

Computer Science A

Hardware Design Exercise 6

Handout V3.30 December 20th, 2021

CSAHW Computer Science A, Meiji University

CSA_B3_EX6.pptx 41 Slides
Renji Mikami

内容

時間信号(実数)

1.線形時不変システム:

- 線形性、線形システム
- 時不変性、時不変システム
- 因果性、因果システム
- 安定性、安定システム

2.インパルス応答

- 単位インパルス応答
- 連続インパルス入力に対する出力

3.デジタル・フィルタ:

- FIR
- IIR

複素領域信号(複素数)

1.Z変換:

- 線形性
- 推移定理

2.伝達関数

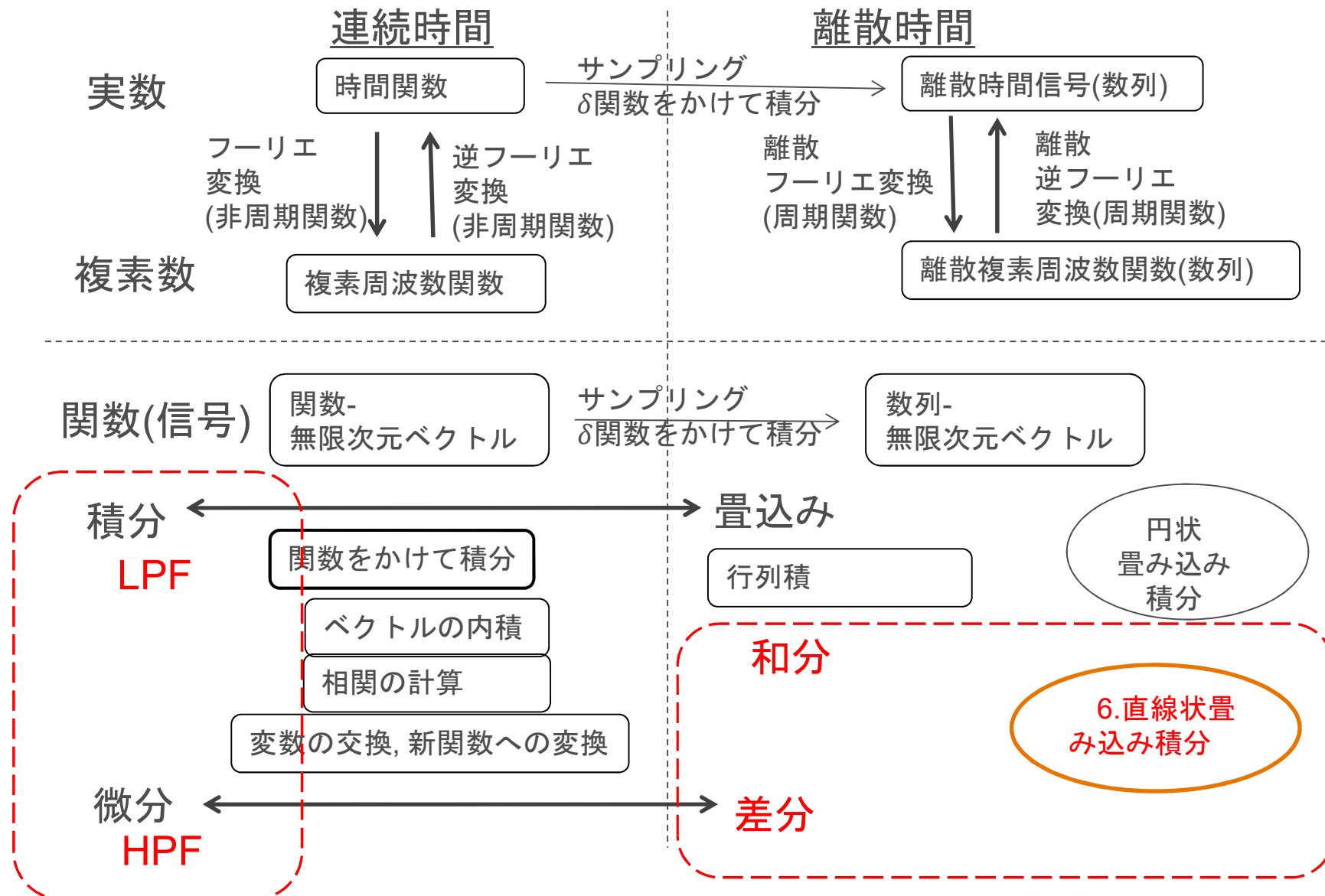
- 極と安定性
- 収束、振動、発散
- ゼロ点

3.デジタル・フィルタ

- 設計法
- 使い方

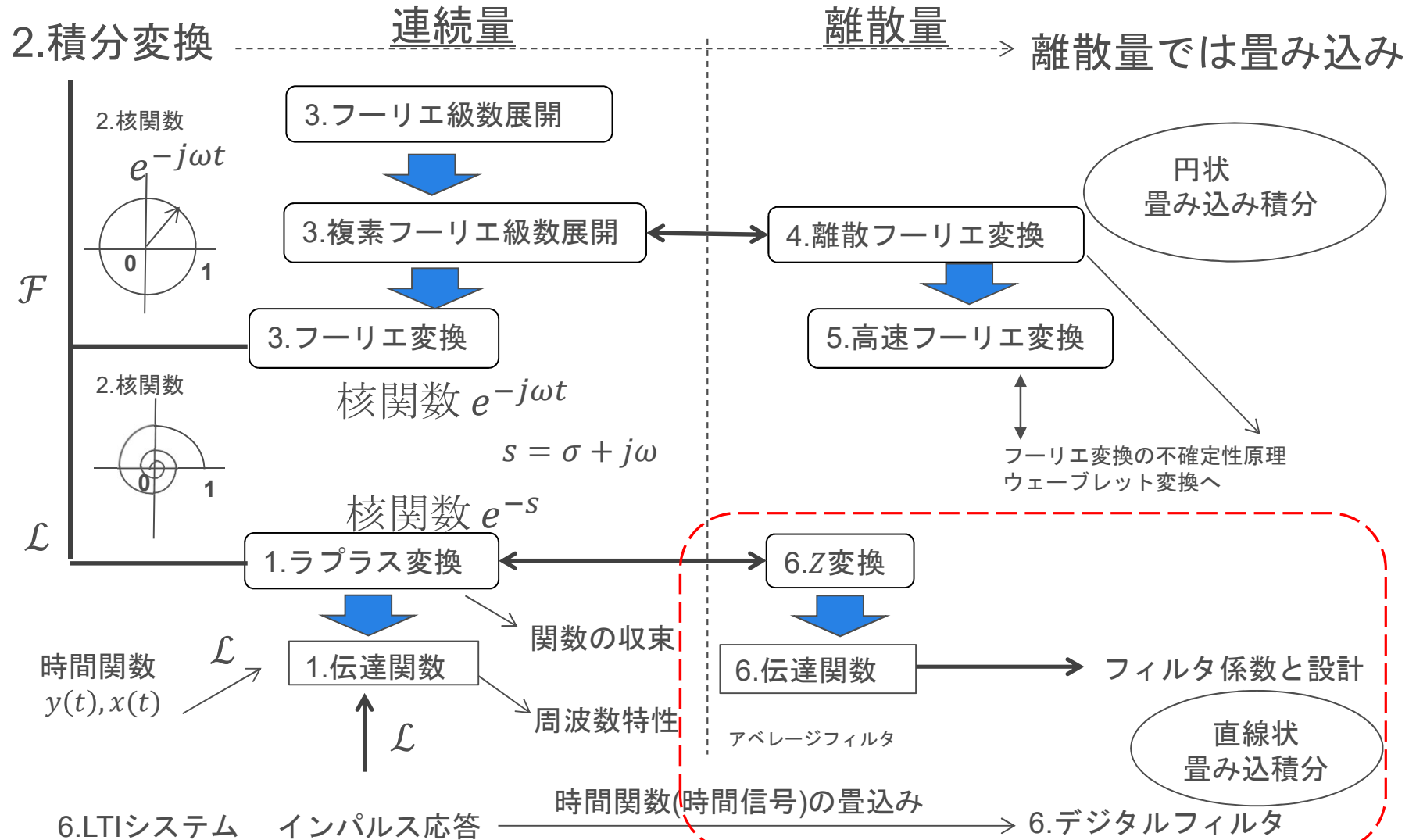
冒険マップ：実数,複素数,連続,離散,積分

点線枠の部分をやっていきます



冒険マップ：関数(信号)の変換

点線枠の部分をやっていきます



数字はテキスト(テーマ)番号, $\mathcal{F}$:フーリエ変換, $\mathcal{L}$:ラプラス変換

フィルターをイメージする

- 1 : フィルタとは、特定の周波数帯域の信号を通過させる装置
- 2 : アナログのフィルタとデジタルのフィルタがある
- 3 : LPF (Low Pass Filter) 低い周波数の信号を通過させる
- 4 : HPF (High Pass Filter) 高い周波数の信号を通過させる
- 5 : BPF (Band Pass Filter) 特定周波数帯域の信号を通過させる。
- 6 : Notch Filter : 通過できない信号は、阻止されていると考えられるので、特定周波数帯域の信号が通過できないようにするフィルターもある。(バンドパスフィルタの逆の機能)
- 7 : フィルターのQ: 中心周波数/-3dB落ちのバンド幅(フィルタの特性をあらわす)。フィルタは、何段も重ねることができ、重ねるとQは高くなっていく。
- 8 : フーリエ変換によって、周波数スペクトルが得られたことを思い出す。(フーリエ変換には、逆フーリエ変換が成り立つことを考える。)
- 9 : 波形のとんがり具合いと周波数スペクトラムの関係を視覚的にイメージしておく。

6.1.1 高周波雑音の除去

EX 6.1.1

ゆっくり(なだらかに)と変動するデータをサンプリングして、その周波数スペクトラムを得ようとしたが、信号に高周波の雑音が混入し、サンプリング値が雑音によって、大きく乱されている。雑音の除去方法を考えてください。



6.1.2 FT / IFT とアナログフィルタ

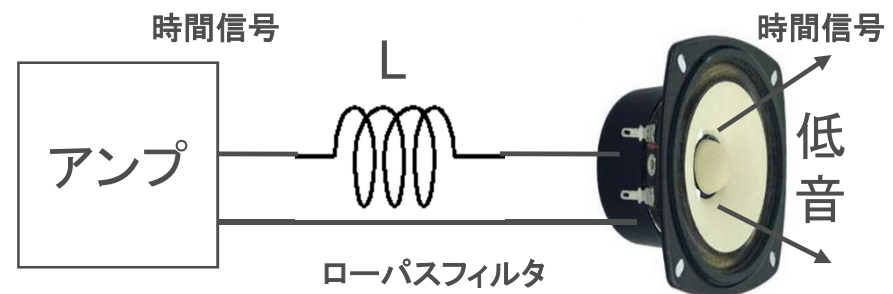


電気信号のアナログ回路では、インダクタ(コイル)を直列に挿入してアナログフィルタでLPFを構成して高周波信号を減衰させます。時間信号(関数)領域で処理をしている。

離散量に対して処理できないだろうか。

この時間信号(関数)には、フーリエ解析すると複数の周波数のスペクトルが含まれています。この中で、必要な周波数は、低い周波数の成分で、雑音は高い周波数の成分として含まれます。

これまでの学習の成果を使うと、混入雑音周波数の2倍以上でサンプリングし、これをフーリエ変換(周波数関数に変換)し、不要な高周波スペクトル項を削除し、その後に再び逆フーリエ変換をすれば、時間信号(関数)が得られるはずです。



高周波信号はLのインダクタンスで熱に変換される

6.2.0 信号値の平均化

通常実習では、授業HPから
AF_48_Form.xlsxをDLしてください

系列1-黒: 直線的(線形)に変化する連続信号を元信号と想定します。

系列2-赤: は、サンプリングによって得られた離散化した信号(数列)とします。

この離散化信号(数列)を平均化してみます。平均化によって、信号の変化は、よりなだらかなになるはずですが、ここで、元の信号の変化に対して、雑音による変化が十分に大きければ、平均化によって、得られた離散信号(数列)は、より元の信号に近づくと期待されます。

想定されるなだらかな元信号変化
(系列1-黒)

0 1 2 3 4 5 6 7 8 9 10 11

サンプリング信号例(系列2-赤)

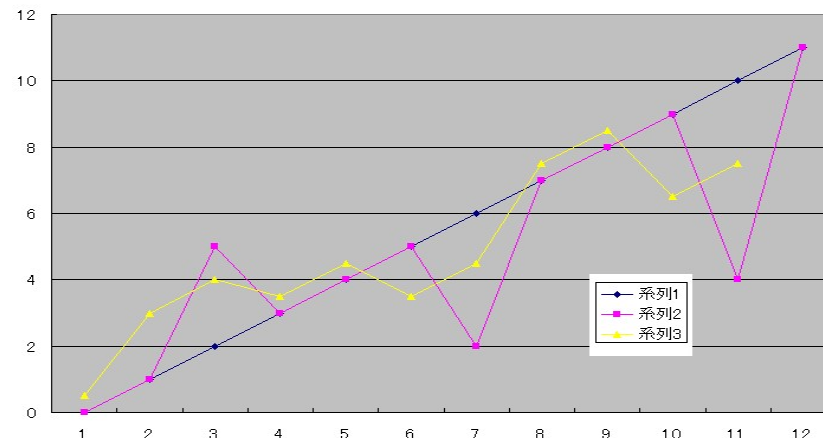
0 1 5 3 4 5 2 7 8 9 4 11

これを入力とします。

となりあう値の平均値をとると(系列3-黄)

0 1 5 3 4 5 2 7 8 9 4 11

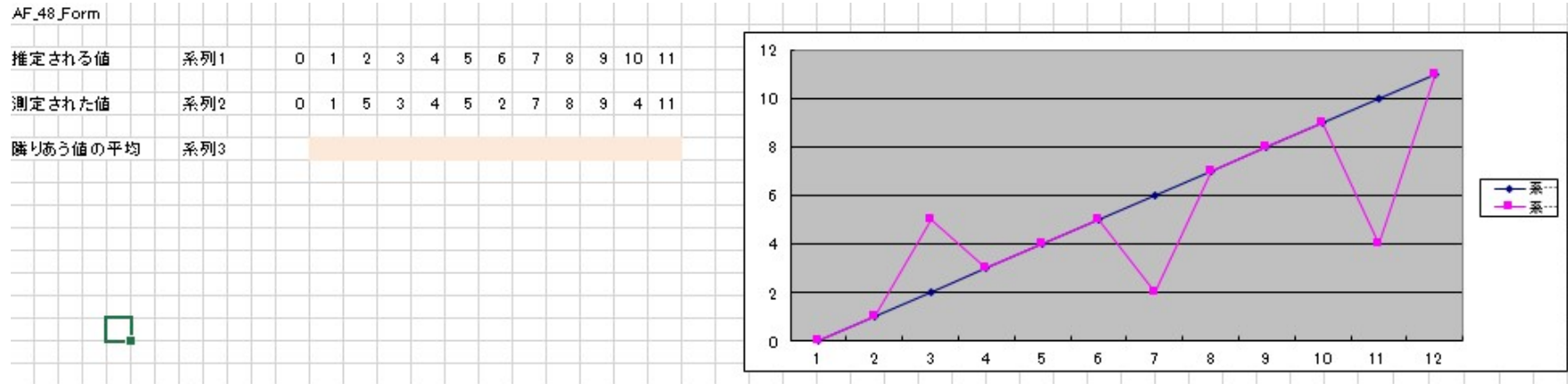
0.5 3 4 3.5 4.5 3.5 4.5 7.5 8.5 7.5 7.5 得られた結果



EX621 エクセル(またはMaxima)
で系列1,系列2,系列3のグラフ
を作成し、結果を考察してください。
(AF_48_Form を使わない場合)

AF_48_Form.xlsx フォームを使う
場合はEX631に進んでください
次のスライドでフォームの解説をし
ます

6.2.1 AF_48_Form の解説 1

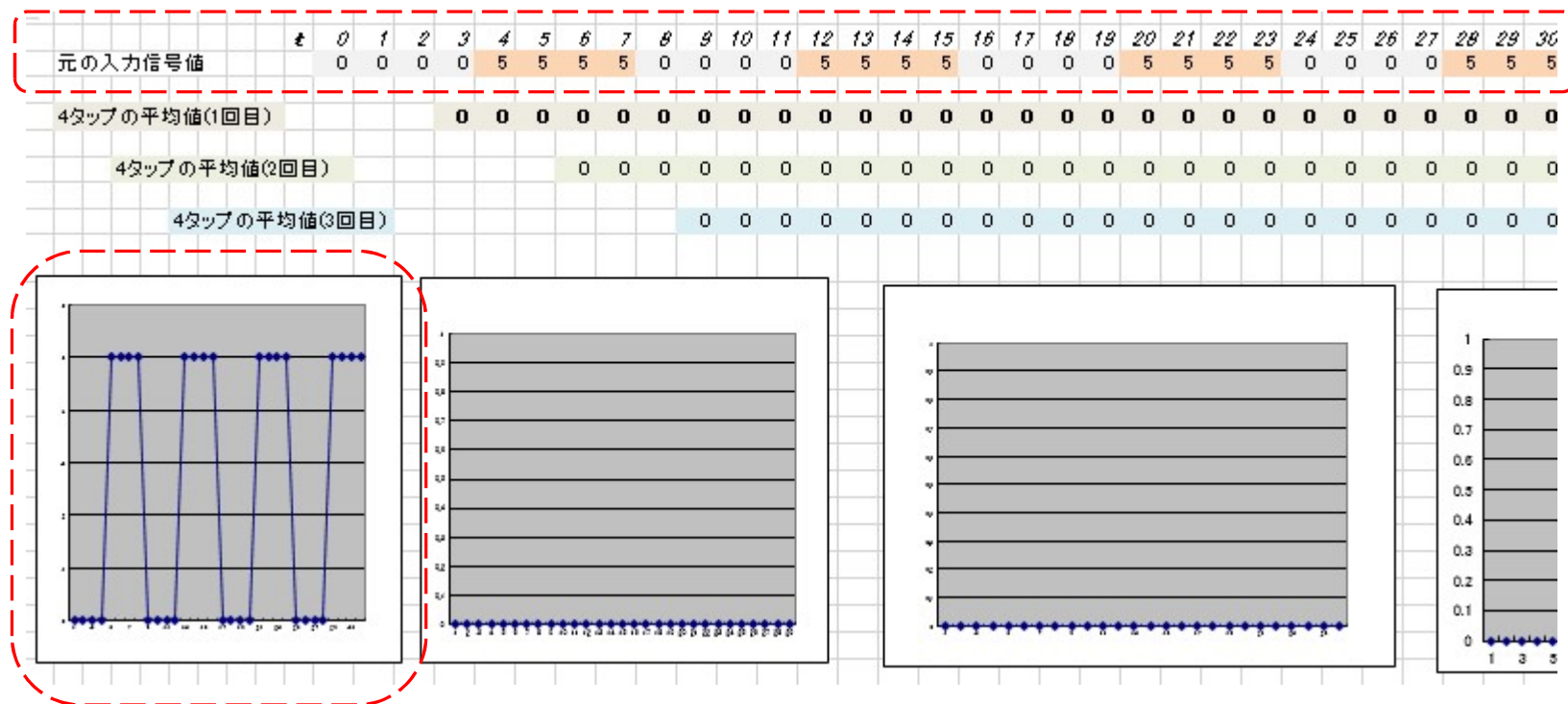


EX631 の実行には
図示された上のフォーム
のオレンジ色の部分に計算式を入
力します。
式が計算されると右のグラフに黄
色い線で結果が表示されます

左端の計算式を右方向にコピーしてすると、自動計算されるように作られています

6.2.2 AF_48_Form の解説 2

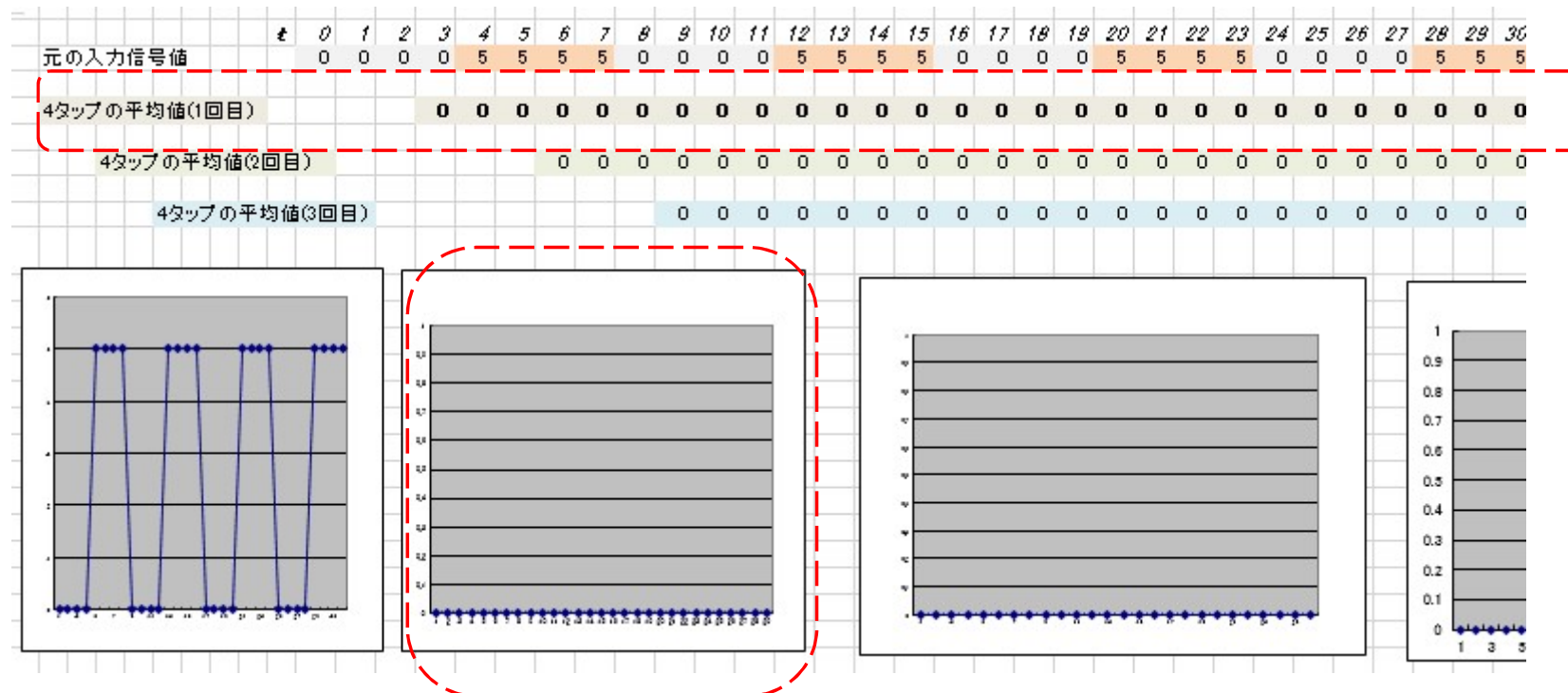
EX641 で使用するフォームの解説です。赤点線枠で囲まれた部分の数値でグラフが自動表示されます



元の信号値がグラフに表示されています。

6.2.3 AF_48_Form の解説 3

4タップのアベレージング・フィルタを作ります。赤い点線枠に計算式を入力すると左から2番目にグラフが表示されます。数式は次のスライドで解説します。



次々と計算していきます。計算フォームはコピーすると自動計算されるように作られています

6.2.4 AF_48_Form の解説 4

最初の項の計算式を入力します。

係数 $0.25 = 1/4$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	

次々と計算していきます。
 計算フォームはコピーすると
 自動計算されるように作られています

0	0	0	0	0	5
					0 1

0	0	0	0	0	5
					0 1

次々と計算

6.2.5 AF_48_Form の解説 5

最初の項の計算式をコピーして残り全部にペーストします。

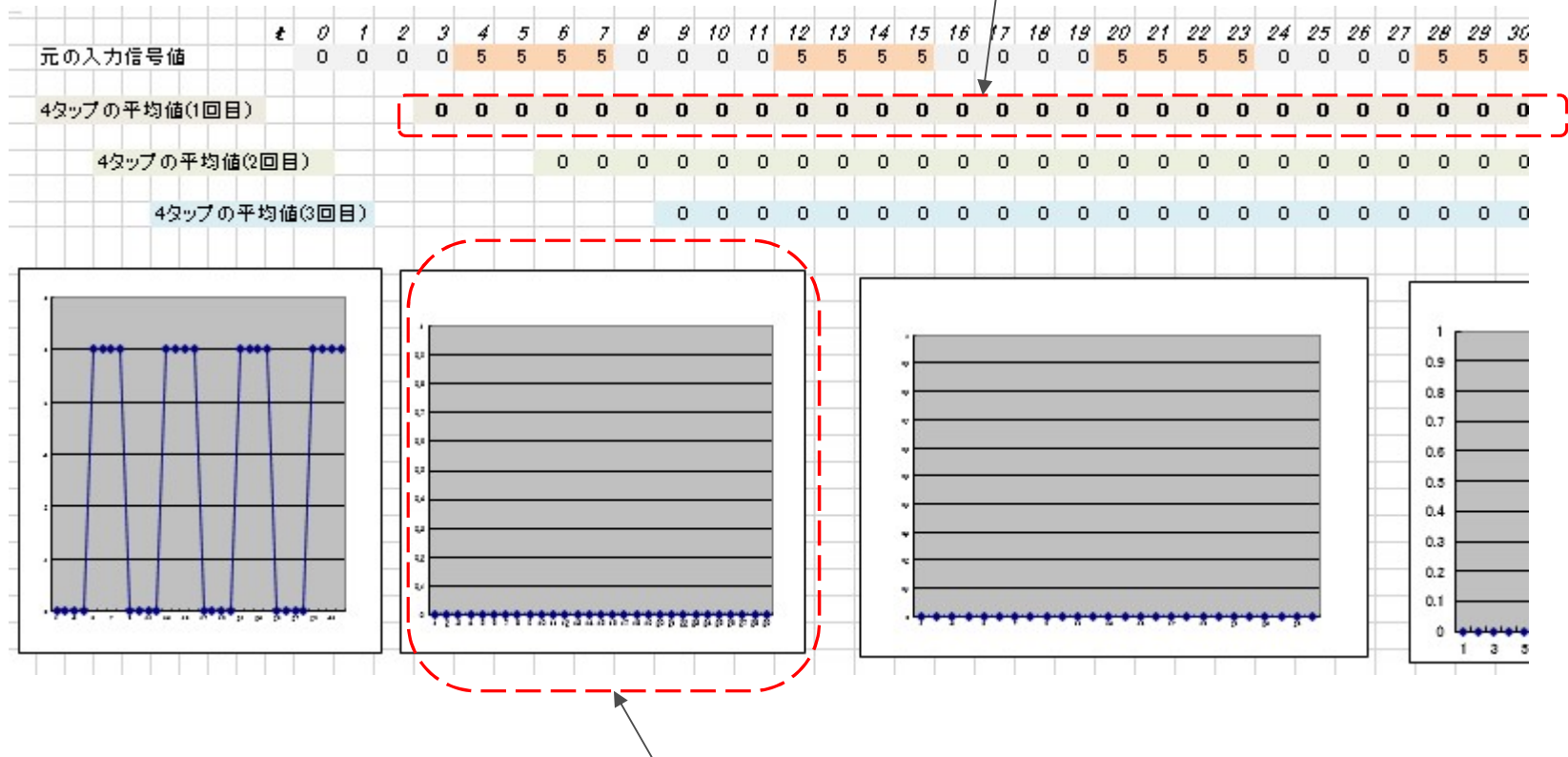
係数 $0.25 = 1/4$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23																						
24																						
25																						
26																						
27																						
28																						
29																						

次々と計算していきます。計算フォームはコピーすると自動計算されるように作られています

6.2.6 AF_48_Form の解説 6

計算値が表示されます

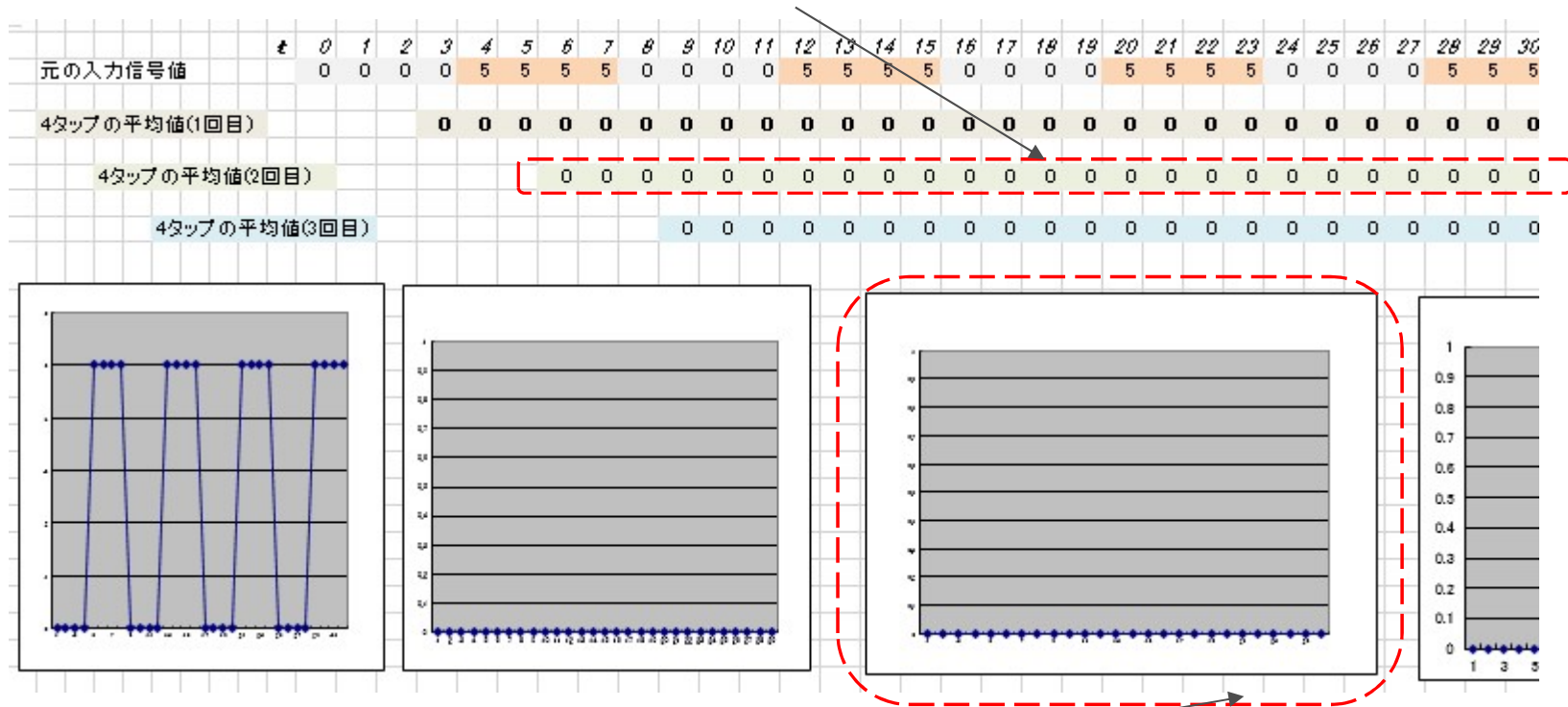


左から2番目にグラフが表示されます

次々と計算していきます。計算フォームはコピーすると自動計算されるように作られています

6.2.7 AF_48_Form の解説 7

1回目の計算結果にさらに2回目のフィルタをかけます



左から3番目に2回目のグラフが表示されます

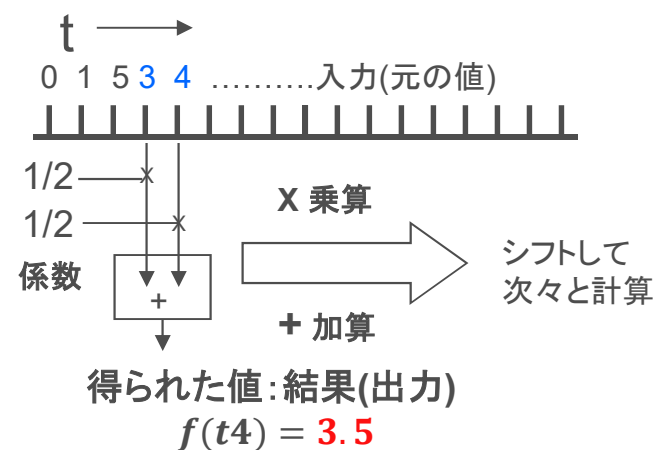
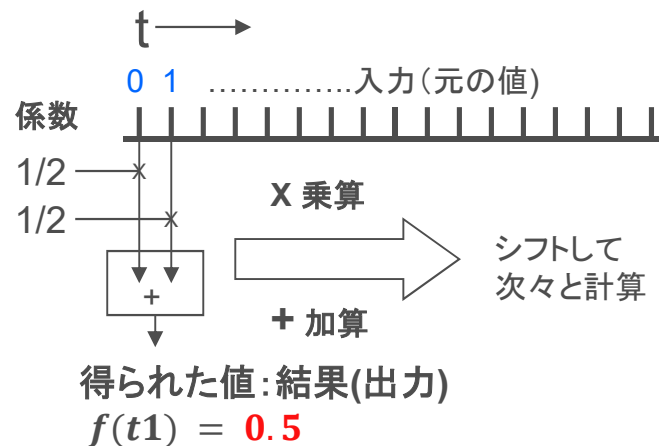
次々と何回もフィルタをかけてグラフを表示してみます。

6.2.8 平均化と係数積和計算

平均化の意味は入力時間信号 $f(t1)$ の次の入力時間信号 $f(t2)$ の平均だから、入力が、 $1/2(t1)$ と $1/2(t2)$ の加算を出力とする関数です。
ポイントは、①入力を Δt 時間シフトして加えること、②入りに係数($1/2$)をかけていること、③加算結果(出力)は Δt 後に得られることです。

となりあう値の平均値をとると(系列3-黄)

0 1 5 3 4 5 2 7 8 9 4 11
0.5 3 4 3.5 4.5 3.5 4.5 7.5 8.5 7.5 7.5 得られた結果

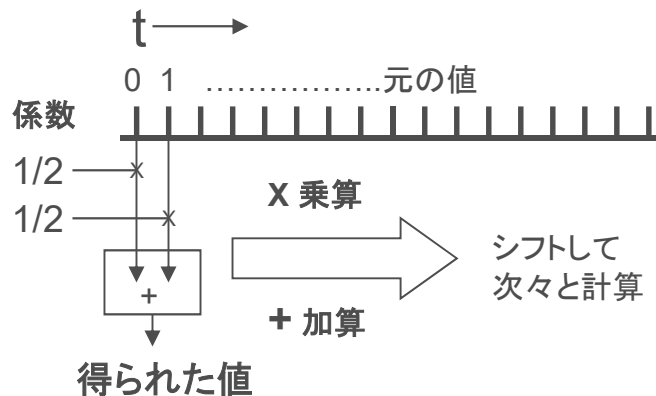


後述しますが、①はLTIシステムの信号の重ね合わせ、②はインパルス応答、デジタル・フィルタの係数、③は因果システムに関係しています。

6.3.1 アベレージング・フィルタ

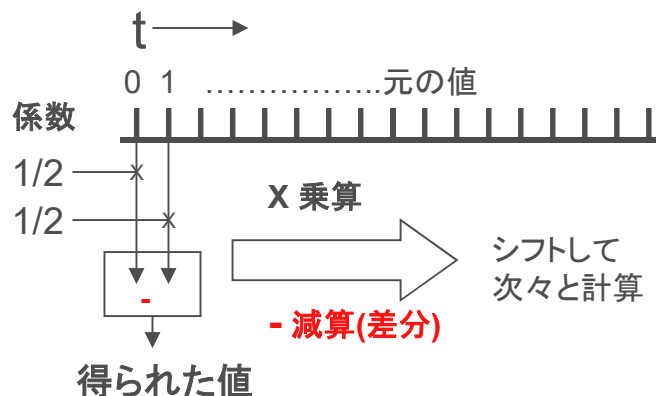
AF_48_Form左上を使用してください

EX631 系列2を入力、系列3を出力としてグラフを表示させてください。系列3の平均を求める計算は0.5を掛けた2項の和として計算してください



EX632 :

左の演算は、どのような機能と考えられるかを述べてください。



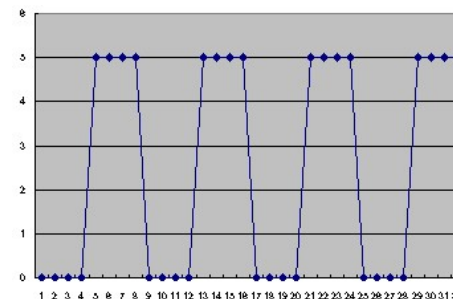
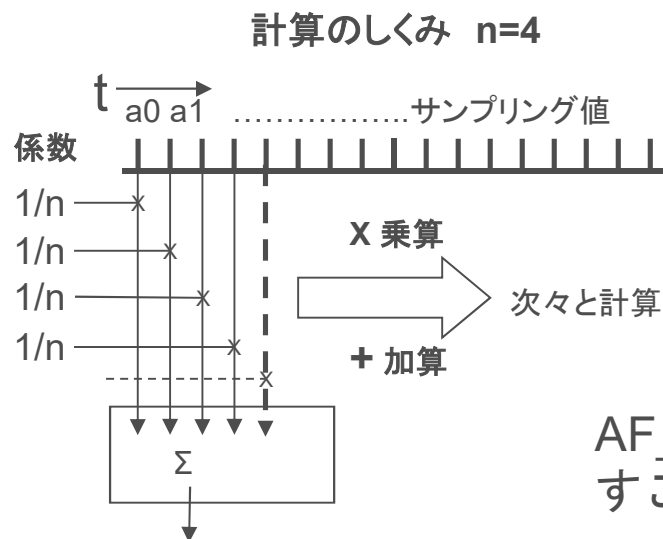
考察633 :

上の演算の、加算を**差分**に置き換えるとどうなるか、考察してください。

6.4.1 平均化項数とタップ 1

EX641(AF_48_Form中央部を使用します。数値と左端のグラフは表示済です。)
平均化する項の数を4に増やします。積和計算の項数をタップ数 n とすれば、
係数は $1/n$ になります。係数と計算式どうなるでしょうか？

AF_48_Formがない場合はエクセルに以下の数値を入力して、結果を(折れ線
線)グラフで表示してください。(フィルタの次数は、タップ数-1になります。)



左端に表示す
みのグラフ

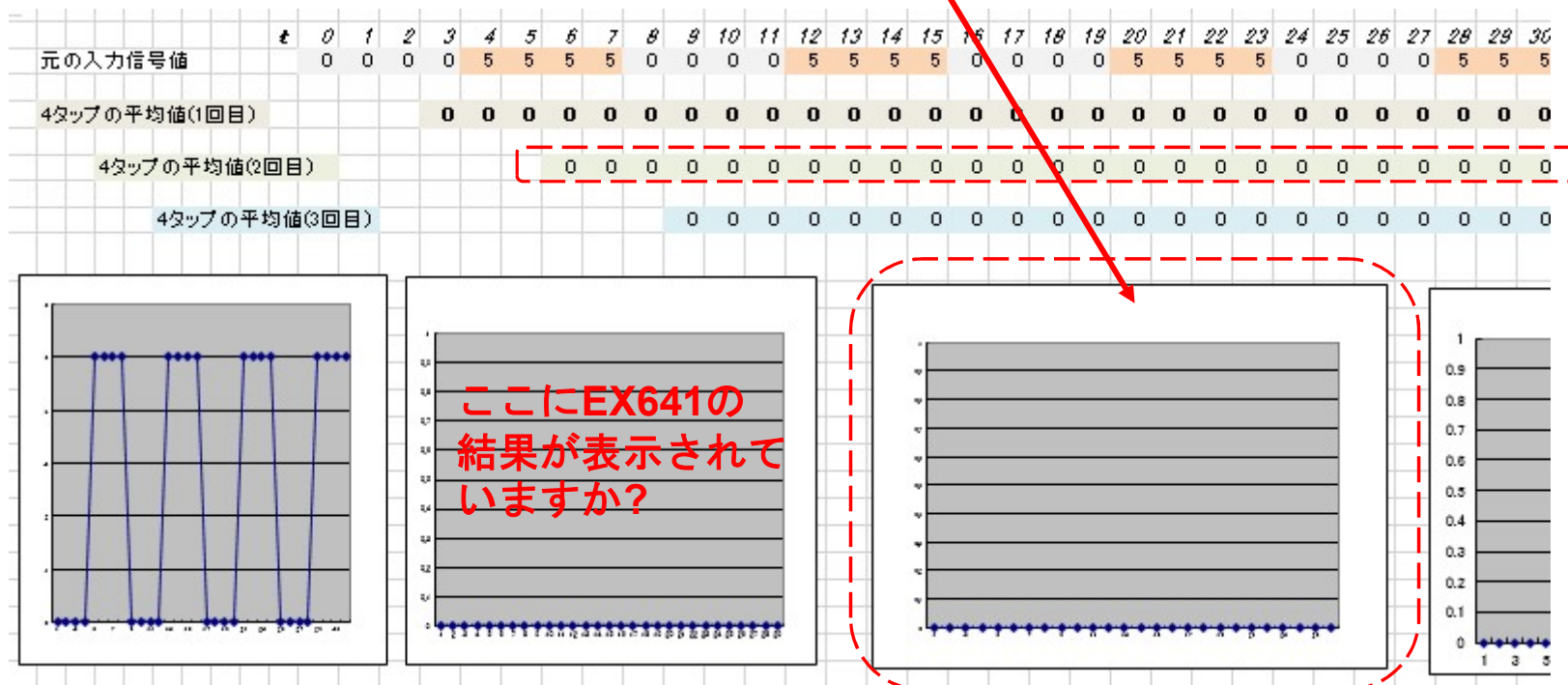
AF_48_Form.xlsxには、数値が入力済みで、
すこのときのグラフが自動表示されます。

次々と計算していきます。計算フォームはコピーすると自動計算されるように作られています

6.4.2 平均化項数とタップ 2

EX642

EX641 数の結果に対して、さらに4タップのアベレージングフィルタ(2回目)をかけて、結果をグラフ表示させてください。(6.2.7 を参照してください。)



6.4.3 平均化項数とタップ-グラフ化

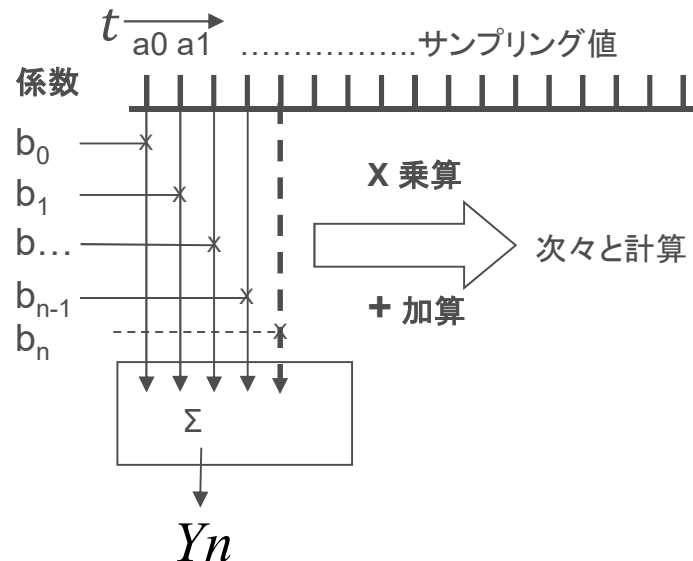
EX643

6.4.2の結果をさらに4タップのアベレージ・フィルタにかけ、結果をグラフ化してください。(3回の平均化を実行)

EX644 : **EX642/643**のグラフは、どのような関数(信号波形)に似ているか、またその関数(信号)のスペクトルの特性から、この平均化フィルタの機能について述べてください。またこのフィルタをさらに複数段重ねていった場合についてどのようなようになっていくかを考察してください。

6.4.4 係数列の変更

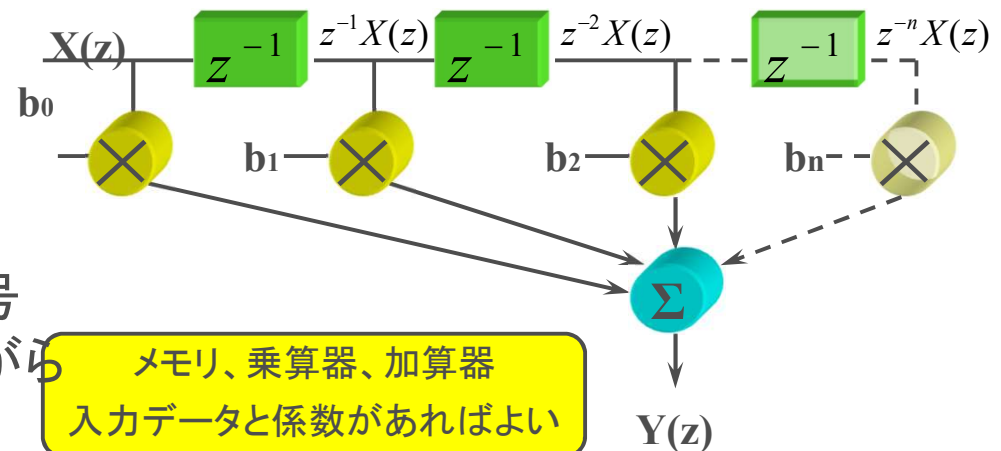
アベレージ・フィルタでは、係数はすべて $1/n$ でした。ここで積和計算の係数に特定の係数列を使用したらどうなるかを考察します。



デジタル・フィルタは、右のような構成を取り、入力サンプリング信号に係数をかけ、次々にシフトしながら加算していきます。このときフィルタの特性は、係数列で決定されます。

FIR Filter一般式の例

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) \dots$$
$$= (b_0 + b_1 z^{-1} + b_2 z^{-2} \dots) X(z)$$

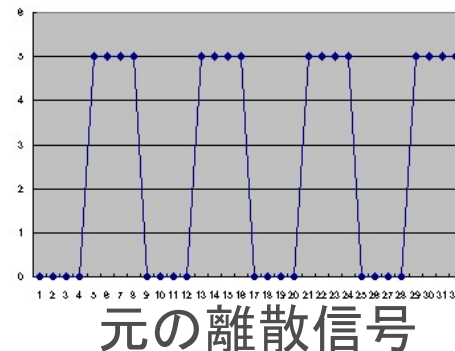
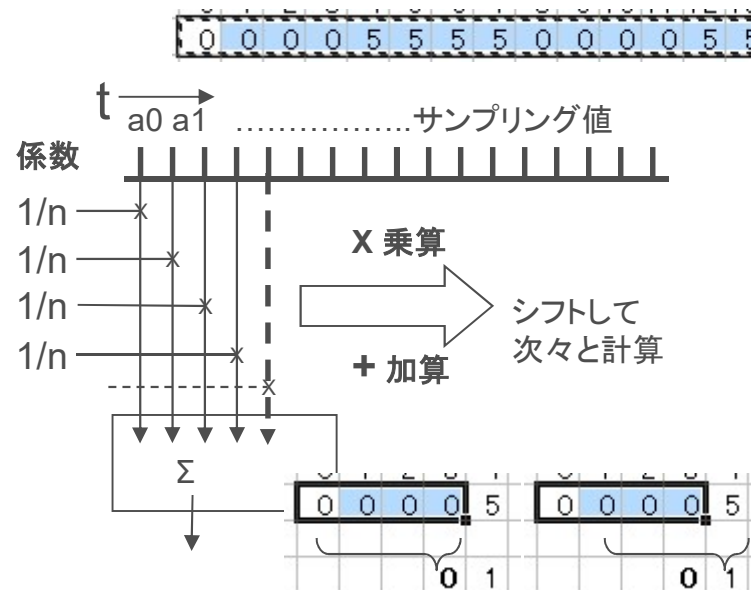


FIR: Finite Impulse Response

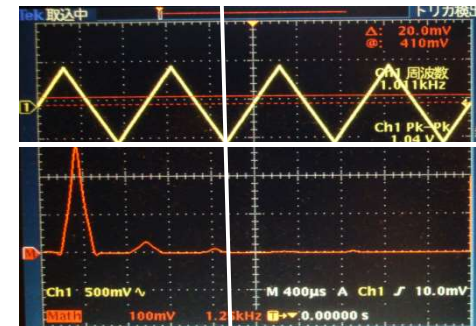
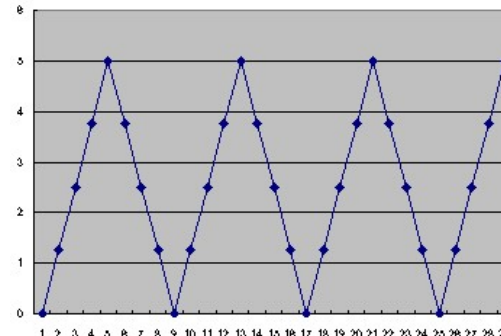
6.4.5 641から645の総合考察

積和計算の項数をタップ数 n とすれば、係数は $1/n$ になります。

エクセルに以下の数値を入力して、結果を(折れ線)グラフで表示



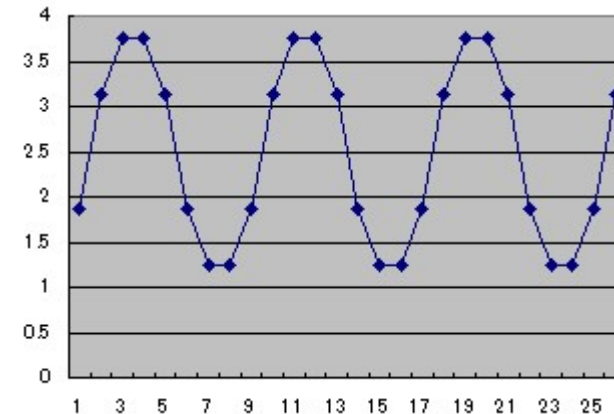
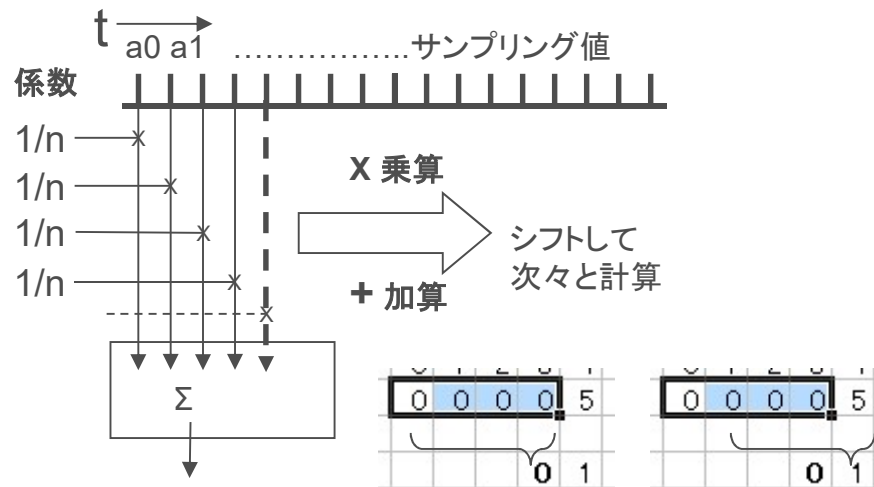
EX645 : 4タップ アベレージ・フィルタの複数回処理の結果をグラフ化してください



アベレージ・フィルタを通した後の波形は、高周波スペクトル項が減少しています。これは(ハイカットなので)ローパスフィルタです。

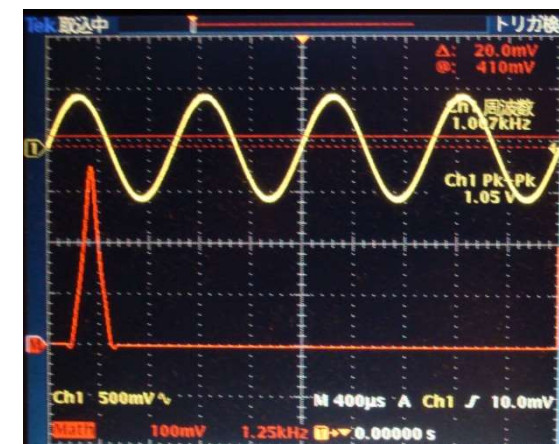
6.4.6 タップ アベレージ・フィルタ 複数段

考察646 : EX641から645 の結果を総合的に考察してください。



この操作は、LTIシステムの、時間軸上での直線畳み込み積分Convolutionです。

アベレージ・フィルタを2回通してさらにグラフは、“なだらか”になります。



これを何度も繰り返していったらどうなるかを考察してください。

6.5.1 本論へ:デジタル・フィルタ

アベレージ・フィルタの例で、デジタル・フィルタの基本形が理解できたと思います。アベレージ・フィルタは、最もシンプルな形のフィルタで、特性に制約があり、有効に使える範囲が限定されます。

実際にはさまざまなデジタル・フィルタが考案されており、多様な特性を実現できるようになっています。フィルタの特性は、フィルタの形式を選択し、係数を求めることで決定します。実装演習のデジタル・フィルタの項でフィルタの型を選択し、必要な特性を入力することで、係数が計算され、フィルタの周波数特性グラフを表示させることができます。

デジタルフィルタの実装に関しては、各自でトライしてください。

この講義では、以降デジタル・フィルタ構成の原理的な面について解説していきます。基礎となるのは、LTIシステムです。このシステムでは、システムのインパルス応答が求まれば、時間関数領域で直線畳み込み積分で、計算ができます。

(フィルタの特性は、連続量では、ラプラス変換で求めた伝達関数で把握できます。離散量の場合は、ラプラス変換の離散版とも言えるZ変換で、伝達関数を求めます。)

6.5.2 線形時不変システム



線形(Linear)性：直線、一次関数であること。

$$H\{ax_1[t]\} = aH\{x_1[t]\} = ay[t],$$

$$H\{ax_1(nT) + bx_2(nT)\} = aH\{x_1(nT)\} + bH\{x_2(nT)\}$$

よって(インパルス応答などの)**重ね合わせ**ができる。

時不変:時間によって性質が変わらないこと。

$$H\{x(nT - \tau_0)\} = y(nT - \tau_0)$$

入力を τ_0 遅らせれば、出力も τ_0 遅れる。

シフティング-**時間をずらした積和演算**を行える。(後述:Z変換で z^{-m})

因果性:出力が現在の値と過去の値にのみ依存する。

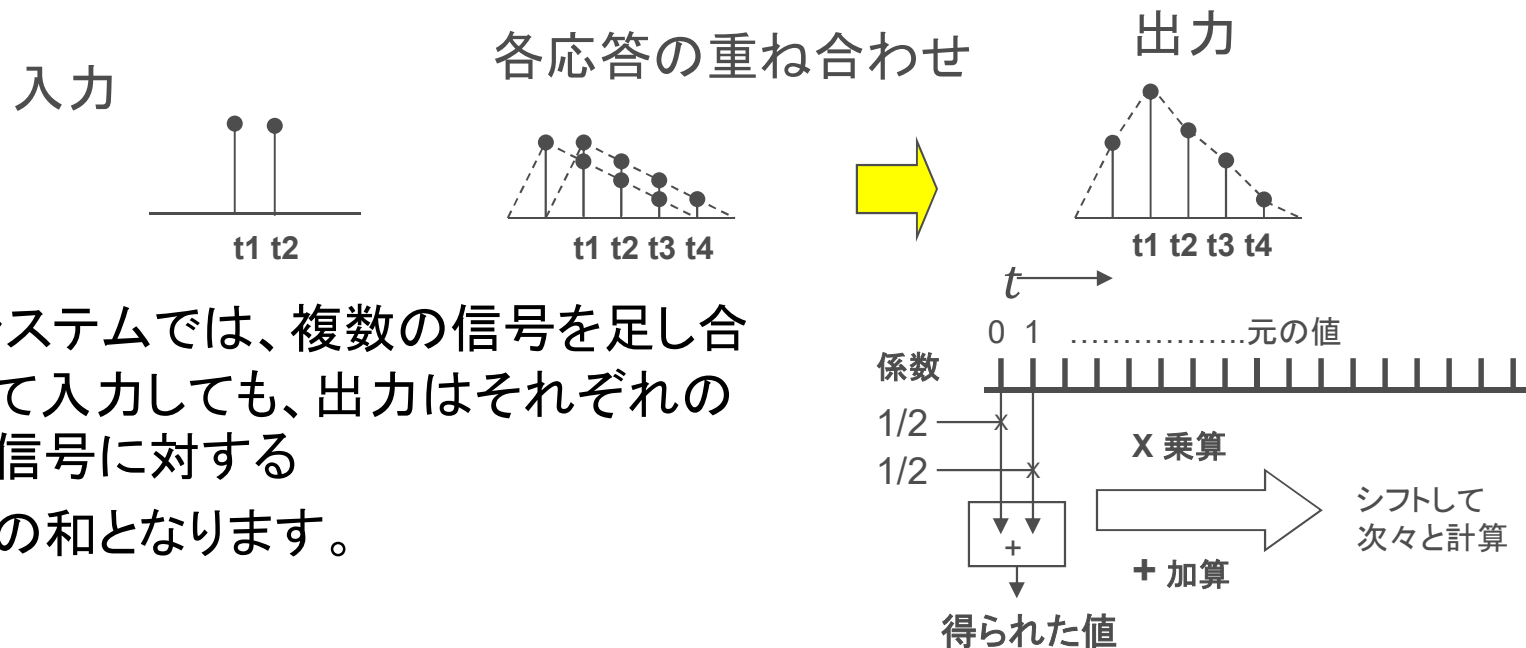
安定性:発散しないこと(入力が有界のとき出力も有界となる。)

演習のアベレージング・フィルタの例で考えてみます。

6.5.3 インパルス応答の重ね合わせ



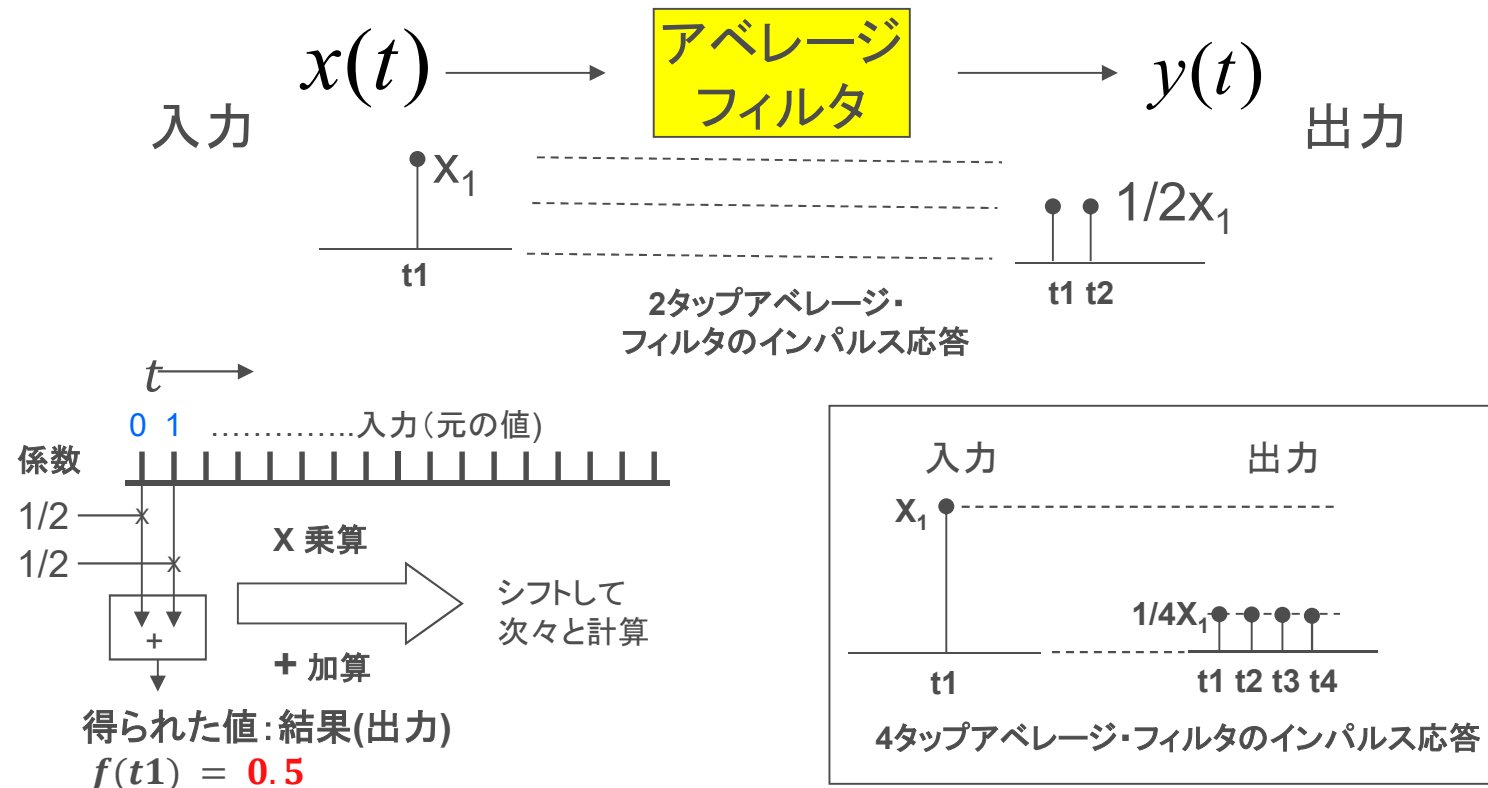
単位インパルスを入力したときに、既知の応答が得られます。
連続したインパルス列を入力し、得られた応答を重ね合わせることができます。



LTIシステムでは、複数の信号を足し合わせて入力しても、出力はそれぞれの入力信号に対する出力の和となります。

6.5.4 アベレージ・フィルタのインパルス応答

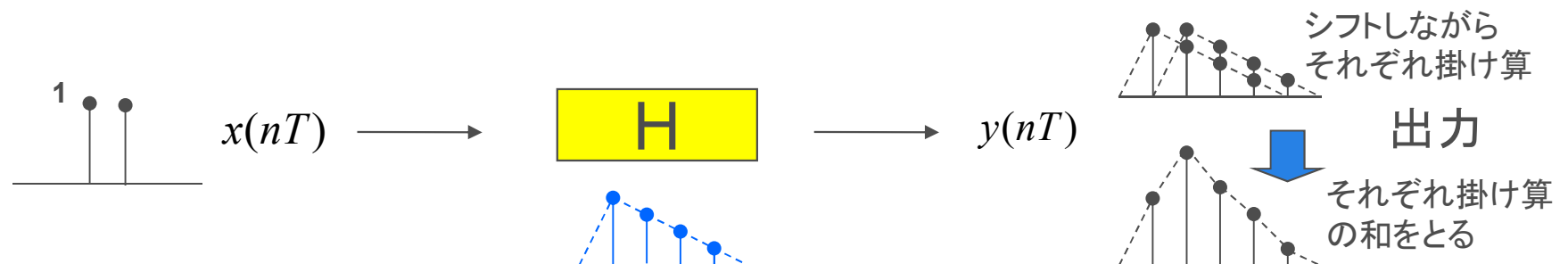
2タップのアベレージング・フィルタをインパルス応答で考えてみると、2タップの平均化は、 t_1 のインパルス応答が、 $1/2$ の値になって(係数 $1/2$ を乗算する)、 $2\Delta t$ 間継続します。よって、このインパルス応答をシフトしながら加算すれば、LTIシステムとしてフィルタ機能を実現できます。



6.5.5 LTIシステムHのインパルス応答とシフト積和



$y(nT) = H[\delta(nT)] =: h(nT)$ システムのインパルス応答



単位インパルス入力に対して、**Hシステムの応答**が既知 **$h(nT)$** とすると、
任意の大きさのインパルス列の連続入力に対しての出力は、それぞれの
入力インパルス値(サンプリング値)に既知のインパルス応答をシフトしながら
かけて、それぞれを加算していけば求められます。

$$y(nT) = \sum_{k=-\infty}^{\infty} x(kT)h(nT - kT) \text{ 畳み込み}$$

時間をシフトしながらの積和演算を(直線状)Convolution-畳み込み積分といいま

6.5.6 インパルス応答と畳み込み

FIRフィルタの構成

$$y(nT) = \sum_{k=0}^n x(kT)h(nT - kT)$$

Z変換では、 z^{-n} は、 n サンプル遅れに相当します。

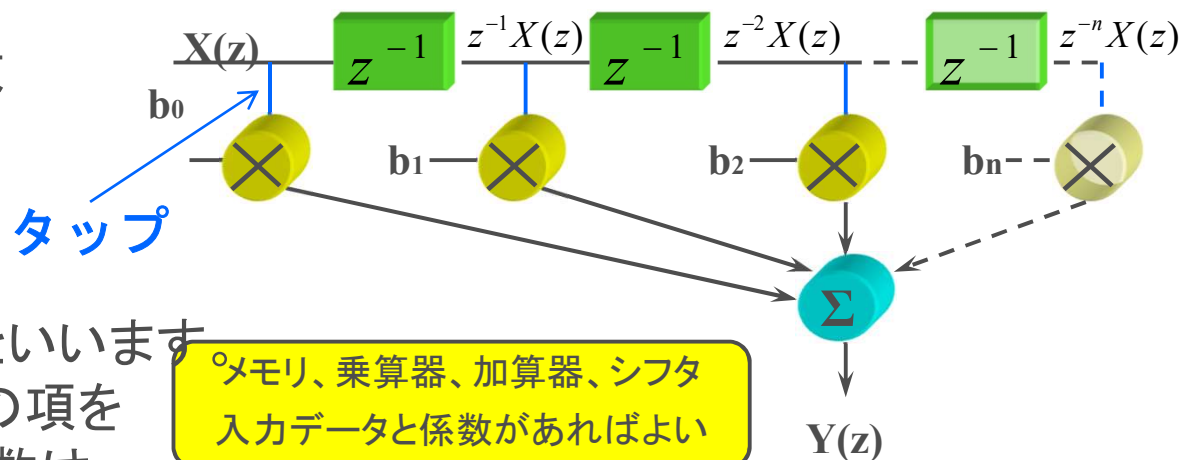
フィルター設計は、システムの伝達関数をインパルス応答に対応させた係数を求めることになります。

n をフィルタの**次数**といいます。係数との乗算は、 z^0 の項を含めますから、**タップ**数は、 $n + 1$ となります。

フィードバックを加えたIIRフィルタもありますので各自で調べてみてください

FIR Filter一般式の例

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) \dots \\ = (b_0 + b_1 z^{-1} + b_2 z^{-2} \dots) X(z)$$

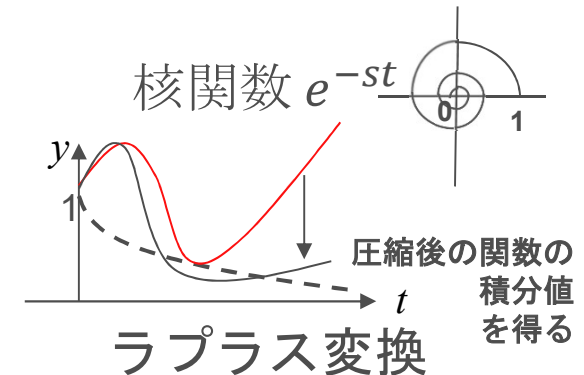


ここまでは、実数の時間関数であることを確認しておきます

6.6.1 Z変換(複素領域への変換)

ラプラス変換では、もとの関数に、核関数(右図点線---、左巻収束関数、実数イメージ)として、 e^{-st} をかけて積分しました。

もとの関数は、圧縮され収束可能になり、変換後の関数は、複素数 s を変数とする関数に変換されました。



Z変換では、もとの関数は離散量(数列)です。この数列に対して、核関数に相当する z^{-n} ($n = 0, 1, 2, \dots$) を項 n に順にかけて、加算して、新しい数列(級数)を作ります。ここで Z は、複素数のため、新しい数列(級数)も複素数になります。

Z変換では、 z^{-n} は、 n サンプル遅れに相当します。これは回路上では、データのシフト操作になります。(Z変換の推移定理)

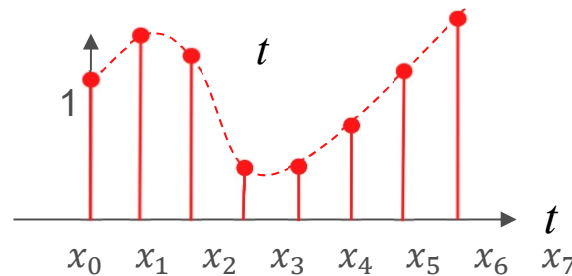
Z変換では
サンプリングされた数列を
 $\{x(nT)\} =$
 $\dots x(-T), x(0), x(T), x(2T) \dots x(nT)$
と表します。

Z変換の定義

$$Z\{x(nT)\} = X(z)$$

$$= \sum_{n=0}^{\infty} (x(nT)) Z^{-n}$$

6.6.2 ラプラス変換との対比でみるZ変換

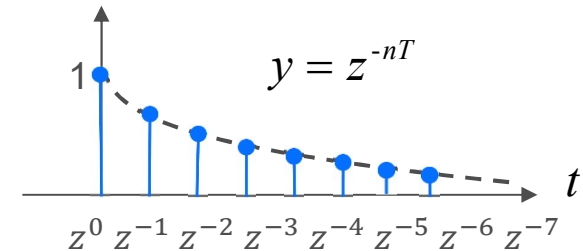


もとのサンプリング数列(実数系列)

Z変換の定義

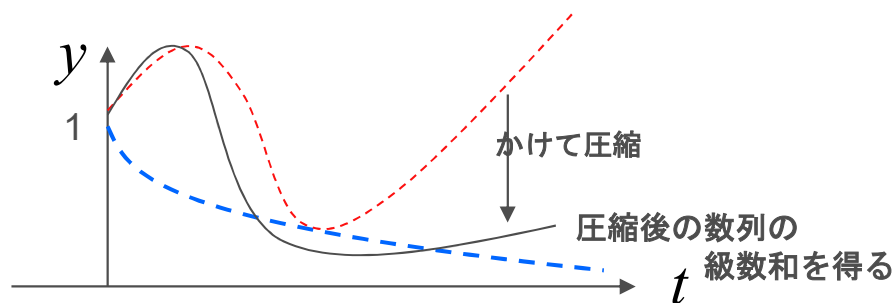
$$Z\{x(nT)\} = X(z)$$

$$= \sum_{n=0}^{\infty} (x(nT))Z^{-n}$$



核関数に相当する、 $\frac{1}{z}$ ずつ収縮
($Z > 1$ の場合)する級数列(複素系列)

x_n と z^{-n} を次々にかけて足しこんで、Z変換を行います。



どのような場合でも収束する
数列(級数)が得られるとは限りません。
Zの値により、級数は、
収束($Z > 0$)、発散($Z < 0$)します。

イメージとして捉えやすくするために、Zを実数としていますが、
実際には複素数です。(複素平面では単位円の外側になります。)

6.6.3 Z変換の重要な性質

- 線形性：二つの級数列のZ変換 $X_1(z), X_2(z)$ を以下とします。

$$X_1(z) = Z\{c_1 x_1(nT)\}, X_2(z) = Z\{c_2 x_2(nT)\}$$

これらの級数列を加算してから求めたZ変換 $X(z)$ は、

$$\begin{aligned} X(z) &= Z\{c_1 x_1(nT) + c_2 x_2(nT)\} \\ &= Z\{c_1 x_1(nT)\} + Z\{c_2 x_2(nT)\} \\ &= c_1 Z\{x_1(nT)\} + c_2 Z\{x_2(nT)\} \end{aligned}$$

以下のとおり別々に求めたものと等しくなります。

$$X_1(z) + X_2(z) = X(z)$$

- 推移定理： $Z\{x(nT)\} = z^{-m} Z\{x((n-m)T)\}$

$m=1$ で考えると、

$$\sum_{n=0}^{\infty} (x(nT)) Z^{-n} = x_0 Z^{-1} + x_1 Z^{-2} + x_2 Z^{-3} \dots + x_n Z^{-n+1}$$

z^{-1} でくくると

$$= z^{-1} (x_0 z^0 + x_1 z^{-1} + x_2 z^{-2} \dots + x_n z^n)$$

1サンプル時間遅れ(シフト)は、 z^{-1} に相当します。 z^{-m} は m サンプル遅延(シフト)になります。

Z変換の定義

$$Z\{x(nT)\} = X(z)$$

$$= \sum_{n=0}^{\infty} (x(nT)) Z^{-n}$$

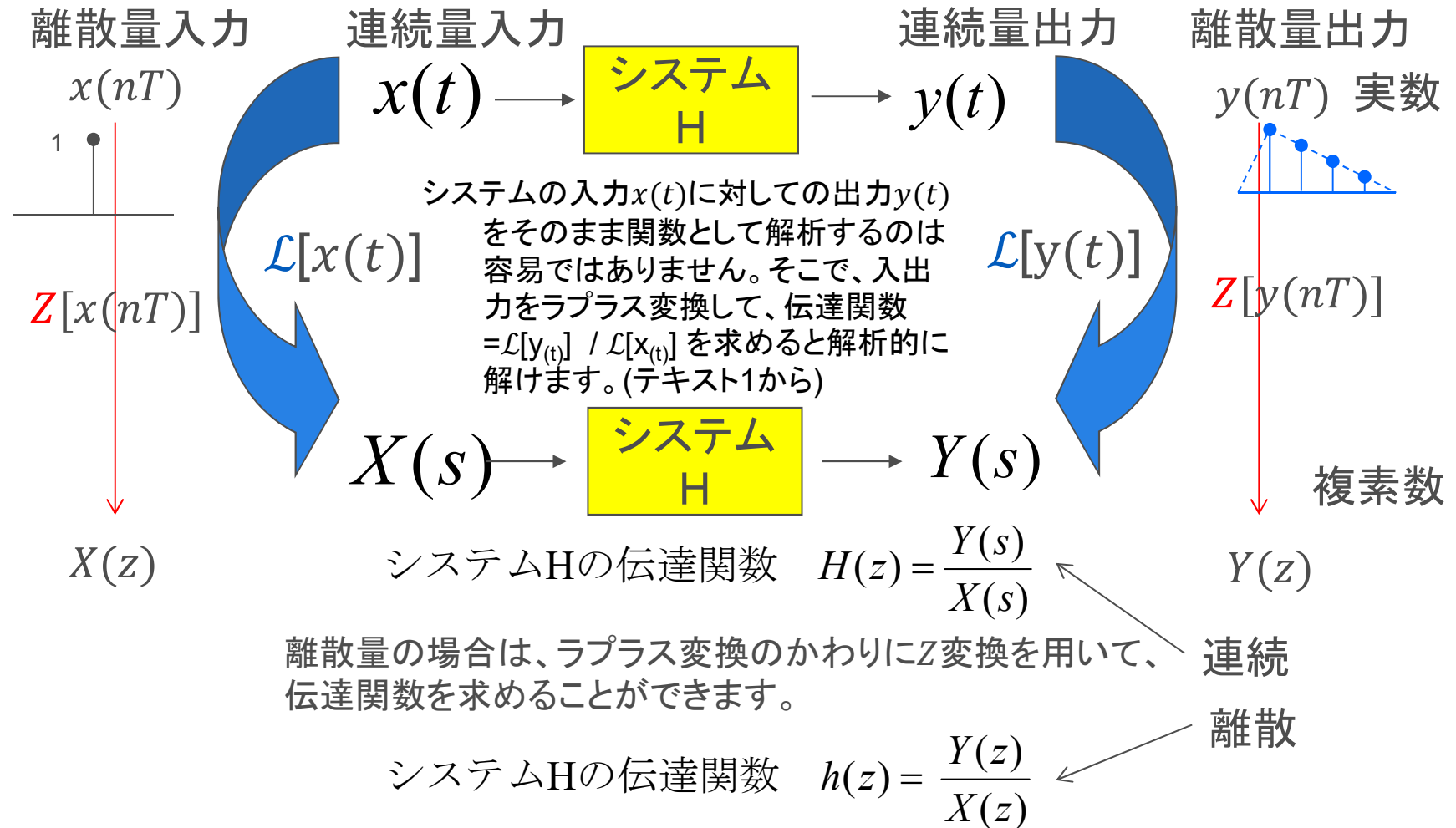
z^{-n} の部分は

初項1公比 $\frac{1}{z}$ の

等比級数和です。

$$\sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - \frac{1}{z}}$$

6.6.4 Z変換と伝達関数



$\mathcal{L}$:ラプラス変換, $\mathcal{Z}$:Z変換, t は時間(実数), s, z は複素数

課題製作のヒント

A. 標準的な実装

PCからWaveGenで発生させた信号をPSoC5に入力しAD変換しFFTを行いスペクトルをLCDに表示する。ノイズ除去にLPFを用いるのもよい。LPFはソフトウェアとDSPの二方法がある。

展開応用: ADにDMAを使用する。FFTをウェーブレット変換やDCT変換に置き換える。周波数領域に変換後に特定の信号処理を行いこれを再度時間領域に戻しその変化と対応を考察する

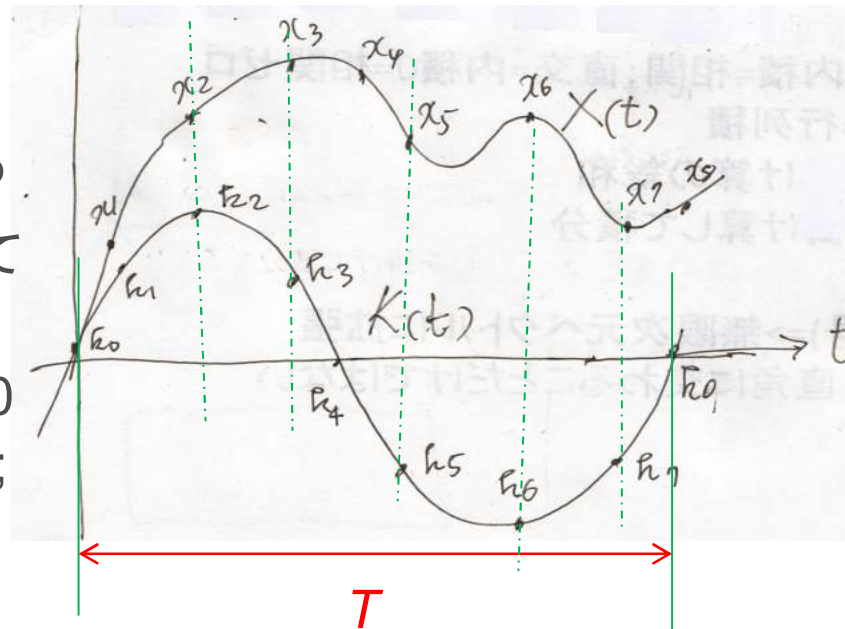
B. 画像を扱うもの(画像は2次元-直交している)

画像データをFFTし、これに特定の処理を行い再度逆FFTして考察する。あるいは画像データにフィルタリング処理(カーネル演算)を行い考察する。FFTを組み合わせでもよい。

行列の計算

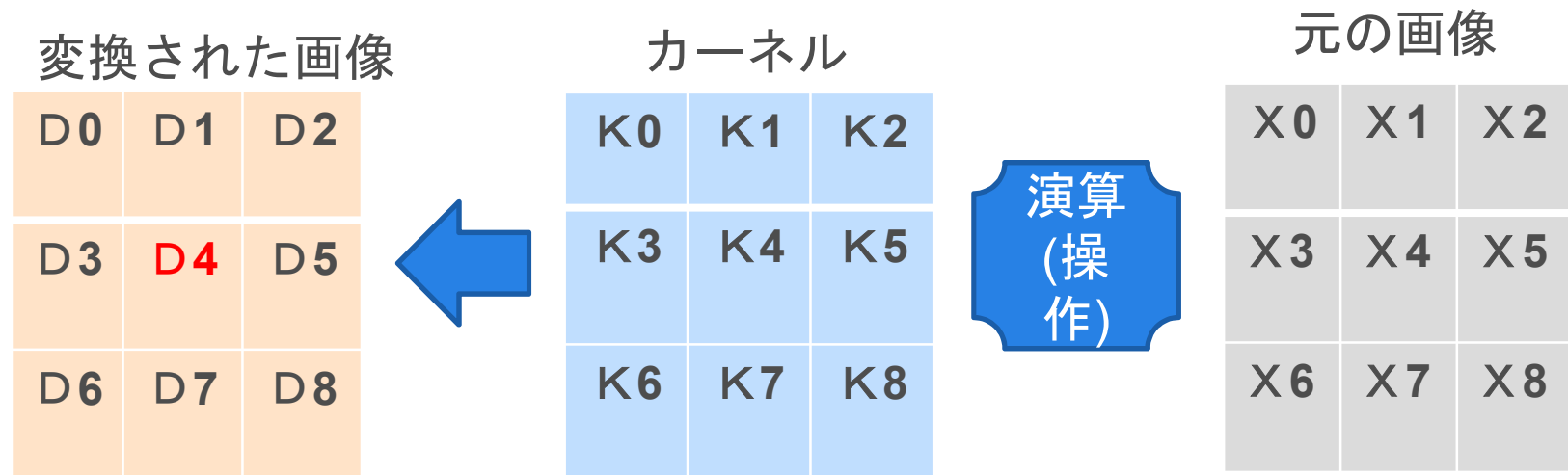
$$\int K(t)X(t)dt \approx \sum_0^{n-1} k_n x_n \begin{bmatrix} \square & \square & \square & \square & \square & \square & \square & \square & \square \end{bmatrix} \begin{bmatrix} \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{bmatrix} \vec{X}(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$

右図は、フリーハンド
なので、 X_n に対応する
 k_n の時間位置がずれて
いますが、実際は
同一時間 t_n における 0
値と考えてください^^;



連続関数の内積(相関)を求める -> 2関数を掛けて積分する ->
関数を離散化すれば離散化数列の積和で内積(相関)が求まる ->
これ(積和計算)は行列の計算になる

画像処理における演算(操作)



画像処理におけるフィルター

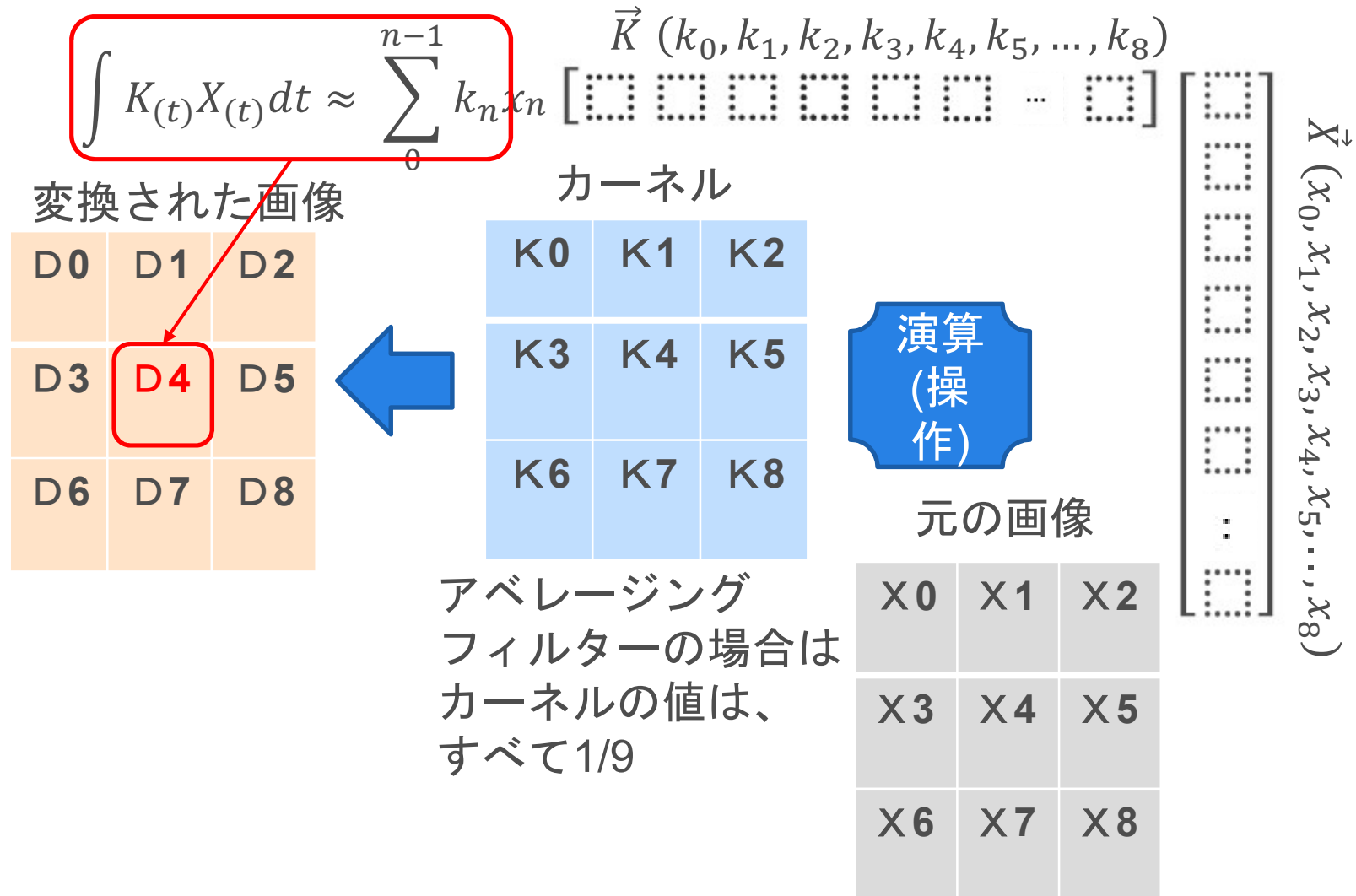
アベレージングフィルターを考えてみる

カーネルは係数行列、元の画像に対しての演算(操作)で変換された画像が得られる。ここでカーネルのKの係数は、 $1/9$ 元の画像X0からX8に対してカーネル係数を掛けて中央**D4**の値を計算する。同様に元の画像を上下に**シフト(ずら)**して**D0 から D8の値を計算する。**

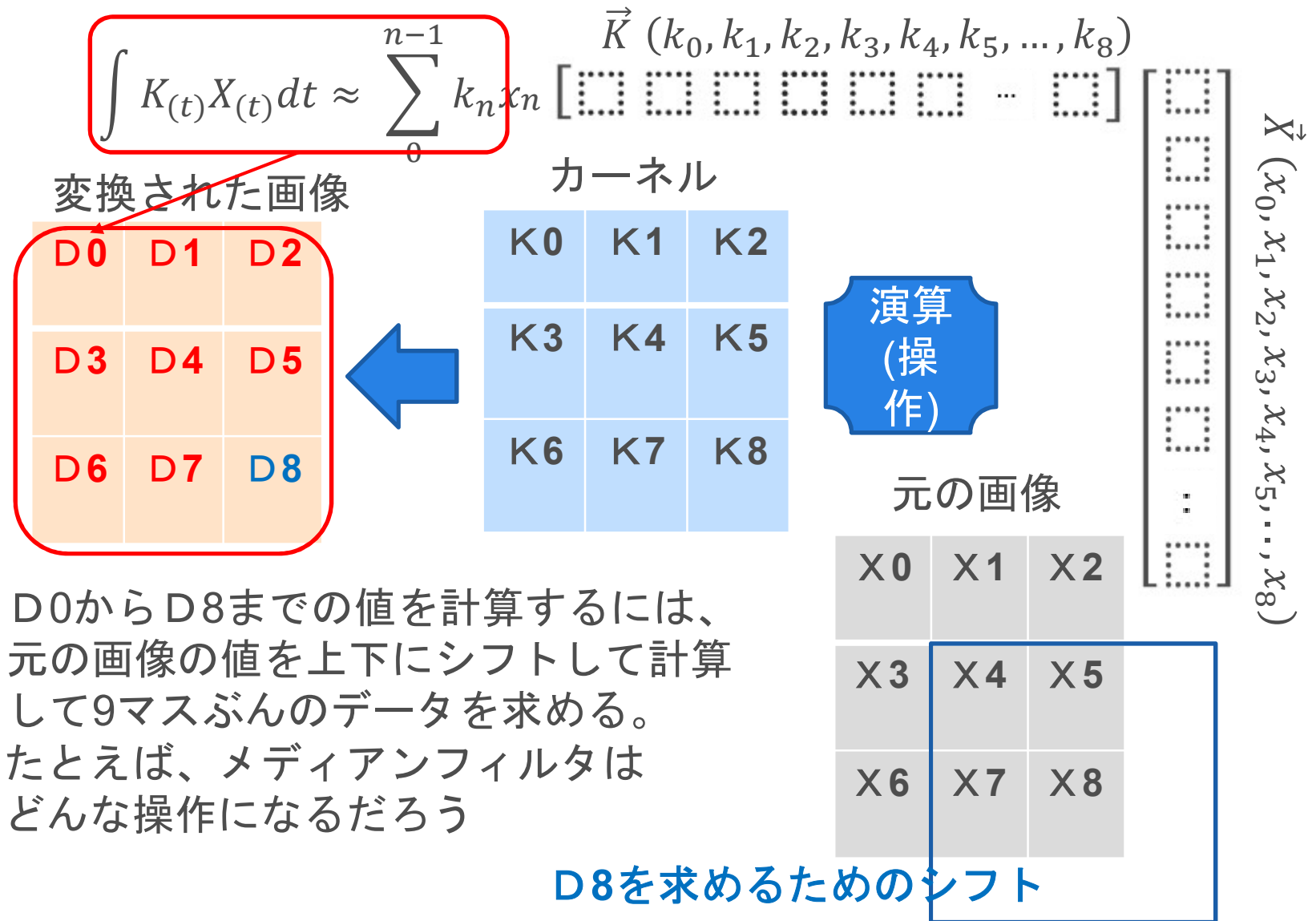
カーネルは係数行列

056

画像処理におけるフィルター カーネルは係数行列



シフトして次々に計算



課題製作のヒント

A. 標準的な実装

PCからWaveGenで発生させた信号をPSoC5に入力しAD変換しFFTを行いスペクトルをLCDに表示する。ノイズ除去にLPFを用いるのもよい。LPFはソフトウェアとDSPの二方法がある。

展開応用: ADにDMAを使用する。FFTをウェーブレット変換やDCT変換に置き換える。周波数領域に変換後に特定の信号処理を行いこれを再度時間領域に戻しその変化と対応を考察する

B. 画像を扱うもの(画像は2次元-直交している)

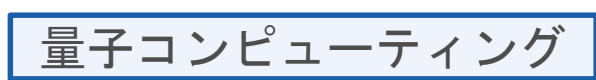
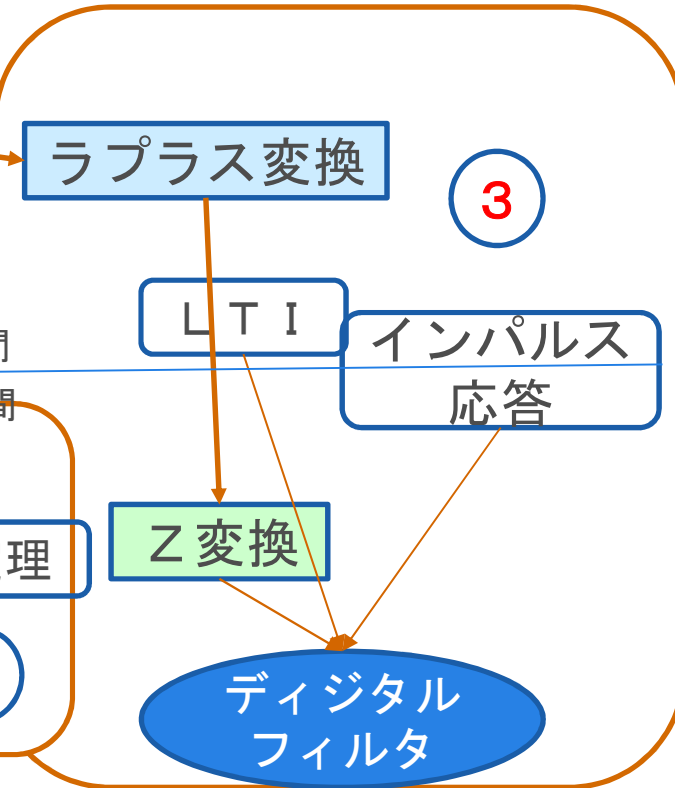
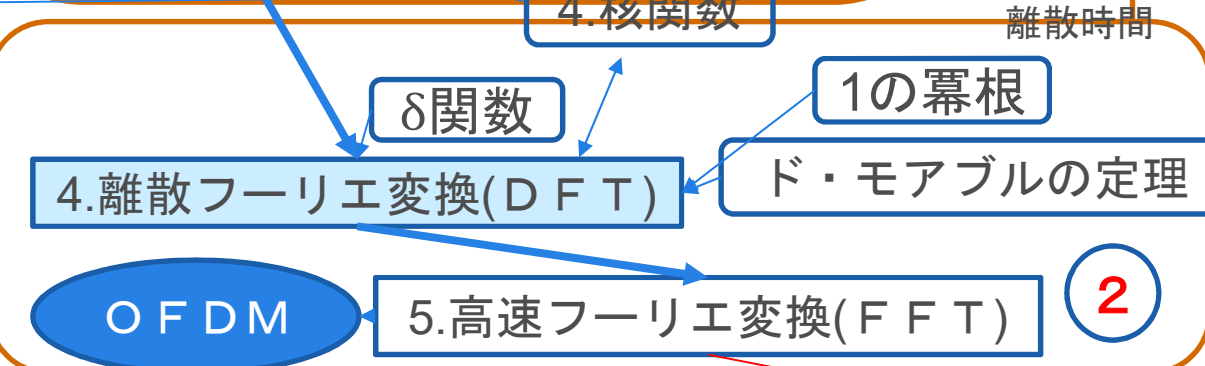
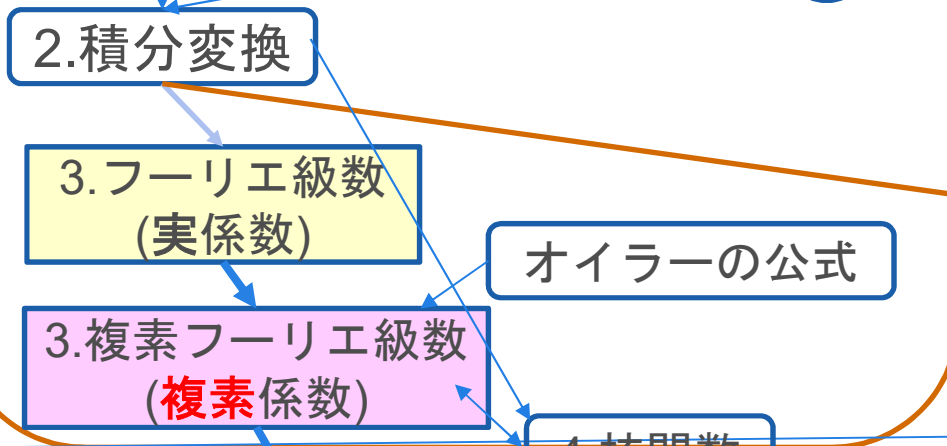
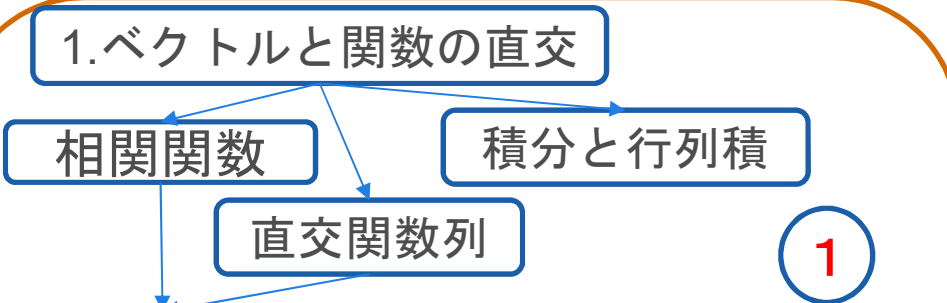
画像データをFFTし、これに特定の処理を行い再度逆FFTして考察する。あるいは画像データにフィルタリング処理(カーネル演算)を行い考察する。FFTを組み合わせでもよい。

講義の流れ(現在位置の確認)

ポイント

- 1.関数を掛けて積分（行列積）
- 2.周期関数と核関数
- 3.連続量の離散化

定理・数理概念



Memo

フォローアップURL (Revised)

<http://mikami.a.la9.jp/meiji/MEIJI.htm>

担当講師

三上 廉司(みかみれんじ)

Renji_Mikami(at_mark)nifty.com

mikami(at_mark)meiji.ac.jp (Alternative)

http://mikami.a.la9.jp/_edu.htm

