

機械知能工学科
メカトロニクス基礎

第10回

MB-10/Rev 15-1.0

センサ信号の処理

工学部 機械知能工学科

熊谷正朗

kumagai@mail.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**

今回の到達目標

○ センサ信号 から 計測値 への処理

◇センサ出力値がそのまま使える情報ではない
ことを説明できる。

- ・センサ出力値からの情報の取り出し

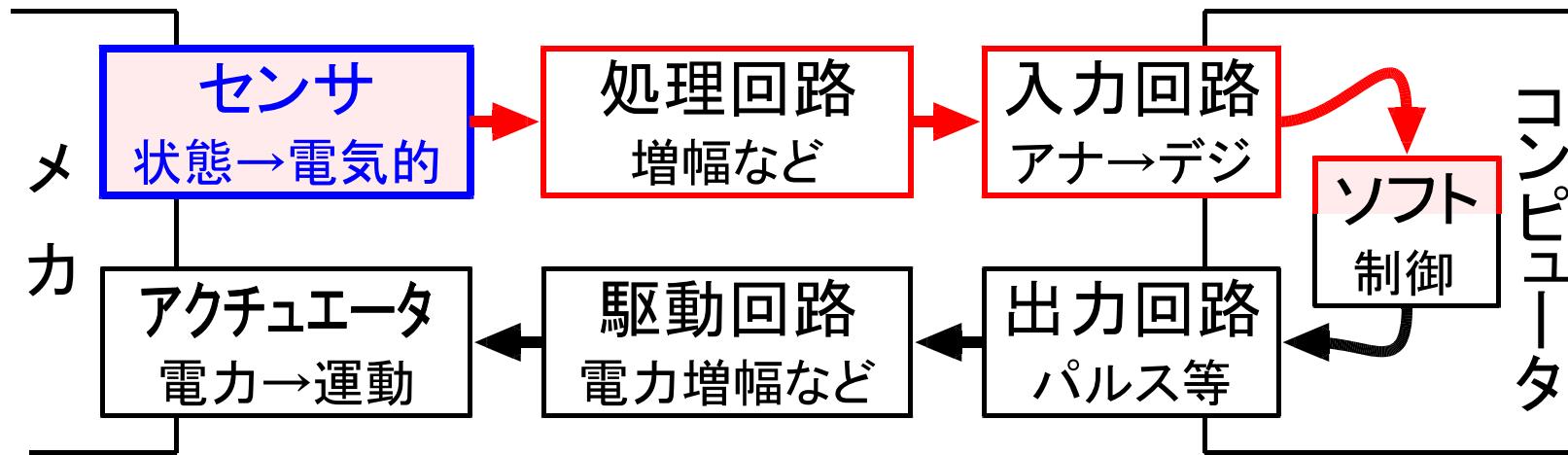
◇情報化するために必要な処理を説明できる。

- ・大きさの調整 / ノイズ除去 / 変換
- ・校正(キャリブレーション)

◇回路の得意なこととソフトの得意なこと
があることを理解できる。

メカトロニクスにおける計測の位置づけ

○メカの状態を数値情報として取得する



- ・計測や制御で必要なのは値ではなく情報
例) 電圧やデジタル値ではない、「温度」
- ・適切に伝達し、情報を復元するための処理

センサ～ソフトで必要な作業

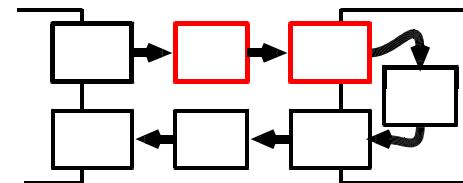
○アナログ回路+デジタル化+演算処理

◇センサ

- ・被測定対象量 → 電気的な変化

◇アナログ回路

- ・電気的な変化 → 電圧変化
- ・増幅（信号の大きさの調整）
- ・フィルタ（ノイズの除去など）



◇アナログ-デジタル変換(AD)（→後期）

- ・アナログ電圧に対応(比例)したデジタル値へ

センサ～ソフトで必要な作業

○電圧変化への変換＋増幅

◇電流変化 → オームの法則 → 電圧変化

- ・抵抗に電流を流す

◇抵抗変化 → オームの法則 → 電圧変化

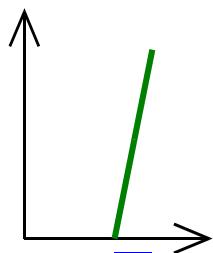
- ・抵抗(センサ)に電流を流しておく

◇増幅・電圧レンジ変換

- ・センサ出力をAD変換の入力に合うサイズに

例) $1.2 \pm 0.1[V] \rightarrow 0 \sim 5[V]$ にする

= 1.1V 引いて25倍する、など



センサ～ソフトで必要な作業

○ フィルタ（主にローパスフィルタ）

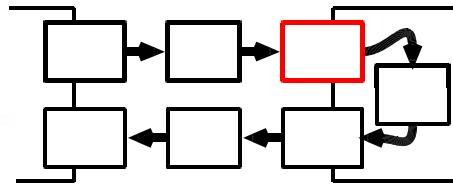
◇ ローパスフィルタ（→参考：学生実験）

- ・ 低い周波数は一定の率で増幅をして
 - ・ 高い周波数は減衰させる
- ことで「ローをパスさせる」回路

◇ ローパスを入れる目的

- ・ 高い周波数のノイズを低減する
- ・ AD変換の前に信号の周波数を制限（→後期）
- ・ 専用回路 or 増幅回路 + コンデンサ

センサ～ソフトで必要な作業



○アナログ→デジタル変換(AD変換→後期)

◇アナログ電圧→デジタル数値

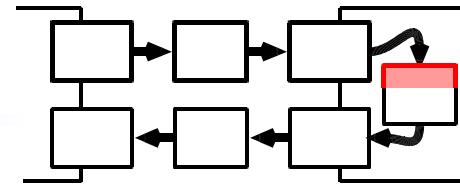
- ・アナログ電圧に比例した整数値(2進数)
 - ・アナログの連続性は、ここでデジタルの離散(とびとび)になる。

例) 0-5[V]を8bitに変換→19.5[mV]刻み
→19.5[mV]よりも細かな変化は無視

◆ 時間の離散化

- ・一定時間間隔で変換=間の変化は不明に

センサ～ソフトで必要な作業

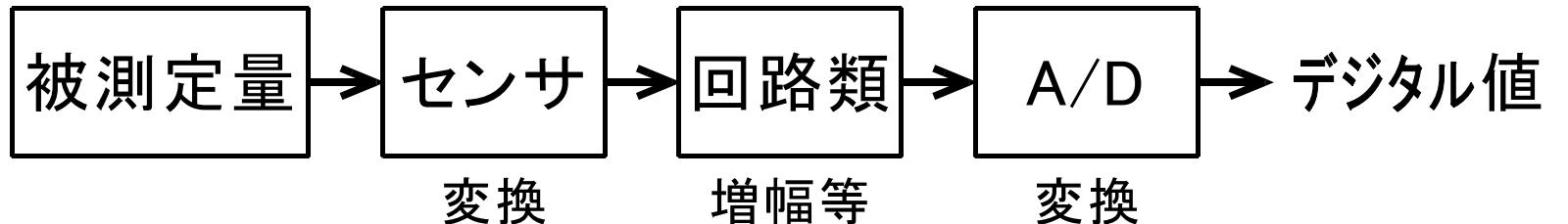


○ソフトウェアの処理：情報の復元

- ◇ただのデジタル値→計測したかった値
- ◇数値の変換作業
 - ・変換後のデジタル値から、単位付きの数値に直す。
↓例：三角関数、 $\sqrt{\cdot}$
 - ・単なる係数／校正曲線／数学的変換
- ◇信号処理（→大学院：計測信号処理特論）
 - ・数学的手法で信号を加工する 例) 微積
 - ・デジタルフィルタ（数式でフィルタ、画像処理も）

センサ～ソフトで必要な作業

○ソフトの処理：情報の復元：係数のみ



量[単位] $S[mV/\text{単位}]$ $K[V/mV]$ $R[digit/V]$ $D[digit]$

得られるデジタル値は

※ mV はmAやΩなどの
場合有り

$$D = R \times (K \times (S \times \text{量}))$$

$[digit/V][V/mV][mV/\text{単位}][\text{単位}]$

※digitはデジタル値

なので

校正でまとめて
対処できる↓

$$\text{量[単位]} = ((D \div R) \div K) \div S = D \times (1/RKS)$$

校正(キャリブレーション)

○センサ～デジタル化の誤差を吸収

◇誤差(ばらつき)の原因

- ・センサそのものの誤差(感度・オフセット)
- ・増幅回路等の誤差
- ・AD変換の誤差 (上二つに比べると小さい)

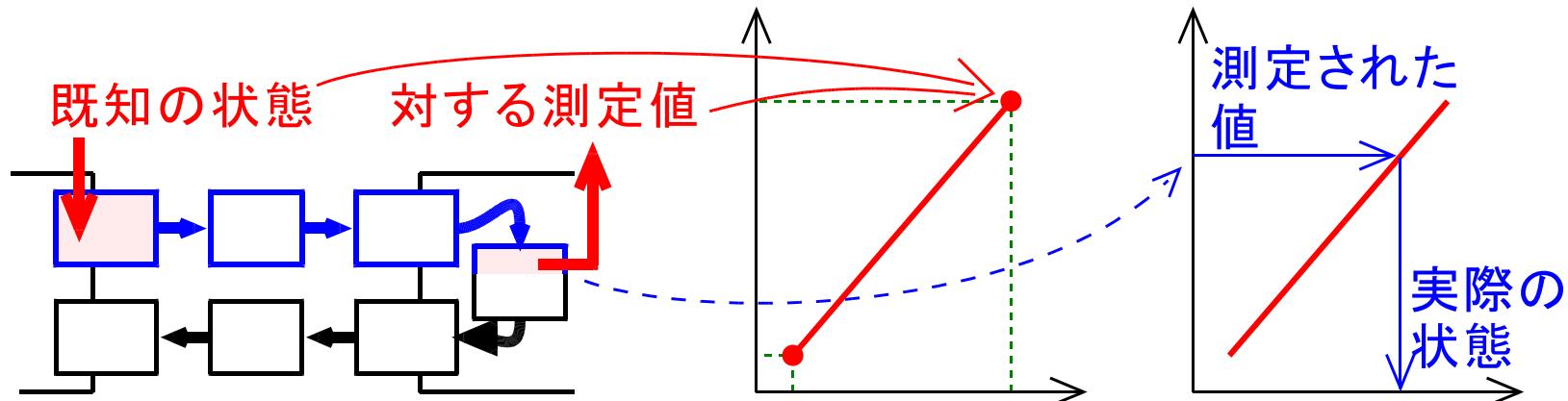
◇誤差の再現性

- ・つくった一式ごとに異なる誤差が出る が
- ・その誤差は固有で変化しない
→誤差をキャンセルすればいい

校正(キャリブレーション)

○校正の基本的な方針

- 1: 測定系で「正しい」被測定量を測定する
- 2: 被測定量→測定値(処理後)の関係を求める
- 3: 利用時に、この関係を逆に使って、
測定値→本来の被測定量 の変換をする

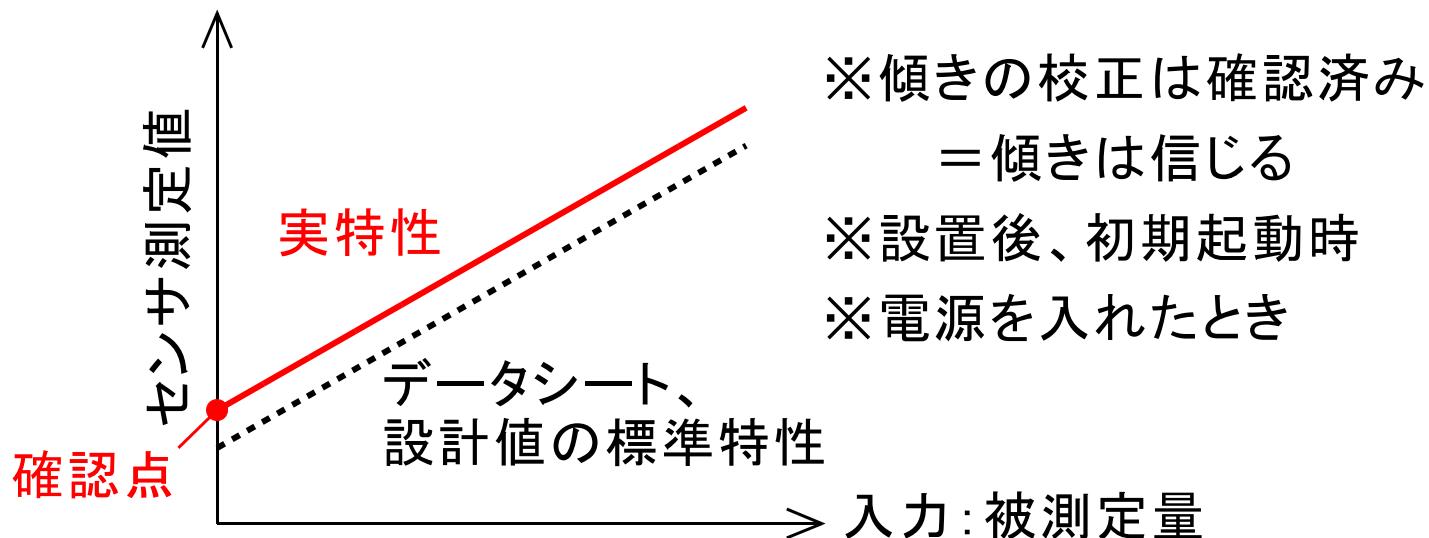


校正(キャリブレーション)

○例1: オフセットの校正

◇オフセット(ゼロ点)だけずれるような場合

- ・センサの設置による、温度による 等
- ・主な例: 電子体重計 ゲームのコントローラ

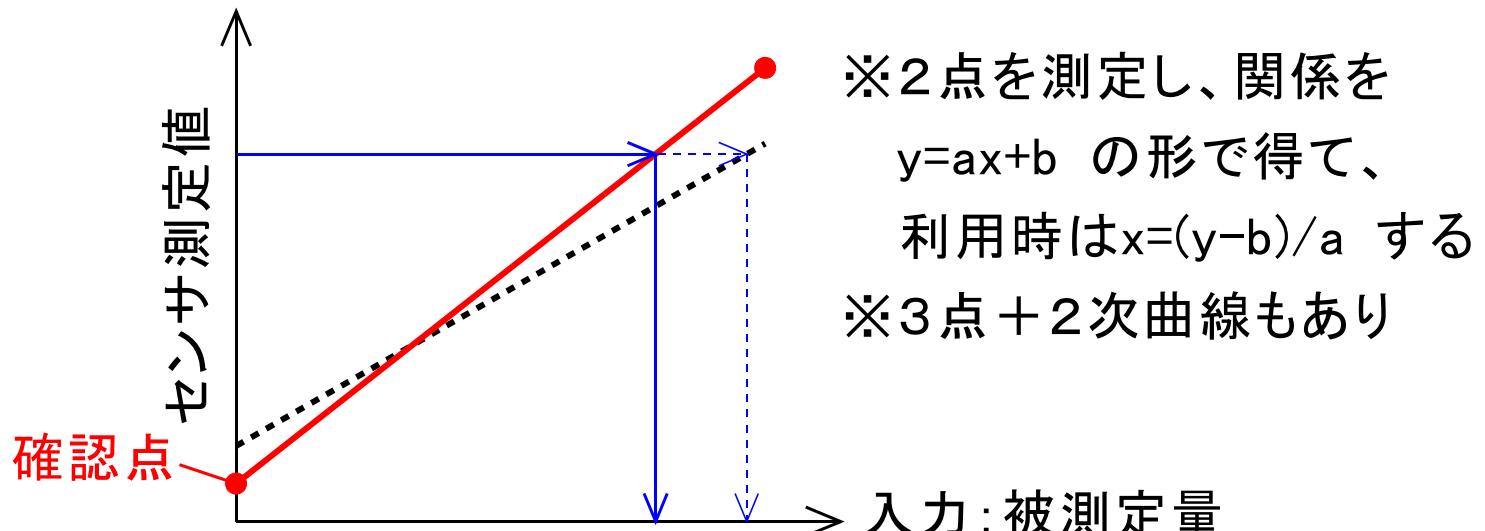


校正(キャリブレーション)

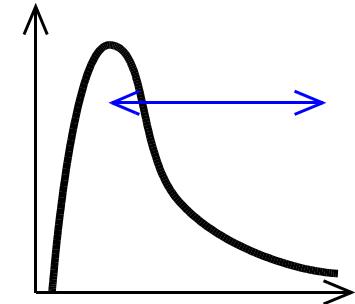
○例2: オフセットと感度の校正（直線的）

◇ 感度の確認が必要な場合

- ・個体間のばらつき、つくったら一度は必要
- ・主な例: 温度計(0,100°C) 傾斜角度センサ



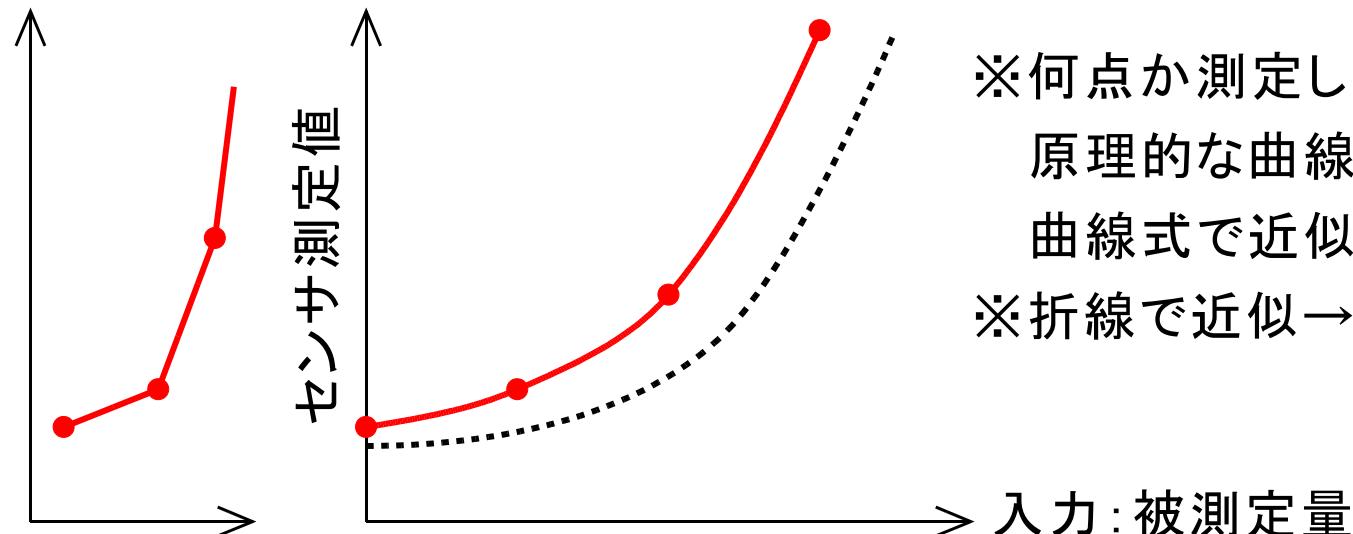
校正(キャリブレーション)



○例3：校正曲線

◇センサの特性が直線的では無い場合

- ・曲線関数で近似、連続直線で近似
- ・主な例：三角測量型の距離センサ



※何点か測定し、センサの
原理的な曲線式、その他
曲線式で近似
※折線で近似→前と同様

センサー→処理の開発手順

○変化を拾う→処理法を考える

- 1: 被測定量の**変化を拾える手段**を見つける
 - ・なるべく**一般的な方法**（「買えば済む」が最強）
 - ・なるべく**大きく変化**を拾える（増幅を減らせる）
- 2: 適切な電圧変化にもちこむ→デジタル化
- 3: 処理方法を検討する
 - ・最初は手動/表計算ソフトなど→プログラム
- 4: 校正と評価

回路とソフトの向き不向き

○どこまでを回路で処理すべきか

◇処理の分担

- ・大きさ調整 → 回路 & ソフト処理(校正)
- ・ゼロ点調整 → 回路 & ソフト処理(校正)
- ・ノイズ除去 → 回路 & ソフト処理
- ・微積、関数変換 → ソフト処理
- ・校 正 → 回路 & ソフト処理

◇なるべくソフト(デジタル)寄り、アナログ減らす

- ・アナログは劣化しやすい & コスト高い