

序 章

ある技術者A君のお話

A君は、とある電機メーカーの新入社員。会社にもだいぶ慣れてきたある日、彼は上司のB氏に呼びつけられました。

B氏：「A君、君は電気科出身だったね」

A君：「はい...そうですが」

B氏：「学生時代、デジタル信号処理とか勉強したでしょ？」

A君：「えーと、単位はとりました」

B氏：「じゃあ今度ゲートアレイ^{注1}を作るから、その中にデジタル・フィルタを埋め込んでくれないかなあ」

A君：「えっ!?(どうしよう...単位は一応とったけれど、過去の問題を暗記して運良く「可」で通っただけだし...自信ないなあ...でもあっさり「できません」とは言えないしなあ)

B氏：「私の時代とは違って、今はみんな大学でデジタルの勉強をするんだろ？ できるよね！」

A君：「だいぶ忘れてしまいましたから、ちょっと勉強の時間が必要だと思いますが...」

B氏：「そうか...まあ、いい機会だから勉強も兼ねてやってみないか？ デジタル・フィルタがわかる人はいないけれど、アナログ・フィルタならそこそこわかる人はいるから、何かあったら聞けばいいよ」

A君：「そうですか...じゃあ勉強を兼ねてということで...やってみます」

B氏：「じゃあ頑張ってくれ！ ちなみに3か月後にゲートアレイが必要になるから」

A君：「納期が決まってるんですか!?! しかも3か月後!!」

A君が設計しなければならないデジタル・フィルタとは、雑音の中から人間の声をフィルタリングし、取り出すものでした。彼には信号処理の知識がほとんどないため、これから勉強をスタートしなければなりません。

注1：ゲートアレイ(Gate Array)：未接続の汎用ロジックをもつチップ。チップの最上位のマスキングのみで配線するため、開発コストが安く、短納期。近年はRAM/ROMなどの記憶素子も搭載可能となってきている。

第1章

もっとも簡単な デジタル・フィルタ

1.1 株価の変動を緩やかにするには

A君はデジタル・フィルタの本を読み始めました。しかし、難しい数式ばかりでわけがわかりません。基本的に数学の嫌いな彼は、本を読むのが嫌になってしまいました。

A君：(デジタル・フィルタって難しいんだなあ。ちょっと疲れたから休もう)

ある日、彼が会社を休んで家にいると、テレビで「株式ニュース」を放送していました。

A君：(株って儲かるんだろうか。株で生活できたら俺はもう技術者なんか辞めてやるんだけどな...)

株式ニュース：「 商事は4日連続の値上がりです」

A君：(待てよ... 会社の業績なんて短期間では大して変わらないから、今日上がった銘柄を買って明日売れば、たいていは儲かるんじゃないかな?)

A君は、さっそく証券会社に勤めている友人のC君に電話しました。

A君：「...俺のアイデアはどう？ たまには損するだろうけど、トータルでは儲かるでしょ？」

C君：「A君、株ってそんな単純なものじゃないよ。今日上がって明日暴落なんて日常茶飯事だよ。そうなりゃ大損害だ」

A君：「じゃあ、過去の株価の上げ下げはまったく参考にならないってことかい？」

C君：「そうでもないけど... まあ、最近の株価の推移が「移動平均線」と近ければ、君のアイデアも少しは正しいかもしれないな...」

A君：「イドウヘイキンセン??? なにそれ？」

C君：「知らないの?? 一定期間の平均を取って株価の変動を緩やかにした線のことだよ。技術者なんだから数学は得意なんだろ？」

A君：「まあ得意...だよ」

A君は思い出しました。移動平均... そういえば、最近読んだデジタル・フィルタの本にそんな言葉があった気がする。もう一度勉強してみよう。

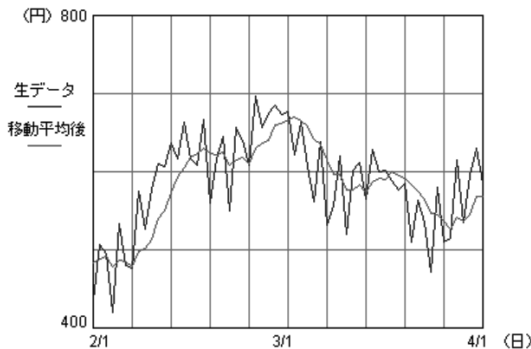


図1-1 株価の生データと1週間移動平均線

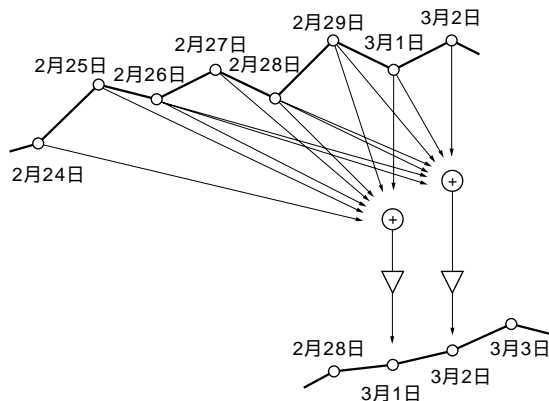


図1-2 1週間移動平均のようす

1.2 移動平均とは？

図1-1に 商事の2か月間の株価推移を示します。ご覧のように、株価の「生データ」は1日ごとに変動があつてギザギザしており、全体的な動きがつかみにくなっています。しかし「移動平均後」を見ると、ギザギザがかなり減ってずいぶん見やすくなっています。

さて、この1週間移動平均線はどうやって導き出されたのでしょうか？ そのようすを図1-2に示します。2月24日から3月1日までの値の平均が、1週間移動平均線における3月1日の値になります。同様に、2月25日から3月2日までの値の平均が、1週間移動平均線における3月2日の値になります。このように、平均を取る期間を移動しながら、順次その結果を新しいデータに対応させます。その結果、移動平均線は生データに比べて、波形の変動が少なくなります。

それでは、もっと長期間の平均をとるとどうなるのでしょうか？ 図1-3に株価の生データと1か月移動平均線を示します。2月1日から3月1日までの値の平均が1か月移動平均線における3月1日の値になり、2月2日から3月2日までの値の平均が、1か月移動平均線における3月2日の値になります。

1週間移動平均線と比べてさらにギザギザが少なくなり、波形が緩やかになったことがわかるでしょう。また、波形の山の位置が右方向にシフトしていることもわかります。これは、1週間移動平均よりも1か月移動平均のほうが遅延が大きいことを意味します。

「移動平均線を求める」という手段は「移動平均フィルタ」と呼ばれ、移動平均フィルタは、もっとも簡単なデジタル・フィルタの一つです。ここで、移動平均フィルタの特徴をまとめると次のようになります。

平均を取る期間を長くすると、一般的に波形が緩やかになる(1週間移動平均線よりも1か月移動平均線のほうがギザギザが少なくなる)。

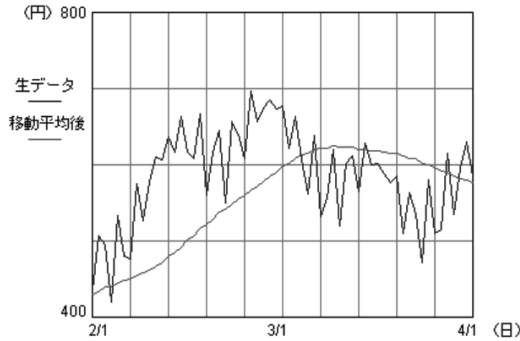


図1-3 株価の生データと1か月移動平均線

平均を取る期間を長くすると、一般的に遅延が大きくなる(1週間移動平均線よりも1か月移動平均線のほうが右にズれる)。

「波形が緩やかになる」ということは、「低周波成分を残して高周波成分を取り去る」ということであり、このようなフィルタはLPF(Low Pass Filter, 低域通過型フィルタ)と呼ばれます。

1.3 移動平均フィルタの数学的表現

株価における1週間移動平均線では、6日前から今日までの株価の平均をとりました。それを、数学的に表現してみましょう。入力(株価の生データ)を $x[n]$ 、出力(1週間移動平均線)を $y[n]$ とすると、6日前の株価は $x[n-6]$ 、今日の株価は $x[n-0]$ なので、1週間移動平均フィルタを数式で表すと、次のようになります。

$$y[n] = (x[n-6] + x[n-5] + x[n-4] + x[n-3] + x[n-2] + x[n-1] + x[n-0]) / 7 \quad \dots\dots\dots(式1-1)$$

式1-1は差分方程式と呼ばれ、1週間移動平均フィルタの時間領域表現です。

1.4 Javaアプレットで移動平均を理解しよう!

図1-4は、株価の移動平均線を求めるJavaアプレットです。

このJavaアプレットは、次のサイトにあります。

<http://digitalfilter.com/jissendf/idoheikin/idoheikin.html> (サイト1-1)

Javaアプレットの説明

「1週間平均」、「1か月平均」または「2か月平均」を選択すると株価の移動平均が計算され、描画されます。

「List in[n]」を選択すると生データ(入力データ)、「List ou[n]」を選択すると移動平均後のデ

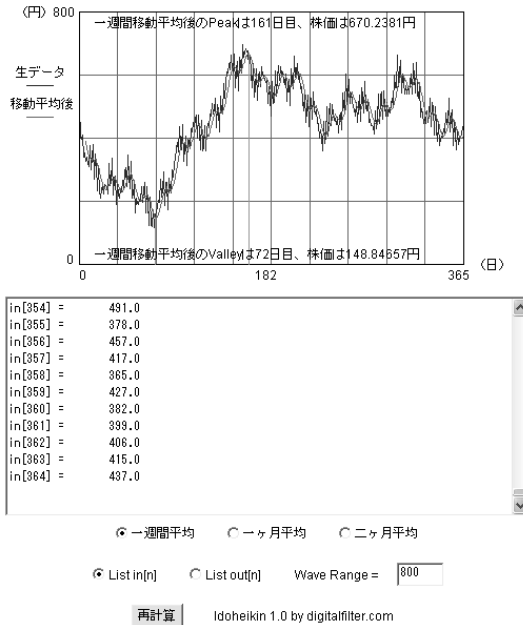


図 1-4 株価の移動平均線を求める Java アプレット

ータ(出力データ)がリストされます。

テキストボックス内のデータ(in[n])を適当に変更し、「再計算」ボタンをクリックすると、それらに対する移動平均が新たに計算されます。

「Wave Range」に数値を入力し、「再計算」ボタンをクリックするとスケールを変更できます。表示がおかしいときは、ブラウザ(Internet Explorer や Netscape Navigator など)の「更新」をクリックしてください。

クイズ1.1

1週間移動平均, 1か月移動平均, 2か月移動平均のうち, 最高値と最低値の差がもっとも小さいものはどれでしょう?

クイズ1.2

1週間移動平均, 1か月移動平均, 2か月移動平均のうち, 最高値, 最低値の日付がもっとも遅いものはどれでしょう?

クイズ1.1の答え

2か月移動平均がもっとも小さくなります。

図 1-5 に 1 か月移動平均の結果を示します。最高値と最低値の差は 625 円 - 198 円 = 427 円です。図

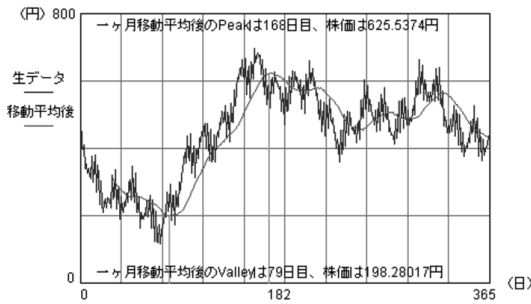


図1-5 1か月移動平均の結果

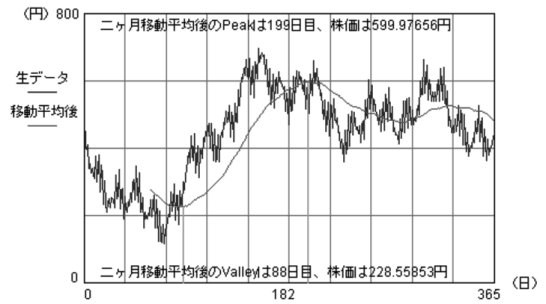


図1-6 2か月移動平均の結果

1-6は2か月移動平均で、最高値と最低値の差は599円 - 228円 = 371円になります。移動平均フィルタの特徴、「平均を取る期間を長くするほど波形が緩やかになる(変動が少なくなる)」がよく表れていますね。このように、平均期間をどんどん長くすると、波形の全体的な振幅がどんどん小さくなっていきます。したがって、1年移動平均、10年移動平均、と平均期間を増やしていくと、移動平均線はほとんど平坦になってしまいます。

クイズ1.2の答え

2か月移動平均がもっとも遅くなります。

図1-5の1か月移動平均線を見ると、最低値の日付は79日目、最高値の日付は168日目です。図1-6の2か月移動平均線では、最低値の日付は88日目、最高値の日付は199日目です。移動平均フィルタの特徴、「平均を取る期間を長くするほど遅延が大きくなる」が表れています。このように移動平均の結果「遅延」が生じ、それは平均期間が長いほど大きくなります。専門的な用語を使うと、「フィルタ長(タップ数ともいう)が長くなると遅延が大きくなる」となります。

1.5 Visual C++で移動平均フィルタを実行しよう!

次のURLに、移動平均フィルタを実行するVisual C++プロジェクト(Doheikin.zip)があるので、ダウンロードしてください。

<http://digitalfilter.com/jissendf/vcpro/idoheikin.html> (サイト1-2)

Zip圧縮ファイルを解凍した後、適当なディレクトリにコピーしましょう。Doheikin.dswをダブルクリックするとVisual C++のプロジェクトがスタートします(図1-7)。

「ビルド」「実行」を選択してください(図1-8)。

するとアプリケーションがビルドされ、実行されます(図1-9)。

まず、Waveファイルを読み込みましょう(図1-10)。

Ding.wavを選択してください(図1-11)。

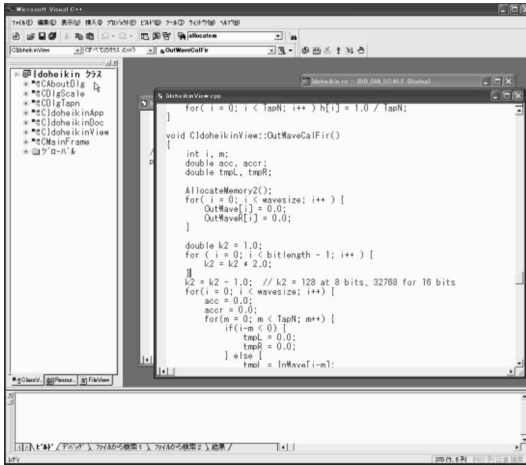


図 1-7 Visual C++ プロジェクト Idoheikin

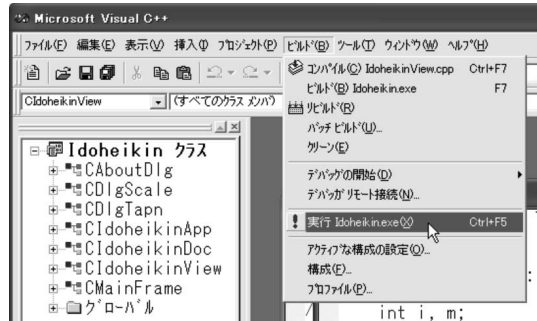


図 1-8 Idoheikinのビルドと実行



図 1-9 Idoheikin.exeの初期画面



図 1-10 Wave ファイルの読み込み

するとDing.wavに、移動平均フィルタ(タップ数の初期値は10)が施されます。図1-12に示すように、上段に原信号(Ding.wav)が、下段に移動平均フィルタを施した波形が描画されます。ご覧のようにギザギザが減って、波形が滑らかになっています。また、移動平均後の波形は、原信号より若干遅延しているのがわかります。

フィルタリング結果をWaveファイルにセーブしましょう(図1-13)。ファイル名はtest1.wavとしてください。

Waveをセーブしたらtest1.wavの音とDing.wavの音を比較してみましょう。フィルタリングの結果、音が柔らかく(低く)なっているはずですが、移動平均フィルタによって、音の高周波成分が取

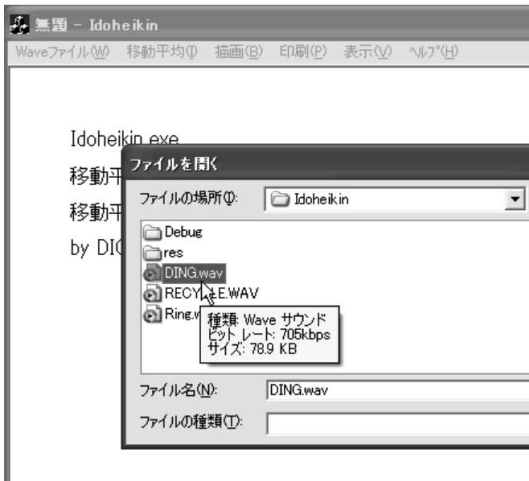


図 1-11 Ding.wav の選択

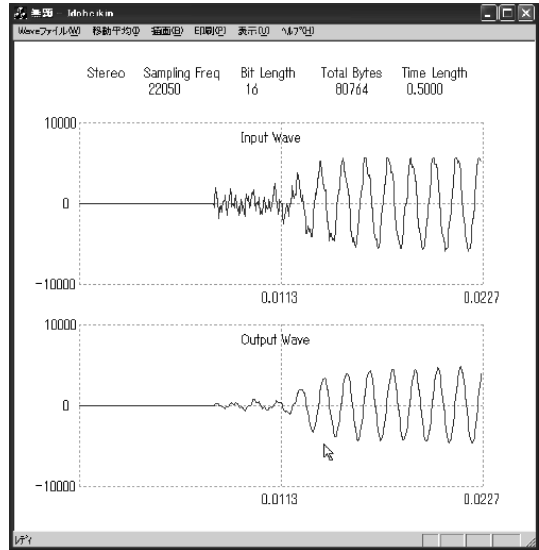


図 1-12 Ding.wav に移動平均フィルタを施す前と後の波形

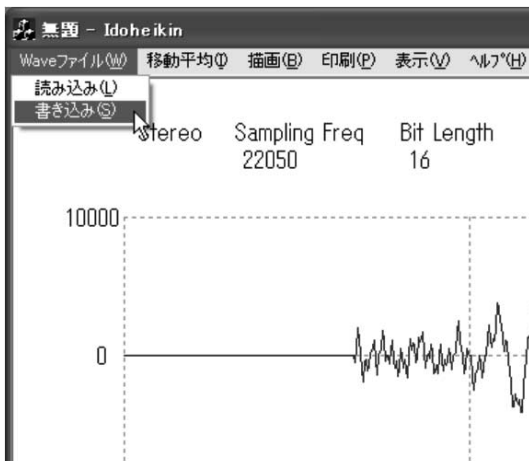


図 1-13 移動平均後の Wave ファイルの書き込み

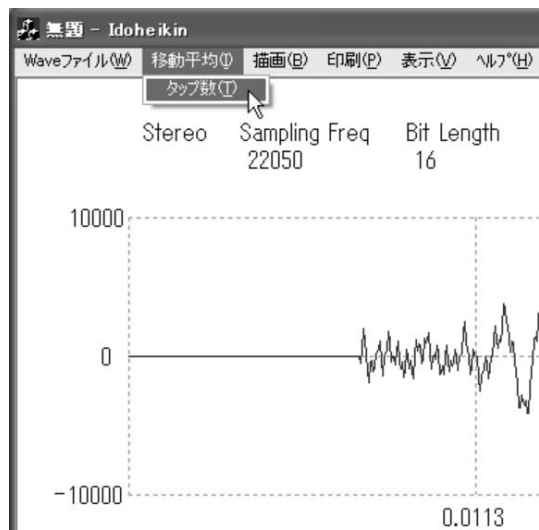


図 1-14 移動平均フィルタのタップ数変更

り去られたからです。

さて、移動平均フィルタのタップ数(フィルタ長)をもっと増やすとどうなるでしょう? 「移動平均」「タップ数」を選択しましょう(図 1-14)。

タップ数を 30 としてみます(図 1-15)。

その結果は、図 1-16 のようになります。移動平均フィルタのタップ数が長すぎて、波形が減衰し



図1-15 タップ数を10から30へ増やすと...

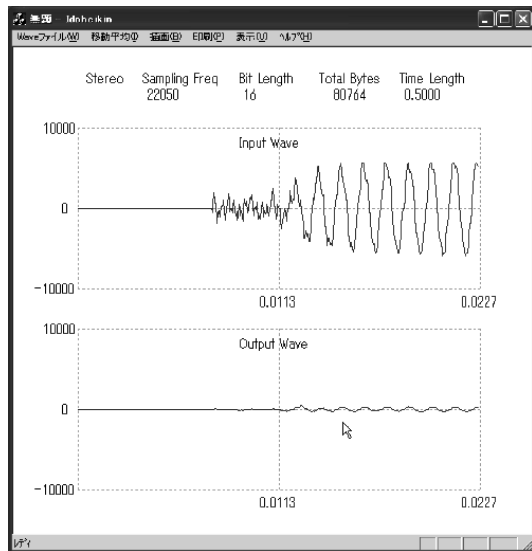


図1-16 長いタップ数での移動平均の結果

てしまいました。Waveファイルの音を再生すると、音が相当小さくなったことがわかるので試してみてください。

さて、もう一度移動平均フィルタの特徴を思い出してください。

平均を採る期間を長くすると、一般的に波形が緩やかになる。

平均を採る期間を長くすると、一般的に遅延が大きくなる。

移動平均フィルタの入出力を波形で表すと、図1-12のように、波形のギザギザが少なくなります。すなわち波形が緩やかになり、音が柔らかく(低く)なるということです。そして、移動平均フィルタのタップ数(平均を採る期間)を長くしていくと、図1-16のように波形が緩やかになりすぎて減衰してしまいます。この現象は、移動平均フィルタの特徴 を表しています。

また、移動平均フィルタのタップ数を長くしていくと、出力の波形がどんどん右にシフトしていくことがわかります。この現象は、移動平均フィルタの特徴 を表しています。

Visual C++ ソース・コードの説明

CIdoheikinView::InitCoeffs 関数：移動平均フィルタの係数を決定する関数 (リスト1-1)

CIdoheikinView::OutWaveCalFir 関数：移動平均フィルタリングを行う関数 (リスト1-2)

CIdoheikinView::GetWavInfo 関数：Waveファイルの情報を読み込む関数 (リスト1-3)

CIdoheikinView::PutWavInfo 関数：Waveファイルに情報を書き込む関数 (リスト1-4)

CIdoheikinView::WaveDraw 関数：Waveの時間軸波形を描画する関数

CIdoheikinView::CIdoheikinView 関数：各変数の初期値を設定する関数。アプリケーション立ち上げ時に一度だけ実行されます。