仙台市/仙台市産業振興事業団

ロボット博士の基礎からのメカトロニクスセミナー

C07/Rev 1.0

第7回

センサ信号の処理の基礎

仙台市地域連携フェロー

熊　谷　正　朗

kumagai@tjcc.tohoku-gakuin.ac.jp

東北学院大学工学部
ロボット開発工学研究室 RDE

今回の目的

○ センサ信号の処理の基礎

テーマ１：センサの信号と情報

- センサの信号は処理が必要
- 値の変換処理・微分積分

テーマ２：フィルタ＝時間変化する信号の処理

- ノイズ除去系のフィルタ
- 周波数抽出・分析型

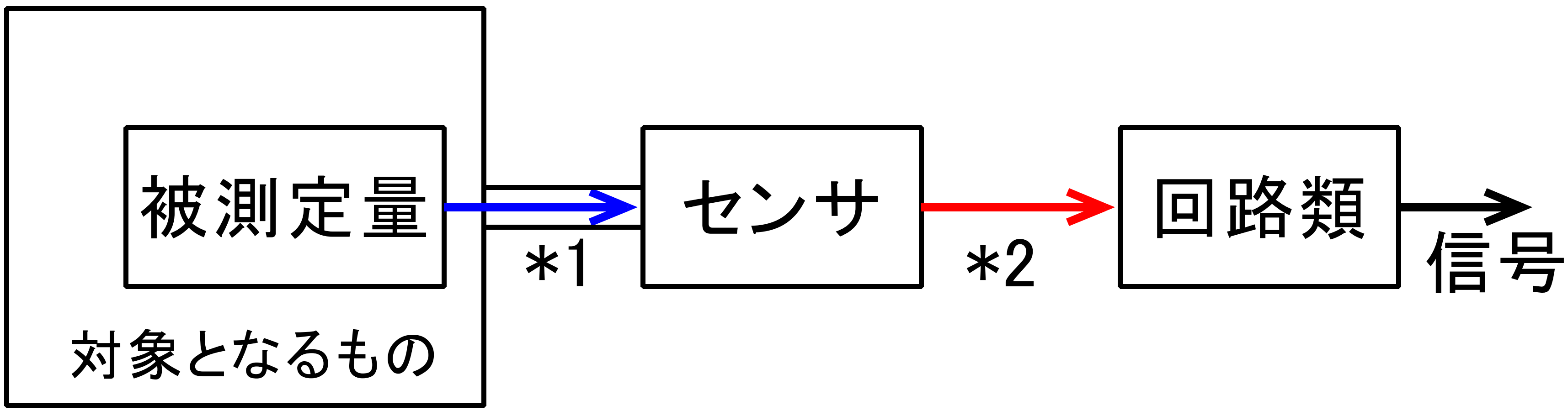
テーマ３：信号処理の実例

- ロボット姿勢センサ等

イントロダクション

○ センサの役割

物理的・化学的現象(*1)を電気的変化(*2)に。



*1　光、温度、圧力、速度、加速度、角速度、
電圧、電流、抵抗、pH、化学物質、等

*2　電圧変化、電流変化、抵抗変化、
電気容量変化、インダクタンス変化、等

イントロダクション

○ センサの信号処理の役割

ただの数値データを情報に

数値データ≠情報

コンピュータに入力したのみでは、
ただのデジタル数値であって、
情報への変換、情報の抽出が必要。

イントロダクション

○ センサ信号の処理の例

- AD変換後のデジタル値→測定値
- 値の微分積分
- フィルタ（時間変化信号の加工）
- 周波数分析
- 画像処理
- 認識(文字、音声)

今日はここまで

※制御も数式上は類似するが、区別されている

値の変換処理

○ センサからの値を被測定量に（個別）

被測定量	→	センサ	→	回路類	→	A/D	→	デジタル値
		変換		増幅等		変換		
量[単位]		S[mV/単位]		K[V/mV]		R[digit/V]		D[digit]

※mVはmAやΩなどの場合有り

得られるデジタル値は

D＝R×（K×（S×量））

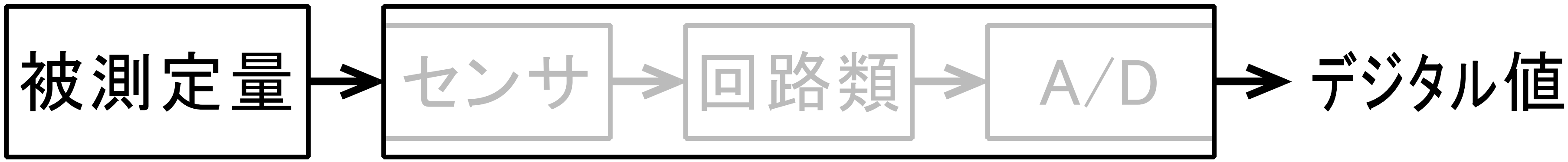
[digit/~~V~~][~~V~~/~~mV~~][~~mV~~/~~単位~~][~~単位~~]

なので

量＝（（D÷R）÷K）÷S

値の変換処理

○ センサからの値を被測定量に（まとめて）



全部まとめたブラックボックス

量[単位]　　K[digit/単位]　　D[digit]

個別に考えて計算するのではなく、被測定量と得られるデジタル値の関係だけを考慮する。

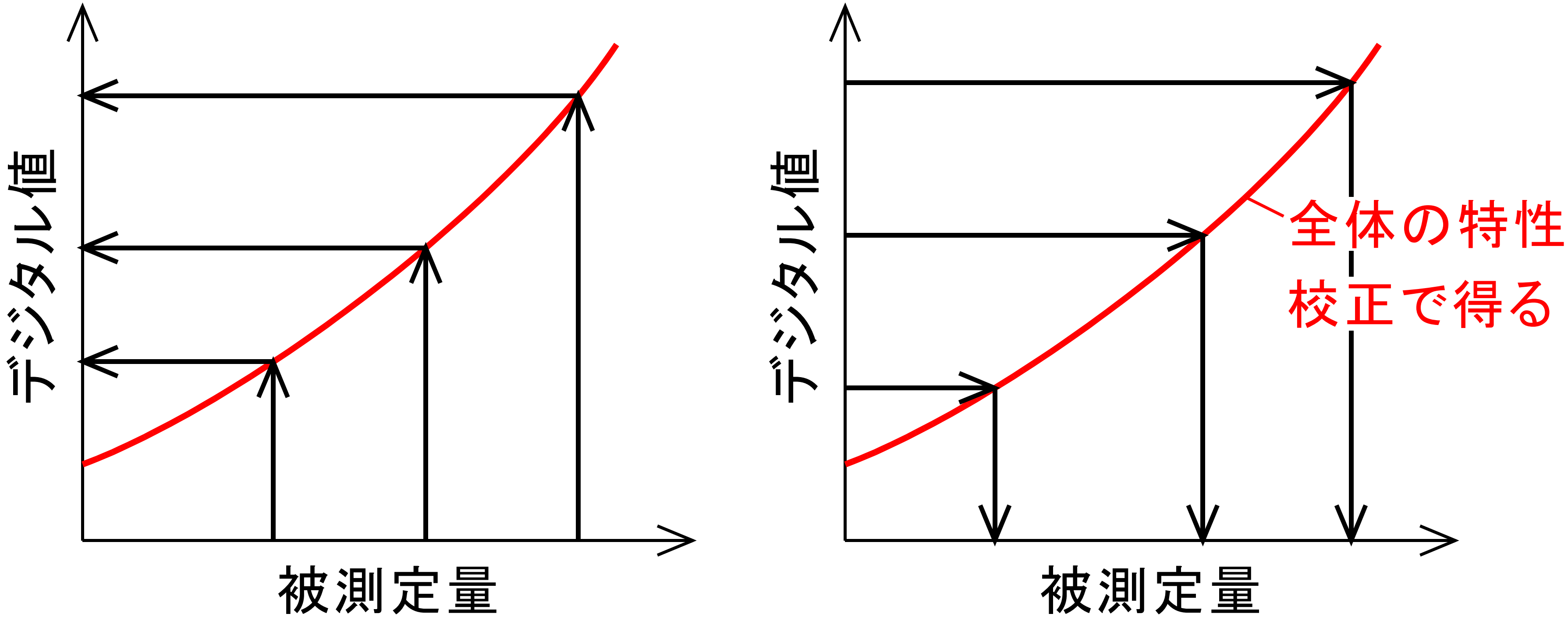
D＝K×量

量＝D÷K

※センサまでだけではなく、機構(回転―直動)や計測法(風速→風車回転数)まで含められる。

値の変換処理

○ センサからの値を被測定量に（比例せず）



測定時:被測定量→変換特性→デジタル

復元:デジタル→変換特性逆読み→被測定量

データ列の処理

○ セットとして、変化傾向として

◇時間的な変化

・時々刻々変化する値を扱う。

・「今の値」でなにか(制御等)する。

・「これまでの値の傾向」から、なにかする。

◇空間的な変化

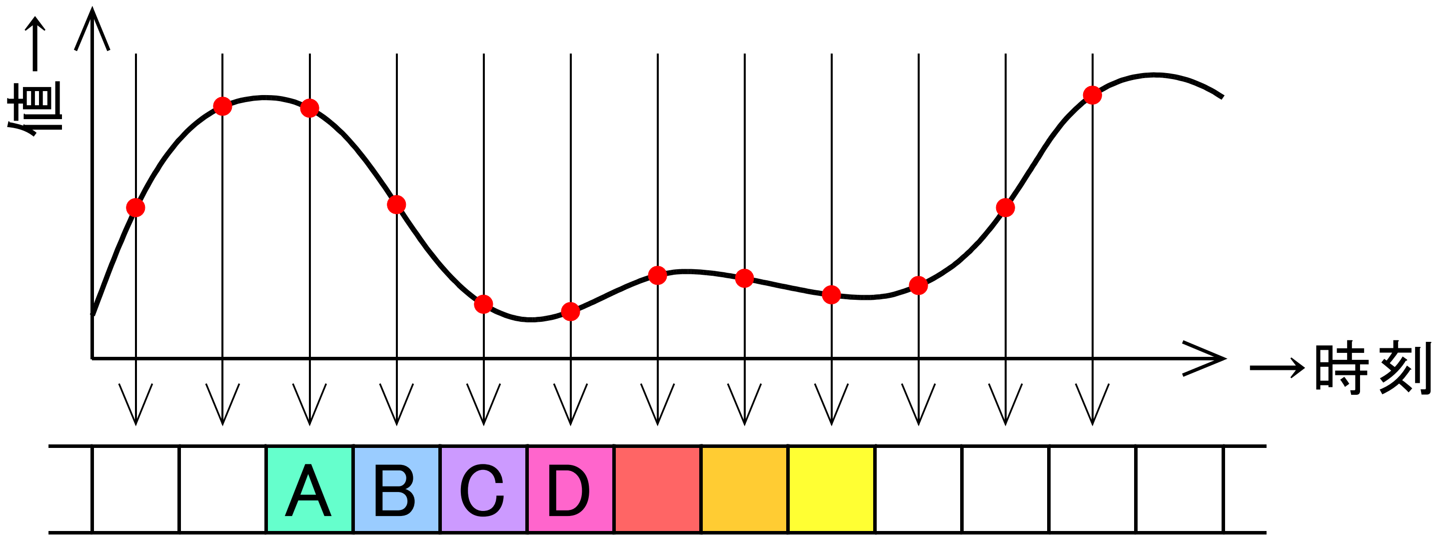
・長さ方向、面方向の値の変化。

・画像データ(画像処理)。

・地図と標高。

時間の系列のデータの処理

○ データのサンプリング（→第５回）



・時間変化するデータを取るときは、一般に一定の時間間隔で行う(サンプリング周期)。

・取得したデータの系列を処理する。

微分と積分

○ センサの選択肢を増やす処理

◇微分

・信号の単位時間あたりの変化を得る。

・瞬間的な速度。

・位置センサ→速度→加速度

◇積分

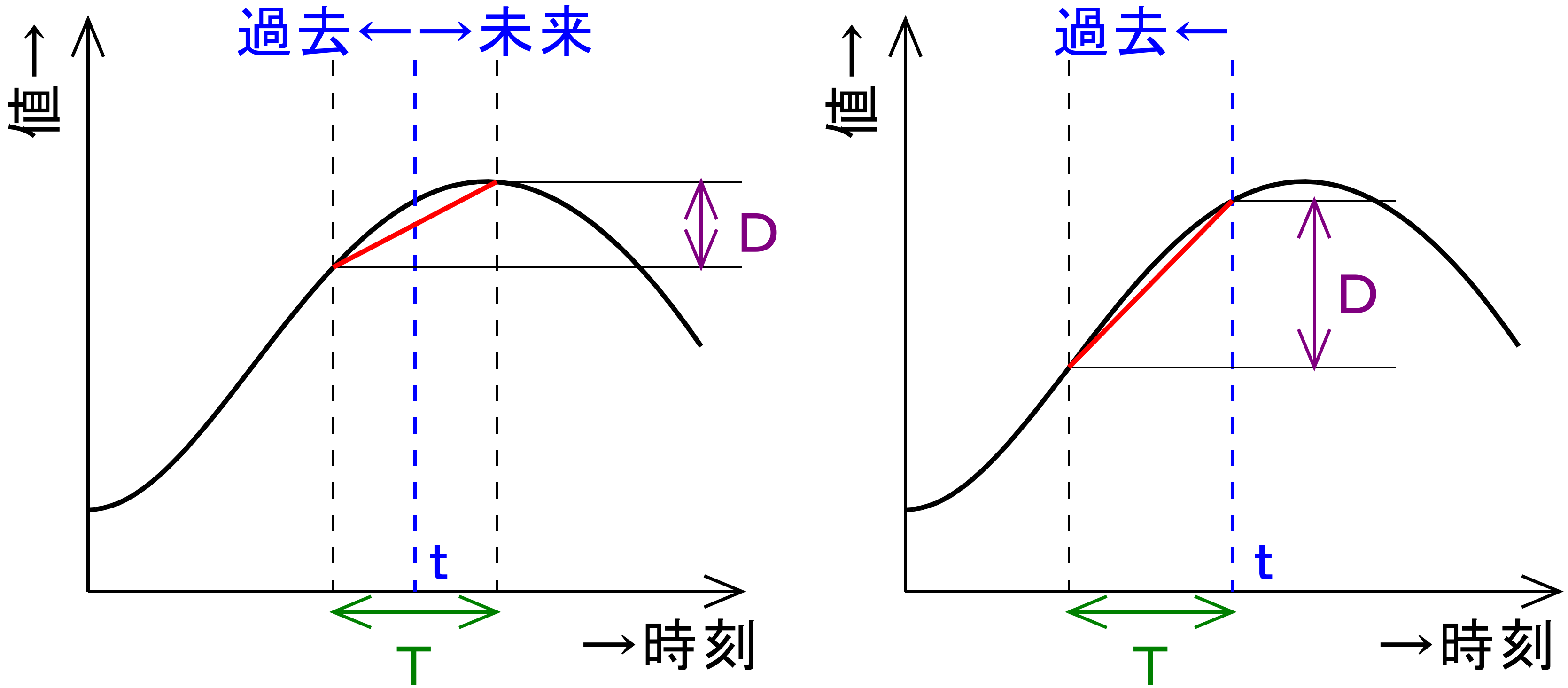
・微分の逆、値を積み上げる。

・速度に対する位置の関係。

・角速度センサ→角度

微 分

○ 瞬間ごとの速度を求める

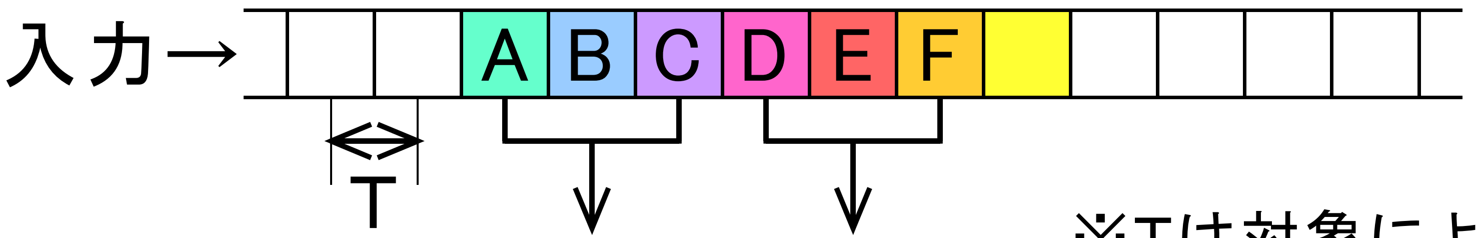


・ある時刻tで間隔Tの間の変化Dを求める。

・D/Tがその瞬間の速度。

微　分

○　系列データからの算出　その1



※Tは対象によって決定

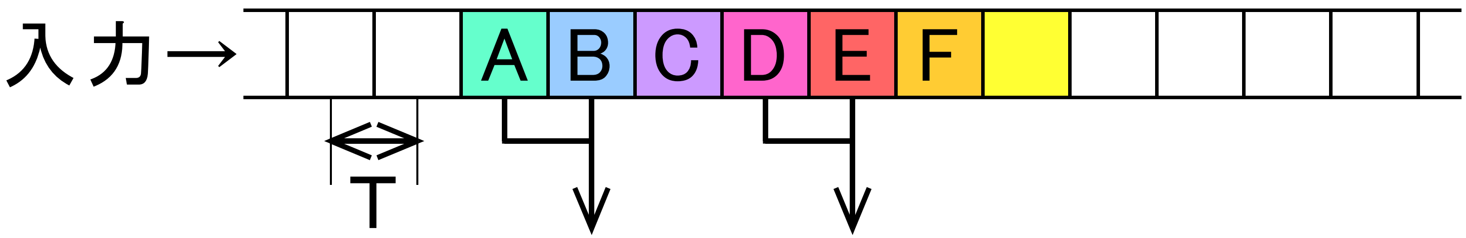
・データは時間T間隔でサンプルされている。

・Bの時点での微分値：(C−A)/2T
　Eの時点での微分値：(F−D)/2T

・「未来の値」を使う。
　＝Bの時点ではCはまだ得られていない

微　分

○　系列データからの算出　その2



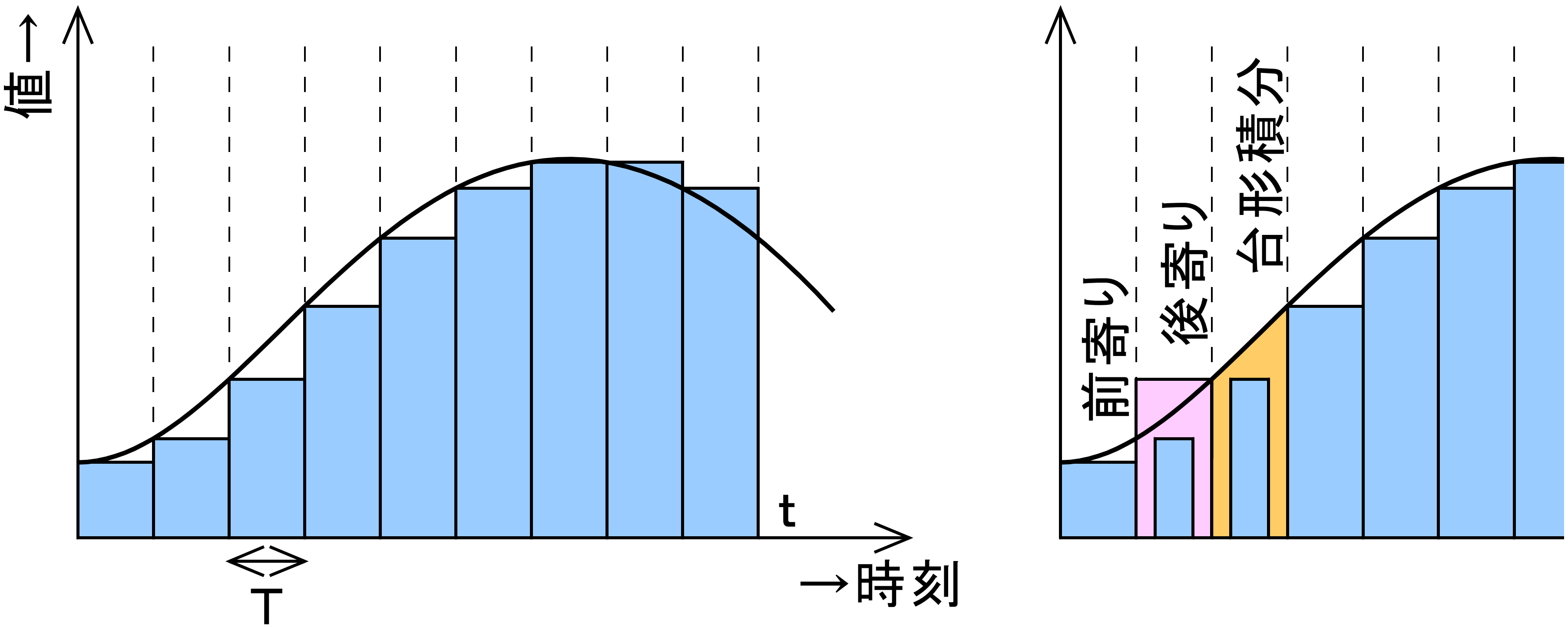
・データは時間T間隔でサンプルされている。

・Bの時点での微分値：(B−A)/T
　Eの時点での微分値：(E−D)/T

・未来の値を使わない。
　＝Bまでのデータで問題なく計算できる。

積　分

○　面積を求める

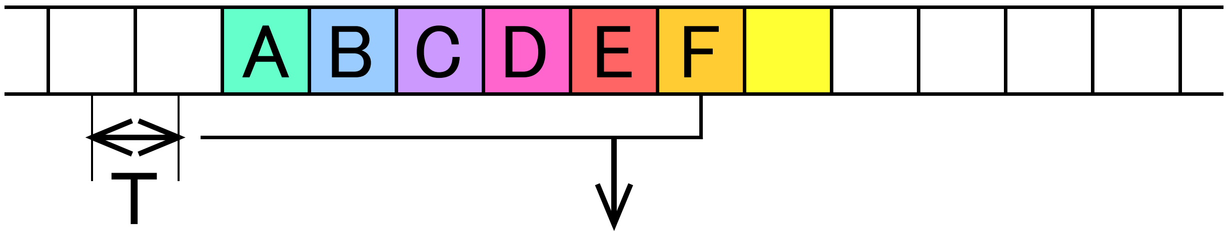


・波形と横軸で囲まれた面積。短冊の和。

・高さ＝短冊区間の前の値、後、両方(台形)。

積　分

○ 系列データからの算出

入力→


・データは時間T間隔でサンプルされている。

・短冊の面積＝…A×T、B×T、C×T…

・積分値＝…(A×T)＋(B×T)＋(C×T)…

・データが得られるたび加算

S＋(A×T)→S、　S＋(B×T)→S

※C言語表記　S+=A*T,　S+=B*T,　..

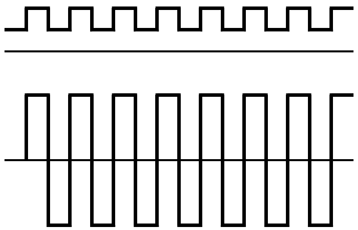
微分と積分

○ 演算上の注意

◇積分

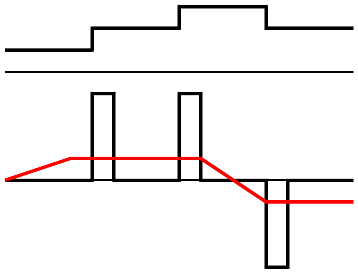
・変数のオーバーフローに注意。

◇微分 (AD値など飛び飛びの値のとき、特に)



・ノイズに注意　　※T=1ms=0.001

{1,2,1,2,1}→{1000,−1000,1000,−1000}



・ゆっくりとした変化に注意

{1,1,1,2,2,2,2,3,3,3,3,4,4,4,4}→

{0,0,0,1000,0,0,0,1000,0,0,0,−1000}

今回の目的

○ センサ信号の処理の基礎

テーマ1：センサの信号と情報

・センサの信号は処理が必要

・値の変換処理・微分積分

テーマ2：フィルタ＝時間変化する信号の処理

・ノイズ除去系のフィルタ

・周波数抽出・分析

テーマ3：信号処理の実例

・ロボット姿勢センサ等

デジタルフィルタ

○ 信号系列に対する処理

◇アナログフィルタとの対比

・目的は同様。主に周波数特性的加工。

・アナログフィルタと似たものが作れる。

◇使用する意義

・主に信号からのノイズ除去。

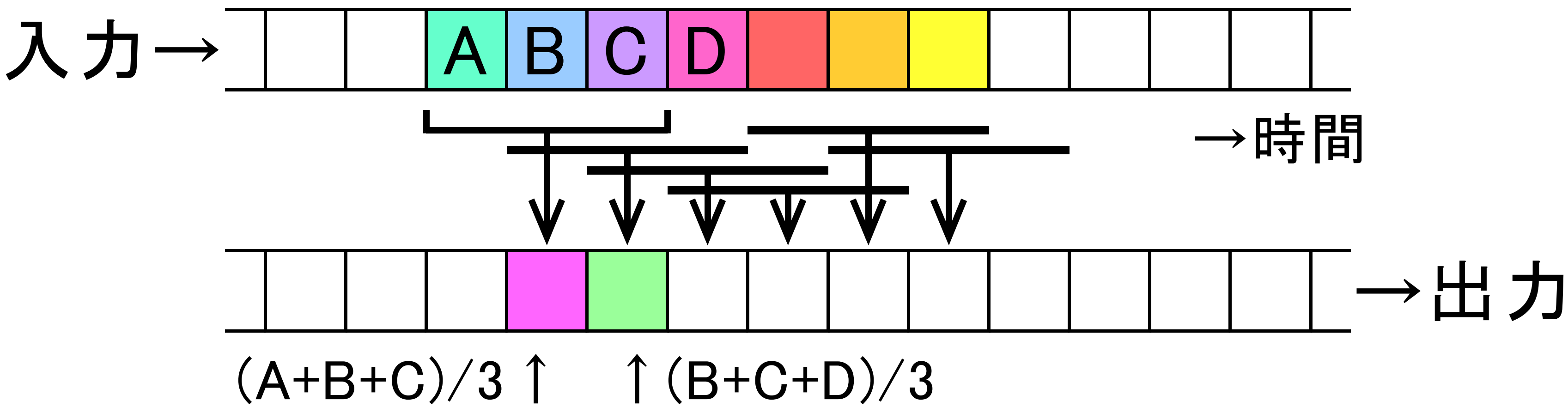
・急激な変化からの制御の保護。

◇処理の実現方法

・C言語等によるプログラム/表計算ソフト

デジタルフィルタ

○ 移動平均フィルタ

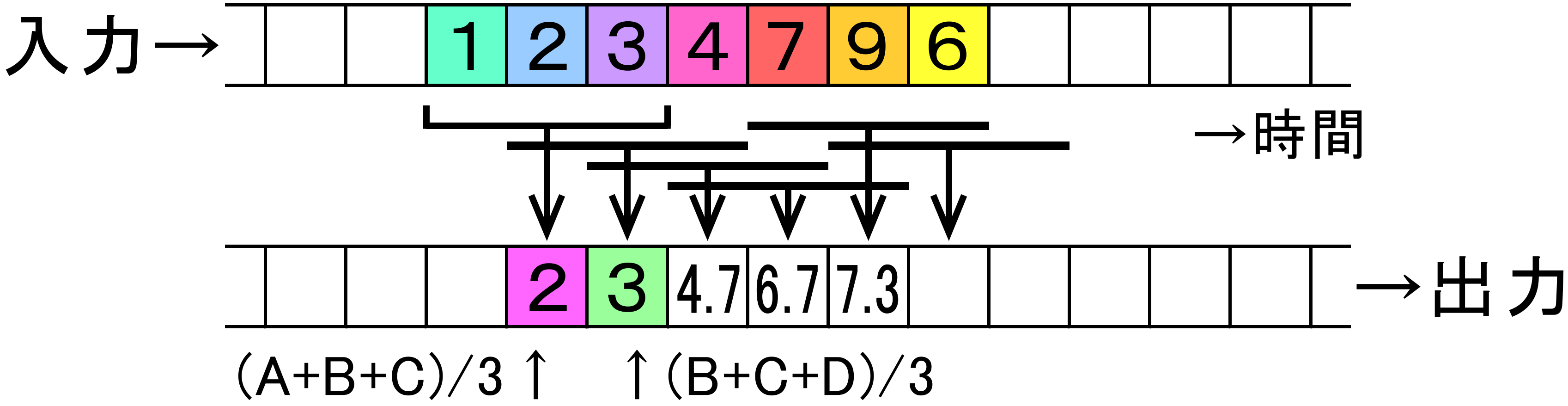


・順番に流れてくる(記録された)データから、
いくつか(上例では3個)を平均したものを
出力とするフィルタ。

・ローパス特性(高い周波数を減らす)がある。

デジタルフィルタ

○ 移動平均フィルタ



・ノイズや小さな変動を消すのには簡便。

・リアルタイム用途(制御など)には不向き。

・あまりローパス能力は高くない。

・特定の周波数に強力な除去能力。

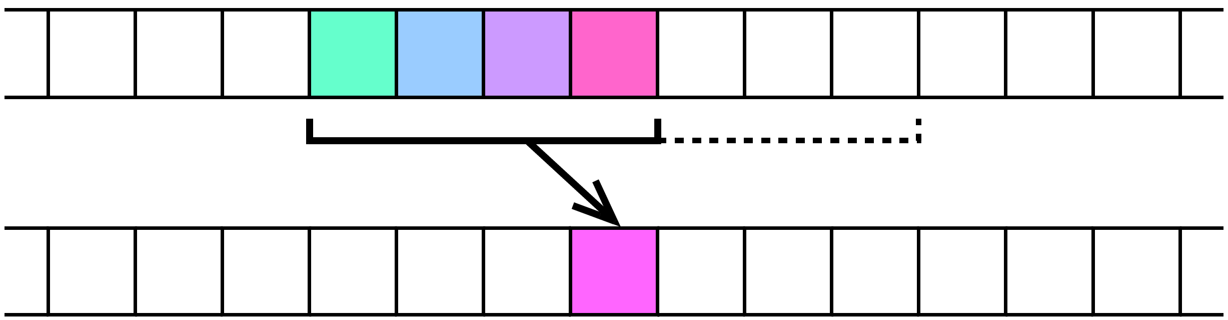
信号処理の「過去・今」と「未来」

○ リアルタイム処理か計測後処理か

◇リアルタイム処理

・制御で使用する信号などでは、その場で得られる値＝「今の値」と「過去の値」しか処理に使えない。

→処理をすると必ず時間的遅れを伴う。



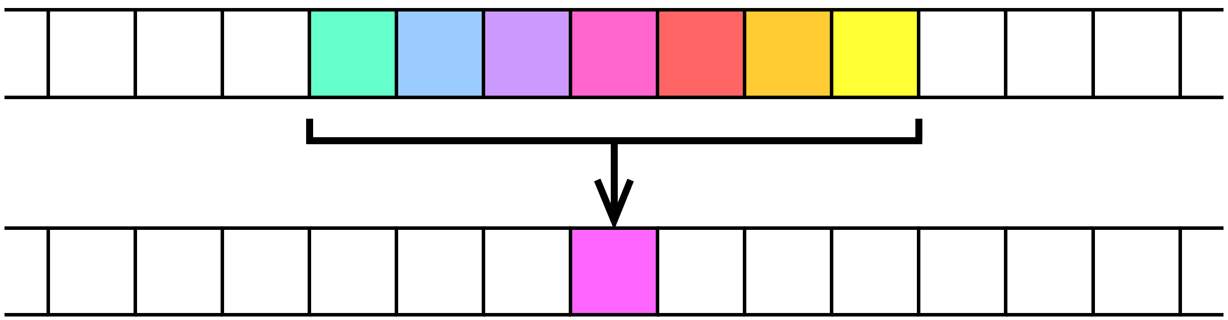
信号処理の「過去・今」と「未来」

○ リアルタイム処理か計測後処理か

◇計測後処理(オフライン処理)

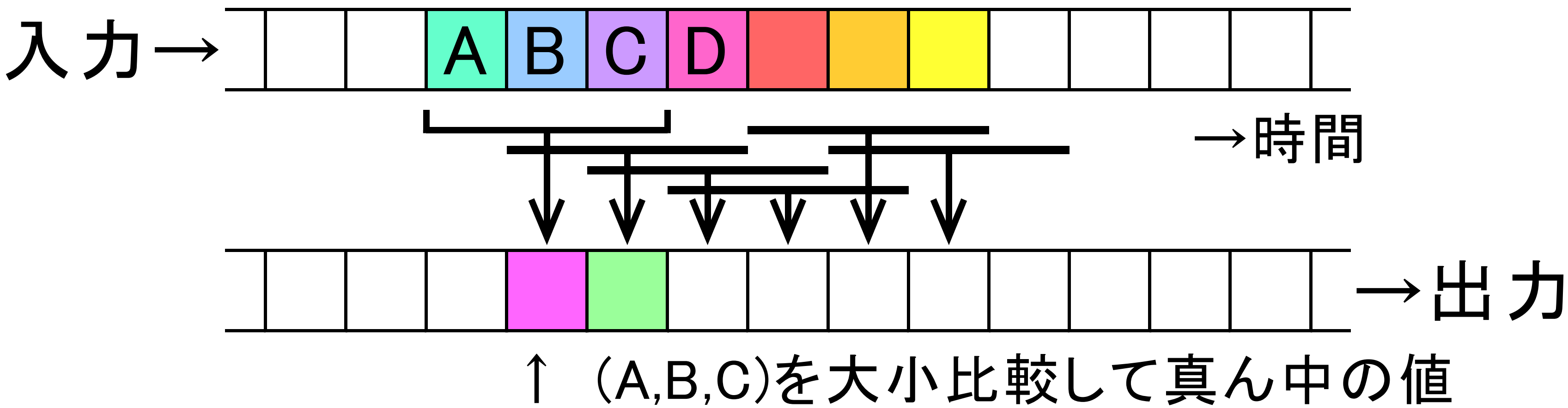
・計測後に処理していいなら、「未来の値」を使うことが出来る。
→時間的遅れのない処理が可能。

・「全体的遅延OK」でも使える(音声、映像等)



デジタルフィルタ

○ メディアン(中央値・中間値)フィルタ

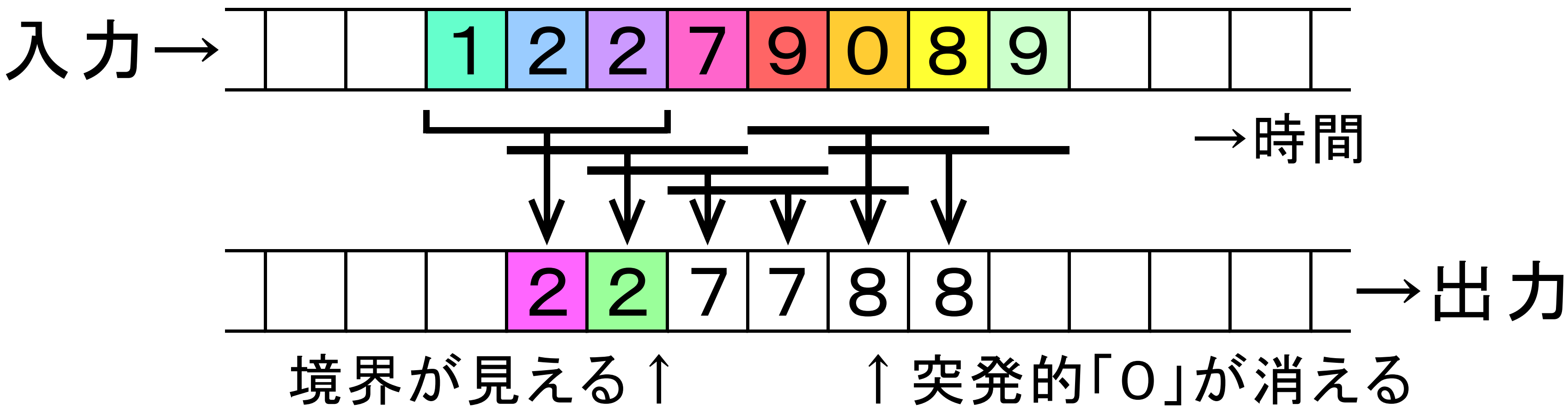


・いくつか(上例3個)のデータを大きい順に並べ、
真ん中の位置の値(中央値)を出力とする。

・突発的な値を除去する一方で、
変化の傾向はそこそこ維持できる。

デジタルフィルタ

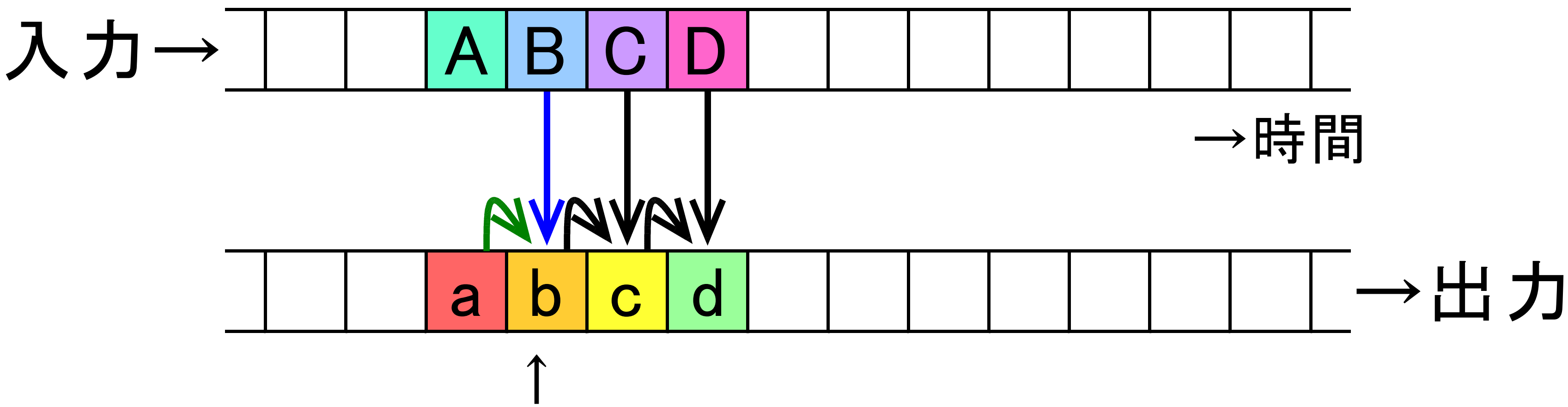
○ メディアン(中間値)フィルタ



- 突発的な大きなノイズ(スパイクノイズ)の除去に効果的で、画像処理(ごま塩対策)で有名。
- 「並び替え」で処理の手間が大きい。
- 表計算ソフトでは手軽に処理できる。

デジタルフィルタ

○ １次ローパスフィルタ



・入力だけではなく、直前の出力も使う。

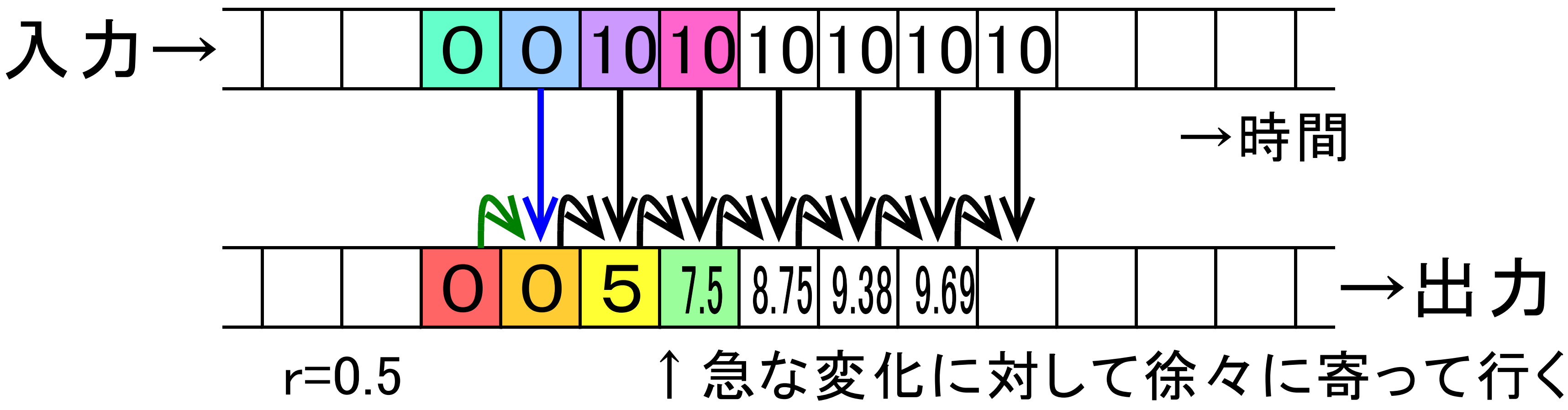
b= r×B ＋ (1−r)×a　　rは0～1

・比率rが0に近いほど、変化が通りにくい。

※r=0→b=a(出力変わらず) r=1→b=B(入力そのまま)

デジタルフィルタ

○ １次ローパスフィルタ



・出力=0.5×入力＋0.5×直前の出力

・徐々に入力値に近づいていく。

・rを小さく(0.01等)すると信号がなだらかに。

・簡単な式で効果がある。

デジタルフィルタ

○ そのほかのフィルタ

◇フィルタの種類

・周波数の低い成分を減らすハイパスなど
様々あり、設計手段も用意されている。

・アナログフィルタより一般に高性能。
（特性、精度）

◇FIR型とIIR型

・FIR型：入力のみから出力を作る。

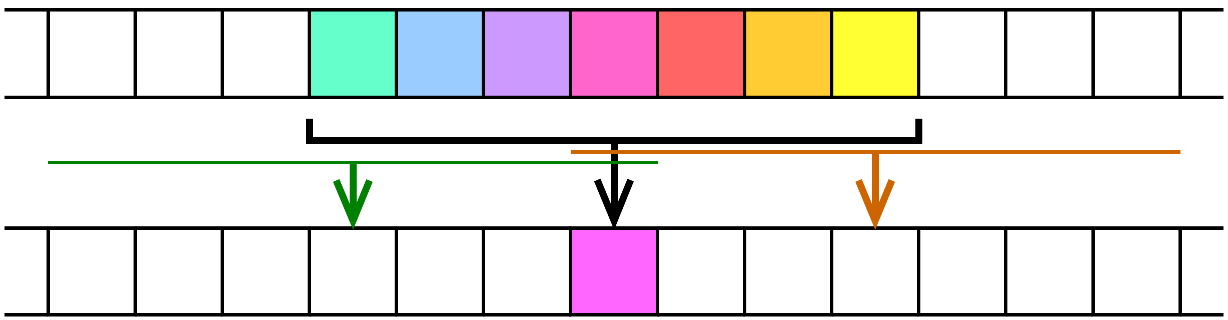
・IIR型： 入力と過去の出力から算出。

※厳密な定義は異なる

デジタルフィルタ

○ FIR(有限インパルス応答)フィルタ

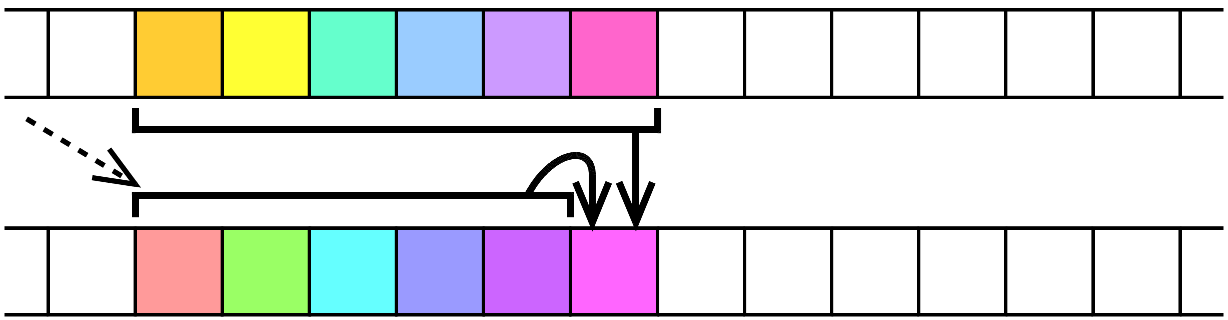
・ある瞬間の入力が出力に及ぼす影響は時間的に限定されている。
・長周期の信号処理には計算量が増大。
・特性の数学的設計をしやすい。
・後処理向き，制御不向き。



デジタルフィルタ

○ IIR(無限インパルス応答)フィルタ

- ある瞬間の入力はその先ずっと出力に影響し続ける。
- 長周期の信号処理が 少ない計算で可能。
- フィルタ設計はアナログの手法を用いる。
- 制御、アナログフィルタの置き換え向き。



周波数分析

○ 信号に含まれる特定周波数の抽出

◇すべての信号は正弦波(＋余弦波)に分解できる

・フーリエ級数、フーリエ変換

・何らかの信号に含まれる、特定の周波数に注目して測定したい。

◇周波数分析の例

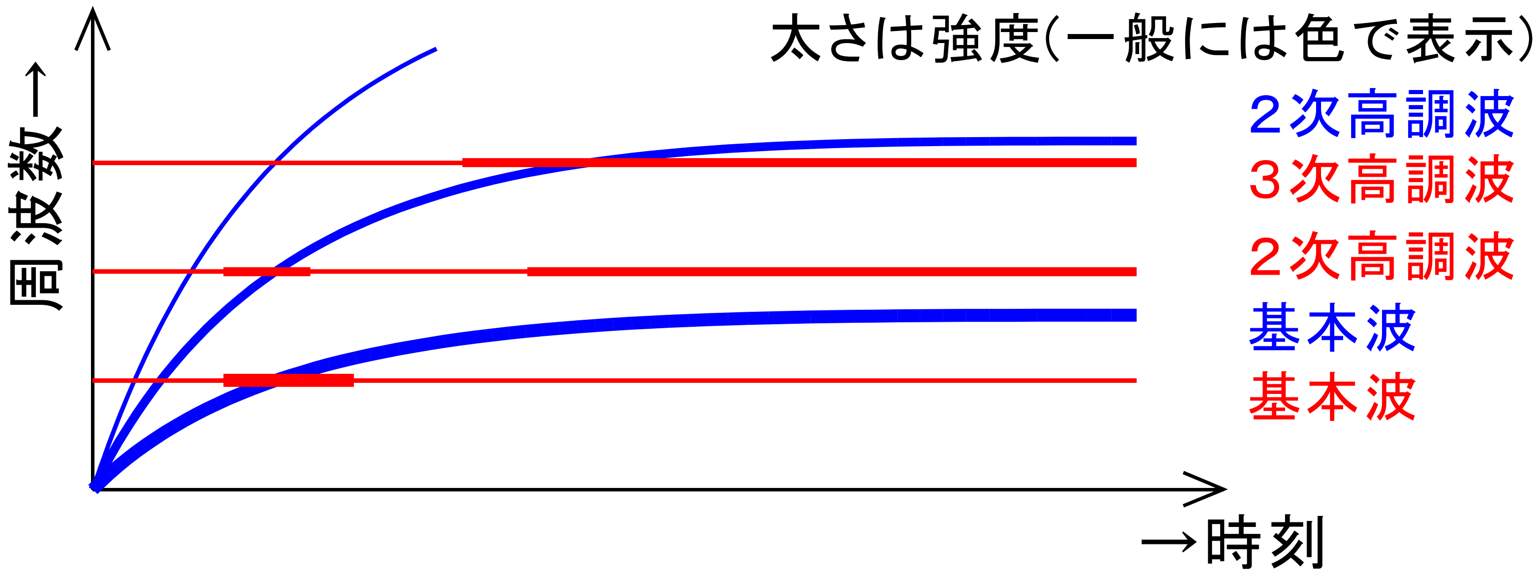
・機械振動やノイズの特性確認→原因調査

・ノイズに強い信号分析

・声や楽器に含まれる周波数成分

周波数分析

○ 機械振動の解析 (自動車等回転機械)



・機械の振動は、回転由来＋固有振動

・時間とともに周波数が変化＝回転による

・周波数一定で強度が変わる＝固有振動

周波数分析

○ ノイズに強い測定 （微弱信号の検出）

ある周期の出力をするセンサ → 特定周波数の信号のみを抽出する

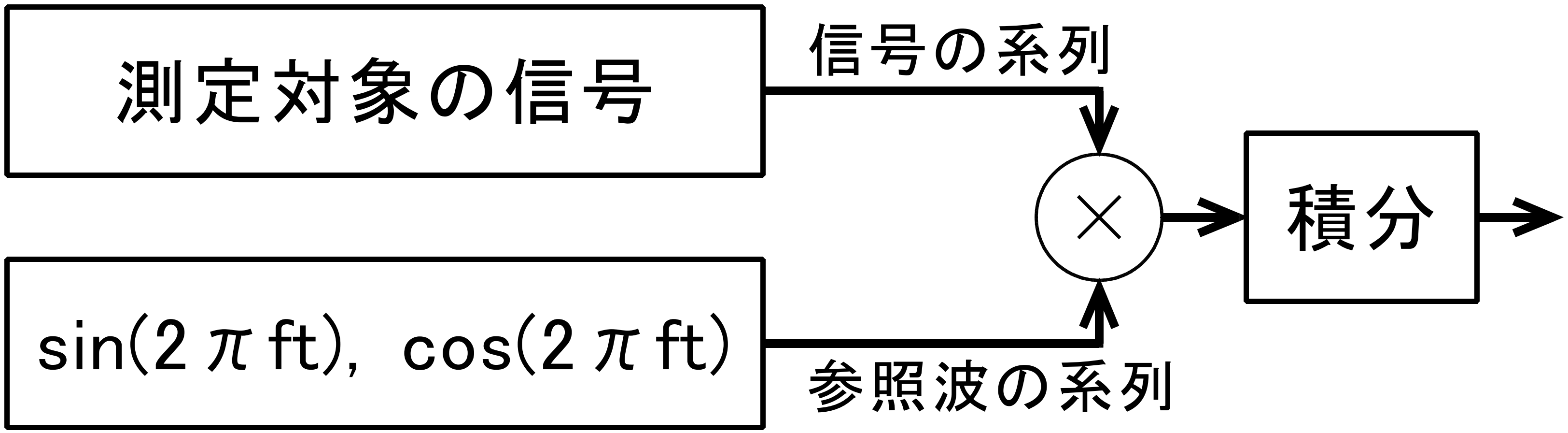
周期的な刺激 → 測定対象 →

・特定の周波数の信号成分のみを測定することで、その他の信号(＝ノイズ等)の影響を受けにくい計測が可能。

※ロックインアンプはこの一種。

周波数分析の方法

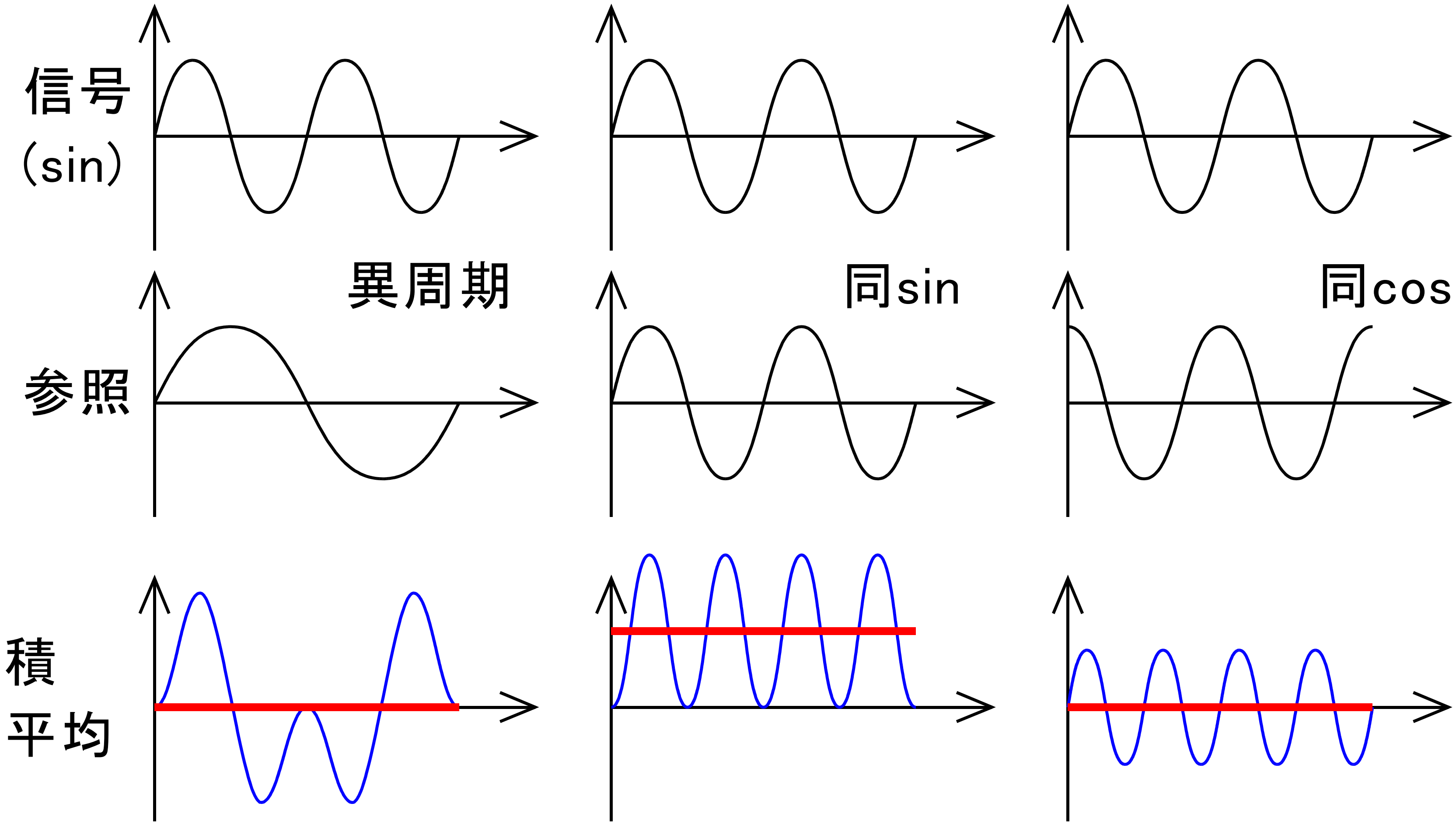
○ 一つの周波数成分の検出



・検出対象としたい周波数のsin(正弦波)とcos(余弦波)を対象信号に乗じて、時間平均を求める (一定期間積分をする)。
→その周波数が含まれる大きさが得られる。

周波数分析の方法

○ 一つの周波数成分の検出：例



周波数分析

○ 基本的性質

・一般の信号には多数の周波数成分が入っているが、混じっていても特定の成分にのみ反応する。

※近い周波数もある程度反応する

・同周波数でも正弦波と余弦波は別扱い。

→ $\sqrt{(\sin\text{成分})^2+(\cos\text{成分})^2}$ で振幅

sin成分とcos成分の比で位相

sinとcosを独立に使うこともできる。

周波数分析

○ FFT(高速フーリエ変換)

・「特定の成分」のみではなく、「含まれる成分の大きさの一覧」を得たいときは、フーリエ変換を利用する。

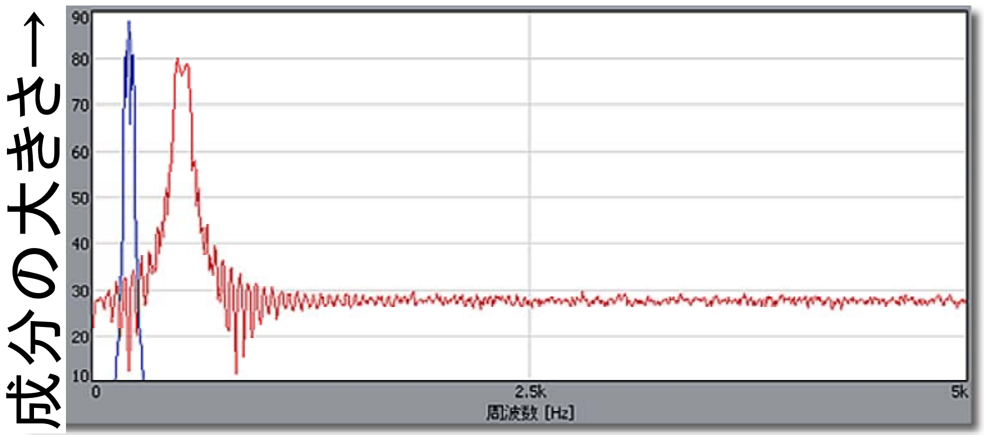
・FFTはDFT(離散フーリエ変換)の演算を工夫して高速化したもので、近年はオシロスコープなどでも標準的機能になった。

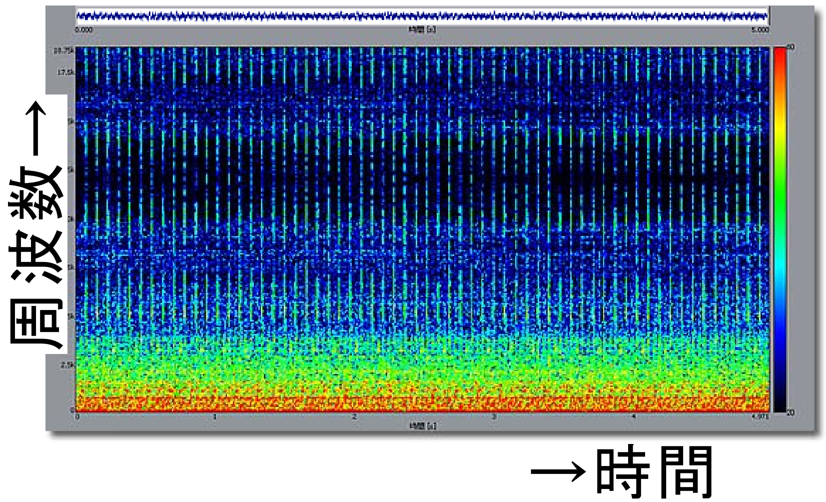
・演算プログラム例なども多数あり。

(注:一般に2^n個のデータに対して適用)

周波数分析

○ FFT(高速フーリエ変換)





FFTの表示例 (小野測器社WEBサイトより)

・周波数－成分の大きさ　:一般的表示

・時間－周波数－大きさ(色)

:時間変化する信号の解析用

周波数分析

○ FFT(高速フーリエ変換)

・周波数成分から「意味」を見つけるには、
別の処理(や人間の判断)が必要。

・なにか対象への、入力と出力の波形の
双方のFFT結果を演算することで、
対象の周波数特性が得られる。

・周波数分解能と時間分解能は両立せず。
例)1Hz単位の周波数分析には1秒間に
わたるデータの計測が必要

整数演算と浮動小数点演算

○ 計算の速度と読みやすさ

整数演算(固定小数点演算)

○ 速度が速い/マイコンでも実用にしやすい。

△ SI単位系で書くことは困難。

× 特有のテクニックが必要。

浮動小数点演算

○ SI単位系による可読性の高いプログラム。

△ 演算コストが大きい

(CPUの演算機能/ソフトウエアエミュレーション)

整数演算と浮動小数点演算

○ 選択の目安

・ソフト開発 and/or 実装すべき内容に
不慣れなら、浮動小数点(SI単位)。

・パソコン級を使えるなら、浮動小数点(SI)。
(演算力に余裕があるなら)

・マイコン組込するなら整数(独自単位)。
(必要ならPC級でコード仮組)

・蛇足：float使うならlongの方が高分解能。

整数演算と浮動小数演算

○ 整数演算での参考テクニック

・非線形な校正曲線はテーブルを使う。
（定数を埋めた配列）

・三角関数もテーブルを使う or
近似式を使う。　例:$(1-79x^2+16x^4-x^6)/64$

・√ は2進数で実装可能。

・桁あふれに注意。

・シフト演算を活用。
（右シフト時は"四捨五入"相当の処理が必要）

信号処理の順番

○ 順番によって妥当性や速度が変わる

非線形性が目立つ場合

D値 → 単位変換 → メディアン → 線形F → 高等処理

→被測定量単位　　　　周波数分析など

・最初に非線形さを校正曲線で取り除く。
・スパイクノイズが目立つならメディアン検討。
・高周波の不要信号はローパスフィルタ。
・周波数分析は前処理抜きで使えること多し。

信号処理の順番

○ 順番によって妥当性や速度が変わる

直線性が高い場合(比例、一次関数的)

D値→メディアン→線形F→高等処理→単位変換

→被測定量単位

・多くの信号処理はデジタル値の整数のままでの演算が可能。

・整数のほうが処理を高速化しやすいので、被測定量(小数)への変換を最後にする。

今回の目的

○ センサ信号の処理の基礎

テーマ1：センサの信号と情報

・センサの信号は処理が必要

・値の変換処理・微分積分

テーマ2：フィルタ＝時間変化する信号の処理

・ノイズ除去系のフィルタ

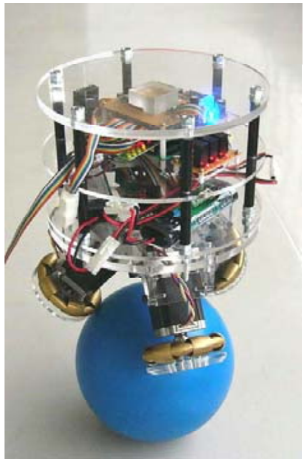
・周波数抽出・分析

テーマ3：信号処理の実例

・ロボット姿勢センサ等

信号処理の実例：ロボットの姿勢センサ

○ 複数のセンサ信号の混合：背景



・ロボット用の主要な姿勢センサ

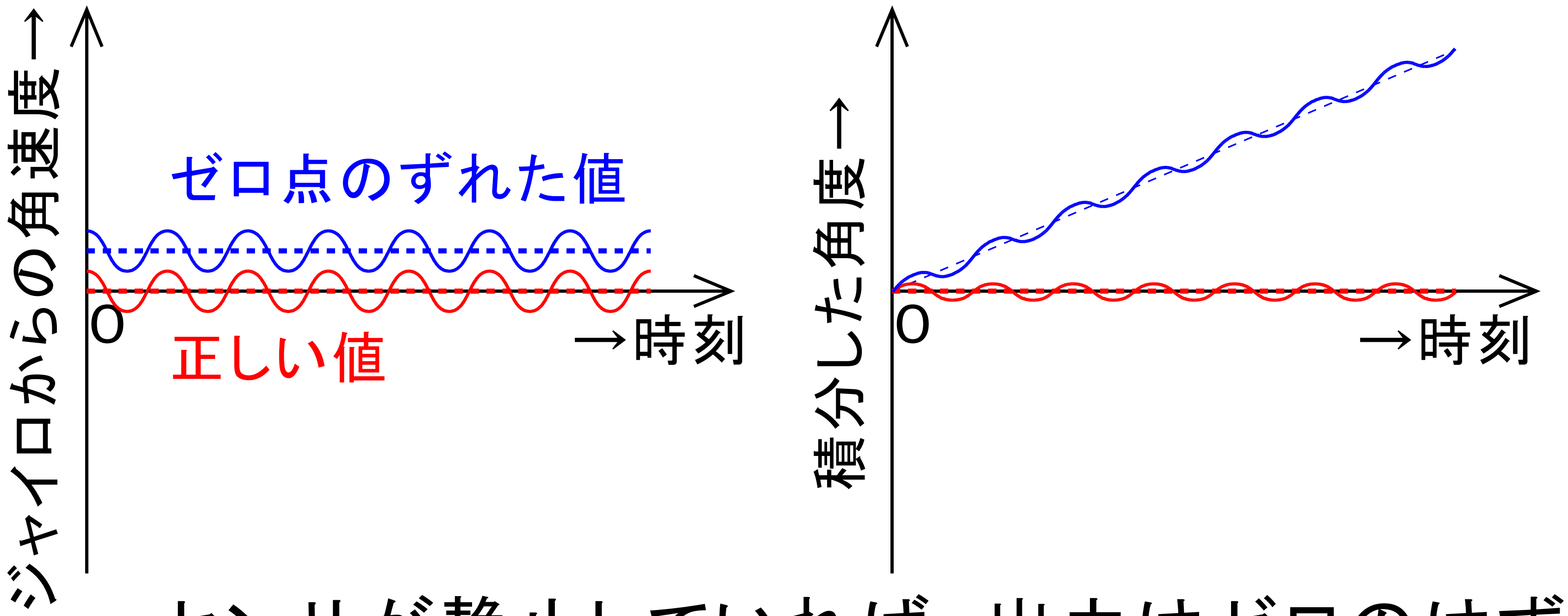
	応答性	安定性
角速度ジャイロ	○	×
加速度センサ	×	○

・"いいとこどり"をしたい

補足：加速度センサは重力加速度の方向を検出。
応答性＝周波数の高い成分、安定性＝直流成分
ジャイロの安定性のなさ ＝ ゼロ点ドリフト→積分
加速度センサの応答性のなさ ＝ ロボットの揺れ

信号処理の実例：ロボットの姿勢センサ

○　積分の「怖さ」



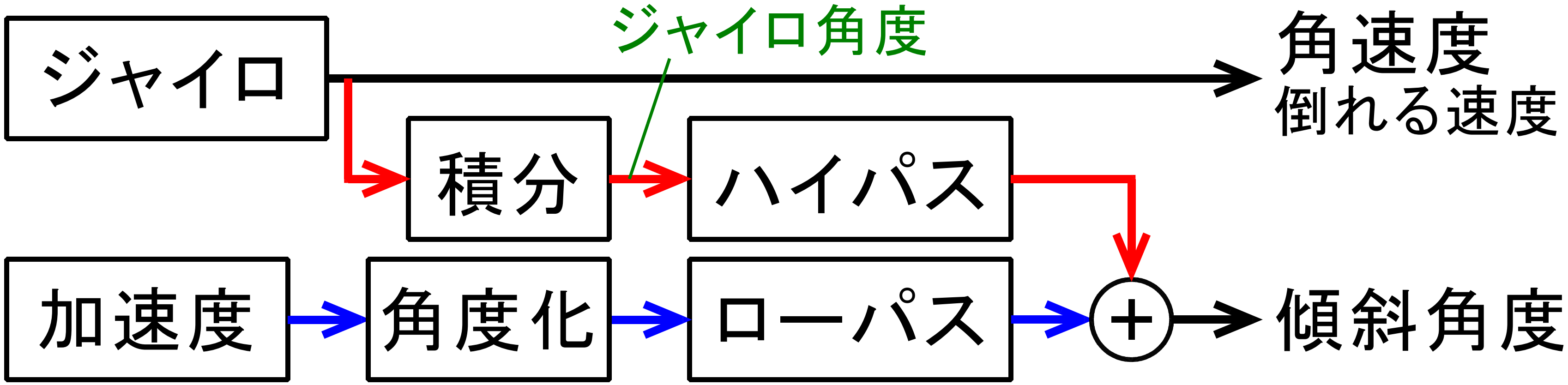
・センサが静止していれば、出力はゼロのはず

→　温度変化などでゼロ点がずれる

→　積分値はどんどんずれていく→役立たず

信号処理の実例：ロボットの姿勢センサ

○ 複数のセンサ信号の混合：アイデア



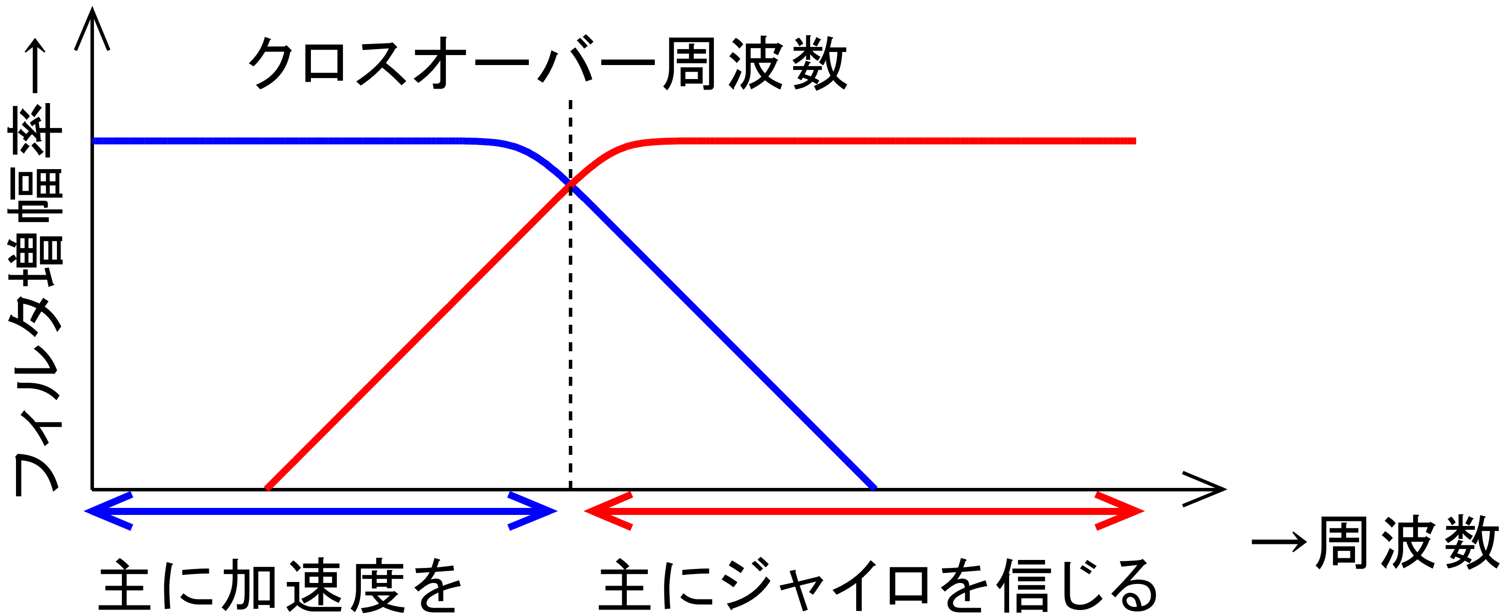
・ジャイロ信号は積分して角度にする。

・加速度信号を角度に変換する。

・ジャイロからハイパスで周波数の高い成分を
加速度からローパスで周波数の低い成分を

信号処理の実例：ロボットの姿勢センサ

○ 複数のセンサ信号の混合：フィルタ特性

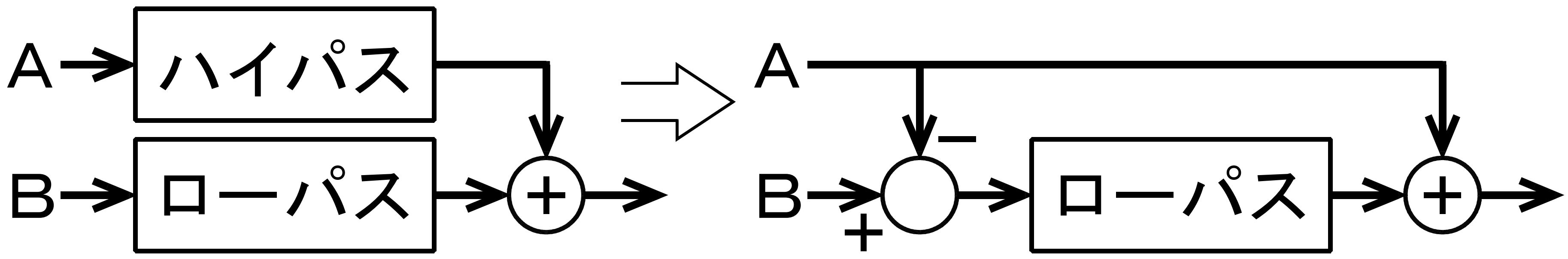


・両センサの特性から、境界となる周波数を基準とした二つのフィルタを検討する。

・重なりすぎ、隙間があくことはNG。

信号処理の実例：ロボットの姿勢センサ

○ 信号の混合用フィルタの一体化



・ハイパス特性 ＝ １ － ローパス特性

・ハイパス(A) ＋ ローパス(B)

＝{A－ローパス(A)} ＋ ローパス(B)

＝A ＋ ローパス(B－A)

※フィルタの「線形性」で演算順序交換

信号処理の実例：ロボットの姿勢センサ

○ ローパスフィルタの実装

◇式変形：一次ローパス

入力:U　出力:Y　比率:r

$Y_{次} = rU + (1-r)Y_{前}$

$Y_{次} = Y_{前} + r(U-Y_{前})$

◇C言語で整数演算実装

$r=(1/2^n)$ とすると、

（r×）は（>>n，nビット右シフト）になる。

Y += (U−Y)>>n;　　　※Y=Y+(U−Y)>>n

信号処理の実例：音楽の周波数解析

○ 学生さん：「楽譜を起こせる処理を」

◇処理の背景

・音楽で聞こえる音高は「周波数」。

→周波数分析できれば音高は分かる？

・音楽では「音符の長さ」「タイミング」が必要。

→時間分解能も必要

◇課題点：両立

・周波数分解能（隣接する音の周波数は1.06倍）

・時間分解能（0.1秒くらいは欲しい）

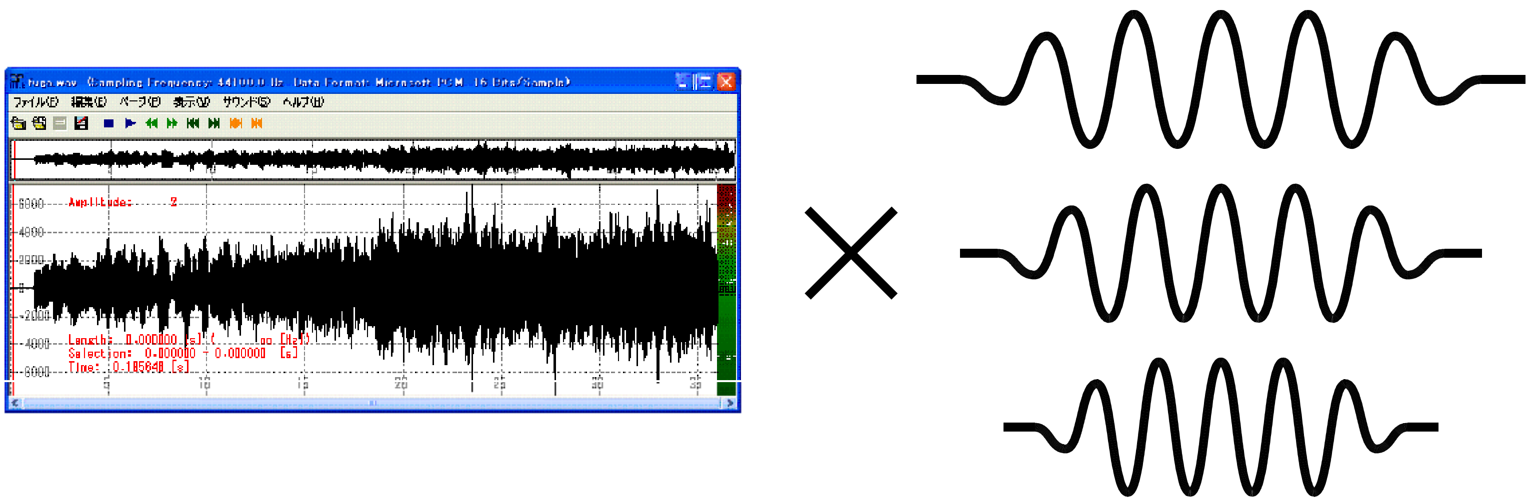
信号処理の実例：音楽の周波数解析

○ 使用した手法：ウェーブレット変換

◇周波数と時間の分解能を、ある程度両立

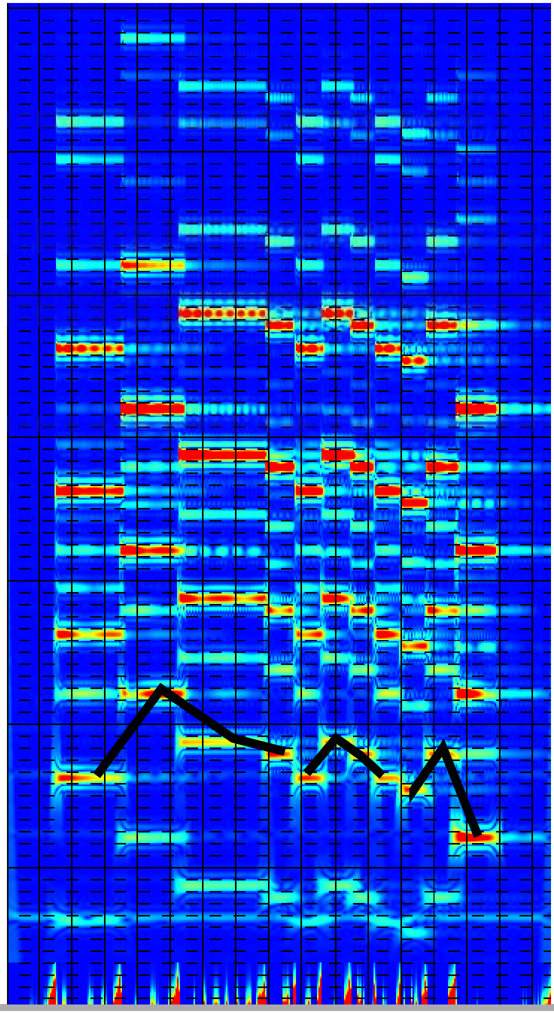
・周波数の低い成分は周波数優先

・周波数の高い成分は時間を優先



信号処理の実例：音楽の周波数解析

○ 処理結果



・それなりに高低の変化は得られた。

・かなりの高調波(倍音)が含まれている。

・ここから楽譜にするのは、別の処理が必要。

まとめ

○ センサ信号の処理の基礎

- センサから取り込み、AD変換した値は
  ただのデータであって、情報ではない。
  有意の情報にするには何らかの処理が
  必要。

- 最低でも、被測定量への変換が必要。

- 簡単な演算で微分積分ができる。
  これにより、センサの選定範囲を増やせる。

まとめ

○ 時間変化する信号の処理

・単発の計測では得られない情報が、
連続した計測データから得られる。

・時間的に前後に関連するデータの処理で、
ノイズを除去したり、不要な周波数の
成分を落とすことができる。

・周波数分析は演算量が多いが、実用的な
用途には強力な手法である。