# MATLAB/Simulink を使ってみませんか?

## ～ 信号処理編 ～

# はじめに

## 1. 本資料の対象者

・興味はあるが、なかなか触れる機会がない方
・購入前に使用イメージを明確にしておきたい方
・実際に操作しながら、どのようなメリットがあるか確認したい方

本資料は「MATLAB®/Simulink®を使ってみたいのだけれど、なかなか触れる機会がない・・」、「資料やマニュアルを読みながら、ゼロから自分で試す気が起こらない・・」など、日々のご業務の中でツール評価の時間を捻出するのが難しいといったお悩みにお応えします。信号処理分野の主要製品にフォーカスした例題を用いて、一連のフローを段階的に体験していただくことにより、実際の作業をイメージしながら、より短時間でツール評価を行うことができます。 本資料がお客様のツール評価にお役立て頂ければ幸いです。

## 2. 本書で使用するツールについて

本書はWindows 環境にインストールされた **MATLAB R2008b**をベースに記述されています。このため、UNIX 環境にインストールされたMATLABとはインタフェースが若干異なります。なお、お客様の環境で本資料のファイルを実行される際には以上のMATLABバージョンについて一度ご確認ください。

<使用ツール>

| | |
|---|---|
| MATLAB | Version 7.7 |
| Simulink | Version 7.2 |
| Data Acquisition Toolbox | Version 2.13 |
| Filter Design Toolbox | Version 4.4 |
| Fixed-Point Toolbox | Version 2.3 |
| Image Processing Toolbox | Version 6.2 |
| Signal Processing Blockset | Version 6.8 |
| Signal Processing Toolbox | Version 6.10 |
| Simulink Fixed Point | Version 6.0 |
| Video and Image Processing Blockset | Version 2.6 |

<補足のデモで使用するツール>

| | |
|---|---|
| Communications Blockset | Version 4.1 |
| Image Acquisition Toolbox | Version 3.2 |
| Real-Time Workshop | Version 7.2 |
| Real-Time Workshop Embedded Coder | Version 5.2 |
| SimPowerSystems | Version 5.0 |
| Statistics Toolbox | Version 7.0 |

# 目 次

# 第1章　開発環境の概要

## 1.1　MATLABプロダクトファミリ

ディジタルシステムは、音響・映像、情報・通信分野等、非常に幅広く用いられ、多くの研究者・技術者が携わっています。またシステムレベルのアルゴリズム開発や実現設計など、様々なステージがあり、その目的により開発ツールに対する要求は異なります。MATLAB 製品ファミリはその目的に合わせた様々な開発環境を提供します。図 1-1 にその環境の概略図を示します。

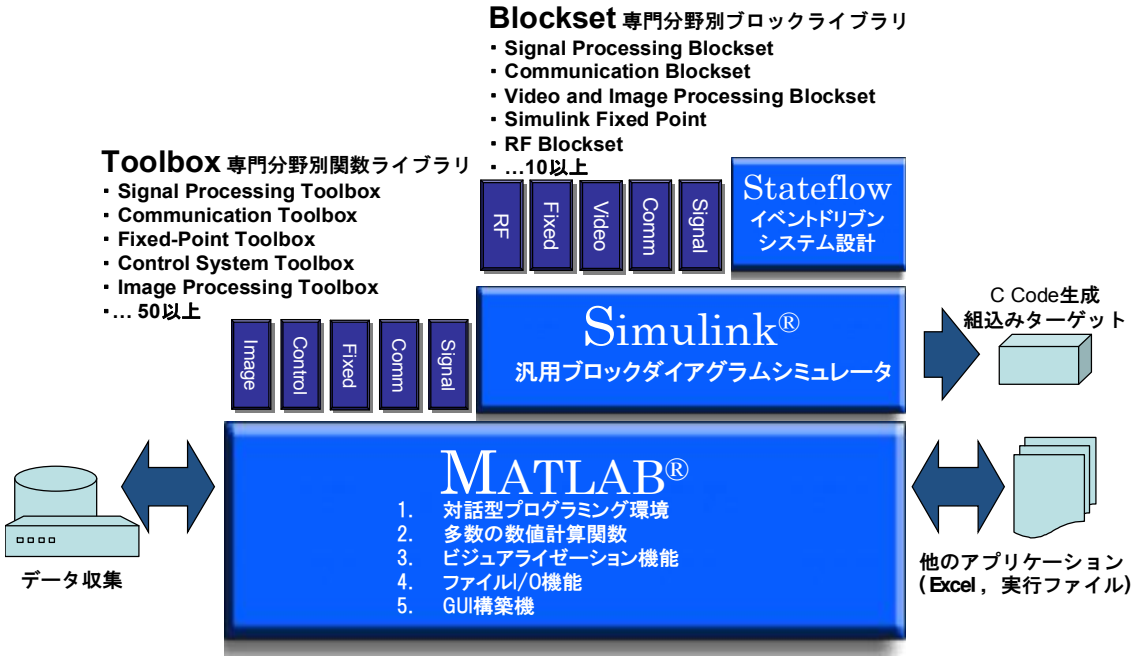



図 1-1　MATLAB プロダクトファミリの概略図

## 1.2　信号処理系オプションツール

MATLAB 製品ファミリには、汎用的なアナログ・ディジタル信号処理を行う Signal Processing Toolbox をはじめ､ディジタル信号処理システムの開発を支援する様々なツールが用意されています。ここでは、本資料で使用するツールを主に紹介します。

### Signal Processing Toolbox

汎用的なアナログ・ディジタル信号処理を行う関数を提供します。また、一般的に良く用いられる機能に対しては、GUI 環境が用意されています。信号管理ツール、フィルタ設計ツール、スペクトル推定ツール、窓関数設計ツールなどがあります。

- ☞信号と線形システムモデル
- ☞FIR・IIR ディジタルフィルタの設計、解析、実行
- ☞スペクトラム推定
- ☞時間領域のパラメトリックモデリング
- ☞ウィンドウ関数
- ☞アナログフィルタ設計
- ☞FFT、DCT などの変換
- ☞統計的信号処理
- ☞波形の生成

## Filter Design Toolbox

Filter Design Toolbox は、ディジタルフィルタ設計の専用関数ライブラリです。Signal Processing Toolboxのディジタルフィルタ設計機能をさらに拡張する関数群が用意されています。

- ☞最小次数、最小位相、halfband、補間 FIR を含む高度な FIR フィルタ設計手法
- ☞完全再構成型（Perfect reconstruction）2 チャンネル FIR フィルタバンク設計
- ☞任意振幅特性フィルタ、群遅延等化器、櫛形フィルタを等の高度な IIR フィルタ設計手法
- ☞単精度浮動小数点、固定小数点によるディジタルフィルタの解析と実現
- ☞2 次カスケード型 IIR フィルタのスケーリング、オーダリング機能
- ☞ 単精度浮動小数点あるいは固定小数点(※ 1)で実現されたフィルタの丸め誤差解析
- ☞ローパスからローパス、ローパスからハイパス、ローパスからマルチバンドなどの FIR、IIR フィルタの変換
- ☞ LMS、RLS、ラティス、周波数領域、高速トランスバーサル、アフィン投影などの適応フィルタの設計、解析および実現
- ☞ CIC 固定小数点フィルタなどのマルチレートフィルタの設計、解析および実現
- ☞固定小数点フィルタの VHDL、Verilog コード生成(※ 2)

※ 1 固定小数点機能を利用するためには、Fixed-Point Toolbox が必要です。
※2 HDL コードの自動生成には Filter Design HDL Coder が必要です。

## Image Processing  Toolbox

Image Processing  Toolboxは、画像処理、解析、可視化およびアルゴリズム開発のための包括的な関数ライブラリやグラフィカルツールを提供します。

- ☞線形および非線形フィルタリング、フィルタ設計、画像のぼけの修正、自動コントラスト強調などの画像強調機能
- ☞テクスチャー解析、輪郭の検知、形態解析、エッジ検出、区分け、関心領域（ROI）処理、特徴抽出などの画像の解析
- ☞色空間補正や、デバイスに依存しない ICC プロファイルインポート、エクスポートなどのカラー画像処理
- ☞ コントロールポイントを選択するためのグラフィカルツールなどの空間的変換、画像のレジストレーション
- ☞FFT、DCT、ラドン、ファンビーム投影法などの画像変換
- ☞DICOMインポートおよびエクスポート
- ☞ 対話的な画像表示や、画像 GUI 構築のためのモジュラーツール
- ☞多次元画像の処理をサポート

## Fixed-Point Toolbox

Fixed-Point ToolboxはMATLABに固定小数点データタイプと演算機能を提供します。

- ☞MATLABに固定小数点データタイプを追加
- ☞ 固定小数点による演算子、論理演算子をサポート
- ☞MATLABーSimulink間の固定小数点データ交換をサポート
- ☞データのロギング、およびデータタイプオーバーライドを含む、浮動小数点ー固定小数点変換ツールを提供
- ☞MATLABワークスペース内の固定小数点アルゴリズムの実行速度の高速化機能を提供

## Signal Processing Blockset

Signal Processing Blockset は、ディジタル信号処理システムを Simulink 環境でモデル化する場合に有用なブロックライブラリを提供します。これにより、周波数解析、マルチレート処理、各種フィルタの設計・実現、適応フィルタ等を行うことができます。

- ☞ 基本的な信号処理：FFT、DFT、及びその逆変換、窓関数
- ☞ Decimation/Interpolation、線形予測関連、バッファ関連、カウンタ関連、Shift Register
- ☞ ベクトル、行列、線形代数：Convolution、Autocorrelation、Matrix Multiplication、LU、Cholesky、QR 分解
- ☞ スペクトル推定 ：Short-Time FFT、Yule-Walker AR、Burg、Modified Covariance
- ☞ フィルタ設計と実現：浮動小数点あるいは固定小数点(※ 1)での FIR と IIR フィルタリング、適応フィルタ関連、ラティスフィルタ関連、マルチレートフィルタ関連を含みます。フィルタ実現においては直接型の転置型やバイクワッドフィルタを利用することができます。Filter Realization Wizard では、ユーザの仕様をもとにディジタルフィルタの Simulink モデルを自動的に生成します。

※ 1 固定小数点機能を利用するためには、Fixed-Point Toolbox/Simulink Fixed Point が必要です。

## Video & Image Processing Blockset

Video and Image Processing Blockset は航空宇宙・防衛、自動車、通信、エレクトロニクス、教育、医療機器等の広範な適用分野で使用されている画像処理用の組込みシステム設計において､最も基本的なアルゴリズムから高度なアルゴリズムまでカバーします。

- ☞ 任意のワード長の浮動小数点、整数、固定小数点データタイプによるリアルタイム動画像処理システムのモデリング、シミュレーション
- ☞ マルチメディアファイル入出力（I/O）および、シミュレーション中、あるいはシミュレーション後のビデオストリームの状態の表示
- ☞2-D フィルタ、2-D 変換、および基本的な幾何学変換の作成および実現
- ☞ 色空間変換、リサンプリングなどの標準的なカラー動画像変換技術を提供
- ☞ エッジ検出、しきい値、morphology 演算、統計、合成といった画像解析および拡張アルゴリズムの提供
- ☞ Real-Time WorkshopR（オプション製品）との統合による ANSI/ISO C コードの自動生成

## Simulink Fixed Point

Simulink Fixed Pointにより、Simulinkプロダクトファミリで固定小数点機能が利用可能となります。これにより、固定小数点演算を用いた制御および信号処理システムの設計を行うことができます。

- ☞Simulink、Statefl ow、Signal Processing Blockset を用いたモデル上で固定小数点シミュレーションを利用可能
- ☞Real-Time Workshop、Stateflow Coder、Real-TimeWorkshop Embedded Coder を用いて上記モデルから固定小数点コード生成が可能
- ☞xPC Targetや他のラピッドプロトタイピングシステムによる固定小数点ベースのラピッド・プロトタイピングをサポート
- ☞固定小数点データタイプの制御とスケーリング
- ☞ オーバーフロー、飽和エラーの識別ツールの提供
- ☞ レンジおよび精度のトレードオフ確認用の自動スケーリングツールを装備
- ☞ 自動的なスケーリング調整、高度演算、その他のタスクの取扱いが可能

## 1.3　モデルベースデザインとは？

モデルベースデザインによる開発フローではSimulinkモデルをベースに開発を行います。図1-2に示すように、仕様検討から実装やテスト・検証までをSimulinkという同一環境で行うことを可能とし、迅速なプロトタイピングを実現します。また高価なプロトタイプテスト用のハードウエアやプロトタイプテストのためのソフトウエア開発が不要になり、開発期間の短縮にも大きな効果があります。

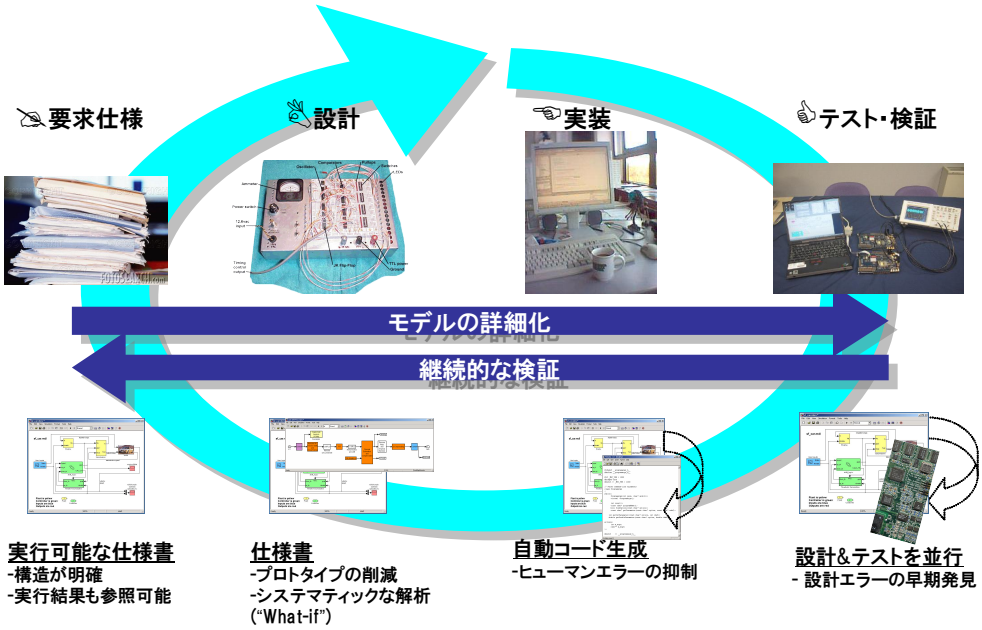



図1-2 モデルベースデザインフロー

図1-3に示すようにモデルベースデザインフローは4段階に分かれています。なお、本書では主に①②③の各ステップにおける例題を取り上げます。

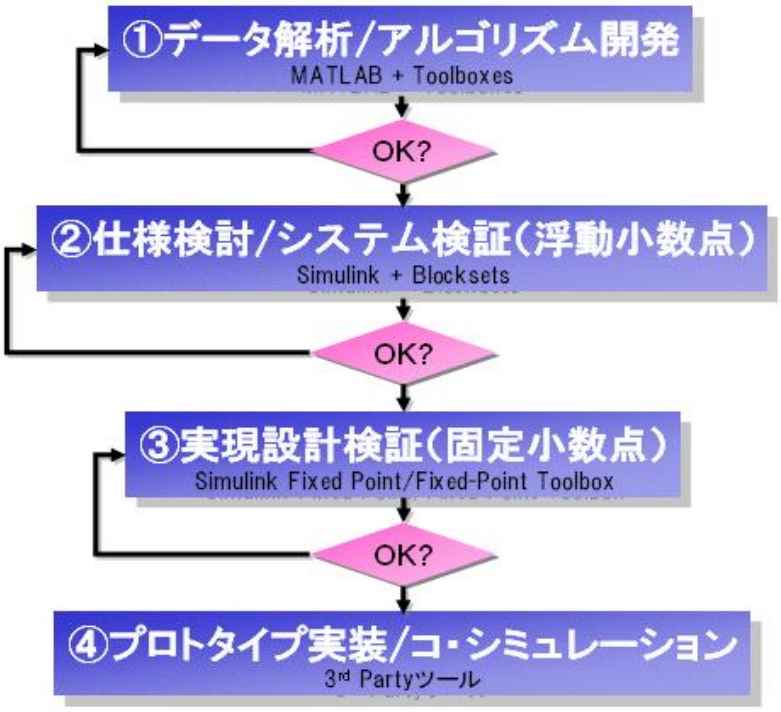



図1-3 モデルベースデザインフロー

# 第２章　１次元信号のスペクトル解析

この章では､MATLABのプログラミング環境におけるデータ解析の例として１次元信号のスペクトル解析を取り上げます。以下はステージと実施手順です。


図2-1 １次元信号のスペクトル解析の実施手順とステージ

## 2.1　Excel ファイルの取り込みとグラフ表示

カレントフォルダ下に格納されているExcelファイル（data1.xls）をMATLABのメモリ内に取り込みます。ここでは、インポートウィザードを使います。以下の手順を実行してください。

### ①インポートウィザードによる取り込み

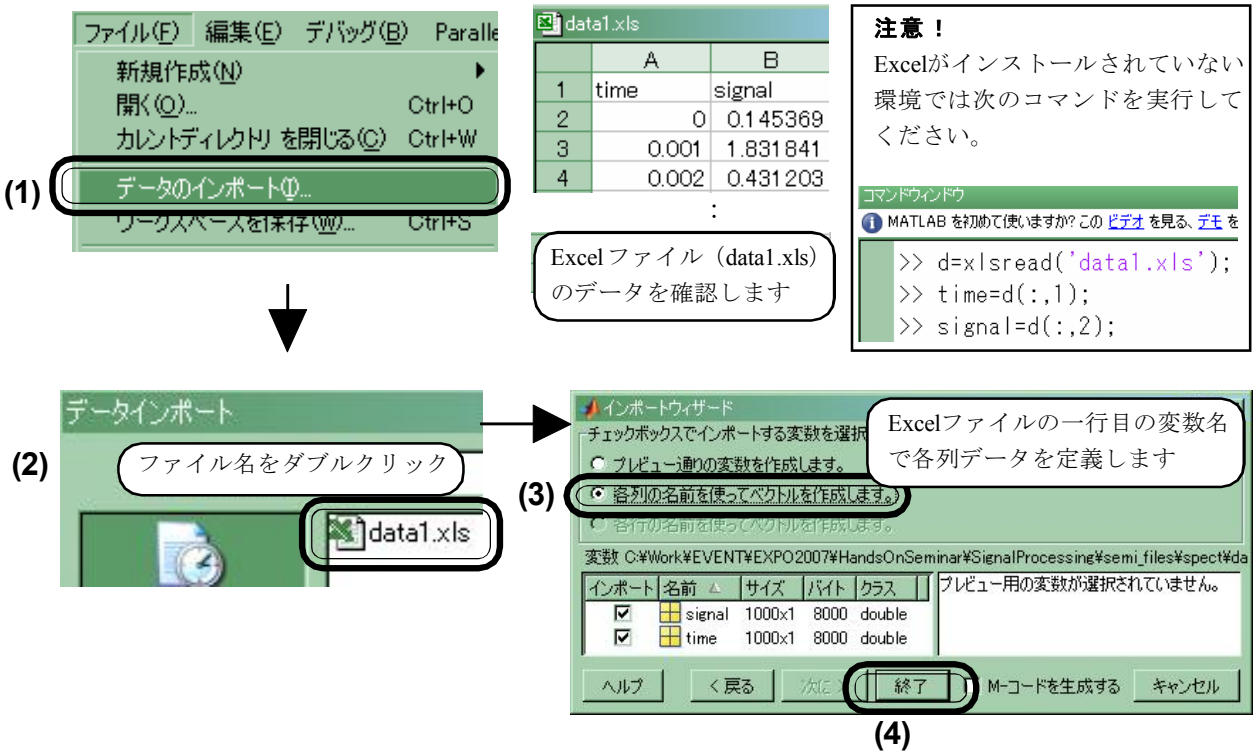



図2-2 インポートウィザードによるExcelファイル（data1.xls）の取り込み手順

これまでの操作で取り込んだデータは、ワークスペースと呼ばれるメモリ領域に格納されます。次に、変数エディタを使ってそのデータを確認します。以下の手順を実行してください。

**②データ要素の確認**


図2-3 ワークスペースと変数エディタによる変数の確認

**Point 1　＜データは配列として管理＞**

MATLABではデータを多次元配列として定義するためデータの参照や変更が容易です。

以上より、変数 time と変数 signal が MATLAB に取り込まれました。次に、変数 time を X 軸、変数 signal を Y 軸として 2 次元グラフを表示してみます。以下の手順を実行してください。

**③2次元グラフの表示**

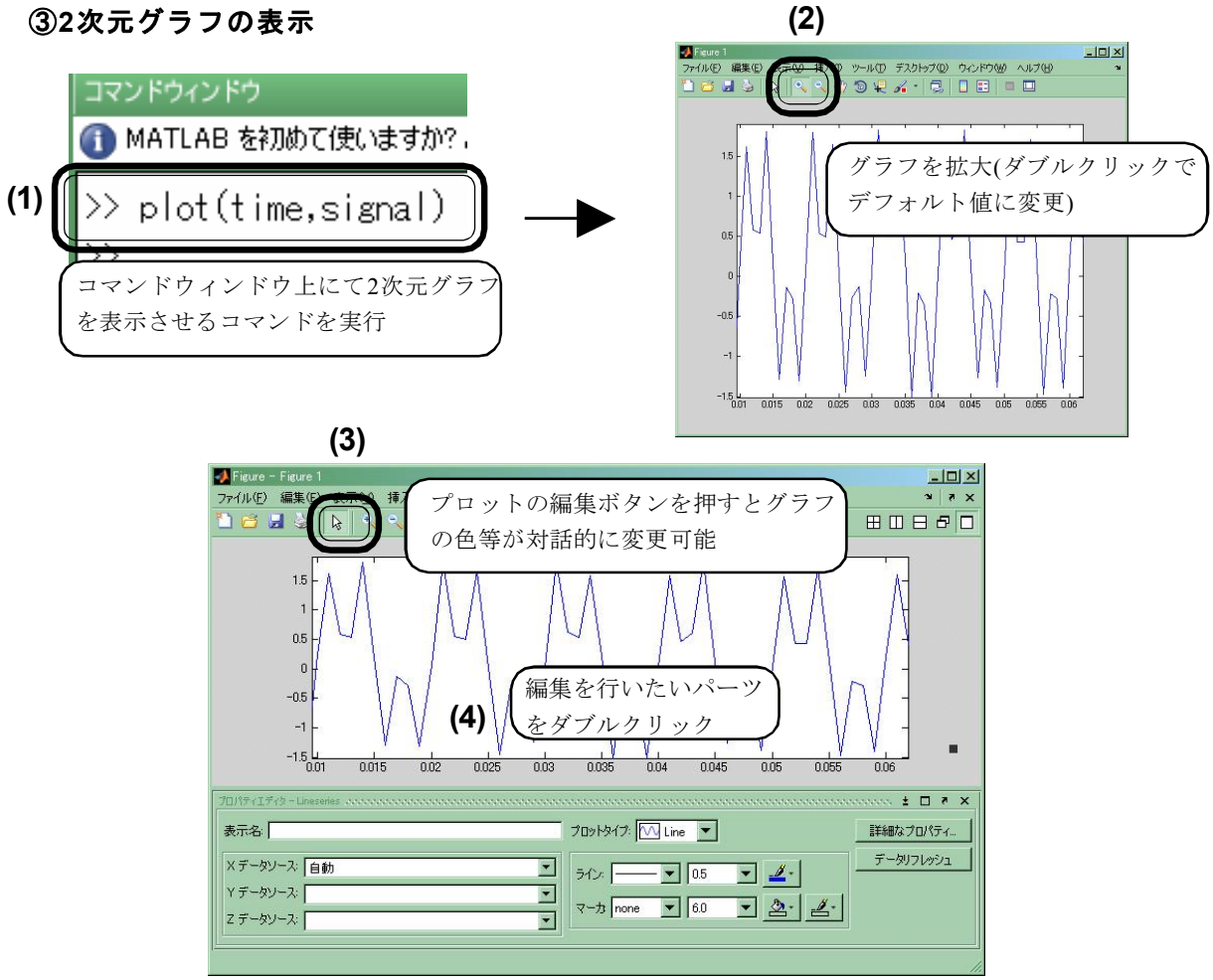



図2-4 2次元グラフの表示とグラフの編集手順

**Point 2　＜柔軟なグラフ機能＞**

MATLABでは2/3次元のグラフをはじめ、ボリュームデータの表示、アニメーション等、様々なグラフィックスをシンプルなコマンドで簡単に描くことが可能です。また、対話的に編集可能なため、即座に編集結果を確認することができます。

## 2.2　Excel データのスペクトル解析

前節のグラフより、変数signalのデータには周期性があることが確認できます。このことから、複数の周波数成分をもつ正弦波が加算されている信号であると予想できます。次ではこの信号にどのような周波数成分が含まれているか？を解析するためにパワースペクトルを計算します。ここでは、MATLABのプログラムファイルであるM-ファイルを使ってパワースペクトル解析を行います。なお、このプログラムには１行追加が必要です。以下の手順を実行してください。

### ①M-ファイルの作成

(1)

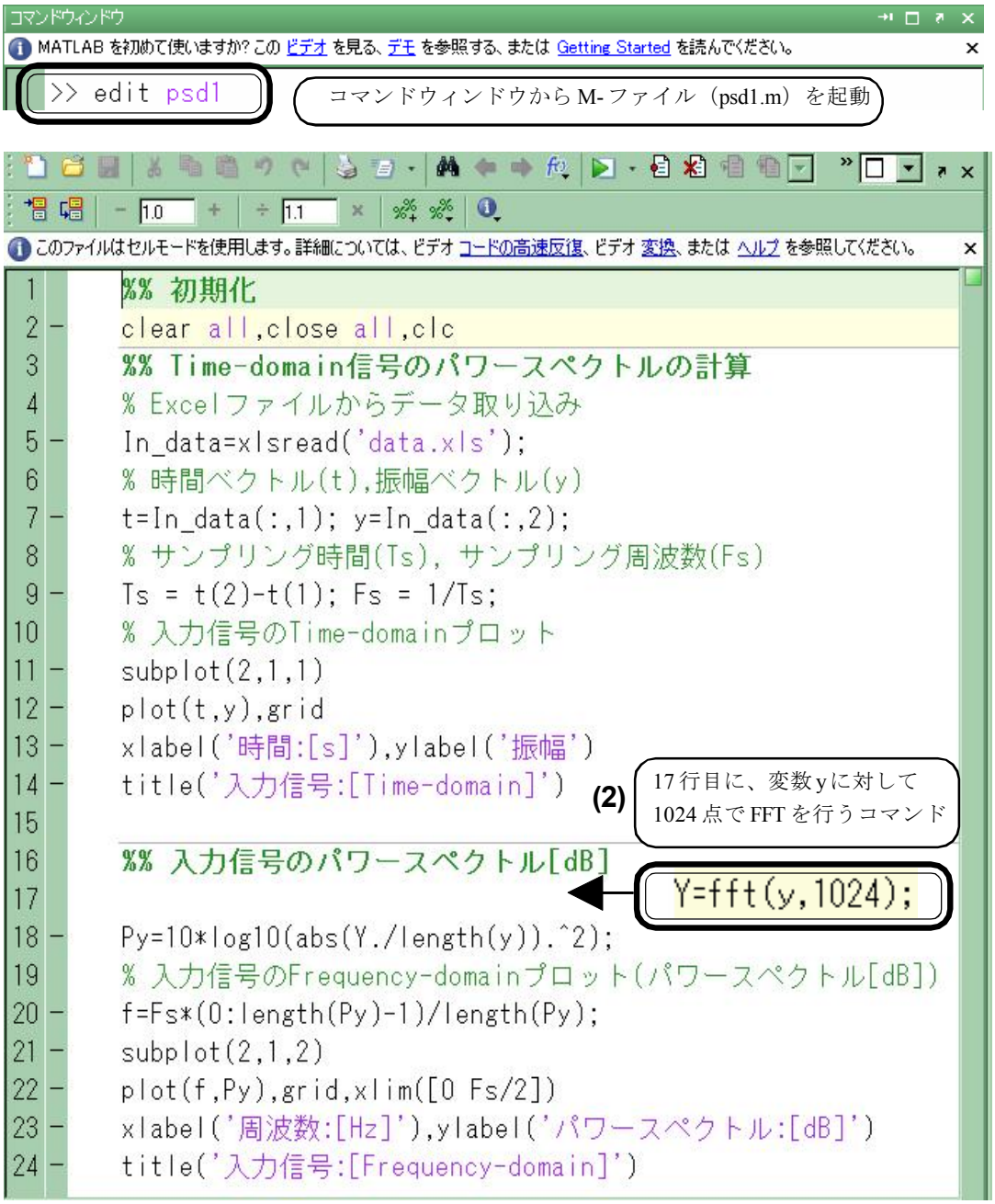



図 2-5　M-ファイル（psd1.m）の起動および修正内容

FFT 処理を追加した M-ファイル（psd1.m）を実行します。以下の手順を実行してください。

**②M-ファイルの実行**

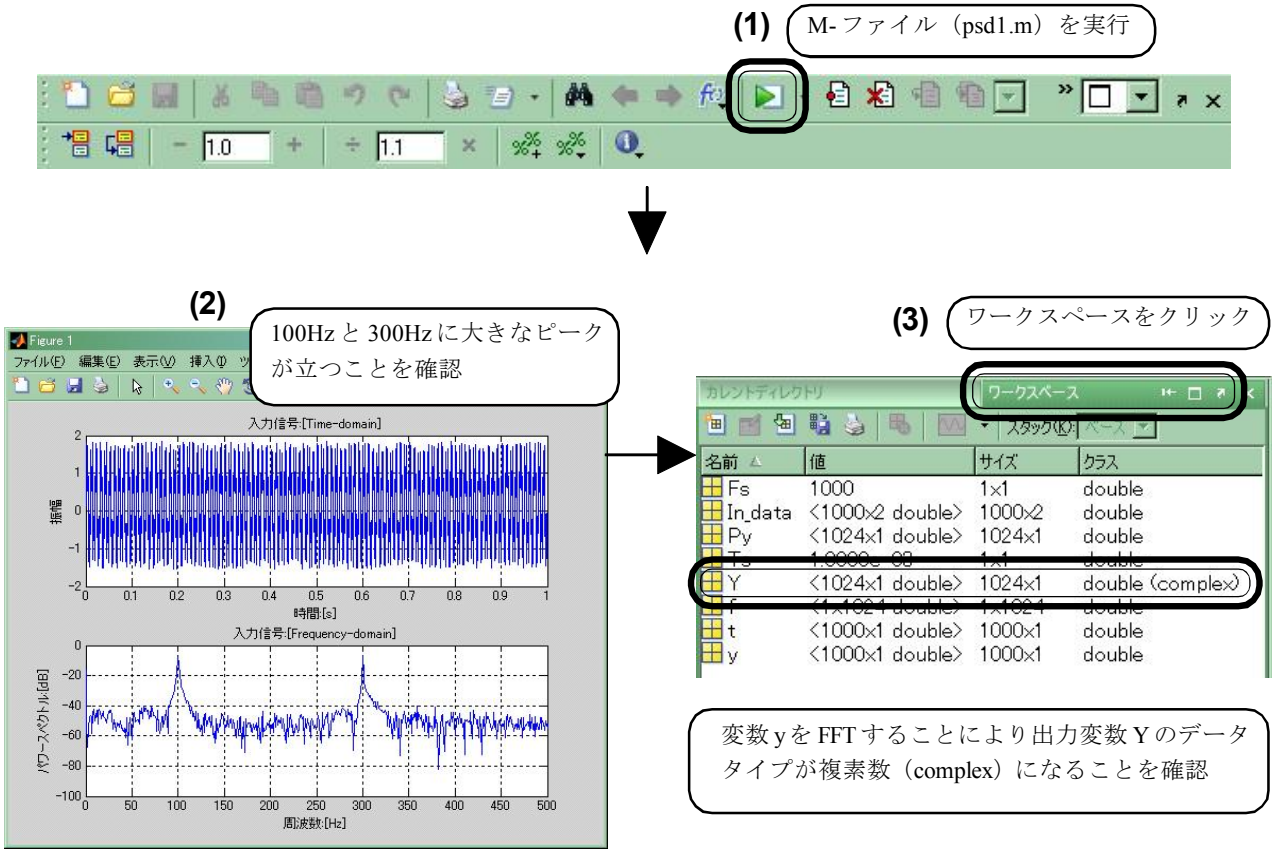



図 2-6 M-ファイル（psd1.m）の実行

以上より、Excelファイル（data1.xls）より取り込んだデータには100Hzと300Hzの正弦波が支配的に含まれていることが確認できます。このようにMATLABではM-ファイルと呼ばれるテキスト形式のプログラムを作成することで複数の処理をまとめて実行することができます。

**Point 3　＜高精度かつ高速な配列演算関数＞**

MATLAB では、FFT をはじめ信号処理分野で一般的に使用される多数の数学関数が提供されています。なお、数値演算のコアライブラリとして各CPU毎に最適化されたLAPACK / BLASを実装しているため、行列/配列演算を簡単かつ高速に実現することができます。なお、数値演算には IEEE の倍精度演算がデフォルトとして適用されます。

**Point 4　＜M-ファイルによるシンプルなプログラミング＞**

MATLAB は、C/Fortran 言語のようなコンパイラ言語に比べて、記述や操作が簡単なのでアルゴリズムの本質部分に専念できます。また、対話型の環境のためプログラム経験が浅い方でも馴染みやすく、デバッグ作業も簡単です。さらに、C/C++言語に比べて大幅にツール習得時間を削減できることも大きな利点といえます。また、次節で述べられているM-Lint機能の情報を活用することにより、早期にエラーを発見し、修正を行うことも可能です。

最後に、M-ファイルを作成する環境であるMATLABエディタの便利機能について紹介します。プログラムを作成し実行する上で最も工数・時間・労力を要する作業がデバッグ作業です。そこで、プログラムを実行する前に、予めコードに不適切な表現や文法ミスがあった場合、これを事前に告知する機能があるとどうでしょうか？この情報を元に早期に誤りを修正することで、デバッグ作業を短時間で行うことが期待できます。MATLABエディタにはM Lintと呼ばれる機能が実装されており、以上の作業を強力に支援します。以下の手順を実行・確認してください。

### ③M-Lint機能

**(1)**

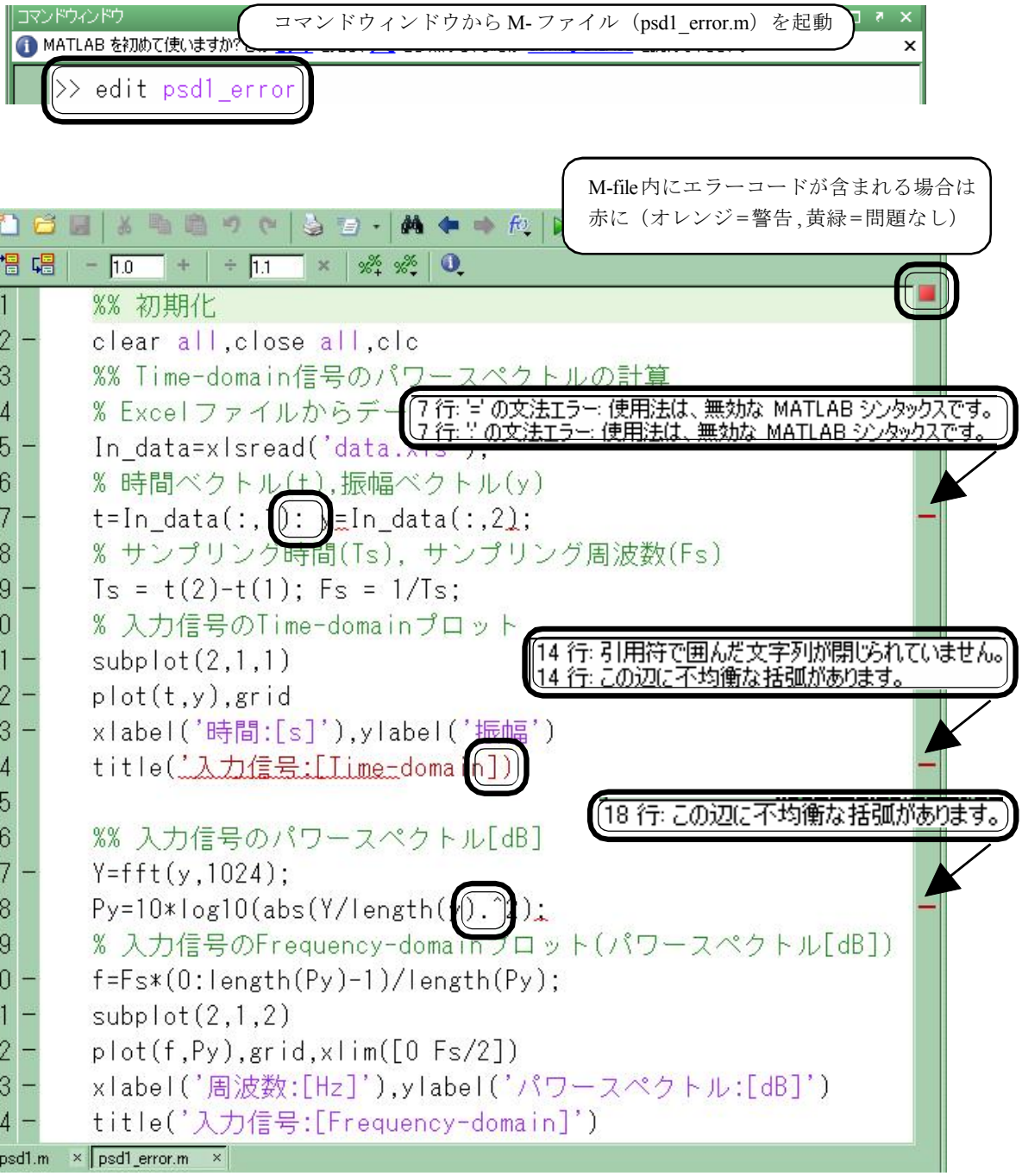
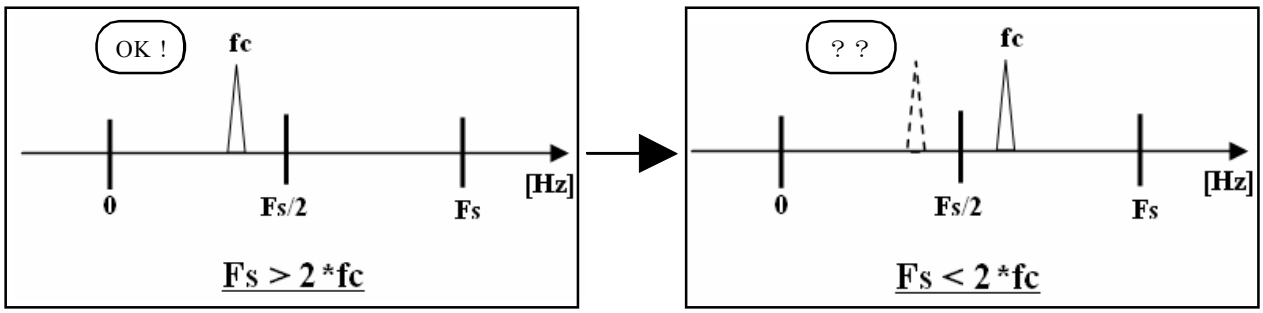



図 2-7 M-ファイル（psd1_error.m）の起動およびM-Lint機能のコメント内容

## 2.3　サンプリング定理の確認

**＜サンプリング定理＞**
信号の中に含まれている有効な信号成分の中で、最も高い周波数が fc の時、
ナイキスト周波数 2fc より大きい周波数でサンプリングする必要がある




サンプリング定理を満たしている場合

サンプリング定理を満たしていない場合
何が起きるのか？

信号をスペクトル解析する際、どの程度のサンプリング周波数でサンプリングするのが妥当でしょうか？また、上記のサンプリング定理を満たしていない場合にどのような現象が発生するのでしょうか？ここでは、上記定理を確認するために、サンプリング周波数(1000Hz)を一定にした状態で、信号の周波数を上げていった場合のスペクトルの変化を確認します。その際、M-ファイル内のパラメータ（周波数）を繰り返し変更しながら実行する必要があります。このような検証を行う際はエディタ機能であるセル実行が有効です。以下の手順を実行してください。

**①M-ファイルのセル実行**

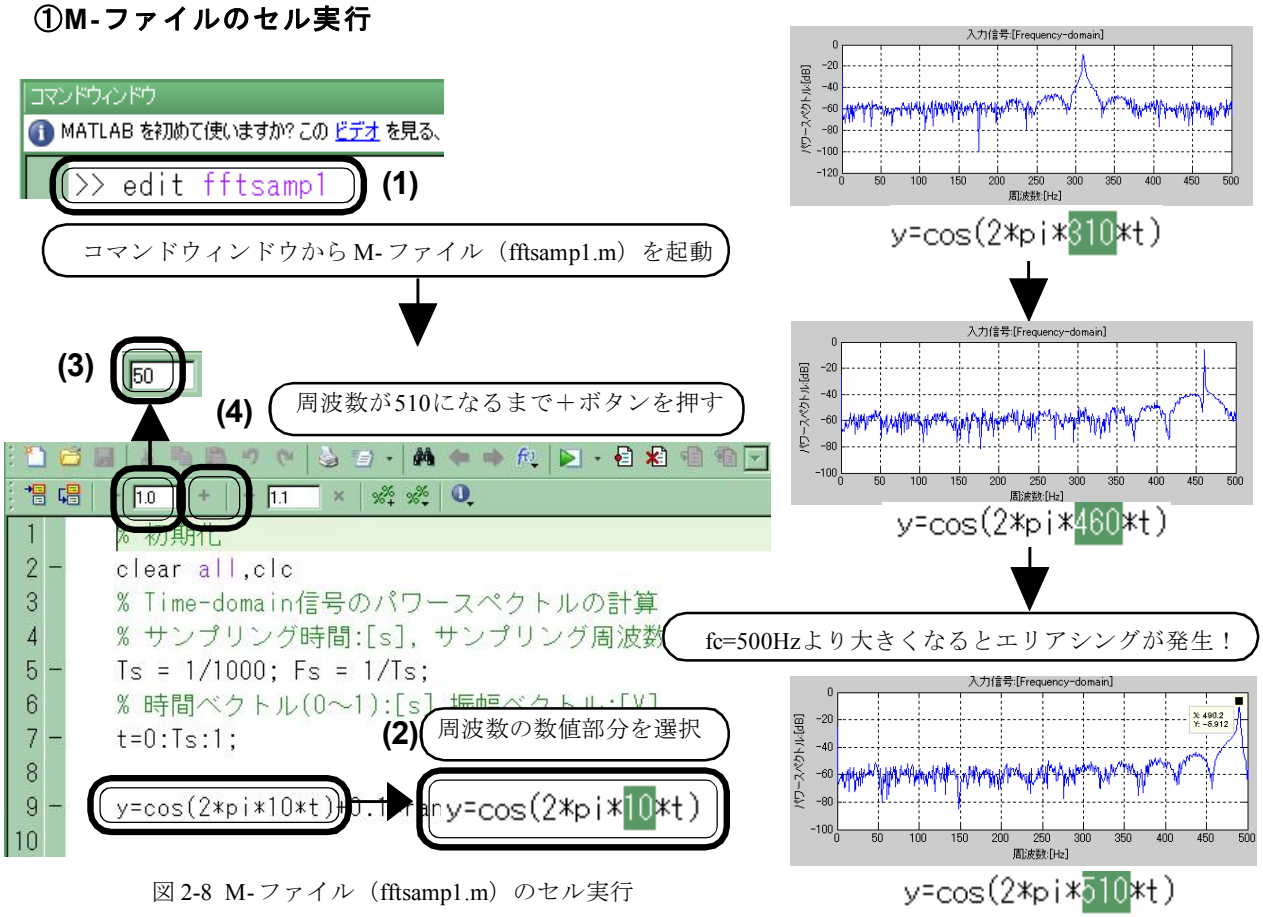



図 2-8 M-ファイル（fftsamp1.m）のセル実行

### 補足：　プロファイラ

MATLABエディタにはM-ファイルのパフォーマンス検証や比較検証を行うためのプロファイラが実装されています。この機能を使うことにより、コードのコール数やカバレッジ、実行時間を容易に表示することができます。ここでは、M-ファイル＝nonallo.m を例に、以下の手順で実行します。

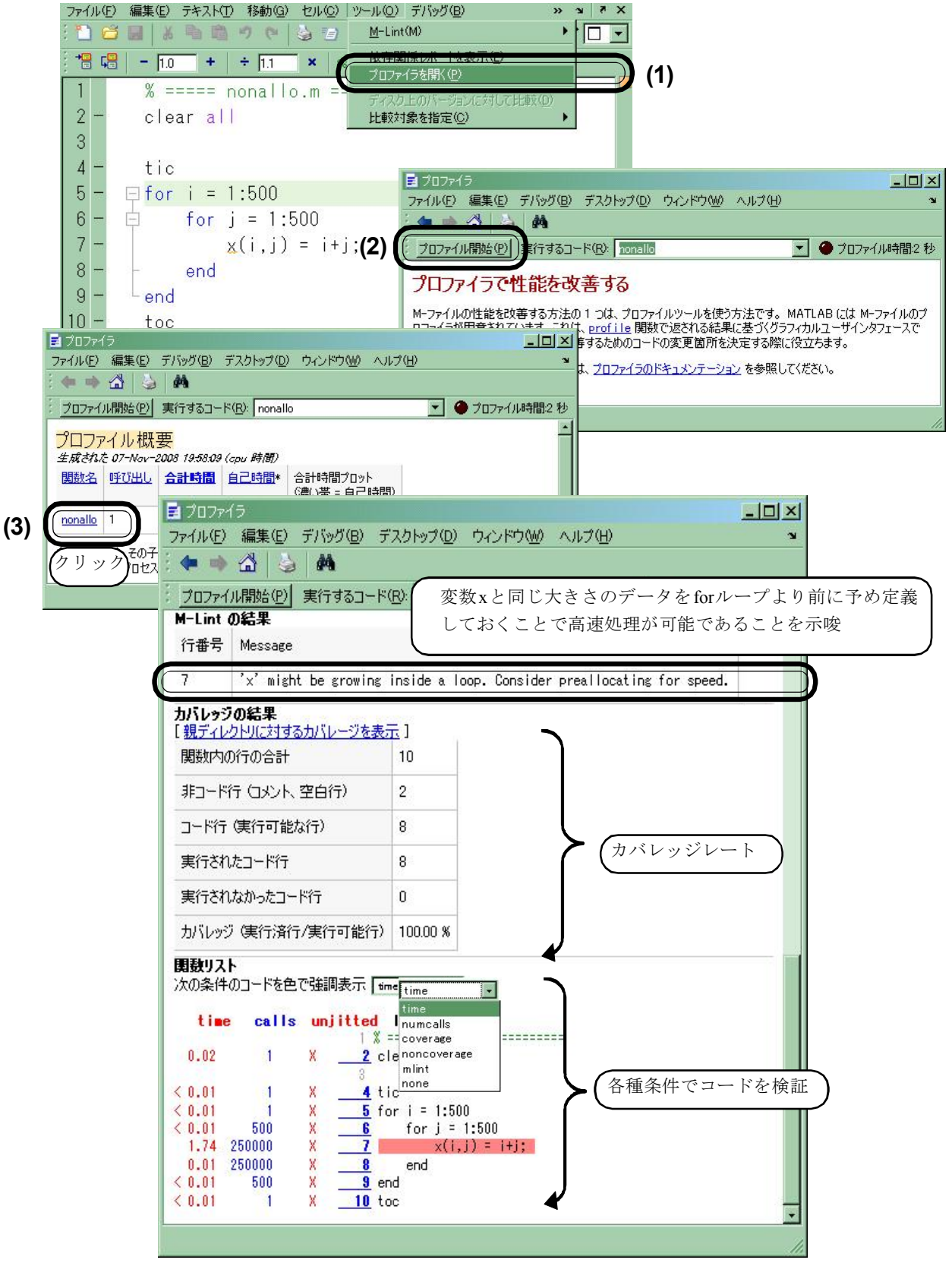



図 2-9 M-ファイル（nonallo.m）に対するプロファイル実行

# 第 3 章　ディジタルフィルタ設計

この章では、MATLABのディジタルフィルタ設計ツール（sptool）によるフィルタ設計・検証を取り上げます。以下はステージと実施手順です。

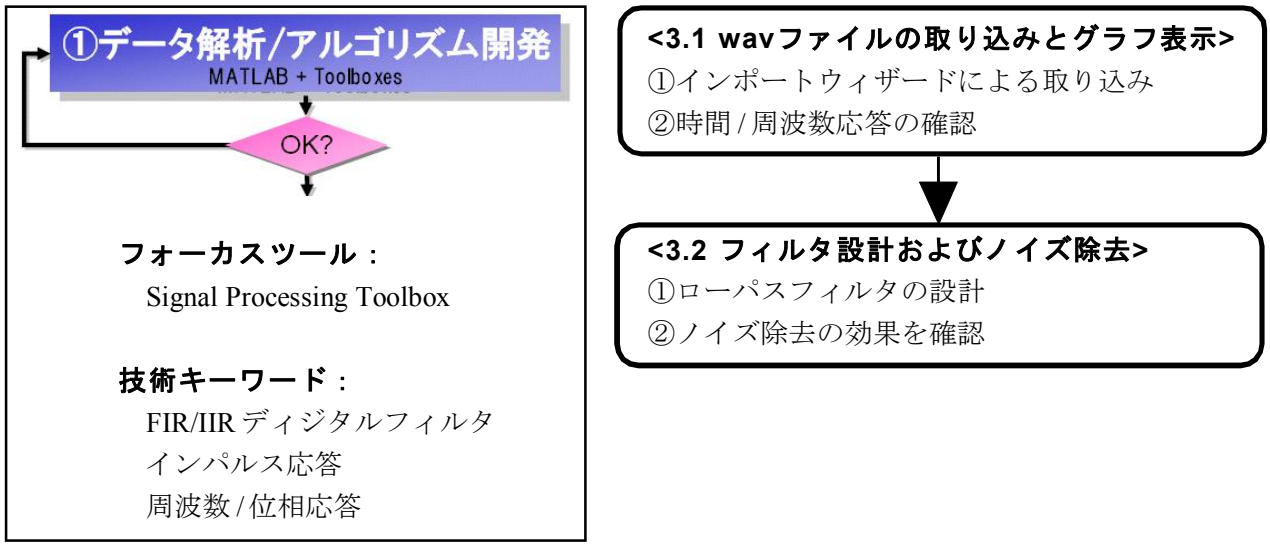



図3-1 ディジタルフィルタ設計の実施手順とステージ

## 3.1　wav ファイルの取り込みとグラフ表示

カレントフォルダ下に格納されているwavファイル（sound1.wav）をMATLABのメモリ内に取り込みます。ここでは、インポートウィザードを使います。以下の手順を実行してください。

### ①インポートウィザードによる取り込み

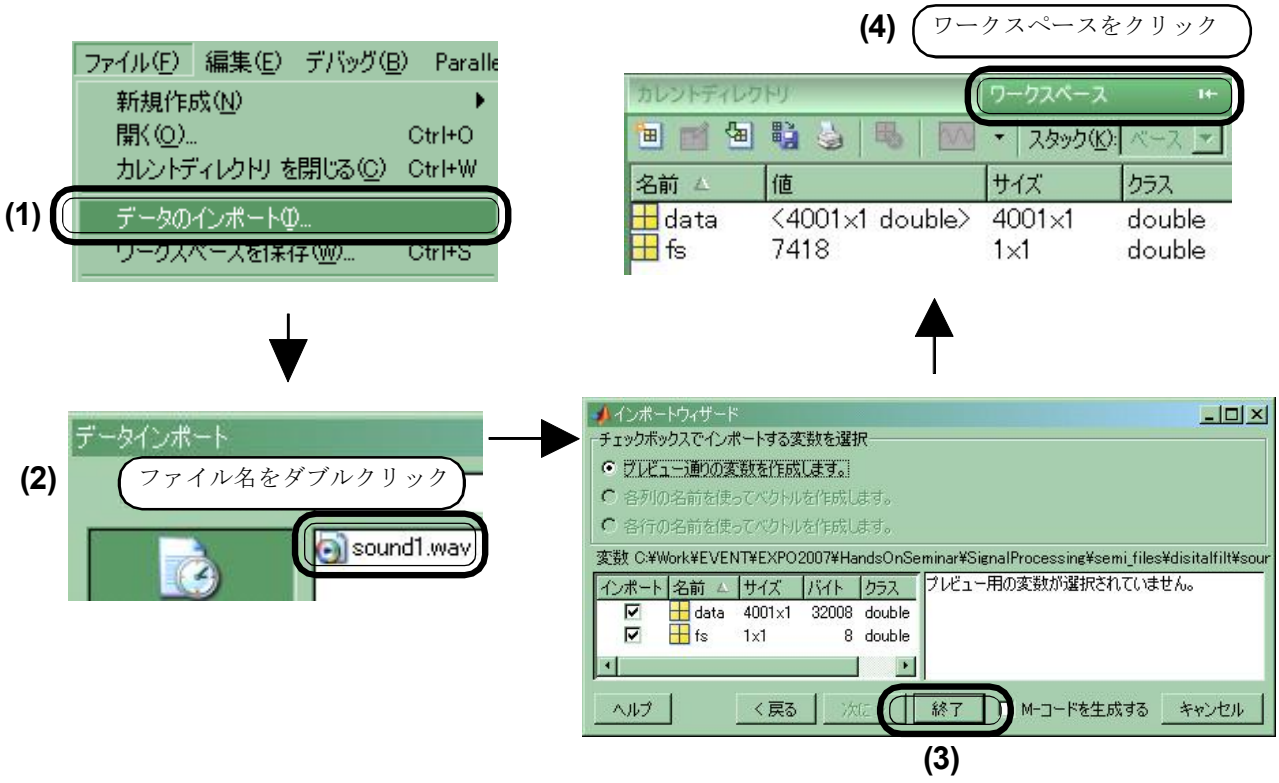



図 2-2 インポートウィザードによるwavファイル（sound1.wav）の取り込み手順

前章ではM-ファイルでの解析を取り上げましたが、ここではsptoolと呼ばれる信号処理解析ツールを使って、時間/周波数応答の確認、ディジタルフィルタ設計を行います。以下の手順を実行してください。

**②時間/周波数応答の確認**

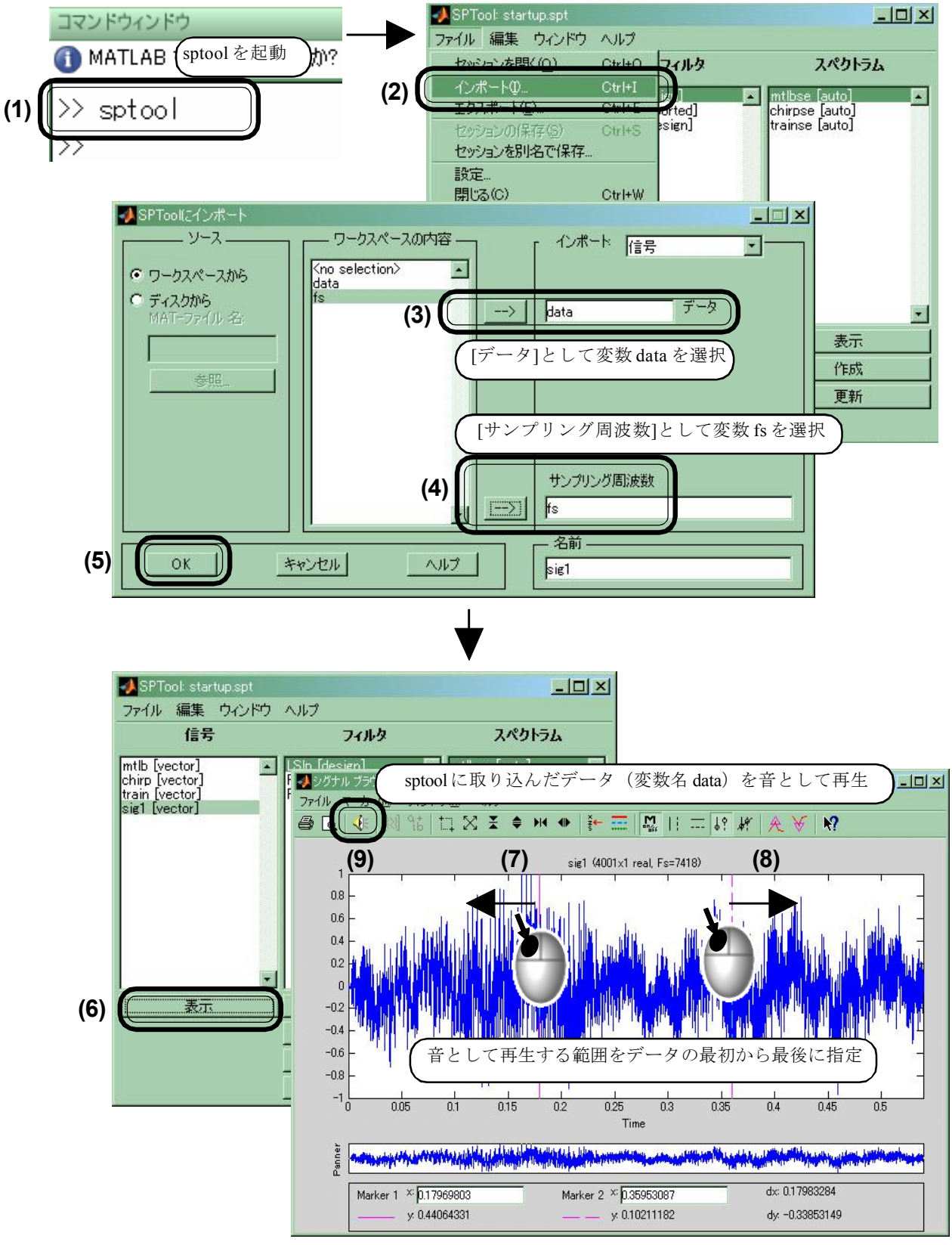



図 2-3 sptoolによるデータの取り込み手順と確認

次に、取り込んだデータ（変数名 data）の周波数応答を確認します。以下の手順を実行します。

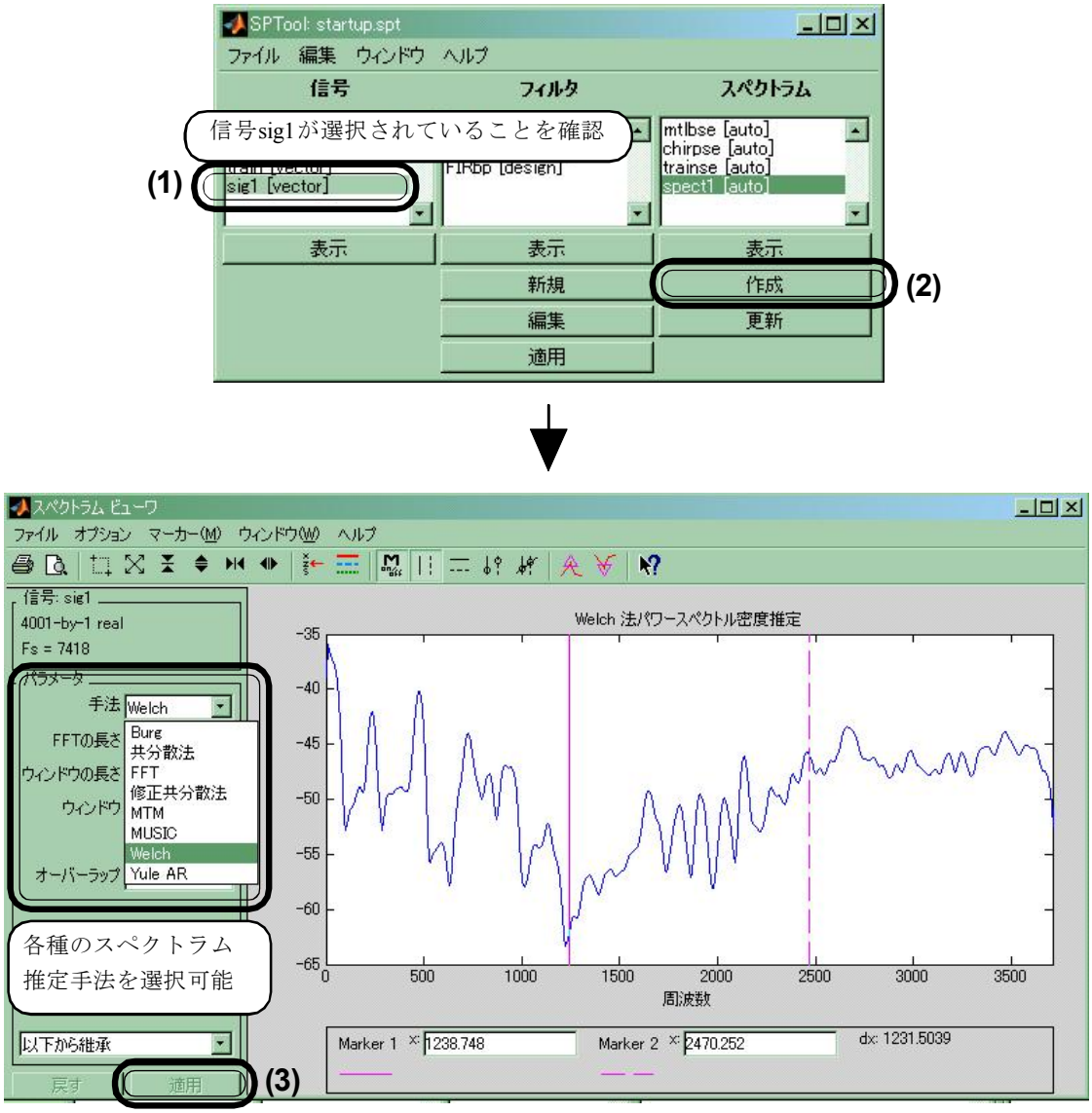



図 2-4　sptool によるパワースペクトラム推定手順と確認

先ほどの音から、データには高域ノイズが付加されている可能性が高いことが予想されます。また、以上の周波数応答からも1500Hz以上の帯域でノイズ成分が付加されていることが解析的に確認されました。次節では、この信号に対してローパスフィルタを適用し、高周波ノイズを除去することを試みます。

## 3.2　フィルタ設計およびノイズ除去

高周波ノイズを除去するためのディジタルフィルタを設計します。では、どのような仕様のフィルタを作成したらよいでしょうか？まず、一つ目の仕様はローパス特性を持っていることです。また、先の周波数特性の結果から、およそ有効帯域（0 ～ Fs/2 = 3709Hz）の半分以降で減衰していく特性であれば高周波ノイズを除去できそうであることが分かります。また、フィルタは線形位相を保持させるためにFIR型を採用します。以上の仕様に基づきフィルタを設計します。以下の手順を実行してください。

応答タイプ　　＝ローパス
設計手法　　＝FIR 型（窓関数法）
次数　　＝20
CutOff 周波数　　＝ 0.5(Fs/2 で正規化した場合)
※この周波数で6dB減衰 ※

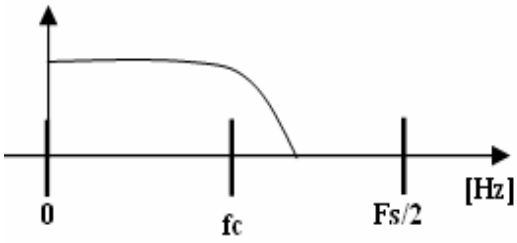



図2-5 設計するディジタルフィルタの仕様

### ①ローパスフィルタの設計

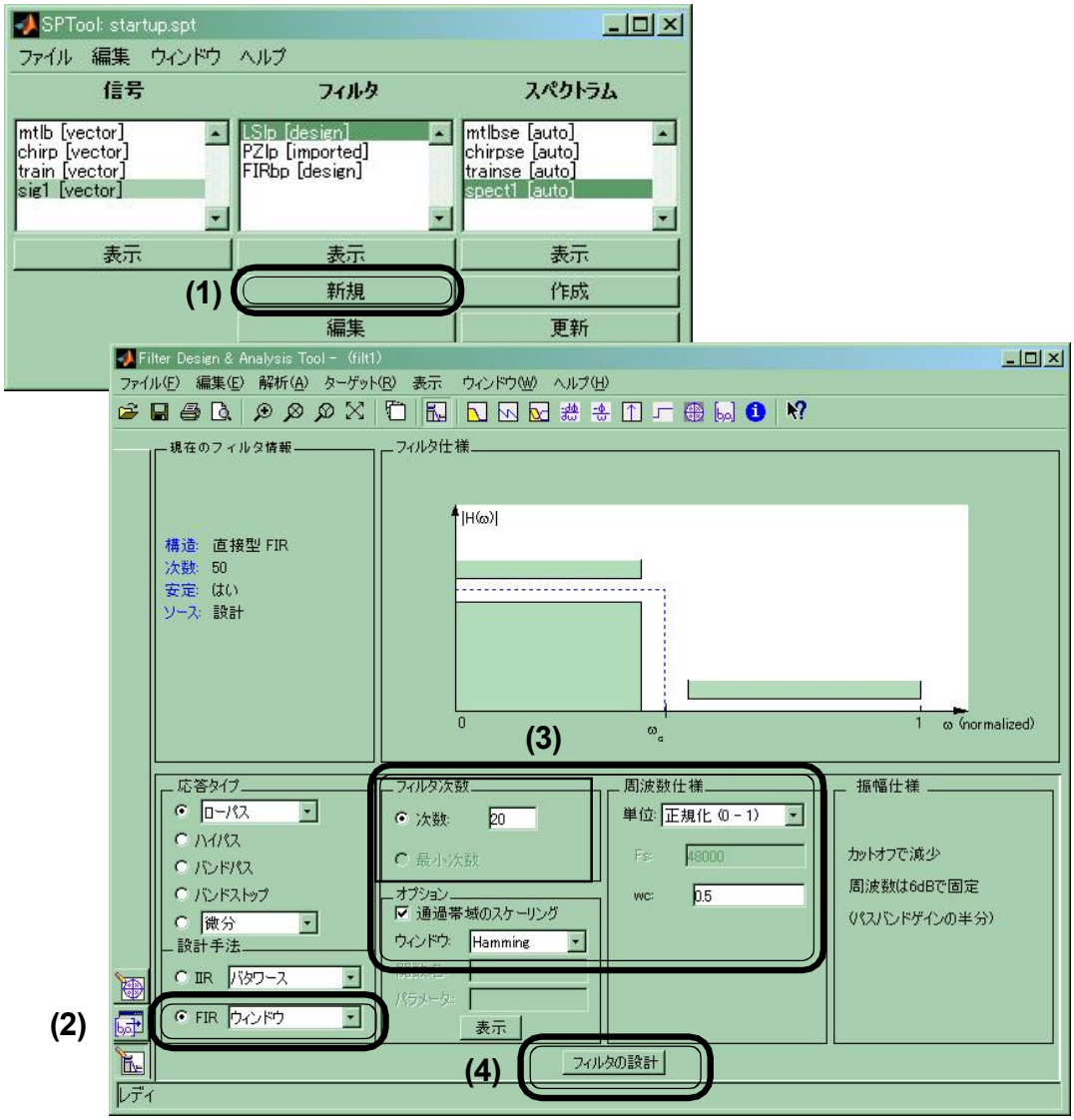



図 2-5 sptool によるディジタルフィルタの設計環境

次に、設計したディジタルフィルタの特性を確認します。以下の手順を実行します。

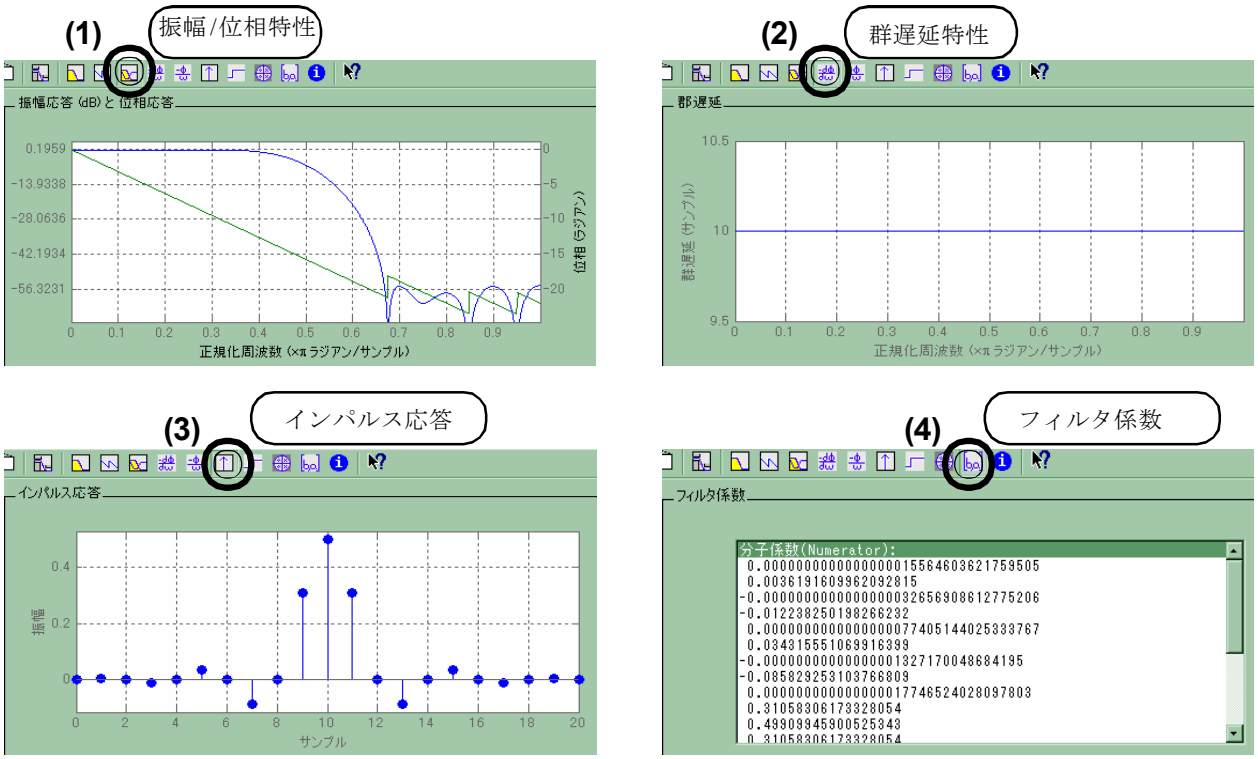



図2-6 設計したディジタルフィルタの特性を確認

以上の特性より、仕様通りのディジタルフィルタが設計できました。次節では、このフィルタに信号を適用しノイズ除去の効果を確認します。以下の手順を実行してください。

### ②ノイズ除去の効果を確認

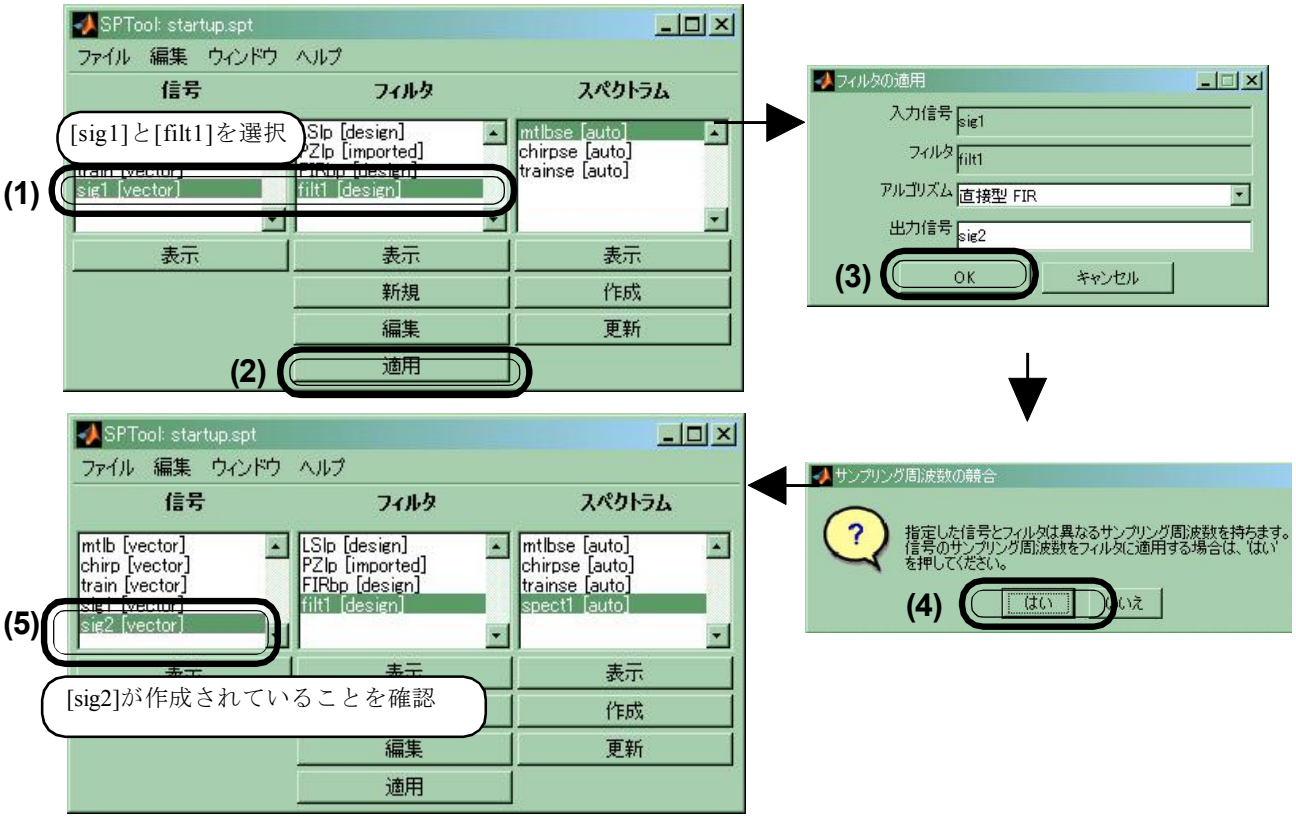



図 2-7　設計したディジタルフィルタを信号(sig1)に適用する手順

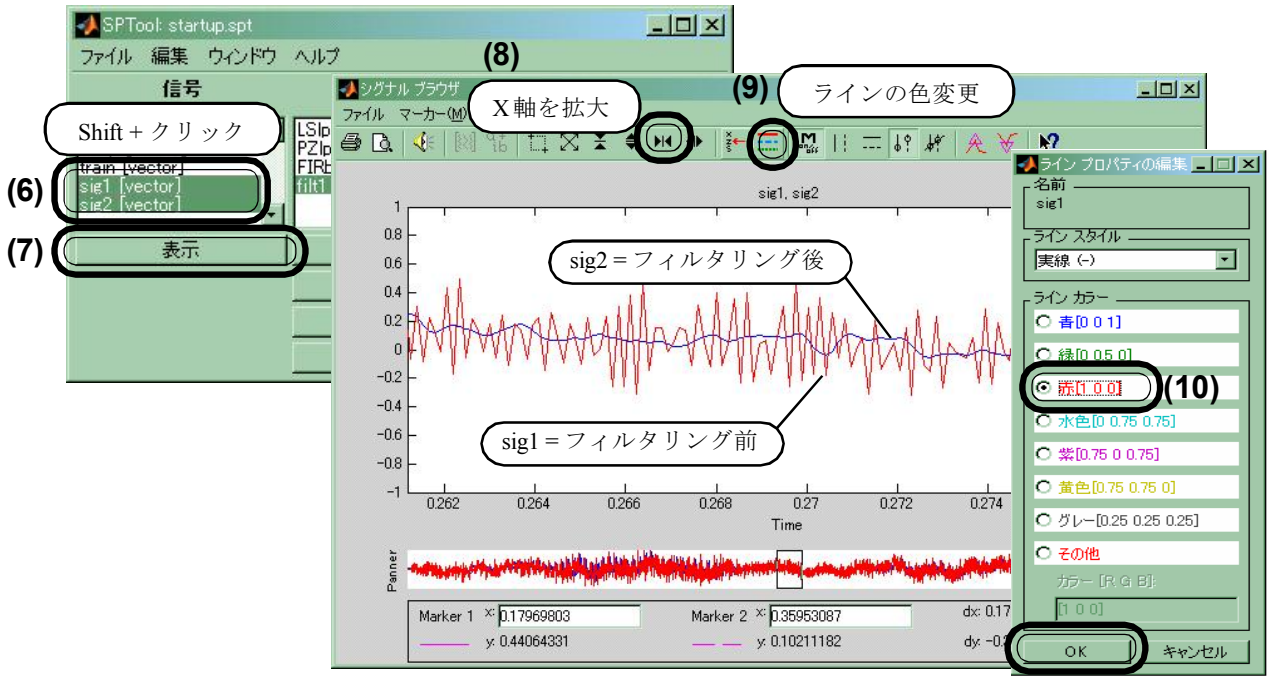



図2-8 フィルタリング前後の時間軸応答

以上の結果より、フィルタ通過後の信号（sig2）は高周波ノイズが除去されていることが確認できます。次に周波数応答も確認してみます。以下の手順を実行してください。

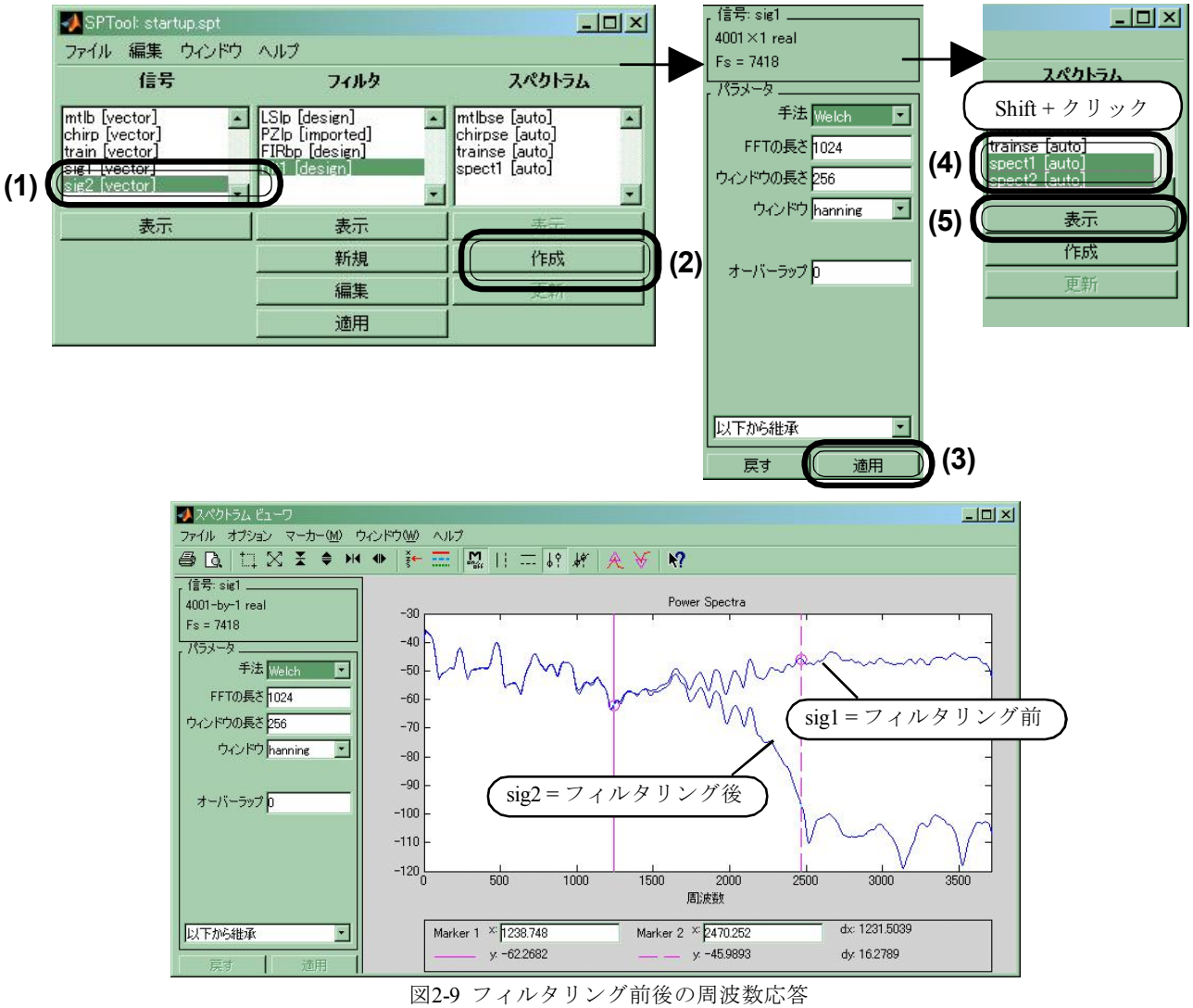



図2-9 フィルタリング前後の周波数応答

**Point 5　＜分野に特化した解析ツールおよび関数ライブラリ＞**

MATLAB では、Signal Processing Toolbox をはじめ信号処理分野で活用される多数の関数ライブラリ（40以上）が提供されています。解析内容に応じて適時利用することで、高度なアルゴリズムを簡単に実行することができます。各Toolboxでは、専門分野にフォーカスした解析関数およびGUIツールが提供されています。

## 補足： sptoolからの出力

先ほどの例題では、ワークスペース上に定義されたデータをsptool上に取り込むことを行いましたが、逆にsptool上で解析した結果（信号、フィルタ、スペクトル）をワークスペースに出力することもできます。また、sptoolは設計したフィルタをM-ファイルで容易に利用できるようにするために、設計内容と等価なM-ファイルを自動生成することもできます。以下の手順で実行します。

①ワークスペースへの出力

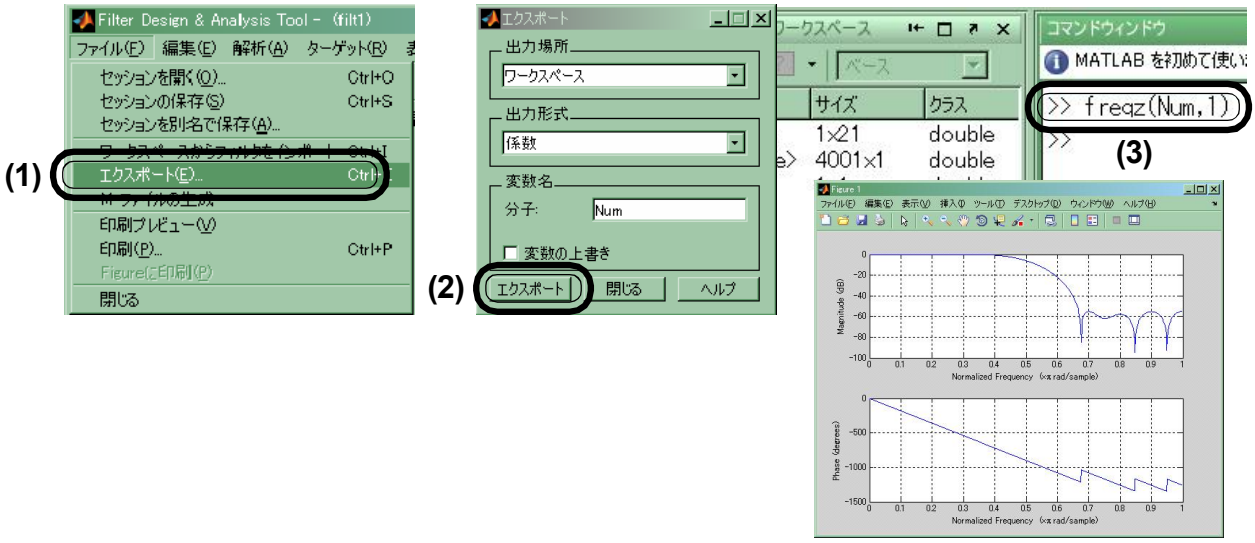



図 2-10　sptoolからワークスペースへの出力

②M-ファイルの出力

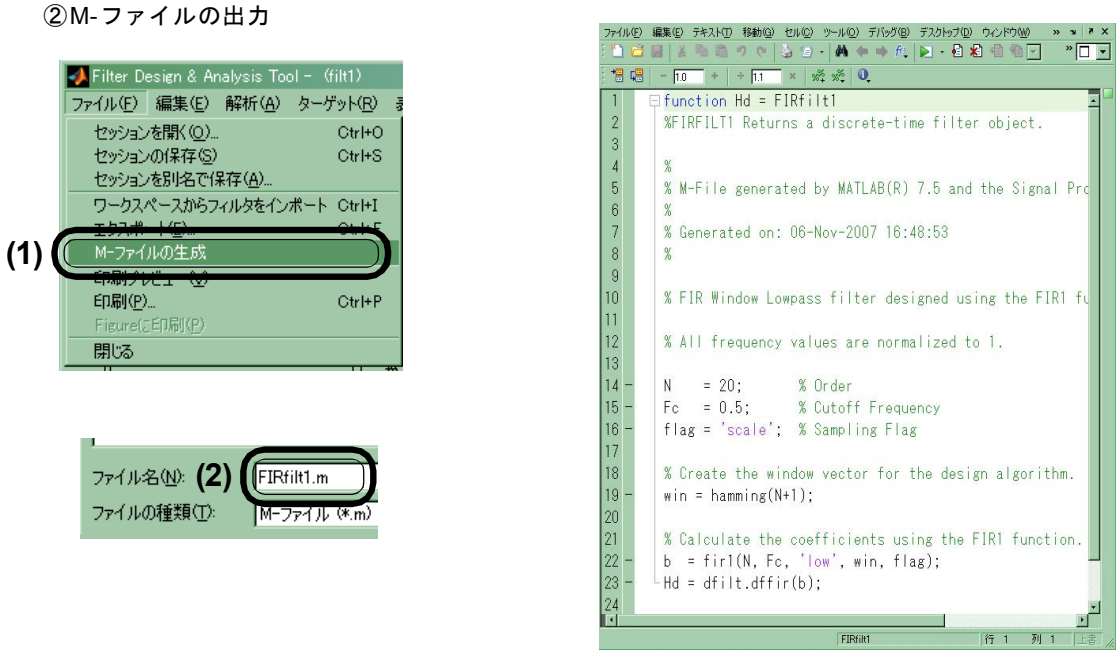



図 2-11　sptool から M-ファイルの出力

## 補足：GUI（Graphical User Interface）の作成

先ほどsptoolを使ってフィルタを設計しましたが、このようなGUIツールをユーザ独自に構築することもできます。これにより、M-ファイルとして実行するのではなく、GUIツールとして操作性を向上させて解析を行うことができます。また、オプション製品であるMATLAB Compilerを利用することでスタンドアロンアプリケーション（*.exe）や共有ライブラリを作成することもできます。以下にGUI構築ツールの起動方法とサンプルGUIを示します。

①GUI構築ツールの起動方法

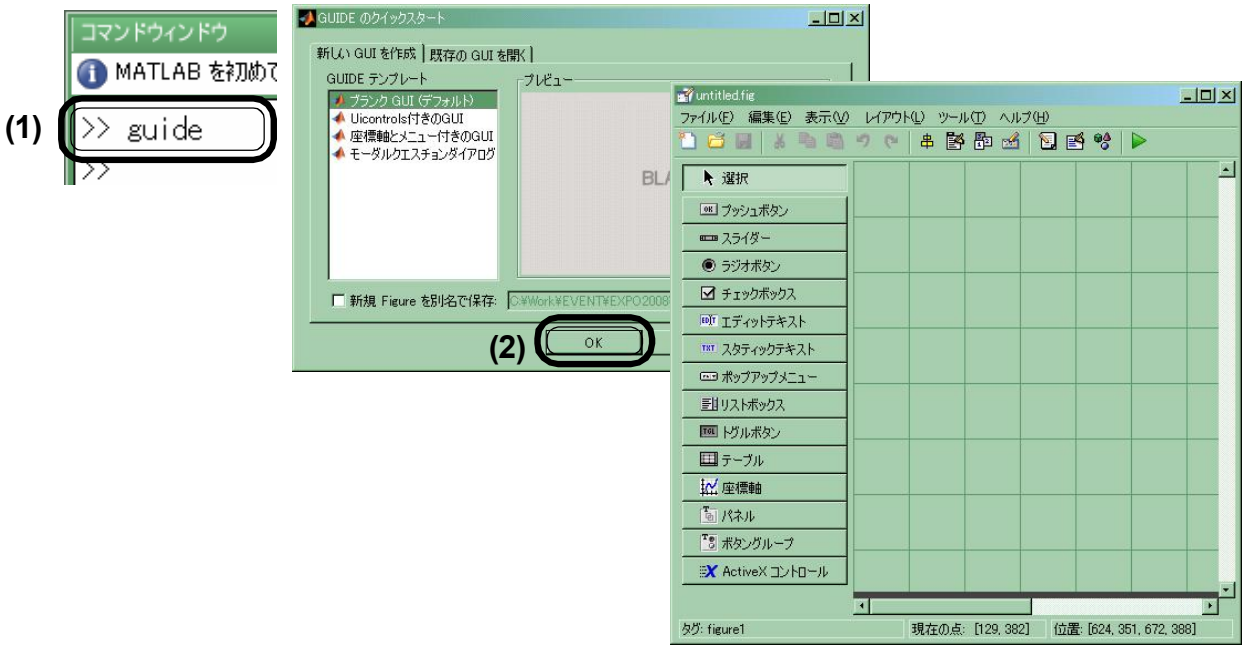



図 2-12 GUI構築ツールの起動

②サンプルGUI（主成分分析法による顔認識GUI）

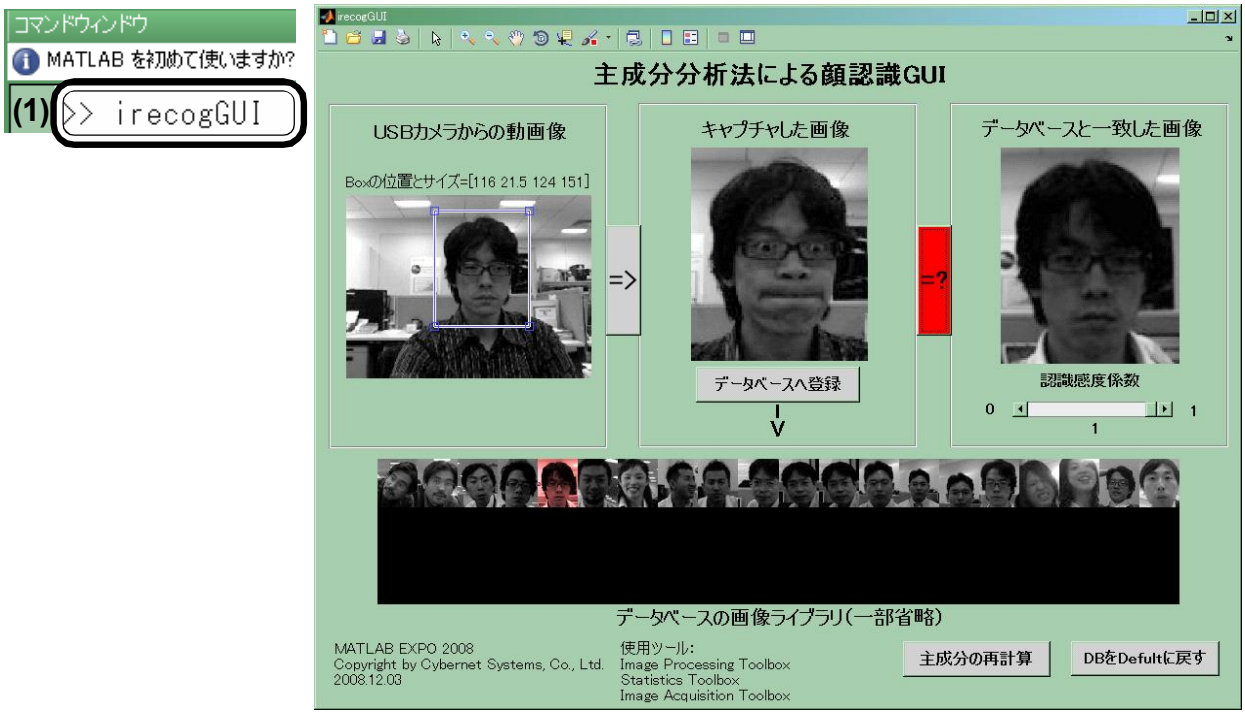



図 2-13 サンプルGUI（主成分分析法による顔認識GUI）の実行

# 第 4 章　マルチレート信号処理システム設計

この章では、Simulinkによるマルチレート信号処理システムの設計を取り上げます。以下はステージと実施手順です。


図4-1 マルチレート信号処理システム設計の実施手順とステージ

## 4.1　ダウンサンプリングによるレート変換

カレントフォルダ下に格納されているダウンサンプリングによるレート変換モデルファイル（rateconv_ver1.mdl）を起動し、シミュレーションを実行します。以下の手順を実行してください。

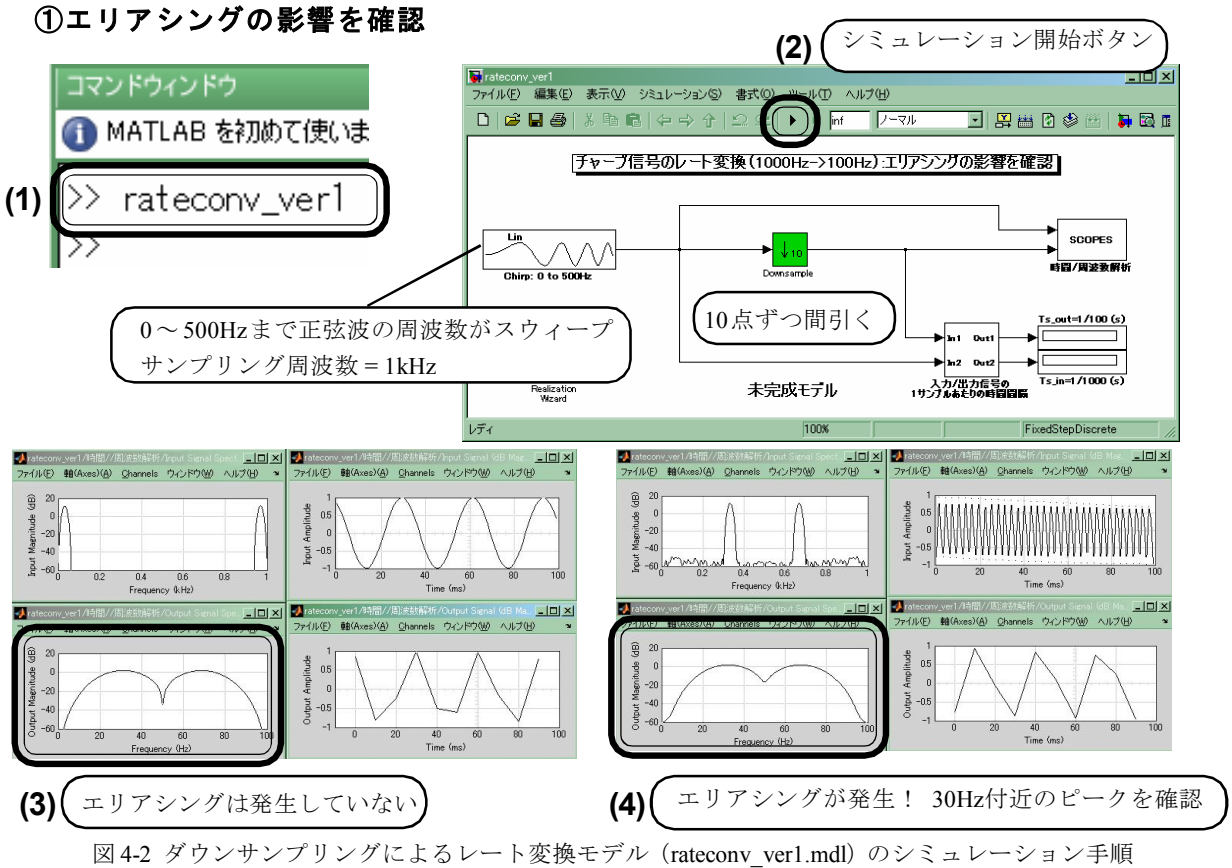



図 4-2 ダウンサンプリングによるレート変換モデル（rateconv_ver1.mdl）のシミュレーション手順

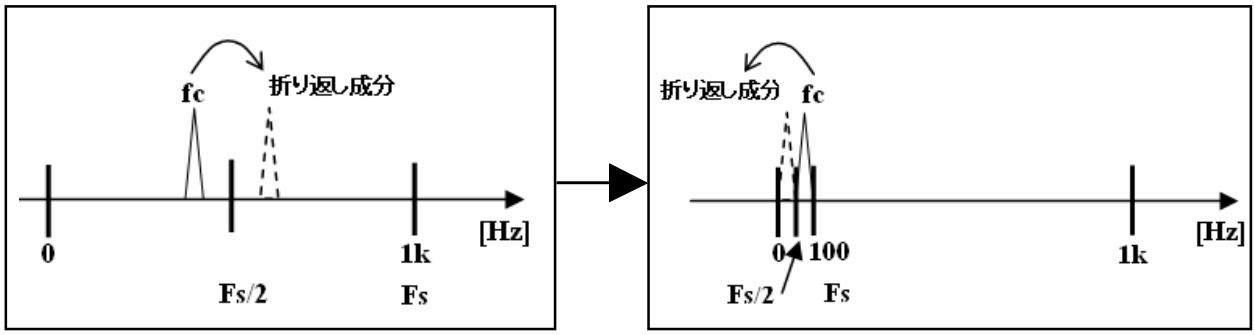



図4-3 ダウンサンプリングとスペクトルの関係

以上のシミュレーション結果より、ダウンサンプリングだけではエリアシングが発生してしまうためレート変換器として利用できないことが分かります。では、どのような処理が必要でしょうか？以上の図4-3を確認すると、今回の場合はエリアシング成分はナイキスト周波数(F2/s)以上の信号の折り返し成分であることが分かります。つまりエリアシング成分を発生させないようにするためには、ナイキスト周波数(Fs/2=100/2=50Hz)以上の信号を減衰させればよいことが分かります。次節では、帯域制限フィルタ（アンチエリアシングフィルタ）とダウンサンプラを組み合わせたシステム（直接構成によるデシメーション）のシミュレーションを行います。

## 4.2　直接構成によるデシメーション

まず、先のモデルの左下にあるフィルタ設計ツールを起動し、以下の仕様の帯域制限フィルタを設計します。以下の手順を実行してください。

### ①帯域制限フィルタの設計

| | | | |
|---|---|---|---|
| 応答タイプ | ＝ローパス | Passband 周波数(wpass) | ＝ 0.077(Fs で正規化) |
| 設計手法 | ＝ FIR 型(等リップル) | Stopband 周波数(wstop) | ＝ 0.12(Fs で正規化) |
| フィルタ次数 | ＝最小次数 | Passband のリプル(Apass) | ＝ 0.001 dB |
| | | Stopband の減衰量(Astop) | ＝ 70 dB |

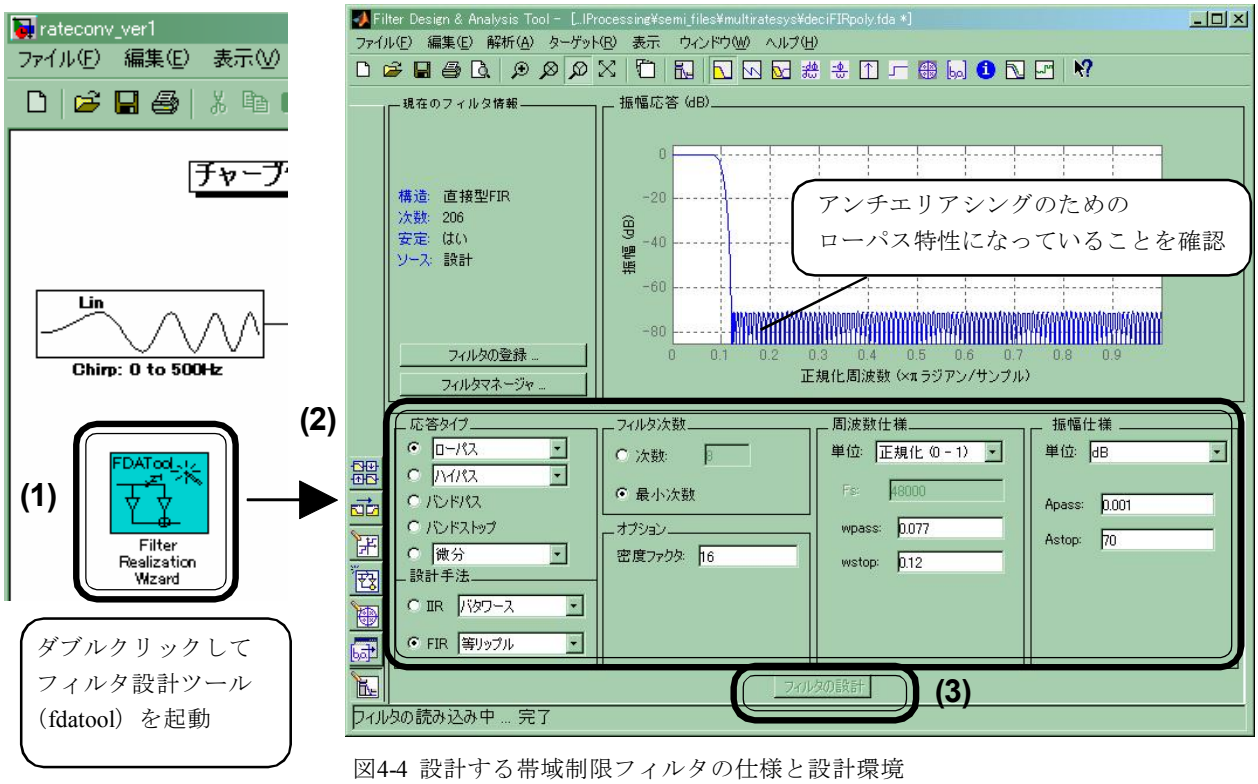



図4-4 設計する帯域制限フィルタの仕様と設計環境

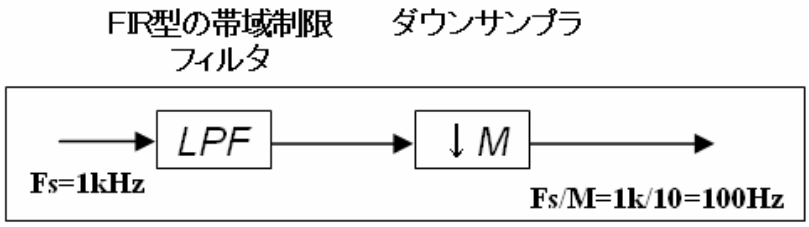



図4-5 設計するレート変換器の構成

次に、以上で設計した帯域制限フィルタからSimulinkモデルを自動生成した後、図4-5と等価なSimulinkモデルを作成しシミュレーションを実行します。以下の手順を実行してください。

### ②アンチエリアシングフィルタの効果を確認

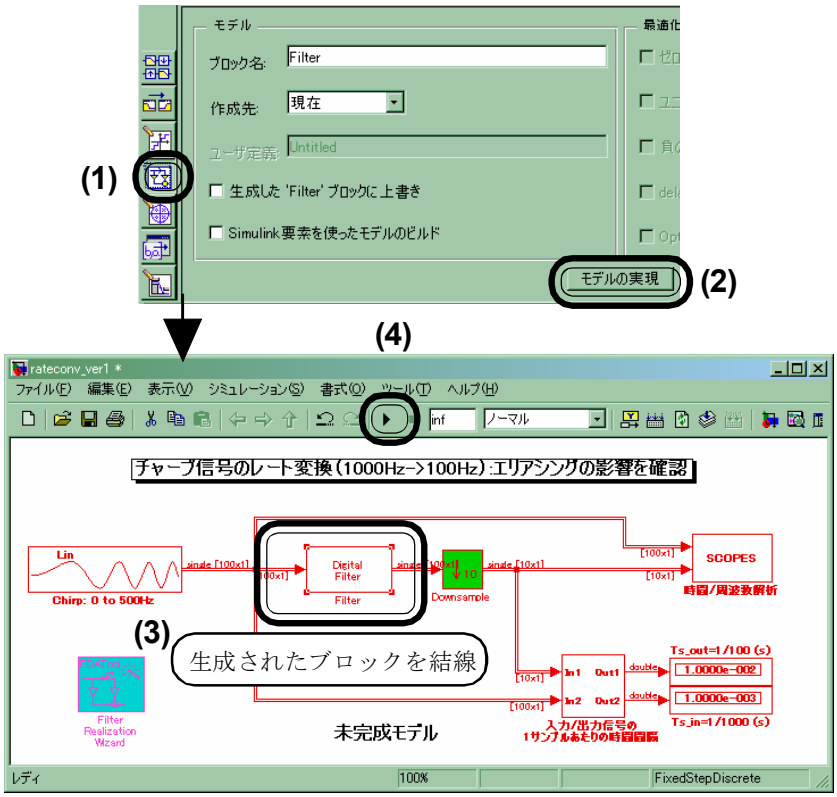



図4-6 設計した帯域制限フィルタを組み合わせたSimulinkモデル

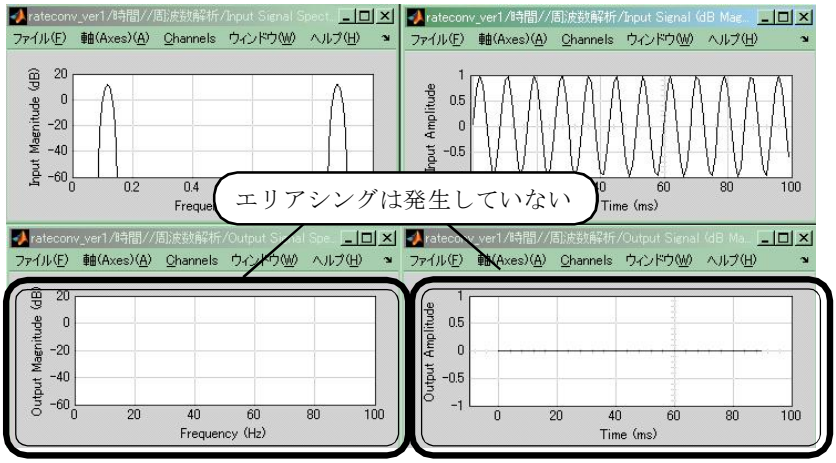



図4-7 レート変換モデル（直接構成によるデシメーション）のシミュレーション手順

以上のシミュレーション結果より、設計した帯域制限フィルタが折り返し成分を減衰している様子を確認することができます。

## 4.3　サウンドカードからのデータ取得

今まではファイル経由でデータをMATLABやSimulink上に取り込みましたが、サウンドカードやデータ収集ボード、計測器/測定器から直接データを取り込むことができます。これにより、実データを使用した解析/シミュレーション/検証作業が容易になります。ここではサウンドカードのサンプリング周波数9600Hzから960Hzにレート変換(ポリフェーズ分解によるデシメータ)するモデル（rateconv_ver4.mdl）を取り上げます。なお、本シミュレーションでは時間が変化するにつれて信号の周波数成分がどのように変化するか？を確認するためにスペクトログラムも表示させています。

### ①リアルタイム信号の確認

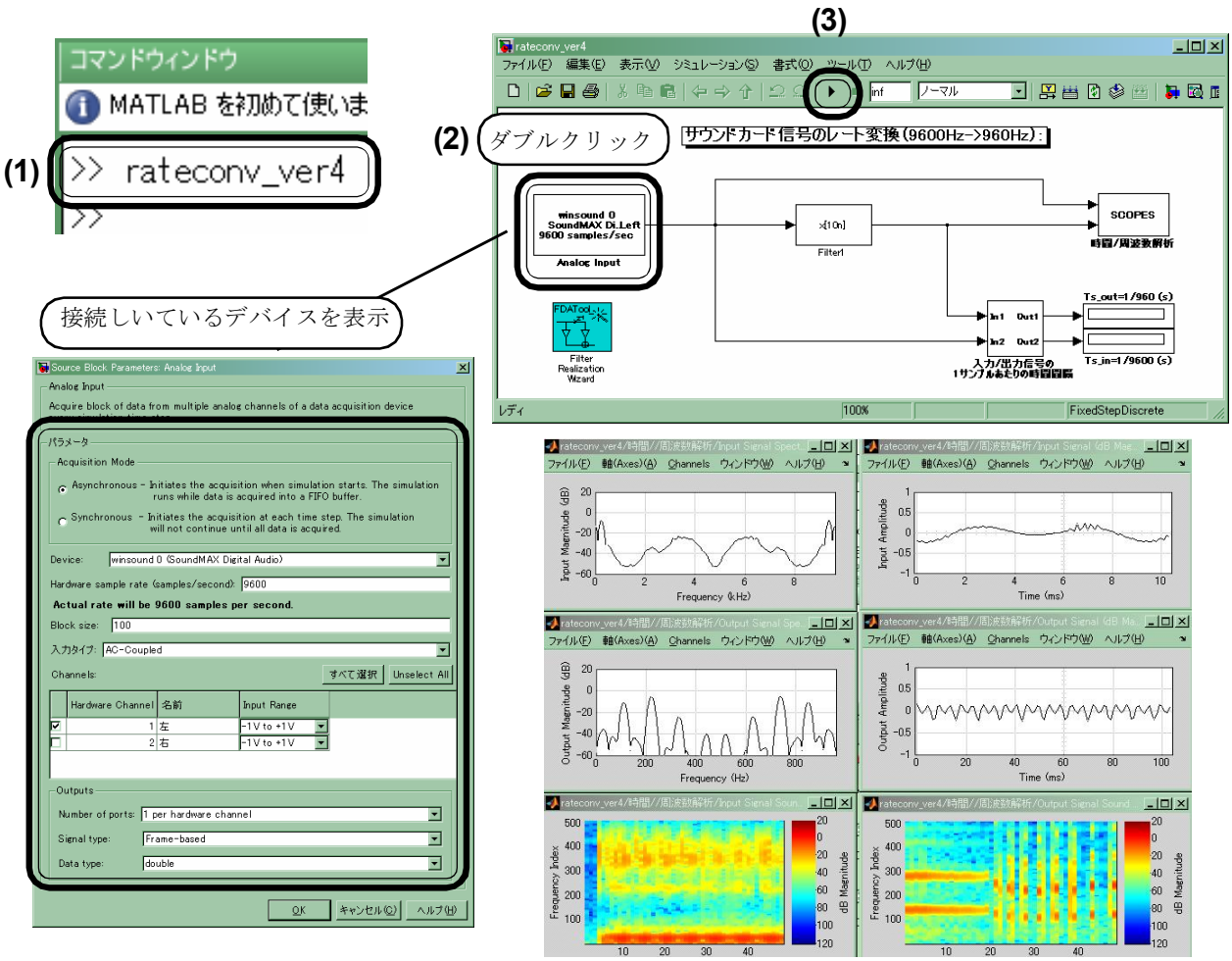



図4-8 サウンドカードからのデータ取得モデル（rateconv_ver4.mdl）のシミュレーション手順

サウンドカード等のハードウェア経由でデータを取得する場合はData Aquisition Toolbox（以下DAQ）が必要となります。現在のサポートしている主要メーカは以下になります。ご使用のハードウェアがサポートされているかどうかについては次の開発元webをご参照ください（http://www.mathworks.com/products/daq/supportedio.html）。

| | |
|---|---|
| ・DAQサポートメーカ | IOTech |
| Acqiris | Keithley Instruments |
| ADLINK | Measurement Computing |
| Advantech | National Instruments |
| CONTEC | sound cards |
| Data Translation | United Electronic Industries |
| g.tec | ... etc |

**Point 6　＜ディジタル・アナログ混在システムシミュレーションの重要性＞**

Simulinkは、解析的に解くことが困難とされる非線形要素を含むシステムやアナログ要素（S領域）とディジタル要素（Z領域）が混在したシステムを直感的に表現し、時間軸上のシミュレーションを行うことができます。特に近年、複雑化・高度化するディジタル・アナログ混在システムの設計・検証においては、Verilog AMS/Spiceレベルのシミュレーションだけでなく、システムレベルの検証を十分に実施し、事前にパラメータの最適値や感度を把握することがますます重要になってきています。

## 補足：　マルチレートの多段フィルタの設計

前節のレート変換の例題では、一つのフィルタで設計しましたが、実際のハードウェア設計を考慮すると、回路面積を削減するために多段のマルチレートフィルタが用いられるのが一般的です。先ほど取り上げたフィルタ設計ツール（fdatool）では複数のディジタルフィルタをカスケード接続し、フィルタ全体の応答解析を行うことが可能です。以下の手順で多段フィルタ設計のサンプルを確認します。

①fdatoolのセッションを開く

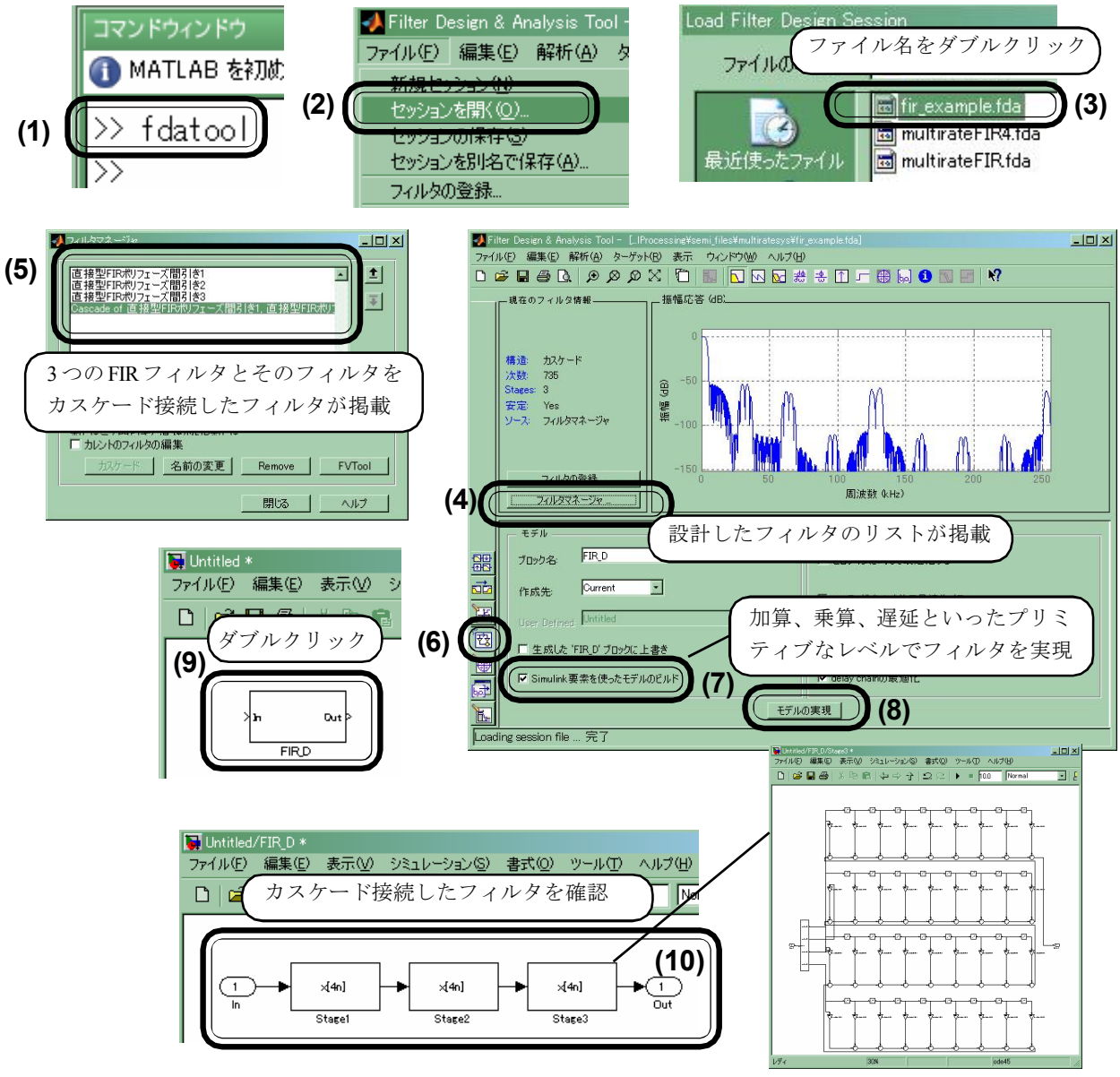
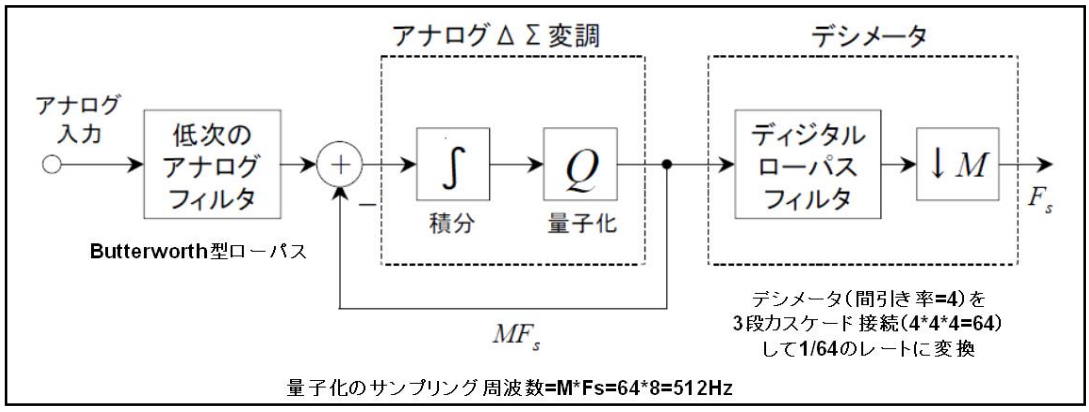



図4-9 マルチレートフィルタのサンプルセッション（fir_example.fda）の起動手順

## 補足： デジタル・アナログ混在システムのシミュレーション

デジタル・アナログ混在システムの例として、シグマデルタA/D変換器を取り上げます。シグマデルタ（Σ Δ）変調器は、リニアリティ、アンチエリアシング特性とアナログ実装への要求の低さから、低コストで高分解能が必要なアプリケーションのA/Dコンバータとして利用されています。特に、オーバーサンプリング技術を使うことにより、緩やかなアナログLPF特性で高いパフォーマンスを実現できるという特徴を持ちます。以下に概念図とモデルの実行手順を示します。




①Σ Δ変換器モデルのシミュレーション

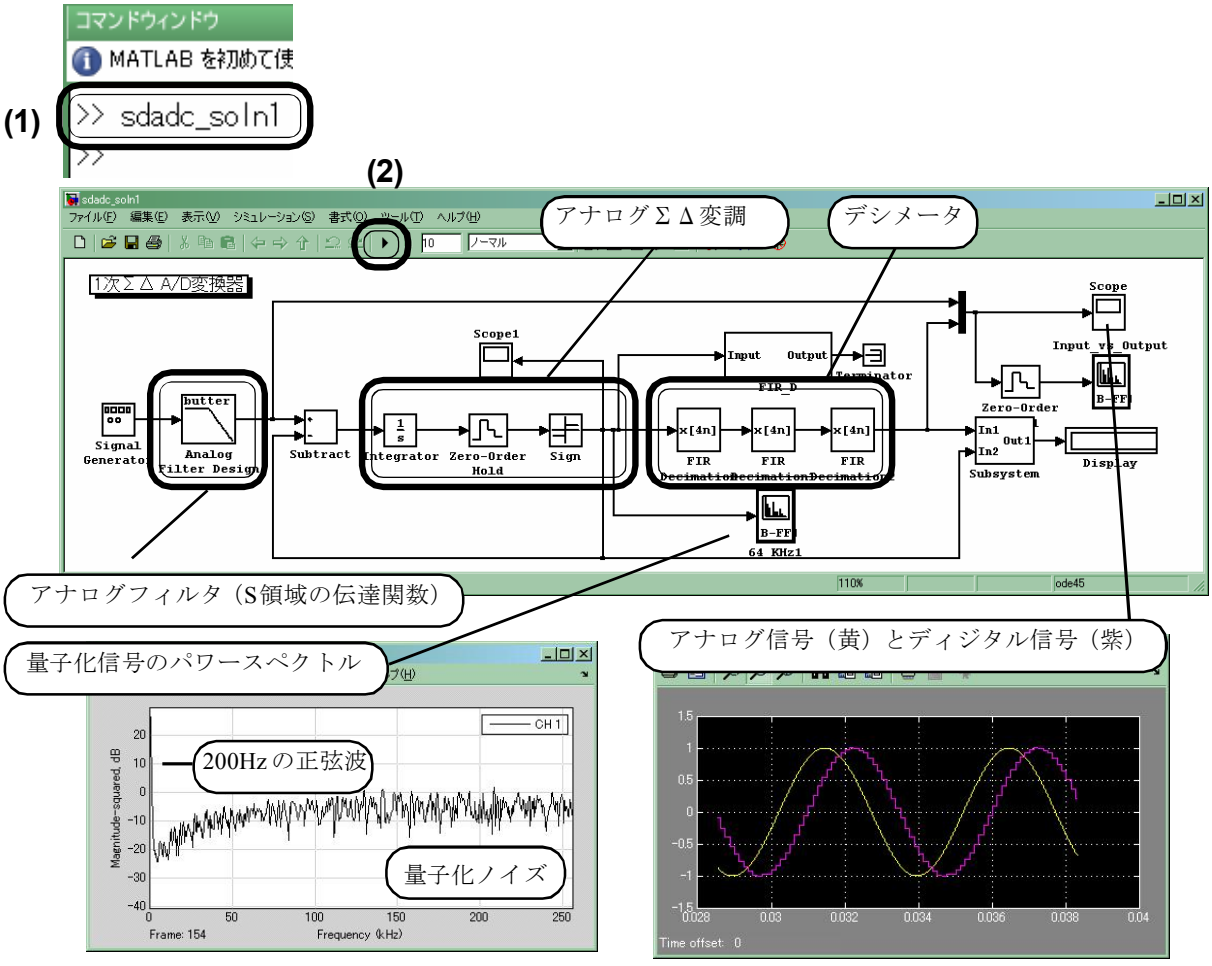


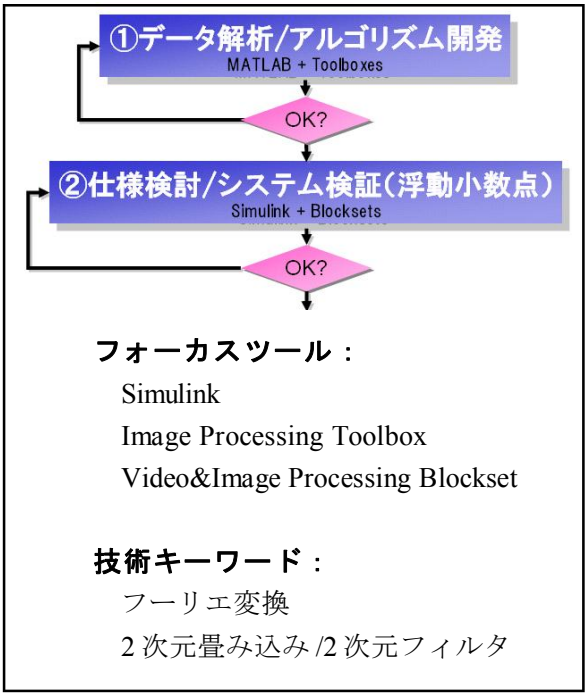

図4-10 Σ Δ変換器モデルのシミュレーション手順とモデル概念図

なお、テキストでは取り上げませんが、カレントディレクトリ内には以下のサンプルも含まれています。

・2次Σ Δ A/D変換モデル（sdadc2ord_soln.mdl）, アクティブ方式 Butterworth LPF（butterlp.mdl）
・Fractional-N（分数分周）方式の周波数シンセサイザ（fractional_4.mdl, fractional_6.mdl）

# 第 5 章　フーリエ変換の性質

この章では、フーリエ変換の性質である、「時間領域の畳み込みと周波数領域の積は等価」を確認します。以下はステージと実施手順です。



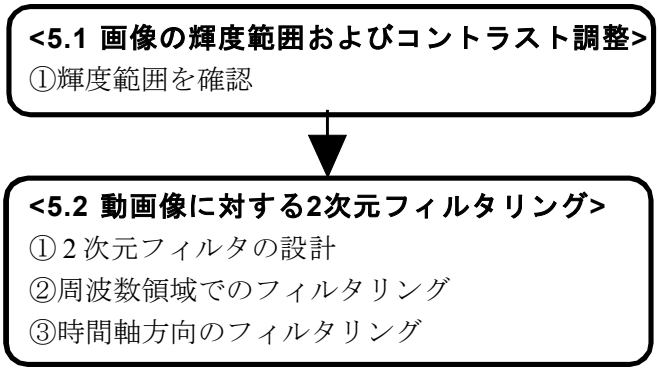




図5-1フーリエ変換の性質の実施手順とステージ

## 5.1　画像の輝度範囲およびコントラスト調整

カレントフォルダ下に格納されている画像の輝度範囲を検証するための M-ファイル（im1.m）を実行し、解析対象となる画像の輝度範囲を確認します。以下の手順を実行してください。

**①輝度範囲を確認**

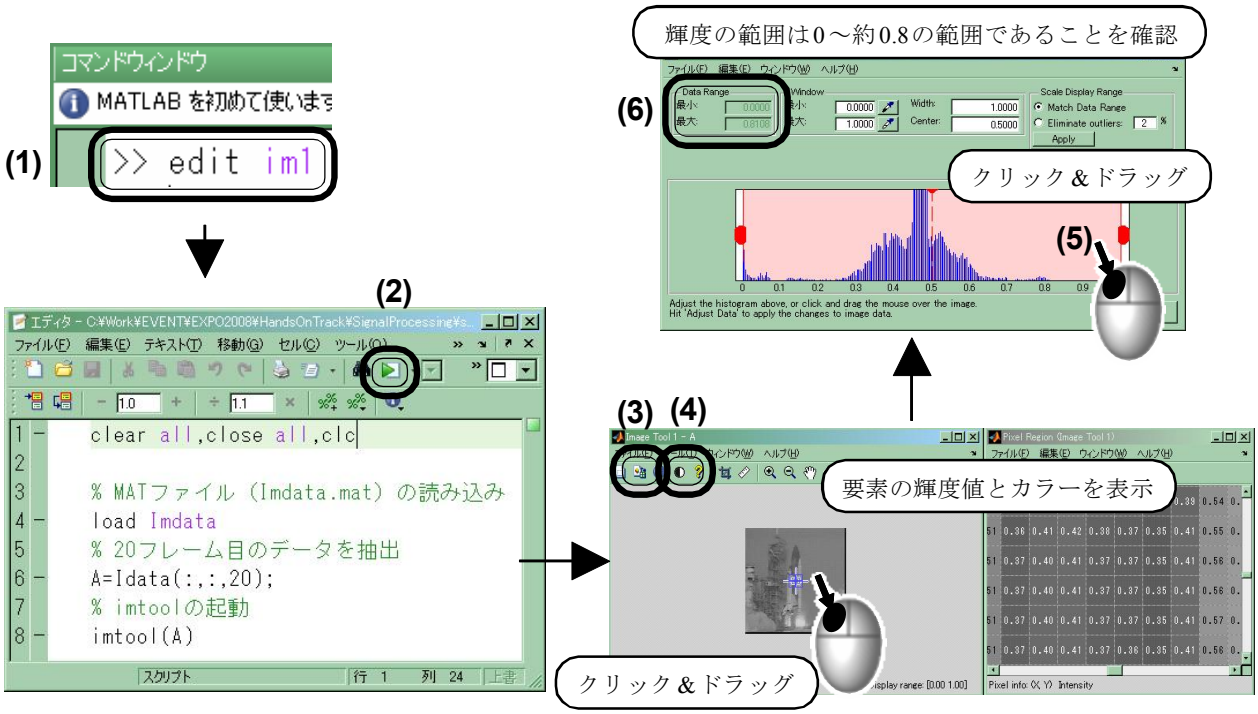



図 5-2 輝度範囲検証用の M-ファイル（im1.m）の実行手順

## 5.2　動画像に対する 2 次元フィルタリング

ここでは、主に2種類のフィルタを適用します。1つ目は、動画像を1枚毎に対して2次元の係数行列を畳み込む方法、2つ目は、動画像の要素に対して時間軸方向にフィルタ係数を畳み込む方法です。以下に概略図を示します。ここではベンチマークとして画像のサイズを 128 × 128 とします。

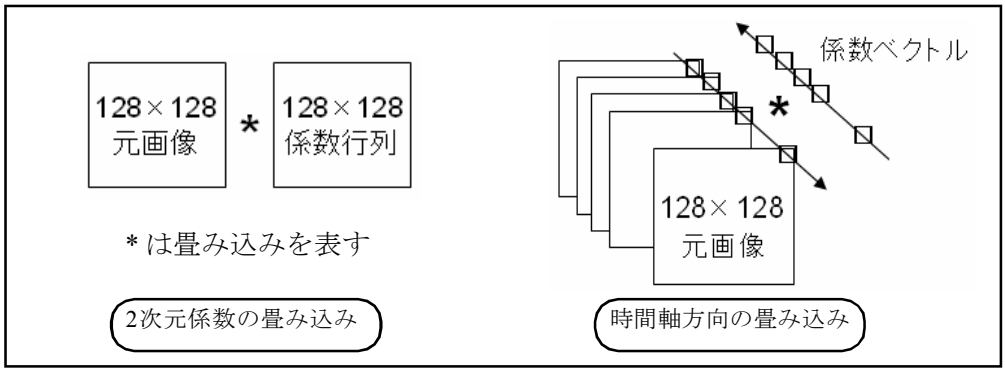



図5-3 2次元フィルタリングの手法

なお、以上で述べた畳み込みは線形システムの応答を時間領域で表現するものですが、この時間領域の畳み込みをフーリエ変換すると以下の関係が得られることが知られています。結論から示すと時間領域で畳み込みとして表現された線形システムの入出力の関係が、周波数領域では積の形で表現できることを表しています。特に2次元信号（画像）の場合は直接的に畳み込みを行うと演算量が膨大になるため、2 次元 FFT による畳み込みは重要な要素となります。

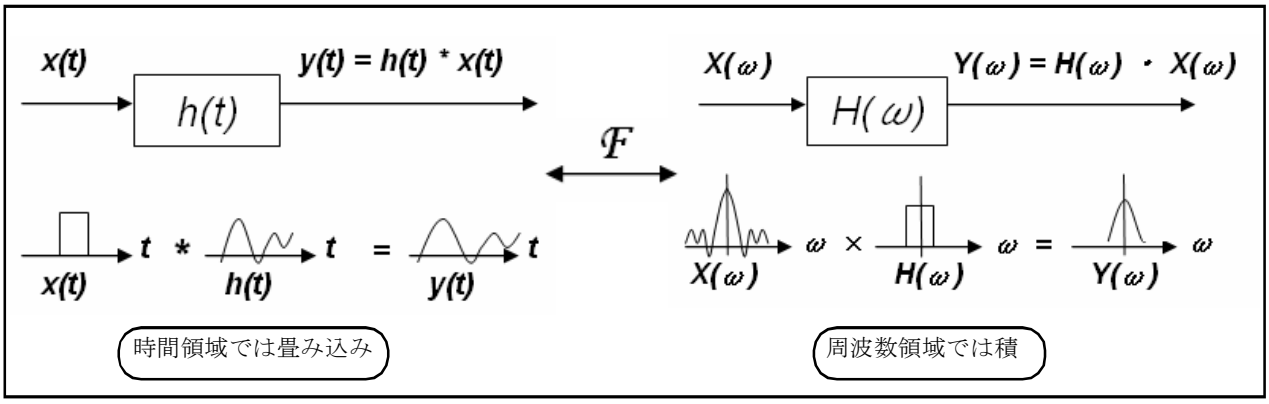



図5-4 フーリエ変換の性質

次節では、以上のフーリエ変換の性質を確認するために、以下のモデリング手法で実現し応答の比較を行います。また、図 5-3 に記載されている２つの手法の違いについても確認します。

・2 次元係数の時間領域での畳み込み　　= Video&Image Processing Blocset
・2 次元係数の周波数領域での畳み込み　= Signal Processing Blockset
・2 次元係数の周波数領域での畳み込み　= Embedded MATLAB Function(M- ファイル)
・時間軸方向の畳み込み　　= Signal Processing Blockset

２次元フィルタの設計においては、まず、1 次元フィルタの係数を計算し、次に変換行列を使って 2次元フィルタの係数を求めます。シミュレーションにおいては高周波ノイズのフィルタリングを行うことを想定しているため、設計するフィルタは以下の仕様に従うものとして設計します。以下の手順を実行してください。

### ①2次元フィルタの設計

| | |
|---|---|
| 応答タイプ | = ローパス |
| 設計手法 | = FIR 型（最小二乗線形位相） |
| 次数 | = 12 |
| PassBand 周波数 | = 0.5(Fs/2 で正規化した場合) |
| StopBand 周波数 | = 0.55 |

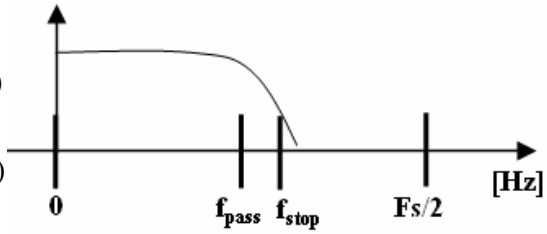



図5-5 設計するディジタルフィルタの仕様

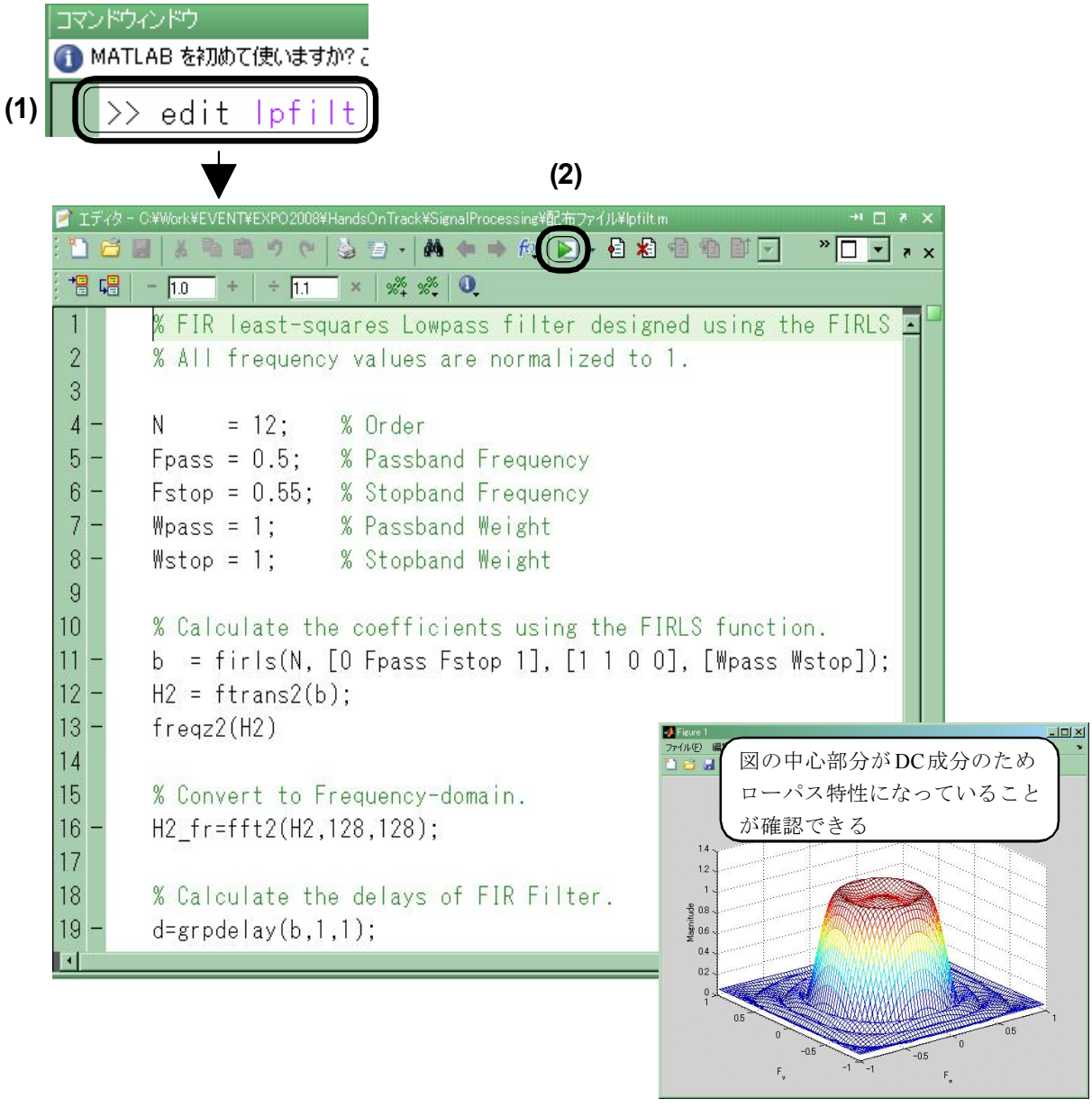



図 5-6 2 次元フィルタの係数を計算する M-ファイル（lpfilt.m）の実行手順

次に、動画像に対する２次元フィルタリングモデル（imagefilt_ver1.mdl）を起動します。このモデル内のEmbedded MATLAB Functionブロックにより周波数軸上での畳み込みを実現していますが、すべてのコードが記述されていません。こちらにコードを追加する必要があります。以下の手順を実行してください。

**②周波数領域でのフィルタリング**

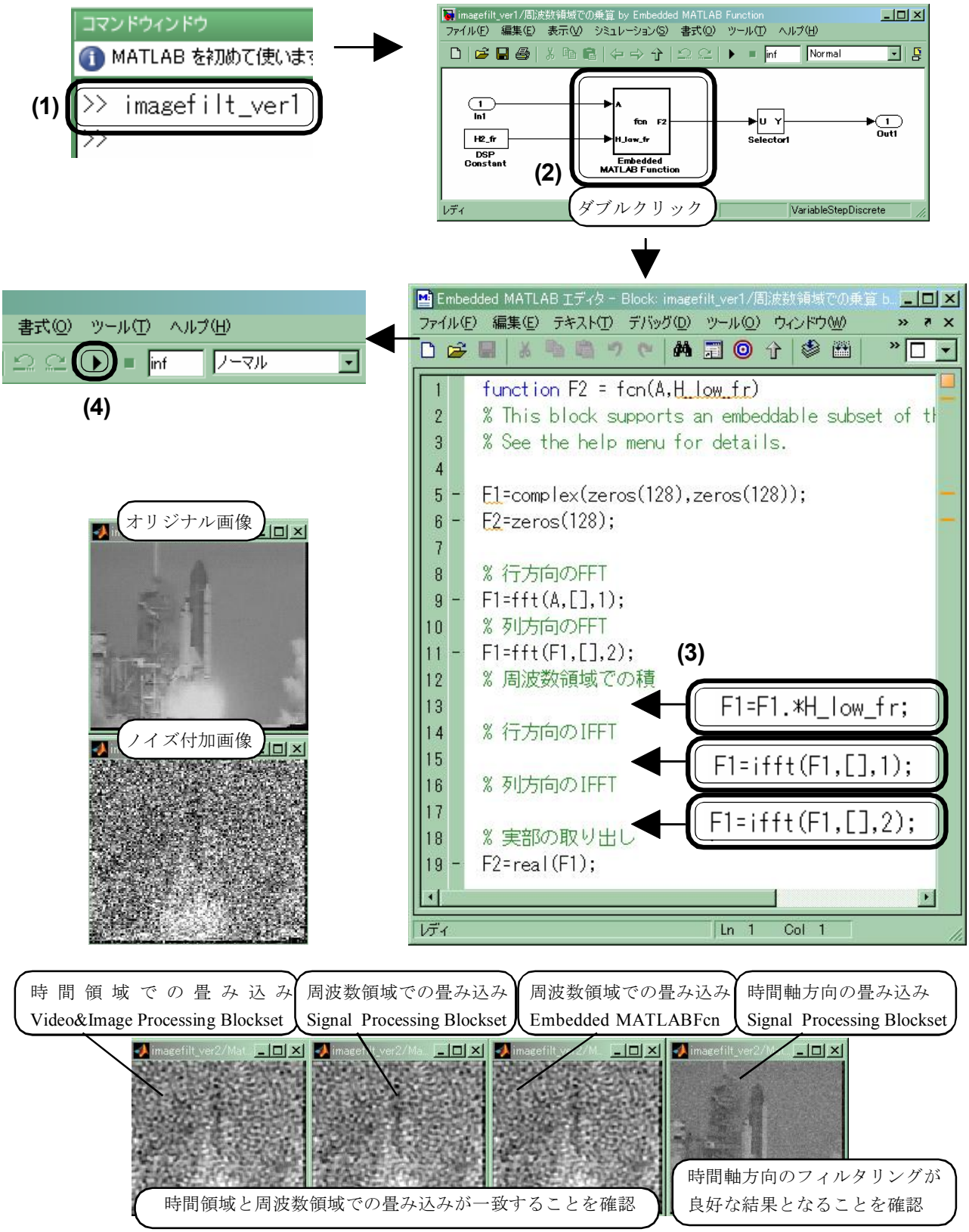



図 5-7 動画像に対する２次元フィルタリングモデル（imagefilt_ver1）のシミュレーション手順

**Point 7　＜SimulinkとM-ファイルとの協調性＞**

Simulink で提供される Embedded Function ブロックによりアルゴリズム開発の段階で検証した資産をシステム設計のフローで容易に流用することができます。なお、このブロックで記述したM-ファイルはシミュレーション前に一度、Cコード生成されコンパイル・リンクを実行します。このため、MATLAB Fcnブロックで M-ファイルをコールするよりも高速にシミュレーションを行うことができます。SimulinkはMATLAB環境との協調により、柔軟かつ自由度の高い解析が可能です。

## 補足：Embedded MATLAB Functionブロックでサポートされる関数リスト

Embedded MATLAB Function ブロックは多数の数学関数をサポートしています。サポートしている関数リストを表示するためには以下の手順を実行します。

①Embedded MATLAB Functionブロックのサポート関数リストの表示方法

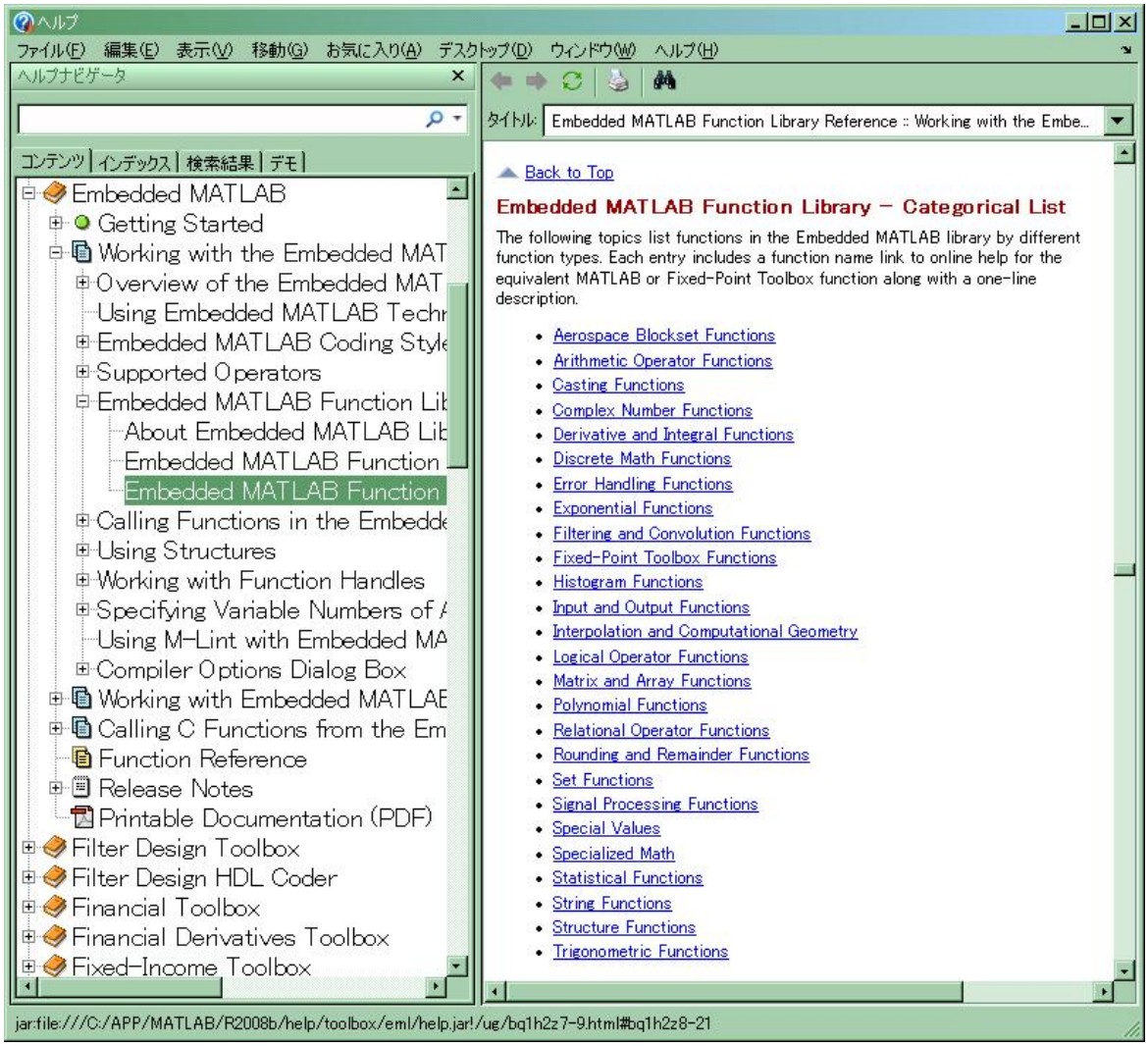




図 5-8　Embedded MATLAB Function ブロックの対応関数リストの表示手順

# 第 6 章　固定小数点ディジタル信号処理システム設計

この章では、前章で取り上げた時間軸方向の畳み込みを行うFIRフィルタを浮動小数点演算させた時と固定小数点演算させた時で画像がどの程度劣化するかを確認するモデルを取り上げます。また、本モデルでは固定小数点モデルに対してオートスケーリングを行い、特性改善を行います。以下はステージと実施手順です。

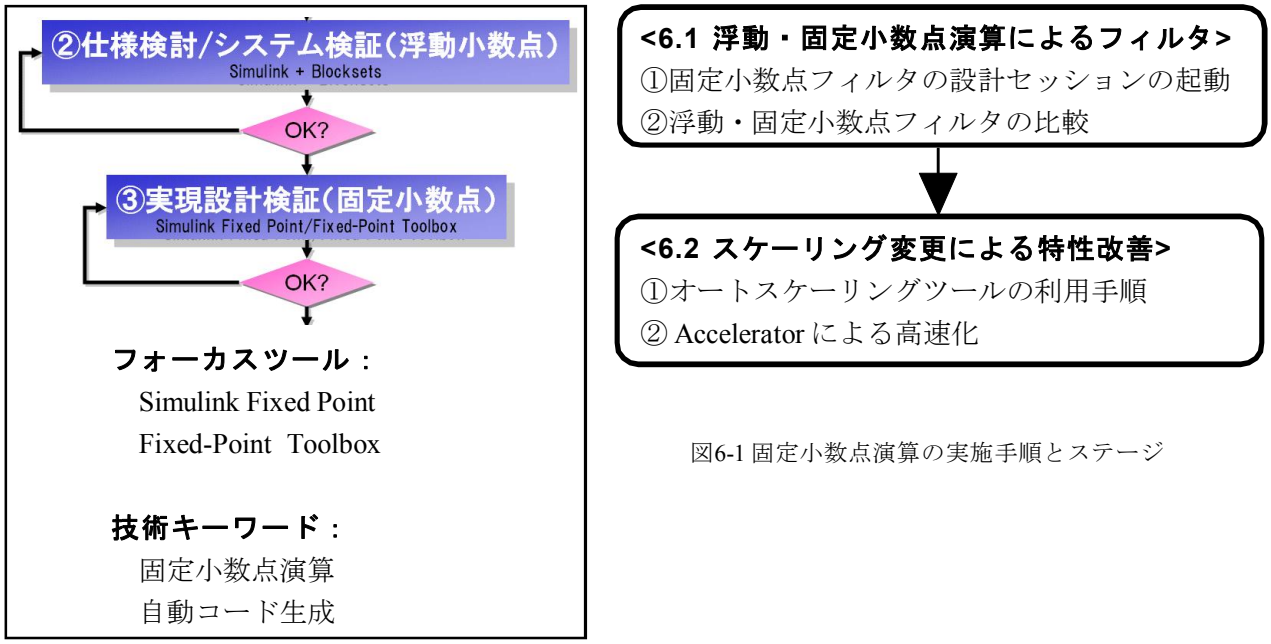
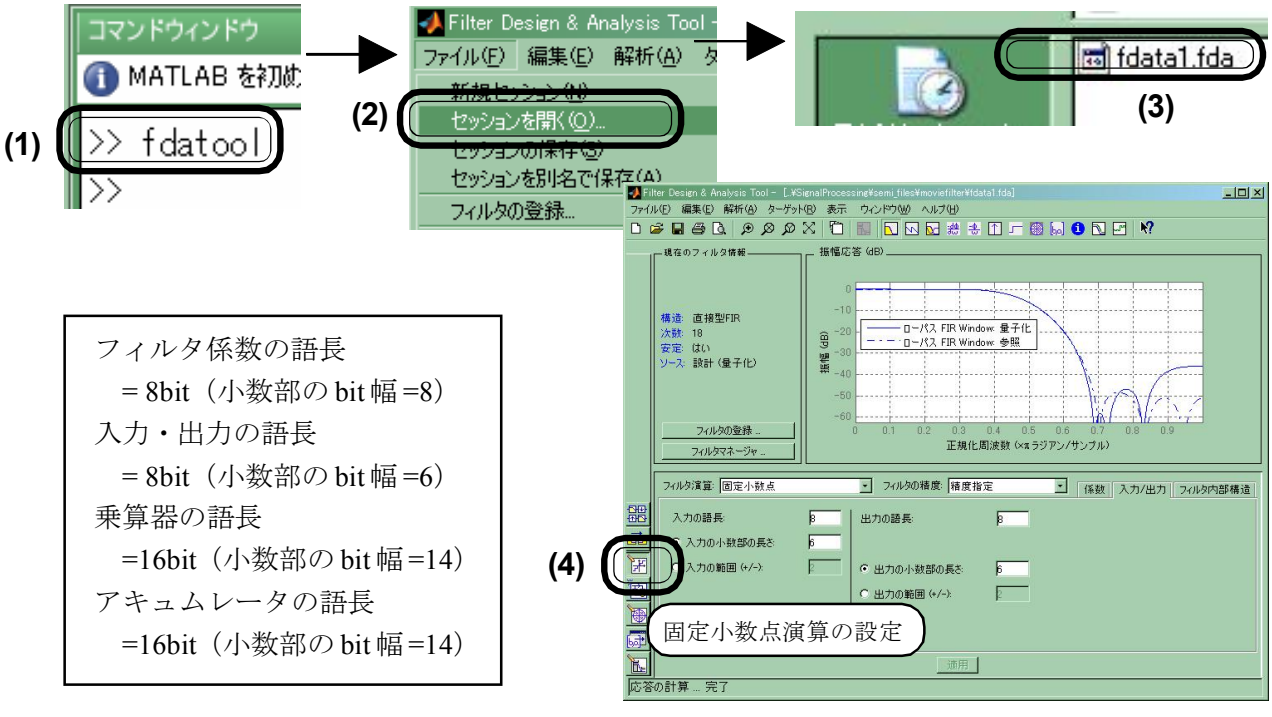



図6-1 固定小数点演算の実施手順とステージ

## 6.1　浮動・固定小数点演算によるフィルタ

カレントフォルダ下に格納されている固定小数点フィルタの設計セッションを開き、固定小数点演算の仕様を確認します。以下の手順を実行してください。

### ①固定小数点フィルタの設計セッションの起動

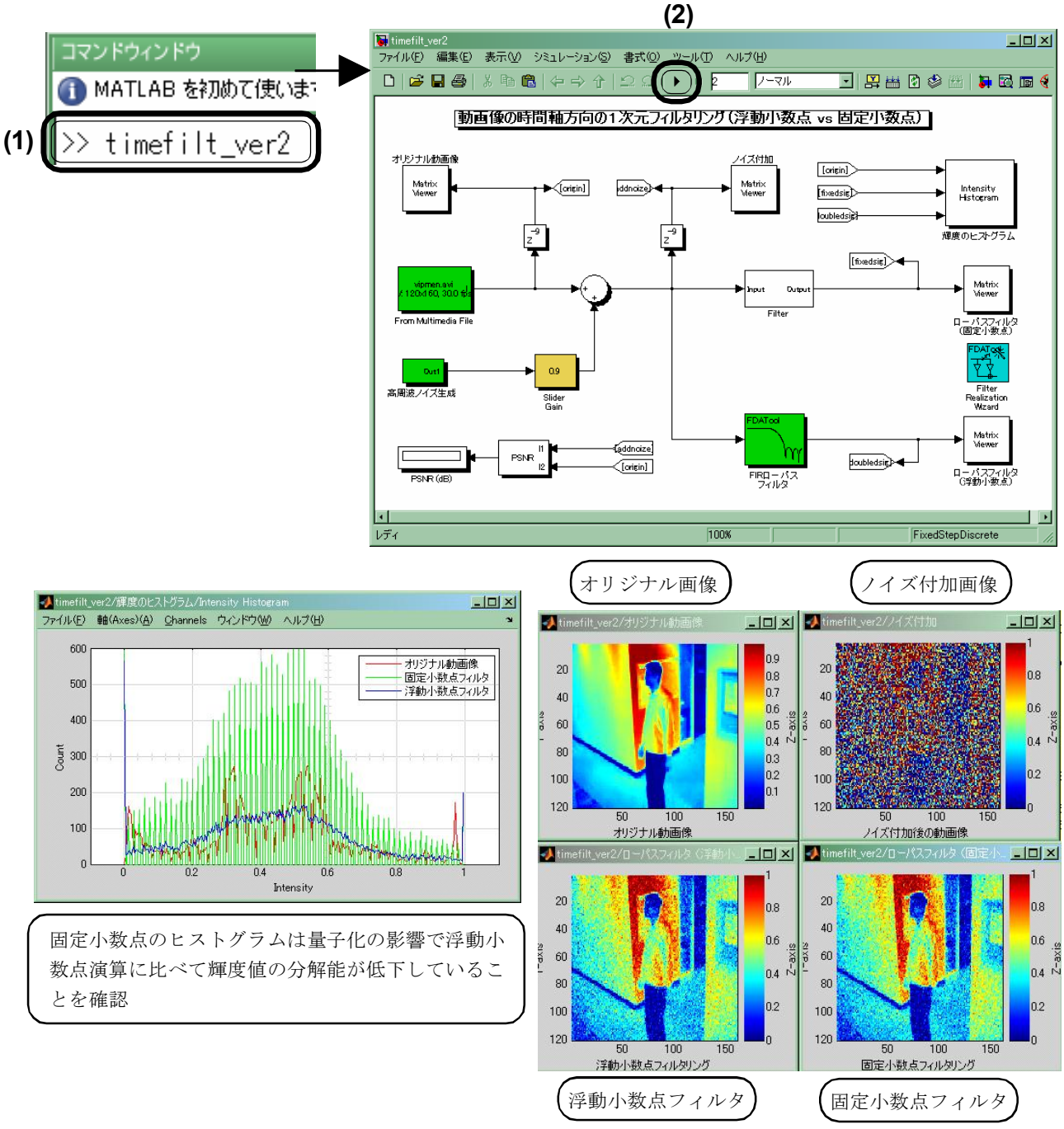



フィルタ係数の語長
　＝8bit（小数部の bit 幅＝8）
入力・出力の語長
　＝8bit（小数部の bit 幅＝6）
乗算器の語長
　＝16bit（小数部の bit 幅＝14）
アキュムレータの語長
　＝16bit（小数部の bit 幅＝14）

図 6-2 固定小数点フィルタの設計セッション（fdata1.fda）の起動手順

前節で設計されている固定小数点フィルタからSimulinkブロックを自動生成して、システム全体を構築したモデル（timefilt_ver2.mdl）を起動します。次に、シミュレーションを実行して固定小数点演算と浮動小数点演算の結果を比較します。以下の手順で実行してください。

### ②浮動・固定小数点フィルタの比較




固定小数点のヒストグラムは量子化の影響で浮動小数点演算に比べて輝度値の分解能が低下していることを確認

図 6-3 固定小数点フィルタリングモデル（timefilt_ver2.mdl）のシミュレーション手順

シミュレーションを実行すると以下のメッセージがコマンドウィンドウ上に表示されます。これは、ブロックに入力される信号が固定小数点化の影響により飽和していることを警告しています。次節ではこの飽和が起きないように自動的にスケーリングポイントを決定します。

```
警告: 飽和が発生しました。. This originated from 'timefilt_ver2/Filter/ConvertIn'.
>>
```

## 6.2　スケーリング変更による特性改善

前節で使用した固定小数点フィルタは画像を見た限りでは浮動小数点フィルタと大きな差は見られませんが、一部のブロックで飽和が発生しています。ここでは、スケーリングポイントの自動変更を行います。以下の手順を実行してください。

### ①オートスケーリングツールの利用手順

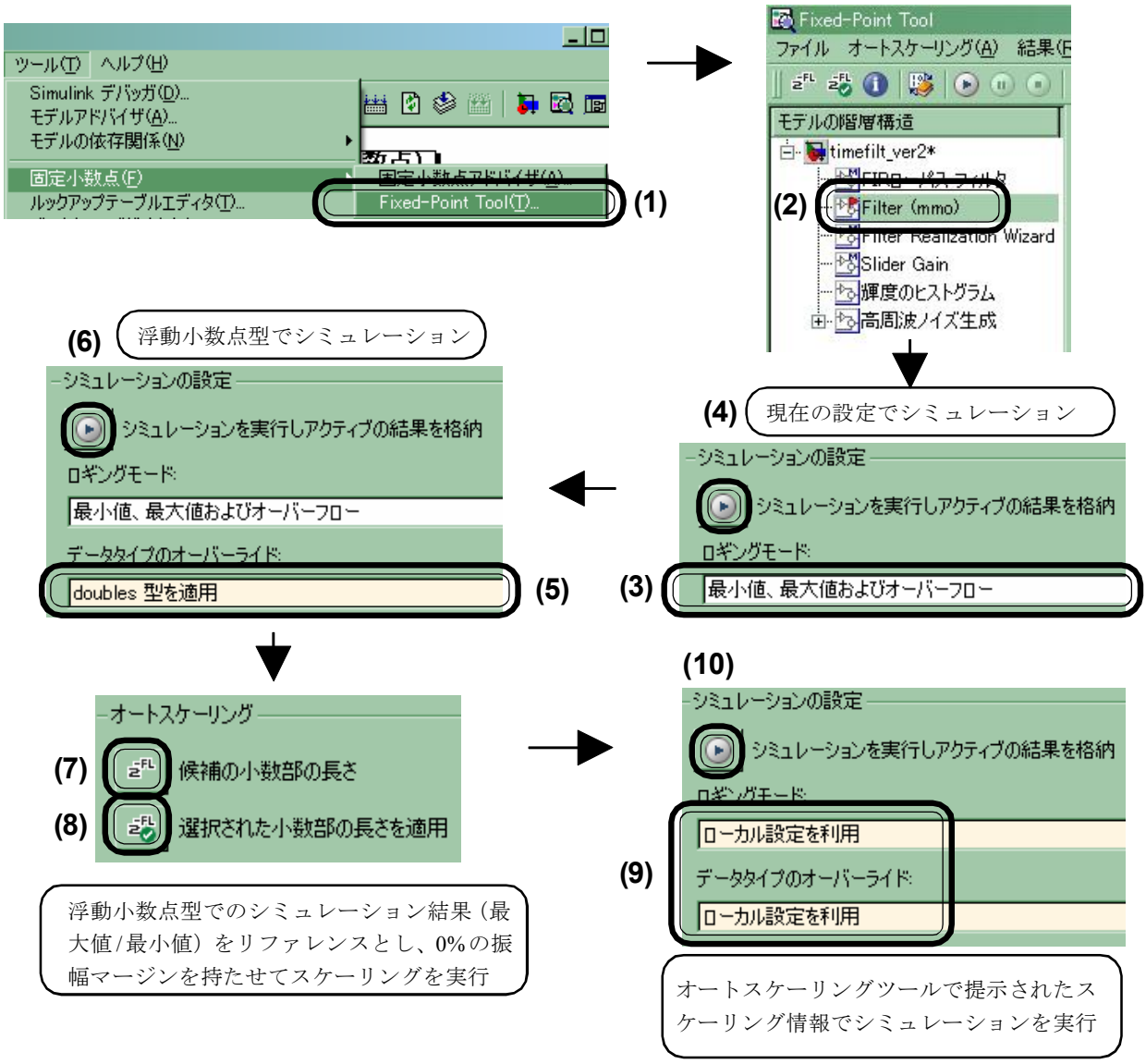



図6-4 オートスケーリングツールの利用手順

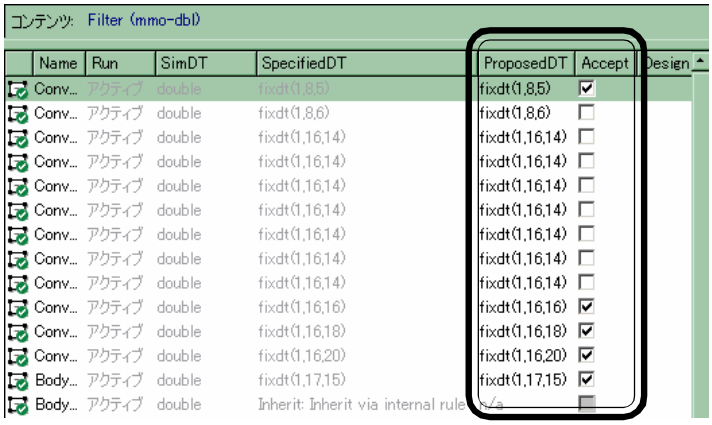



図6-5 オートスケーリングツールで提示されたスケーリング情報

| | データタイプ | 出力スケーリング値 | 信号線の表示 | 概念図と値（符号ビット |
|---|---|---|---|---|
| ① | sfix(10) | 2^-7 | sfix10_En7 | −2.25 |
| ② | ufix(8) | 2^-5 | ufix8_En5 | 5.5 |
| ③ | sfix(10) | 2^2 | sfix10_E2 | 16 |

図6-6　Simulinkの固定小数点データタイプの例

次に、前節で取り上げたシステムを高速にシミュレーションするためのモデル（timefilt1_ver4.mdl）を起動し、シミュレーションを実行します。以下の手順を実行してください。

**②Acceleratorによる高速化**

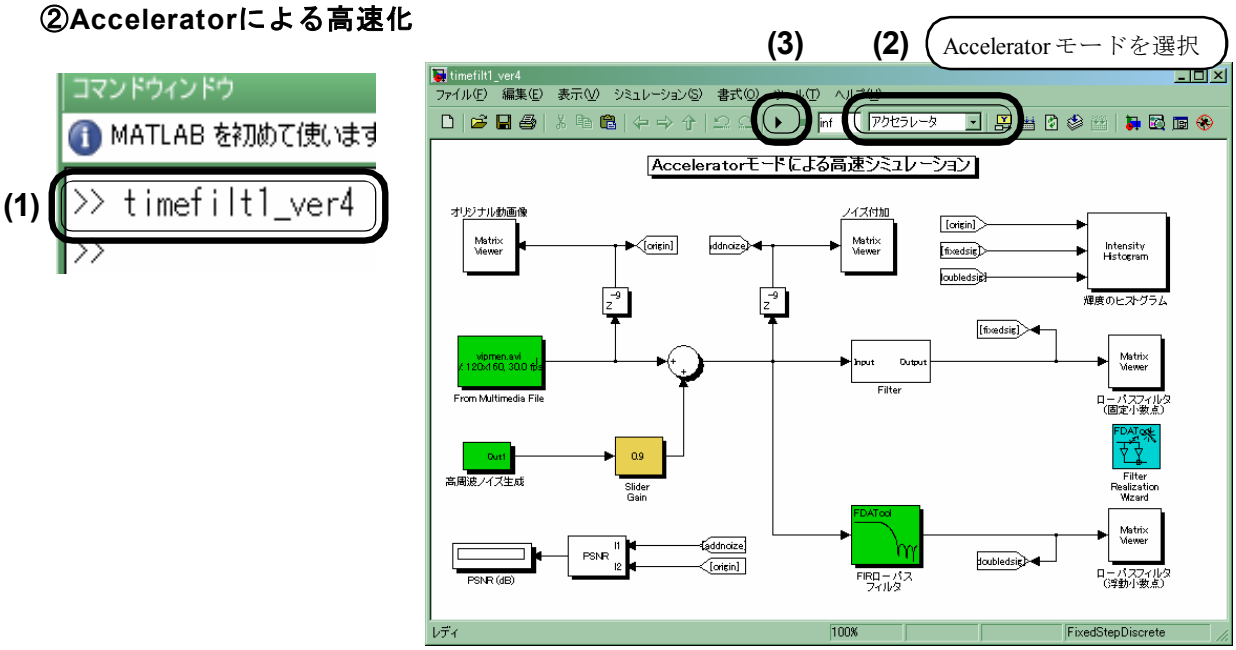



図 6-67 高速シミュレーションモデル（timefilt_ver4.mdl）の実行手順

シミュレーションボタンを押すとモデル全体からCコードを生成し、コンパイル・リンクを行います。シミュレーション時にはコンパイル後のモデル（拡張子=.mexw32）が実行されるため高速にシミュレーションを行うことが可能です。

**Point 8　＜固定小数点システムの評価・検証＞**

Simulink Fixed Point や Fixed-Point Toolbox を使用することで固定小数点化や、最適スケーリングの検証までを短期間で行うことが可能です。これまで取り上げてきたMATLABおよびSimulink環境を共通プラットフォームとしてアルゴリズム設計から固定小数点化までを強力にサポートします。

**Point 9　＜自動コード生成機能＞**

Simulink 環境で作成したモデルは自動コード生成ツールにより組込み用途の C コードや HDL（Verilog/VHDL）コード（Filter Design HDL Coder/Simulink HDL Coder）に変換することができます。また、生成されたコードを 3rd Party 製品により特定のターゲットハードウェアに実装したり、各種シミュレータとコ・シミュレーションすることで設計フローを分断することなく設計・検証・評価を行うことが可能です。

## 補足： Cコード自動生成

Real-time Workshop および Real-time Workshop Embedded Coder を利用すると、Simulink モデルから組込みを考慮した効率的なCコードを自動生成することができます。以下では前節のオートスケーリングを適用したフィルタについて実施します。以下の手順を実行します。

①Filterブロックから Cコードを自動生成

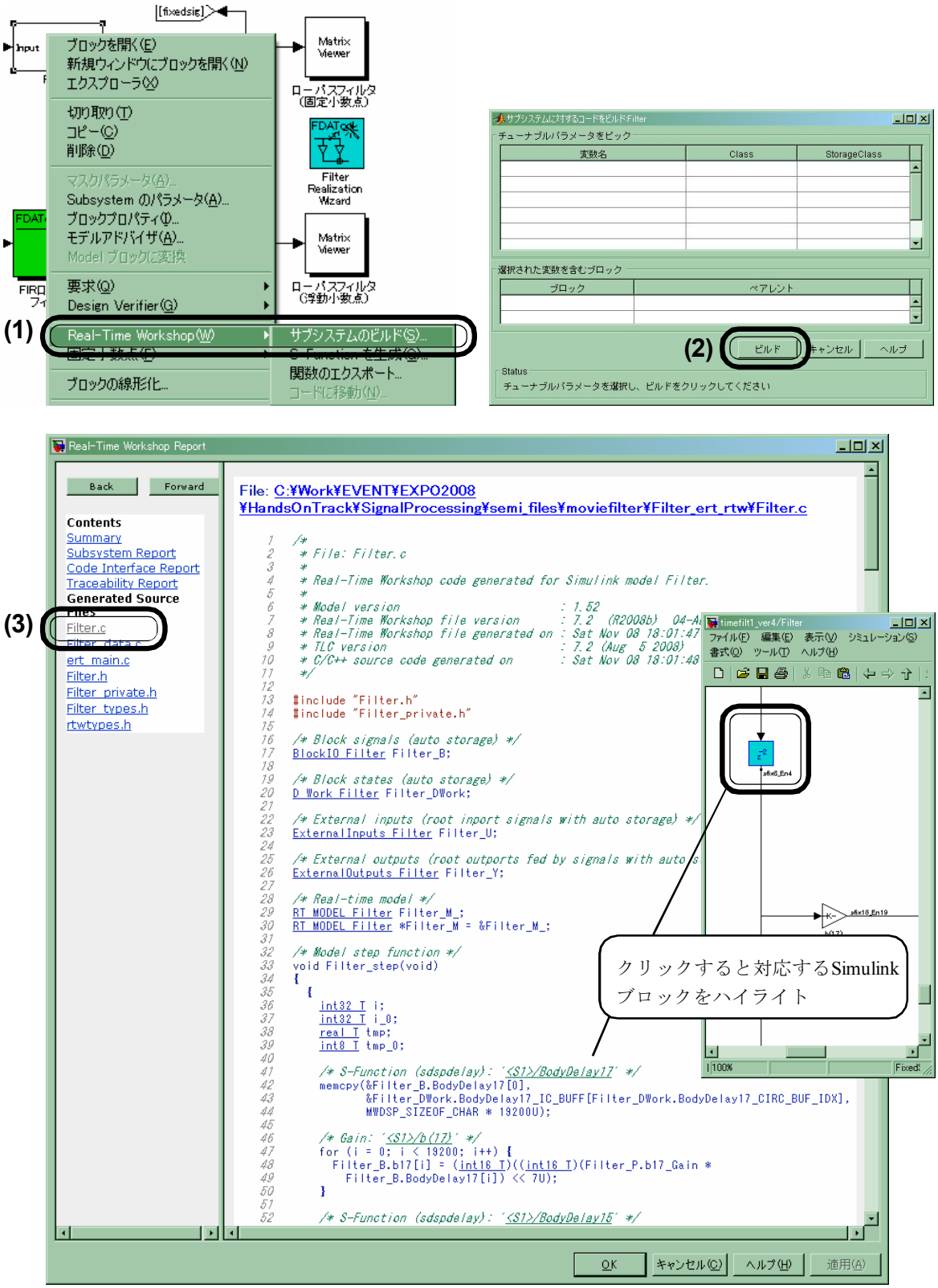


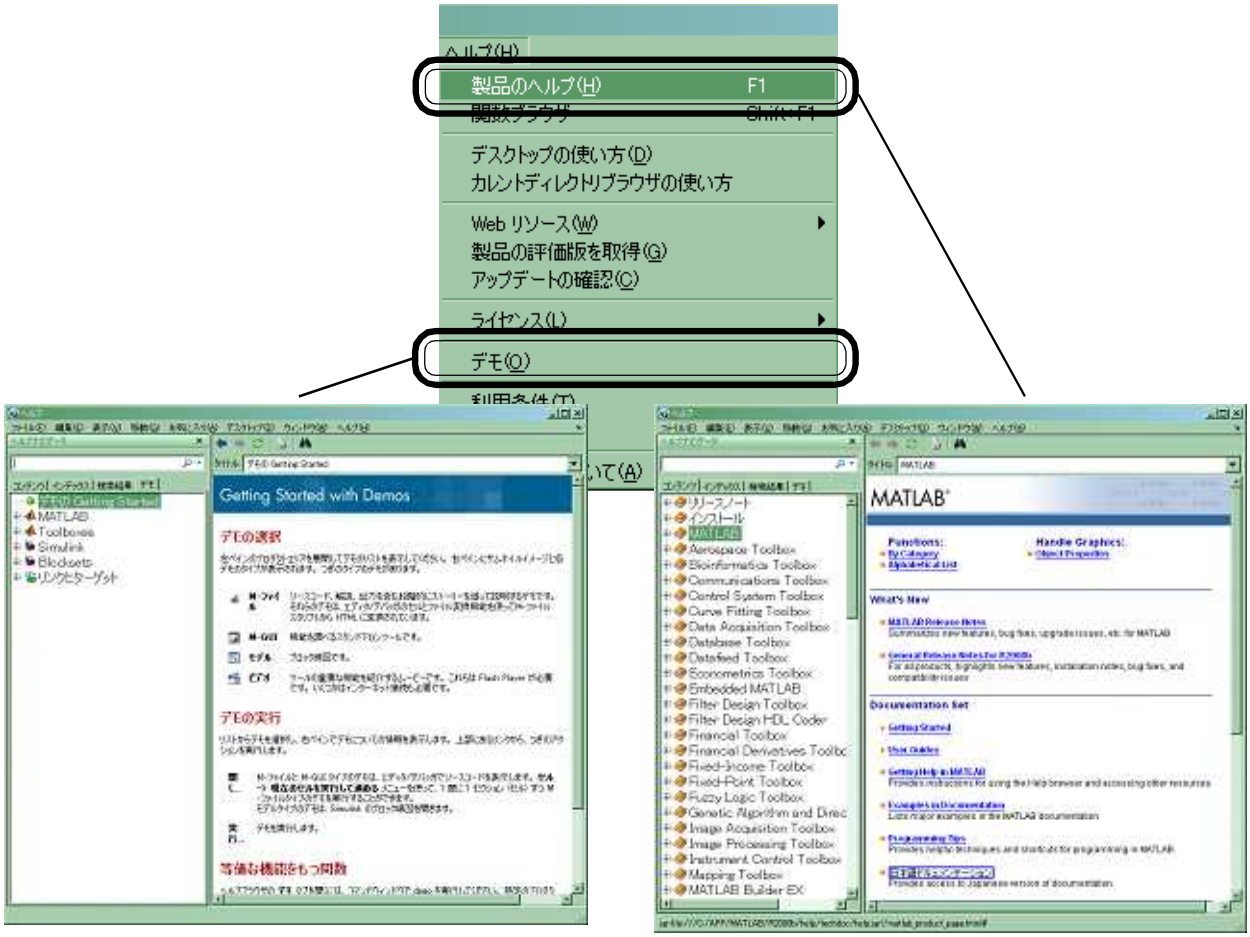

図 6-8 Filterブロックから C コードを自動生成する手順

# 1. MATLAB の情報ソース

MATLABによるプログラム開発において、参考となる資料やホームページを紹介します。

## 1　マニュアル & デモ

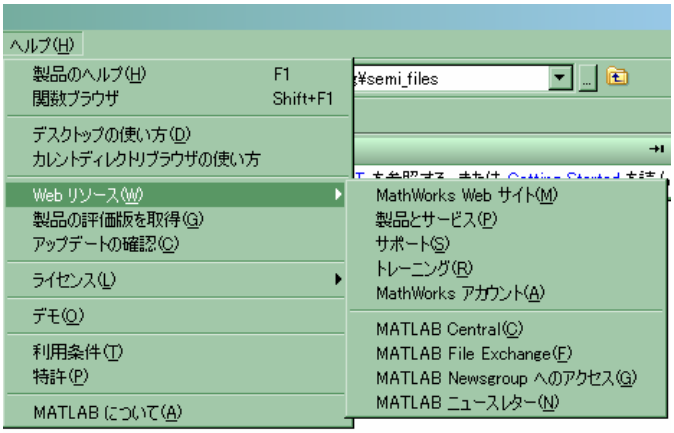
MATLABの基本機能に関するマニュアルは、以下の手順で参照できます。



## 2　MATLAB からの Web アクセス

メニューバー[**ヘルプ**]のサブメニュー[**Web リソース**]から、開発元 The MathWorks 社のホームページへアクセスすることができます。

| | |
|---|---|
| ・The MathWorks Web サイト | The MathWorks 社 HP のメインページ |
| ・製品とサービス | The MathWorks 社のプロダクトページ |
| ・サポート | The MathWorks 社の技術サポートページ |
| ・トレーニング | The MathWorks 社の技術トレーニングページ |
| ・MathWorks アカウント | ライセンス情報の確認、Web ダウンロード等を実行 |
| ・MATLAB Central | MATLABユーザコミュニティのメインページ |
| ・MATLAB File Exchange | ユーザが公開している M-ファイル等をダウンロード |
| ・MATLAB Newsgroupへのアクセス | MATLABユーザのニュースグループへアクセス |
| ・MATLAB ニュースレター | MATLABユーザのニュースレターへアクセス |



## 2. ポイントのまとめ

**Point 1 ＜データは配列として管理＞**

MATLABではデータを多次元配列として定義するためデータの参照や変更が容易です。

**Point 2 ＜柔軟なグラフ機能＞**

MATLABでは2/3次元のグラフをはじめ、ボリュームデータの表示、アニメーション等、様々なグラフィックスをシンプルなコマンドで簡単に描くことが可能です。また、対話的に編集可能なため、即座に編集結果を確認することができます。

**Point 3 ＜高精度かつ高速な配列演算関数＞**

MATLAB では、FFTをはじめ信号処理分野で一般的に使用される多数の数学関数が提供されています。なお、数値演算のコアライブラリとして各CPU毎に最適化されたLAPACK / BLASを実装しているため、行列/配列演算を簡単かつ高速に実現することができます。なお、数値演算にはIEEEの倍精度演算がデフォルトとして適用されます。

**Point 4 ＜M-ファイルによるシンプルなプログラミング＞**

MATLAB は、C/Fortran 言語のようなコンパイラ言語に比べて、記述や操作が簡単なのでアルゴリズムの本質部分に専念できます。また、対話型の環境のためプログラム経験が浅い方でも馴染みやすく、デバッグ作業も簡単です。さらに、C/C++ 言語に比べて大幅にツール習得時間を削減できることも大きな利点といえます。また、次節で述べられているM-Lint機能の情報を活用することにより、早期にエラーを発見し、修正を行うことも可能です。

**Point 5 ＜分野に特化した解析ツールおよび関数ライブラリ＞**

MATLAB では、Signal Processing Toolbox をはじめ信号処理分野で活用される多数の関数ライブラリ（40以上）が提供されています。解析内容に応じて適時利用することで、高度なアルゴリズムを簡単に実行することができます。各Toolboxでは、専門分野にフォーカスした解析関数およびGUIツールが提供されています。

**Point 6　＜ディジタル・アナログ混在システムシミュレーションの重要性＞**

Simulinkは、解析的に解くことが困難とされる非線形要素を含むシステムやアナログ要素（S領域）とディジタル要素（Z領域）が混在したシステムを直感的に表現し、時間軸上のシミュレーションを行うことができます。特に近年、複雑化・高度化するディジタル・アナログ混在システムの設計・検証においては、Verilog AMS/Spiceレベルのシミュレーションだけでなく、システムレベルの検証を十分に実施し、事前にパラメータの最適値や感度を把握することがますます重要になってきています。

**Point 7　＜SimulinkとM-ファイルとの協調性＞**

Simulinkで提供されるEmbedded Functionブロックによりアルゴリズム開発の段階で検証した資産をシステム設計のフローで容易に流用することができます。なお、このブロックで記述したM-ファイルはシミュレーション前に一度、Cコード生成されコンパイル・リンクを実行します。このため、MATLAB FcnブロックでM-ファイルをコールするよりも高速にシミュレーションを行うことができます。SimulinkはMATLAB環境との協調により、柔軟かつ自由度の高い解析が可能です。

**Point 8　＜固定小数点システムの評価・検証＞**

Simulink Fixed PointやFixed-Point Toolboxを使用することで固定小数点化や、最適スケーリングの検証までを短期間で行うことが可能です。これまで取り上げてきたMATLABおよびSimulink環境を共通プラットフォームとしてアルゴリズム設計から固定小数点化までを強力にサポートします。

**Point 9　＜自動コード生成機能＞**

Simulink環境で作成したモデルは自動コード生成ツールにより組込み用途のCコードやHDL（Verilog/VHDL）コード（Filter Design HDL Coder/Simulink HDL Coder）に変換することができます。また、生成されたコードを3rd Party製品により特定のターゲットハードウェアに実装したり、各種シミュレータとコ・シミュレーションすることで設計フローを分断することなく設計・検証・評価を行うことが可能です。