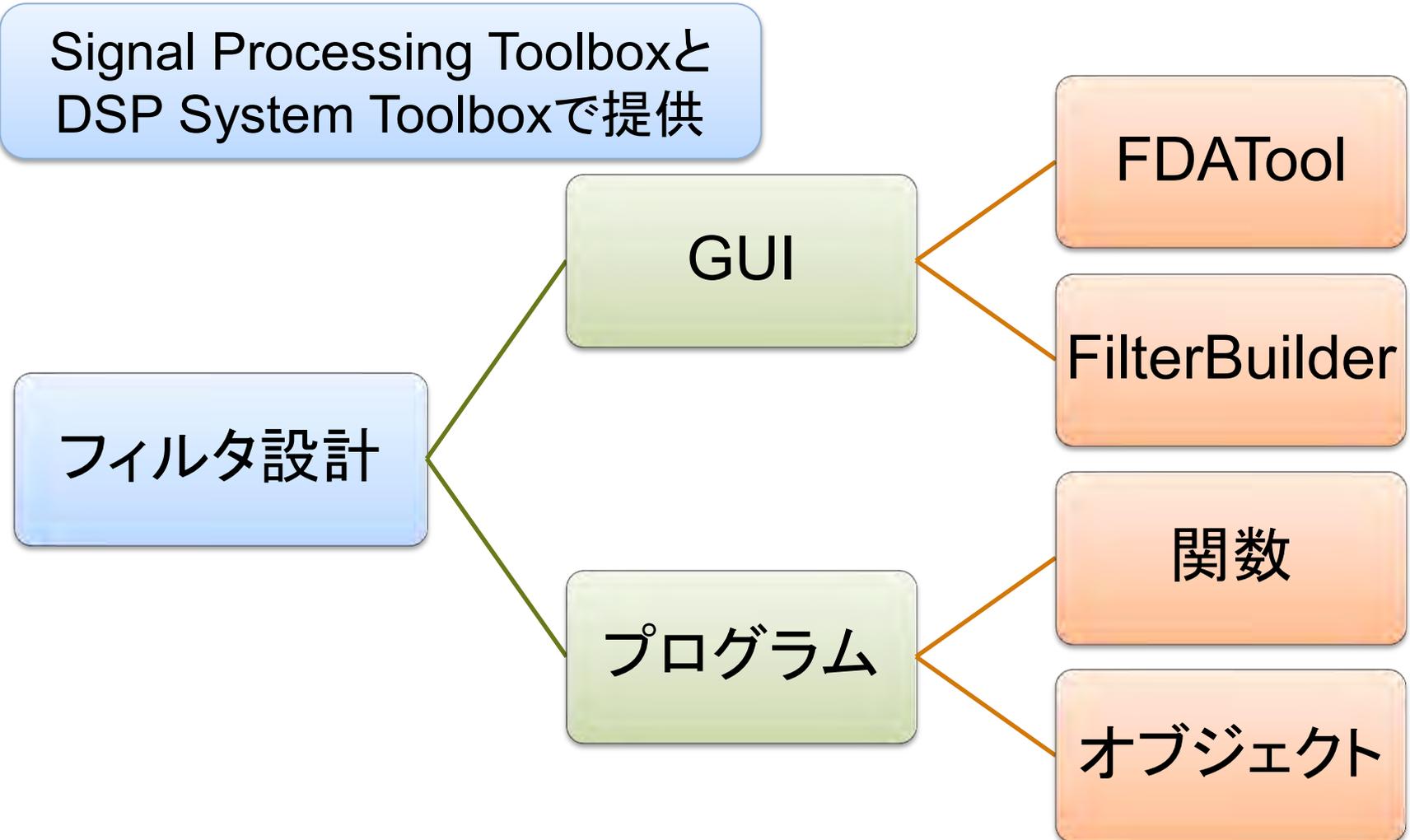
ソリューションやドキュメントの検索

TRIAL SOFTWARE
 MATLAB & SIMULINK

アジェンダ: エンジニアリング・ワークフロー



フィルタ設計機能



設計手法ベースのフィルタ設計解析GUI FDATool (Filter Design & Analysis Tool)

>> *fdatool*

解析ボタン

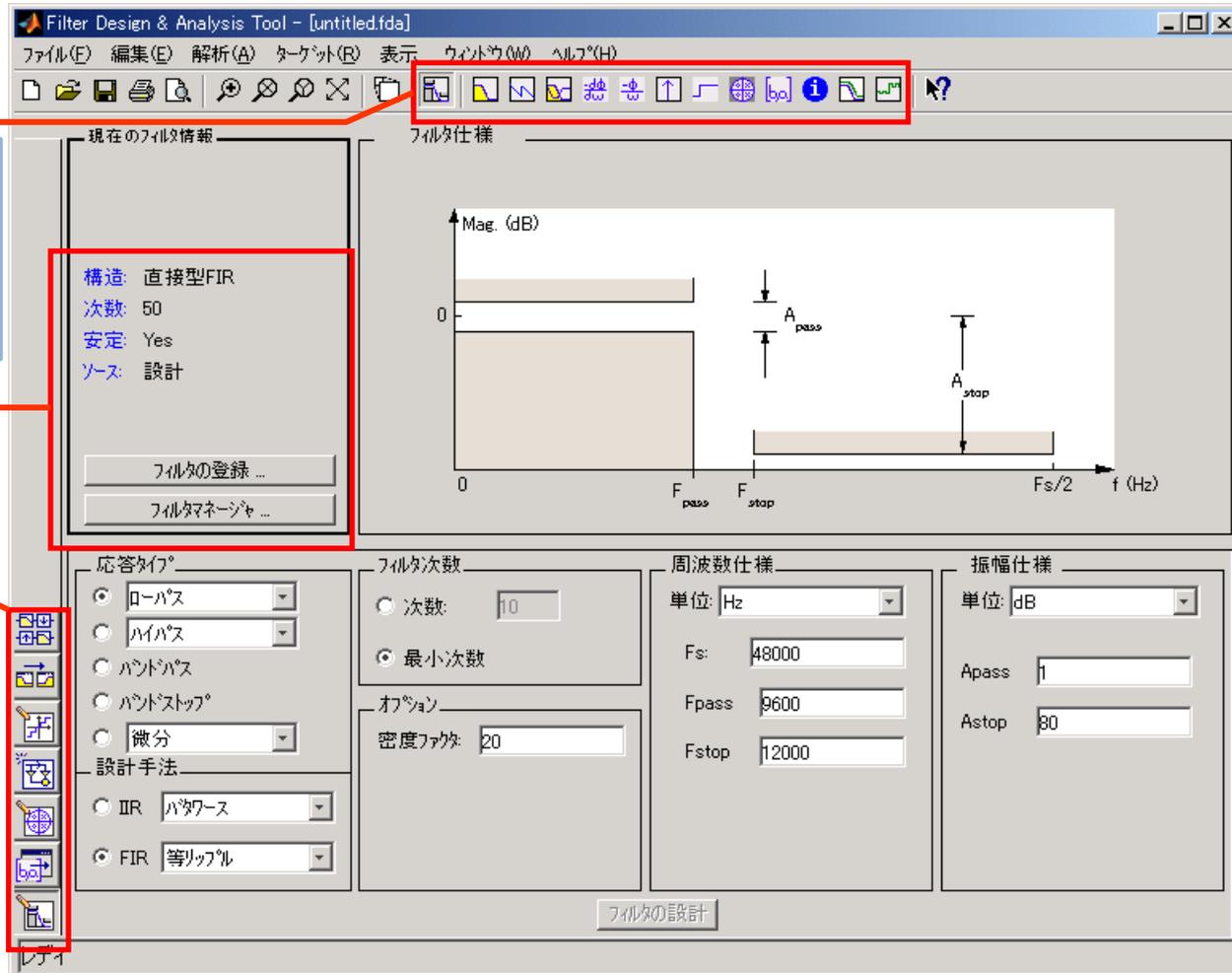
周波数/位相/群遅延/位相遅延/
インパルス/ステップ/零極配置/
フィルタ係数/フィルタ情報/
振幅推定/丸め誤差

フィルタの管理

複数のフィルタを保存、接続

設計ボタン

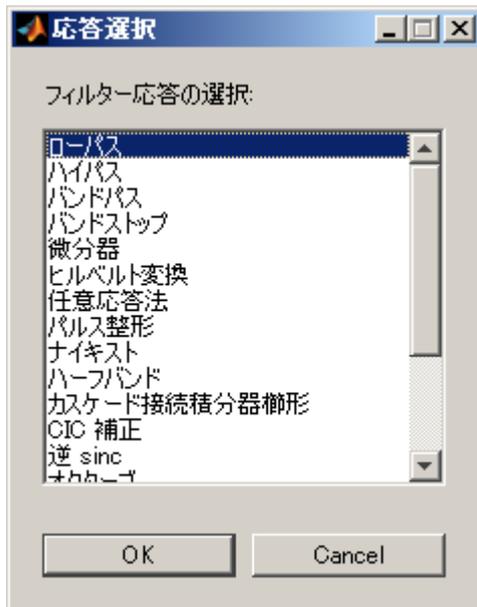
マルチレートフィルタの作成
フィルタの変換
量子化パラメータ設定
モデルの実現
極零エディタ
ワークスペースからインポート
フィルタ設計



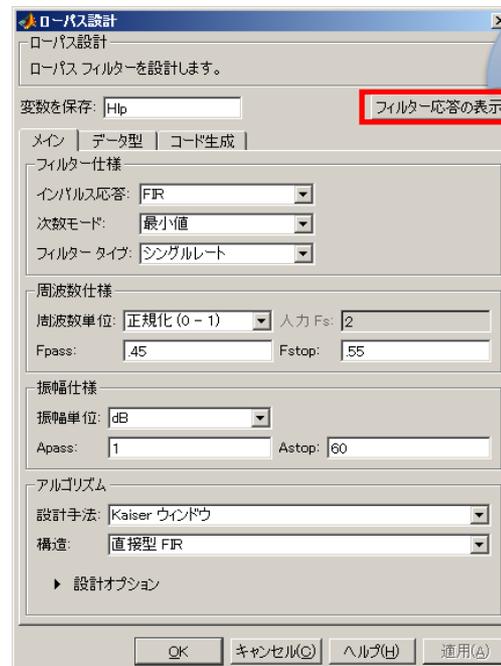
仕様ベースのフィルタ設計ダイアログ FilterBuilder

>> *filterbuilder*

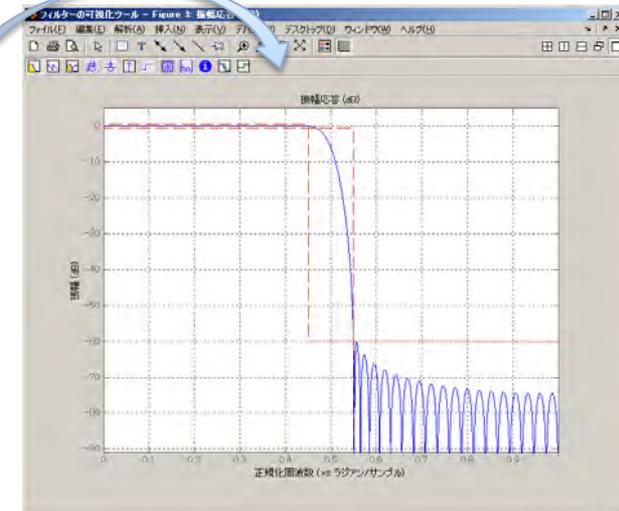
1. 応答を選択



2. 特性と設計手法の設定



3. 応答の確認



設計手法ベースの関数と仕様ベースのオブジェクト

$$[\text{num}, \text{den}] = \text{method}(\text{N}, \text{f}, \text{a})$$

分子
係数

分母
係数

設計手法を指定
(例: *butter*, *fir1*)

フィルタパラメータ
(設計手法によって異なる)

$$\text{Hd} = \text{fdesign.response}(\text{spec})$$

オブジェクト

仕様を指定
(例: *lowpass*, *bandstop*)

周波数やゲインなどの仕様
(設計するフィルタの
応答によって異なる)

FIRフィルタとIIRフィルタの特長

応答	安定性	位相特性	遅延	係数感度	処理負荷	設計方法	係数変更
FIR	必ず安定	直線位相性を容易に実現出来る	大	低いため低ビット幅で実現可能	高い (高い次数が必要)	所望の応答をIFFT	処理中でも可能なため適応フィルタにも適している
IIR	安定性の制約を伴う	非線形 (近似的には直線性を実現可能)	小	高いため高ビット幅で実現する 2次型に等価変換して係数感度を落としてから実現する	低い (高い次数は不要)	アナログプロトタイプをプリワーピング(周波数シフト)して双一次(s-z)変換	処理中はフィードバックループに注意が必要で適応フィルタには向いていない

IIRフィルタ設計関数

Signal Processing Toolbox提供

設計関数	名称	設計パラメータ	Roll Off (肩特性)	リップル	群遅延 / 位相歪	特徴
<i>butter</i>	バターワース	N, Fc	なだらか	なし	小 / 小	全体的にフラット
<i>maxflat</i>	マックスフラット	Nn, Nd, Fc	同上	同上	同上	分母分子で次数の異なるLPF
<i>cheby1</i>	チェビシェフ I 型	N, Fc, Rp	II型より急峻	通過帯域	中 / 中	通過帯域リップル
<i>cheby2</i>	チェビシェフ II 型	N, Fc, Rs	I型に劣る	遮断帯域	中 / 中	通過帯域フラット 遮断帯域リップル
<i>elliptic</i>	楕円 (連立チェビシェフ)	N, Fc, Rp, Rs	急峻	通過/ 遮断帯域	大 / 大	最小次数で仕様を満足
<i>yulewalk</i>	再帰型	N, f, m	最もなだらか	通過/ 遮断帯域	中 / 大	任意振幅応答から設計

IIRフィルタ設計関数

DSP System Toolbox提供

設計関数	設計手法	設計パラメータ	使用用途
<i>iircomb</i>	コム、ノッチ、ピーク	N, BW, ab, type	周期的な周波数を大きく落とすまたは増幅
<i>iirgrpdelay</i>	任意群遅延	N, f, edges, a, w	位相補償
<i>iirlpnorm</i>	最小Pノルム	N, d, f, edges, a, w	周波数、エッジ周波数、振幅、重みベクトルからIIRフィルタ設計
<i>iirlpnormc</i>	制約付最小Pノルム	N, d, f, edges, a, w	極の半径に制約を与えられる
<i>iirnotch</i>	2次ノッチ	w0, bw, ab	ノッチ周波数が単一のノッチフィルタ
<i>iirpeak</i>	2次ピーク	w0, bw, ab	ピーク周波数が単一のピークフィルタ

FIRフィルタ設計関数

Signal Processing Toolbox提供

設計関数	設計手法	設計パラメータ	使用用途
<i>fir1</i> <i>kaiserord</i>	ウィンドウ	N,Wn N,Wn,'ftype' N,Wn,,win	通過帯域リップルが大きい
<i>fir2</i>	周波数 サンプリング	N,f,m / N,f,m,win N,f,m,npt N,f,m,npt,win N,f,m,npt,lap N,f,m,npt,lap,win	任意の周波数と振幅ベクトルを与えて設計
<i>firls</i>	最小二乗 線形位相	N,f,a / N,f,a,w N,f,a,'ftype' N,f,a,w,'ftype'	周波数と振幅ベクトルと重みを与えて設計
<i>fircls</i>	制約付最小 二乗線形位相	N,f,a,up,lo N,f,amp,up,lo,'flag'	振幅の上限、下限の制約を与えて設計

FIRフィルタ設計関数

Signal Processing Toolbox提供

設計関数	設計手法	設計パラメータ	使用用途
<i>fircls1</i>	制約付最小二乗線形位相	N,w0,dp,ds N,w0,dp,ds,'high' N,w0,dp,ds,wt N,w0,dp,ds,wt,'high' N,w0,dp,ds,wp,ws,k N,w0,dp,ds,wp,ws,k,'high' N,w0,dp,ds,...,'flag'	通過/遮断帯域リップルの制約を与えて設計
<i>firpm</i> <i>firpmord</i>	Parks-McClellan最適FIRフィルタ	N,f,a N,f,a,w N,f,a,'ftype' N,f,a,w,'ftype' N,f,a,...,{lgrid} N,f,@fresp,w N,f,@fresp,w,'ftype'	希望する周波数応答と実際の周波数応答が最適になるようにフィルターを設計
<i>intfilt</i>	インターポレーション	l,p,alpha l,n,'Lagrange'	理想的な帯域制限内挿を行う線形位相FIRフィルターを設計

FIRフィルタ設計関数

Signal Processing Toolbox提供

設計関数	設計手法	設計パラメータ	使用用途
<i>cfirpm</i>	複素かつ非線形位相の等リプル	N,f,@fresp N,f,@fresp,w N,f,a / N,f,a,w ..., 'sym' / ..., 'skip_stage2' ..., 'debug' / ..., {lgrid}	複素となる可能性のある任意の周波数領域の制約を指定して設計
<i>sgolay</i>	Savitzky-Golay	k,f k,f,w	多項式kとフレームサイズfを指定して平滑化フィルタを設計
<i>firrcos</i>	コサインロールオフ	N,Fc,df / N,Fc,df,Fs N,Fc,df,Fs,'bandwidth' N,Fc,df,Fs,'type' ..., 'type', delay ..., 'type', delay, window N,Fc,r,Fs,'rolloff' ..., 'rolloff', 'type'	コサインロールオフローパスフィルタを設計
<i>gaussfir</i>	ガウス FIR パルス成形	bt / bt,n / bt, n, o	ローパス FIR ガウス パルス成形フィルタを設計

FIRフィルタ設計関数

DSP System Toolbox提供

設計関数	設計手法	設計パラメータ	その他
<i>firband</i>	制約帯域等リップル	N,f,a,w,c N,f,a,s ..., '1' ..., 'minphase' ..., 'check' ..., {lgrid}	対称/非対称型、最小/最大位相、最小次数、制約リップルフィルタなどをミニマックス法で設計
<i>firceqrip</i>	制約付等リップル	N, f0, Dev ..., 'slope', r ..., 'passedge' ..., 'stopedge' ..., 'high' ..., 'min' ..., 'invsinc', c	<i>firpm</i> と同じ等リップルフィルタを設計するが、フィルタの特性を設定できる
<i>fireqint</i>	等リップルインターポレーション	N,l,alpha N,l,alpha,w 'minorder', l, alpha, r {'minorder', 'initord'}, l, alpha, r	等リップルインターポレーションを設計

FIRフィルタ設計関数

DSP System Toolbox提供

設計関数	設計手法	設計パラメータ	その他
<i>firgr</i>	Parks-McClellan	N,f,a,w / N,f,a,'hilbert' m,f,a,r / {m,ni},f,a,r N,f,a,w,e / N,f,a,s N,f,a,s,w,e / ..., '1' ..., 'minphase' ..., 'check' / ..., {lgrid} N,f,fresp,w N,f,{fresp,p1,p2,...},w	対称/非対称型、最小/最大位相、最小次数、制約リップルフィルタなどをミニマックス法で設計
<i>firhalfband</i>	ハーフバンド	N,fp / N,win / N,dev,'dev' 'minorder',fp,dev 'minorder',fp,dev,'kaiser' ..., 'high' / ..., 'minphase'	等リップルローパスハーフバンドフィルタを設計
<i>firpnorm</i>	最小Pノルム	N,f,edges,a / N,f,edges,a,w N,f,edges,a,w,p N,f,edges,a,w,p,dens N,f,edges,a,w,p,dens,initnum ..., 'minphase'	準ニュートン法により最小Pノルムフィルタを最良近似で設計

FIRフィルタ設計関数

DSP System Toolbox提供

設計関数	設計手法	設計パラメータ	その他
<i>firminphase</i>	最小位相	b b,nz	設計した線形位相FIRフィルタ係数を引数に最小位相フィルタに変換
<i>firnyquist</i>	ローパス ナイキスト	N,l,r 'minorder',l,r,dev N,l,r,decay N,l,r,'nonnegative' N,l,r,'minphase'	lバンド、ロールオフファクタrで等リップルのナイキストフィルタを設計
<i>ifir</i>	インターポレーション	d	フィルタオブジェクトdをインターポレーションフィルタに変換

設計手法ベースの関数と仕様ベースのオブジェクト

$$[\text{num}, \text{den}] = \text{method}(\text{N}, \text{f}, \text{a})$$

分子
係数

分母
係数

設計手法を指定
(例: *butter*, *fir1*)

フィルタパラメータ
(設計手法によって異なる)

$$\text{Hd} = \text{fdesign.response}(\text{spec})$$

オブジェクト

仕様を指定
(例: *lowpass*, *bandstop*)

周波数やゲインなどの仕様
(設計するフィルタの
応答によって異なる)

フィルタ仕様オブジェクト一覧

Signal Processing Toolbox/DSP System Toolbox

フィルタタイプ	設計手法	SPT選択可能な設計仕様	DST選択可能な設計仕様
<i>arbmag</i>	任意振幅応答	N,F,A N,B,F,A	F,A,R / Nb,Na,F,A / N,B,F,A,C / B,F,A,R / Nb,Na,B,F,A
<i>lowpass</i>	ローパス	Fp,Fst,Ap,Ast / N,F3db / Nb,Na,F3dB / N,Fc / N,Fc,Ap,Ast / N,Fp,Ap / N,Fp,Ap,Ast / N,Fp,Fst / N,Fst,Ast	N,F3db,Ap / N,F3db,Ap,Ast / N,F3db,Ast / N,F3db,Fst / N,Fp,Fst,Ap / N,Fp,F3db / N,Fp,Fst,Ast / N,Fst,Ap,Ast / Nb,Na,Fp,Fst
<i>highpass</i>	ハイパス	Fst,Fp,Ast,Ap / N,F3db / N,Fc N,Fc,Ast,Ap / N,Fp,Ap / N,Fp,Ast,Ap / N,Fst,Ast / N,Fst,Fp	N,F3db,Ap / N,F3db,Ast N,F3db,Ast,Ap / N,F3db,Fp N,Fst,F3db / N,Fst,Fp,Ap N,Fst,Fp,Ast / Nb,Na,Fst,Fp
<i>bandpass</i>	バンドパス	Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2 / N,F3dB1,F3dB2 / N,Fc1,Fc2 N,Fc1,Fc2,Ast1,Ap,Ast2 N,Fp1,Fp2,Ap N,Fp1,Fp2,Ast1,Ap,Ast2 N,Fst1,Fp1,Fp2,Fst2 N,Fst1,Fp1,Fp2,Fst2,Ap	N,F3dB1,F3dB2,Ap N,F3dB1,F3dB2,Ast N,F3dB1,F3dB2,Ast1,Ap,Ast2 N,F3dB1,F3dB2,BWp N,F3dB1,F3dB2,BWst N,Fst1,Fp1,Fp2,Fst2,C N,Fst1,Fp1,Fp2,Fst2,Ap Nb,Na,Fst1,Fp1,Fp2,Fst2

フィルタ仕様オブジェクト一覧

Signal Processing Toolbox/DSP System Toolbox

フィルタタイプ	設計手法	SPT選択可能な設計仕様	DST選択可能な設計仕様
<i>bandstop</i>	バンドストップ	Fp1,Fst1,Fst2,Fp2,Ap1,Ast,Ap2 N,F3dB1,F3dB2 N,Fc1,Fc2 N,Fc1,Fc2,Ap1,Ast,Ap2 N,Fp1,Fp2,Ap N,Fp1,Fp2,Ap,Ast N,Fp1,Fst1,Fst2,Fp2 N,Fst1,Fst2,Ast	N,F3dB1,F3dB2,Ap N,F3dB1,F3dB2,Ap,Ast N,F3dB1,F3dB2,Ast N,F3dB1,F3dB2,BWp N,F3dB1,F3dB2,BWst N,Fp1,Fst1,Fst2,Fp2,C N,Fp1,Fst1,Fst2,Fp2,Ap Nb,Na,Fp1,Fst1,Fst2,Fp2
<i>hilbert</i>	ヒルベルト	N,TW	TW,Ap
<i>differentiator</i>	微分	N / N,Fp,Fst / Ap / Fp,Fst,Ap,Ast	N,Fp,Fst,Ap / N,Fp,Fst,Ast / Ap / Fp,Fst,Ap,Ast
<i>pulseshaping</i>	パルス整形	Ast,Beta / Nsym,Beta / N,Beta	

フィルタ仕様オブジェクト一覧

DSP System Toolbox提供

フィルタ仕様	設計手法	設計仕様	その他のパラメータ
<i>arbgrpdelay</i>	任意グループ遅延応答	N,F,Gd / N,B,F,Gd	
<i>arbmagnphase</i>	任意振幅位相応答	N,F,H / N,B,F,H / Nb,Na,F,H	
<i>decimator</i>	デシメーション	Responseによる	Response: Arbitrary Magnitude, Arbitrary Magnitude and Phase, Bandpass, Bandstop, CIC, CIC Compensator, Diffenciator, Gaussian, Halfband, Highpass, Hilbert, Inverse-sinc Lowpass, Inverse-sinc Highpass, Lowpass, Nyquist, Raised Cosine, Square Root Raised Cosine
<i>interpolator</i>	インターポレーション	Responseによる	同上

フィルタ仕様オブジェクト一覧

DSP System Toolbox提供

フィルタ仕様	設計手法	設計仕様	その他のパラメータ
<i>audioweighting</i>	オーディオ重み	WT,Class WT	WeightingType: A, C, C-message, ITU-R 468-4, ITU-T 0.41
<i>ciccomp</i>	CIC補償	fp,fst,ap,astn,fc, ap,ast' n,fp,ap,ast' n,fp,fst' n,fst,ap,ast'	
<i>comb</i>	コム(楕形)	N,Q N,BW L,BW,GWB,Nsh	notch, peak
<i>fracdelay</i>	分数遅延	Delta, N	

フィルタ仕様オブジェクト一覧

DSP System Toolbox提供

フィルタ仕様	設計手法	設計仕様	その他のパラメータ
<i>halfband</i>	ハーフバンド	TW,Ast N,TW N N,Ast	Type: Lowpass, Highpass
<i>isinclp</i>	逆sincローパス	Fp,Fst,Ap,Ast N,Fc,Ap,Ast N,Fp,Ap,Ast N,Fp,Fst N,Fst,Ap,Ast	
<i>isinchp</i>	逆sincハイパス	Fst,Fp,Ast,Ap N,Fc,Ast,Ap N,Fp,Ast,Ap N,Fst,Fp N,Fst,Ast,Ap	
<i>nyquist</i>	ナイキスト	TW,Ast N,TW N N,Ast	

フィルタ仕様オブジェクト一覧

DSP System Toolbox提供

フィルタ仕様	設計手法	設計仕様	その他のパラメータ
<i>notch</i>	ノッチ	N, F0, Q / N, F0, Q, Ap / N, F0, Q, Ast / N, F0, Q, Ap, Ast N, F0, BW / N, F0, BW, Ap N, F0, BW, Ast / N, F0, BW, Ap, Ast	
<i>peak</i>	ピーク	同上	
<i>octave</i>	オクターブ	N, F0	Mask Class 0, Class 1, Class 2
<i>parameq</i>	パラメトリック イコライザ	F0, BW, BWp, Gref, G0, GBW, Gp F0, BW, BWst, Gref, G0, GBW, Gst F0, BW, BWp, Gref, G0, GBW, Gp, Gst N, F0, BW, Gref, G0, GBW N, F0, BW, Gref, G0, GBW, Gp N, F0, Fc, Qa, G0 N, F0, Fc, S, G0 N, F0, BW, Gref, G0, GBW, Gst N, F0, BW, Gref, G0, GBW, Gp, Gst N, Flow, Fhigh, Gref, G0, GBW N, Flow, Fhigh, Gref, G0, GBW, Gp N, Flow, Fhigh, Gref, G0, GBW, Gst N, Flow, Fhigh, Gref, G0, GBW, Gp, Gst	

フィルタ仕様オブジェクト一覧

DSP System Toolbox提供

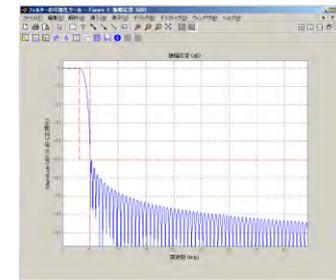
フィルタ仕様	設計手法	設計仕様	その他のパラメータ
<i>polysrc</i>	多項式 サンプルレート コンバータ	Np	Interpolation/Decimation Factor
<i>rsrc</i>	有理ファクタ サンプルレート コンバータ	Responseによる	Response: Arbitrary Magnitude, Arbitrary Magnitude and Phase, Bandpass, Bandstop, CIC, CIC Compensator, Diffenciator, Gaussian, Halfband, Highpass, Hilbert, Inverse-sinc Lowpass, Inverse-sinc Highpass, Lowpass, Nyquist, Raised Cosine, Square Root Raised Cosine

フィルタ設計関数使用例

- サンプルレートコンバータ (8k \Rightarrow 44.1kHz) において
大幅な計算コストの削減

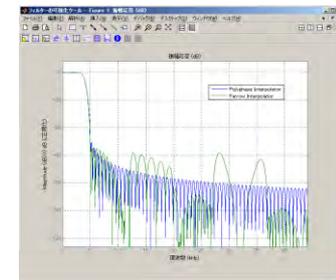
FIRポリフェーズ x 1

- 乗算器: 5178, 加算器: 4738
- 入力サンプルあたり演算器数
 - 乗算器: 64.73, 加算器: 59.26
 - Clock 3.528MHz



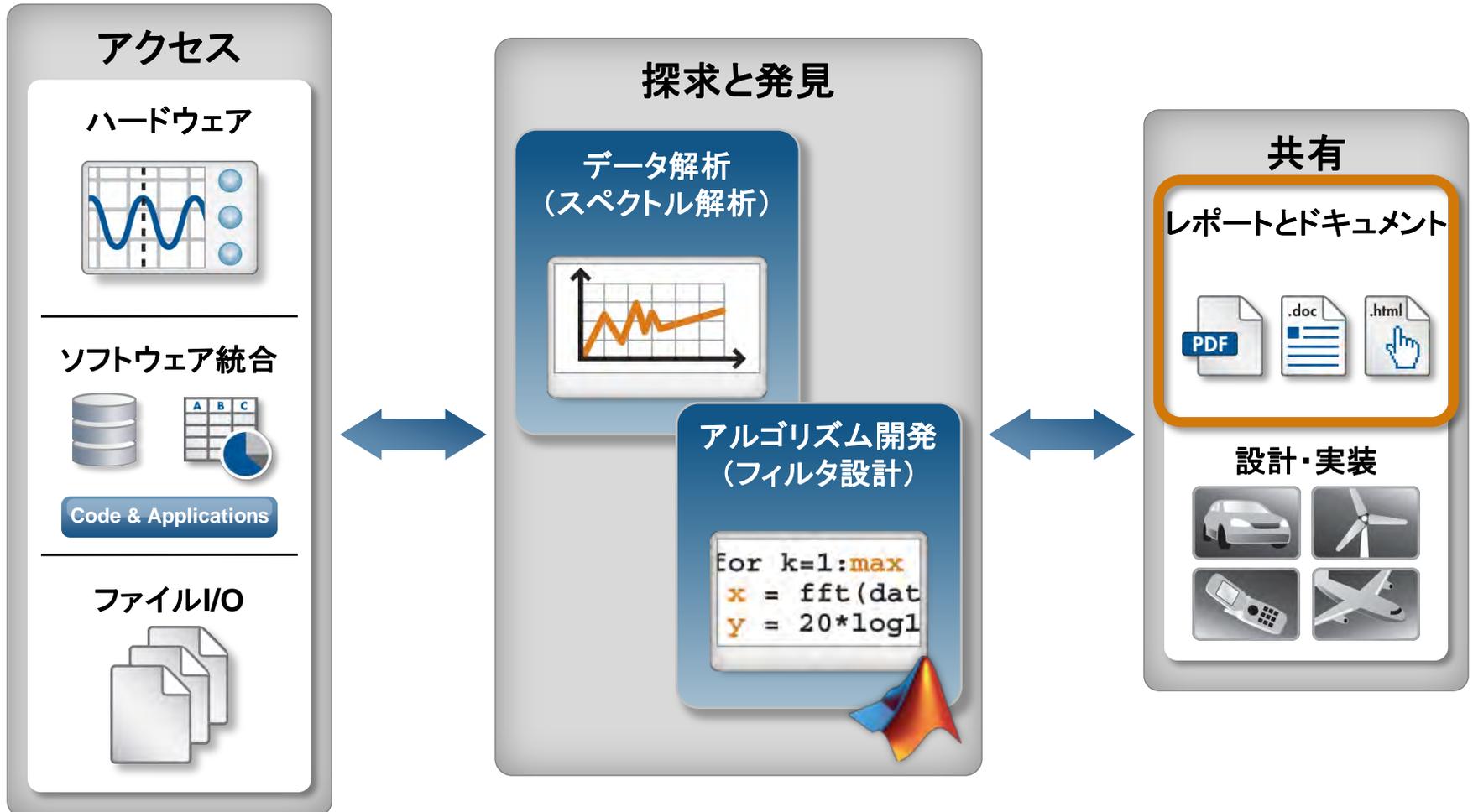
FIRポリフェーズ x 2 + Farrow構造フィルタ

- 乗算器: 32, 加算器: 29
- 入力サンプルあたり演算器数
 - 乗算器: 94.15, 加算器: 85.64
 - Clock 14.112MHz



Demo: `filter_farrow.m`

アジェンダ: エンジニアリング・ワークフロー



レポート生成機能: MATLABエディタ

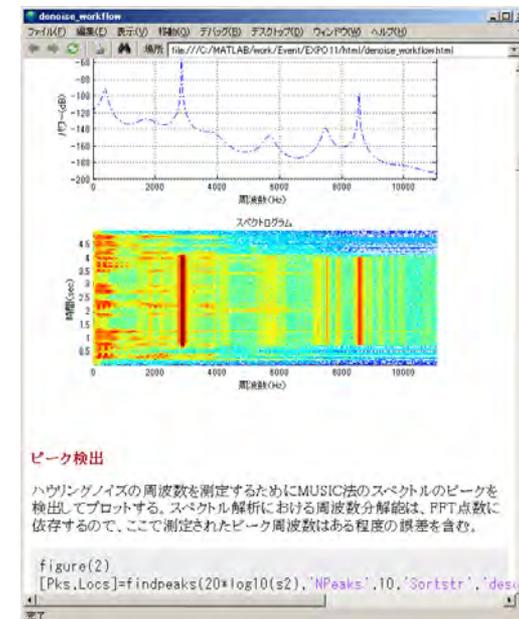
- MATLABエディタから実行結果をレポート生成
 - html, xml, latex, doc, ppt
 - 目次、コメント文、プログラム、Figure、Latexコマンド
 - オプション追加でカスタマイズ可 (MATLAB/Simulink Report Generator)

ボタンを押すと
レポート生成

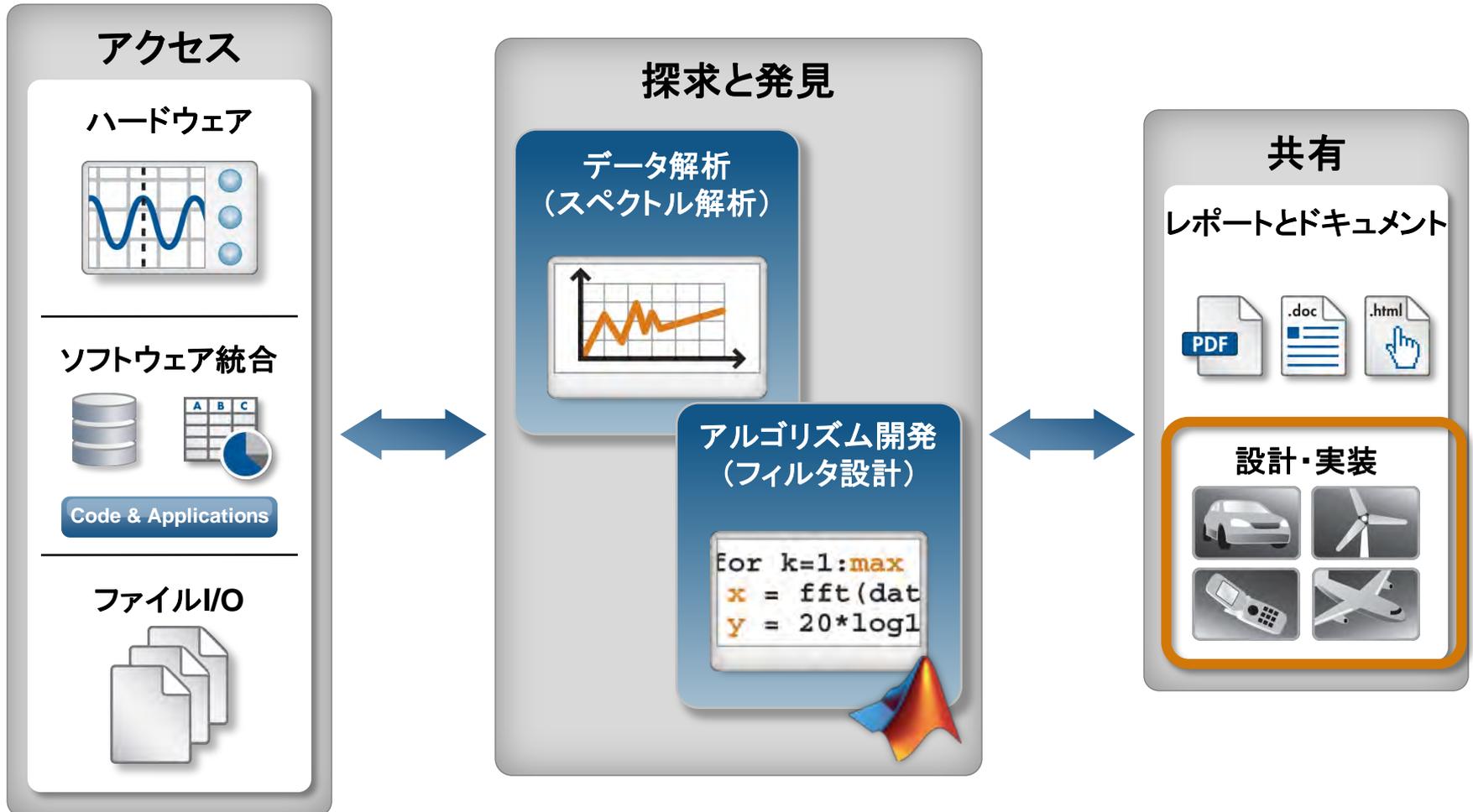
```

19 %% ノイズ信号の周波数応答の表示
20 % いくつかのスペクトル解析手法を用いて周波数スペク
21 % 修正共分散法でのピークとスペクトログラムで一定周波
22 % ハウリングノイズだと予想できる。
23 - figure(2)
24 - subplot(3,1,1)
25 - [s1, f1] = periodogram(noised_sig, hamming(1024), Fs);
26 - plot(f1, 20*log10(s1), 'b-'), grid
27 - xlim([0 f1(end)])
28 - title('Periodogram'), xlabel('周波数 (Hz)'), ylabel('幅 (dB)')
29
30 - subplot(3,1,2)
31 - [s2, f1] = pmcov(noised_sig, 16, 512, Fs);
32 - plot(f1, 20*log10(s2), 'b-'), grid
33 - xlim([0 f1(end)])
34 - title('修正共分散法'), xlabel('周波数 (Hz)'), ylabel('幅 (dB)')
35
36 - subplot(3,1,3)
37 - spectrogram(noised_sig, hamming(512), 256, 512, Fs);

```



アジェンダ: エンジニアリング・ワークフロー

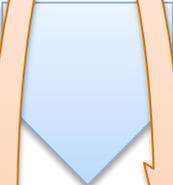


フィルタの固定小数点化



```
Hd = design(Fd, 'FilterStructure', 'df1sos');
```

構造の設定



```
Hd.Arithmetic = 'fixed';
```

オブジェクトの固定小数点化



```
Hd.InputWordLength = 24;
```

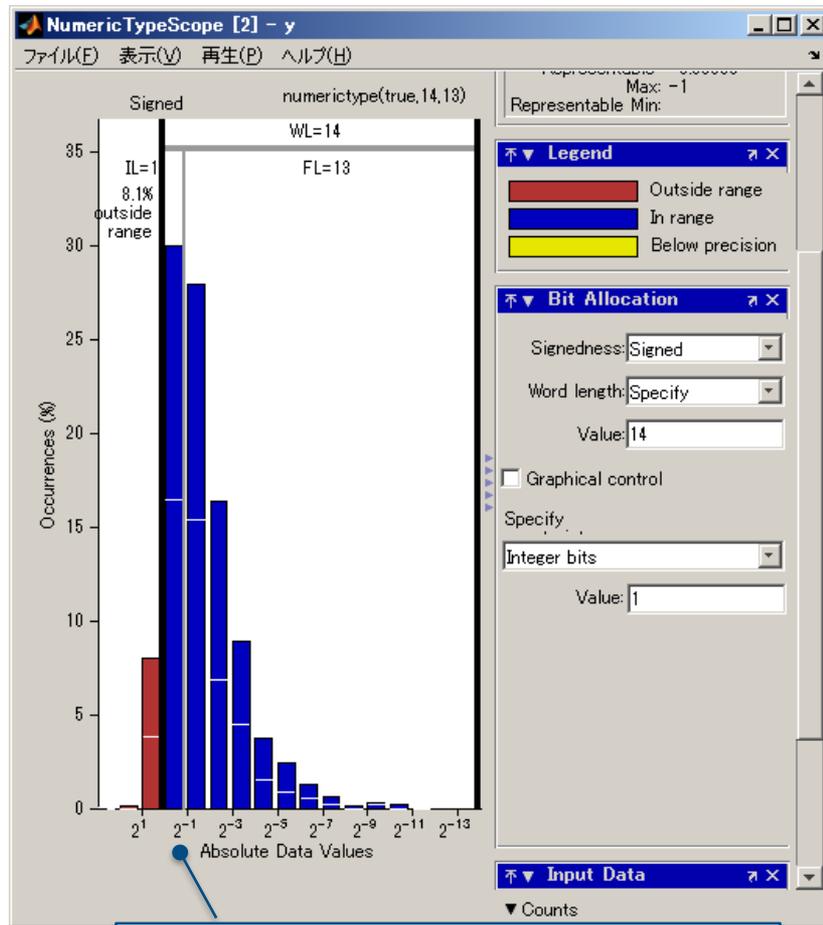
ビット幅や小数点の設定(各項目)



```
Hd2 = autoscale(Hd, x)
```

信号入力してオートスケール

ダイナミックレンジ解析機能: NumericTypeScope



シミュレーションにおける
データのダイナミックレンジ

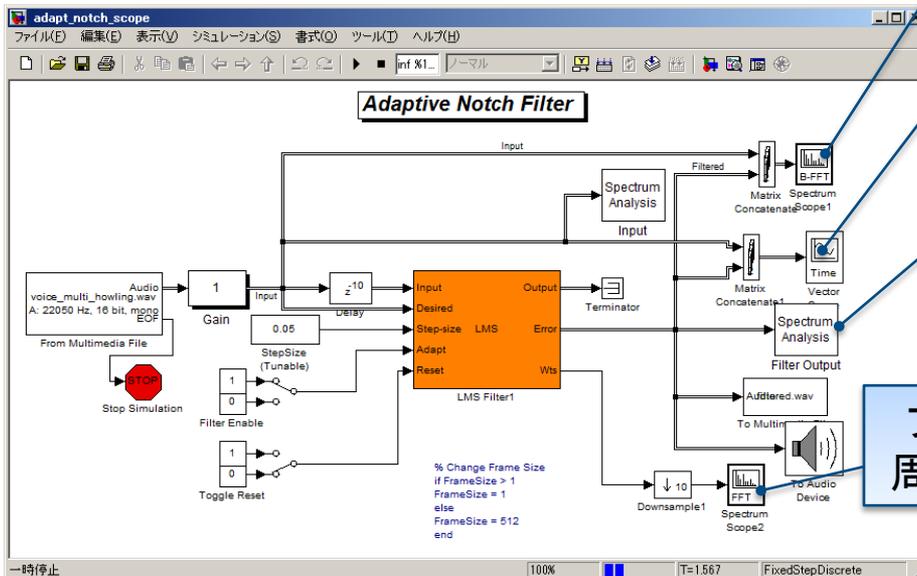
- NumericTypeScopeオブジェクトにより、固定小数点演算時の使用ビット分布を可視化
- データ型を決めるためのダイナミックレンジ解析に有効
- MATLABプログラムの変数測定

```
>> hs = NumericTypeScope
```
- Simulink信号線測定

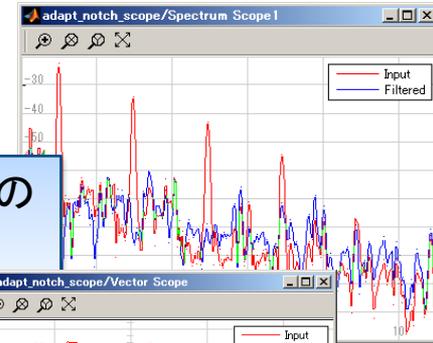
```
>> nts % GUI起動
```

ブロック線図シミュレータ Simulink

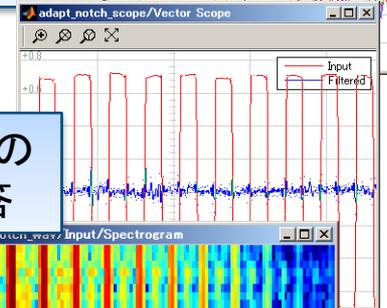
- 動作をグラフィカルに表現
- タイムドリブン・シミュレーション
- From/To Audio Deviceブロックでリアルタイムでサウンド入出力 (DSP System Toolbox機能)



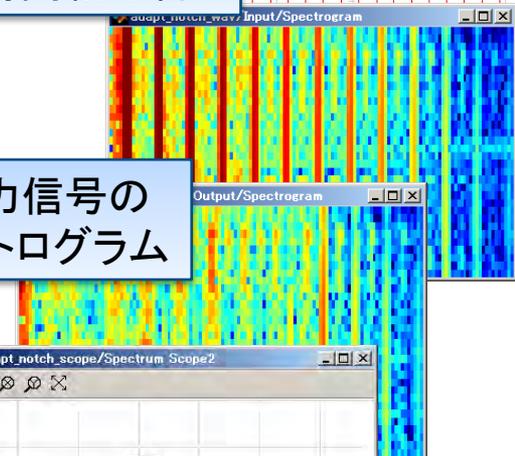
入出力信号の
スペクトル



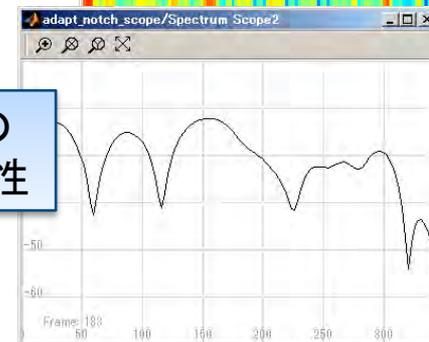
入出力信号の
時間軸応答



入出力信号の
スペクトログラム



フィルタの
周波数特性



固定小数点ツールGUI

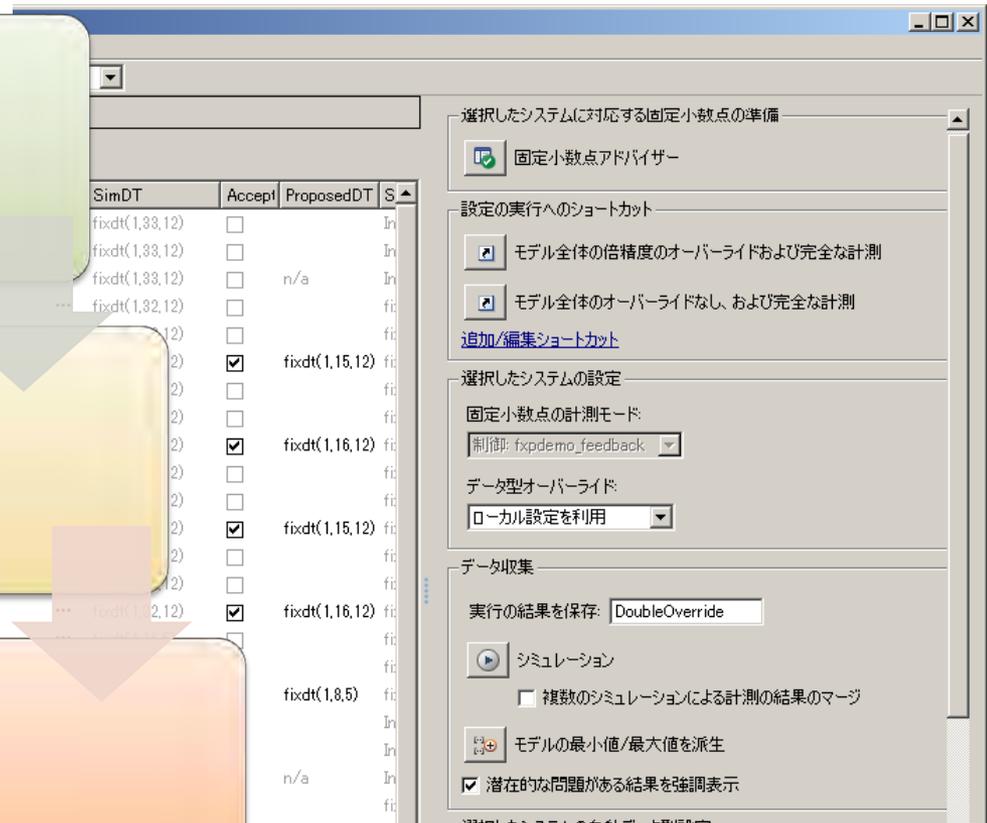
Simulinkモデル固定小数点設定の最適化

- [ツール]メニュー⇒[固定小数点ツール]

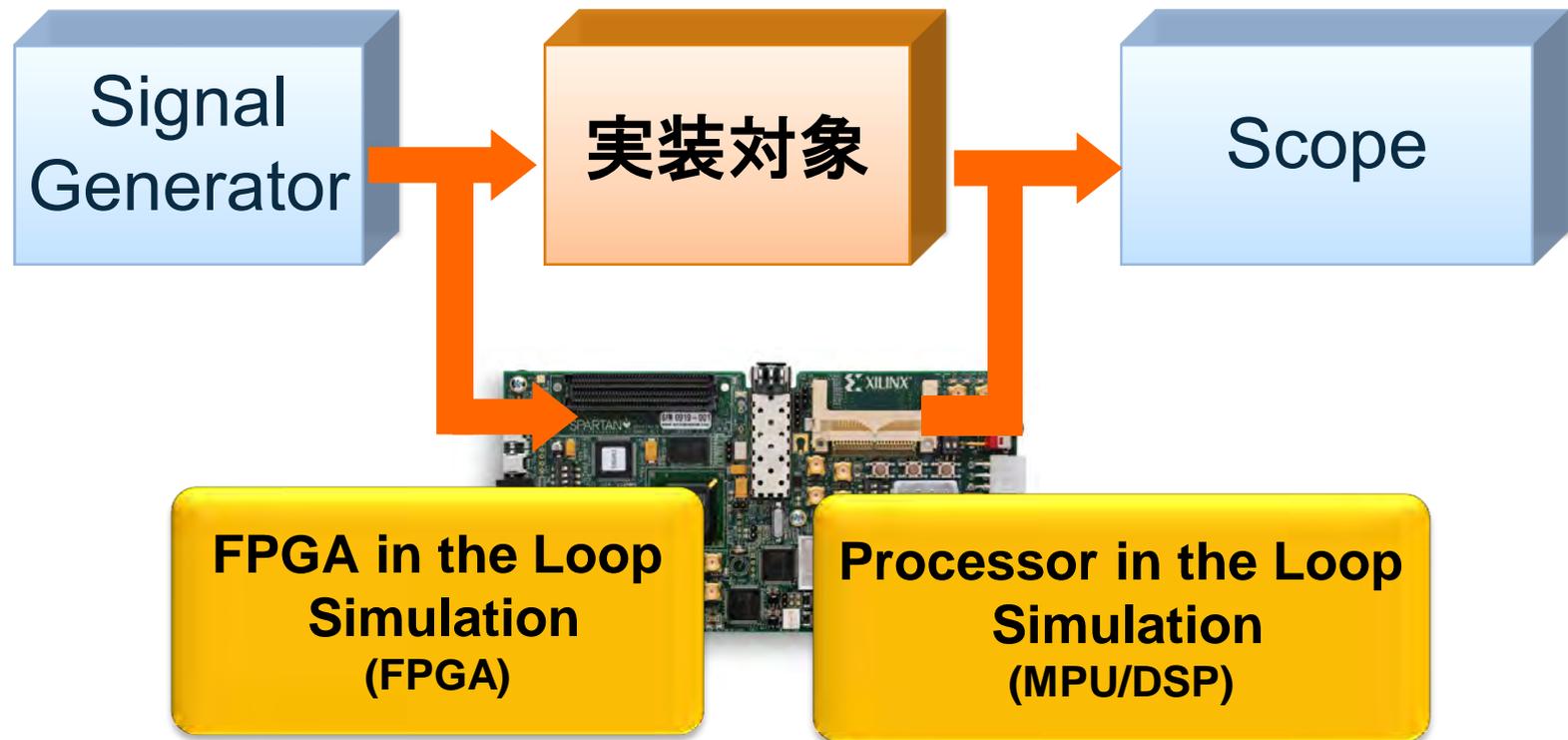
固定小数点
シミュレーション

浮動小数点
(オーバーライド)
シミュレーション

ログから最適化
(小数点・整数ビット)



Simulinkテストベンチによる実装用コード検証： X in the Loop Simulation



まとめ

- 信号処理ワークフローにおける有用な機能を紹介
- 今回のデモリソース

- 製品付属のドキュメント、デモ

- MathWorks Webサイト File Exchange

- MathWorks Webサイト ソリューション

- etc.

- デモファイル



豊富なMATLAB/Simulink機能をご活用下さい😊