電気 311/電情 311
ディジタル制御

教科書: 大野修一線形離散時間システム入門
森北出版 (2008)

- 教科書にはあまり拘泥せずに講義を進める
- 教科書にない話題も取り扱う
- シラバスは各自で読んでおくこと.

第 1 回
ディジタル
制御系の構成

制御とは

- 機械・装置などを目的とする状態に保つために, 適当な操作を加えること (大辞林第 2 版).
- 注目している対象物に属する注目している動作が, なんらかの目標とする動作になるように, その対象物の操作を加える行為 (大須賀, 足立, システム制御へのアプローチ, コロナ社, 1999)

JIS Z8116 自動制御用語–一般より

制御 **(control)** ある目的に適合するように, 制御対象に所要の操作を加えること

制御対象 **(controlled object)** 制御の対象となる系で, 機械, プロセス, プラントなどの全体又は一部がこれに当たる

制御システム

- 制御システムとは, 特定の入力が与えられたとき, 望ましい性能の出力が得られるように, サブシステムやプラントを組み合わせたものである (N. S. Nice, Control Systems Engineering, 7/e, Wiley, 2015)

システム (系) とは…

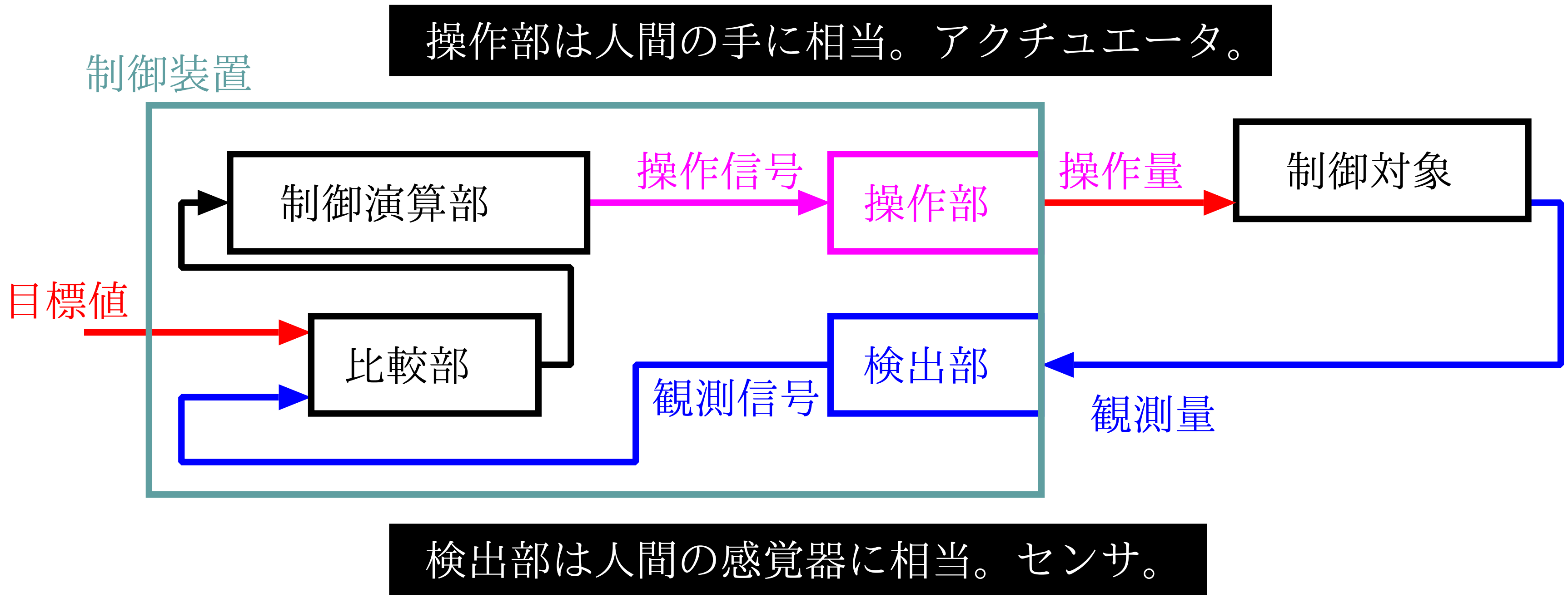
- 個々の要素が有機的に組み合わされた, まとまりをもつ体系 (大辞林第 2 版)
- システムは, その外部とエネルギー・物質・情報をやり取りすることもあれば, しないこともあるが, この講義では前者を議論の対象とする.
- システムと系は同義.

- システムに外部から入ってくるエネルギー・物質・情報を入力, システムから外部に出てゆくエネルギー・物質・情報を出力と呼ぶ. 入力信号, 出力信号という言葉も使われる.
- システムを入力を出力に変換する (抽象的な) 作用素と解釈することもできる.

制御系/制御システム (JIS Z8116 の定義)

- 制御系 **(control systme):** 制御のために制御対象に制御装置を結合して構成された系
- 制御装置 **(controller, control device):** 検出部, 比較部, 制御演算部, 操作部からなり, 操作量を生成する装置

これらを以下の図に示す.

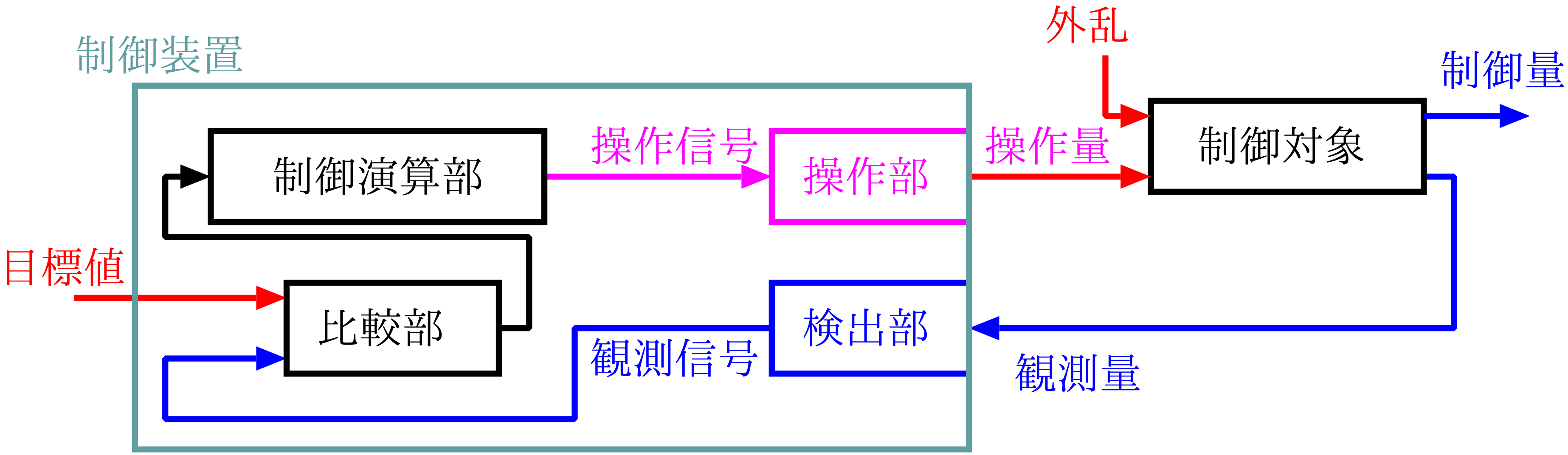


([大須賀, 足立] 図 4.1 を JIS 規格に合うように変更)

- それを制御することが目的となっている量のことを制御量という (JIS Z8116).
- 制御量と関係があり観測できる量を観測量という ([大須賀, 足立]).
- 制御対象に働きかける人為的に操作できない量を外乱という ([大須賀, 足立]).

- 「外乱」「雑音」「ノイズ」はほぼ同義だが使い分けることもある．また, 図の書き方にもバリエーションがある．

- 外乱などを先の図に書き加えると次のようになる．



([大須賀, 足立] 図 4.1 を JIS 規格に合うように変更)

- 外乱をこれ以外の箇所に挿入するモデルもある.

制御システムの具体例

- エアコンによって部屋を適切な温度まで冷房することを考える.
- 室温が設定温度を上回ったらその度合いに応じて冷房装置を動かし, 設定温度を下回ったら冷房装置を止めることで, 室温と設定温度とのずれは一定に収まると期待される.

▷ システムの構成は…

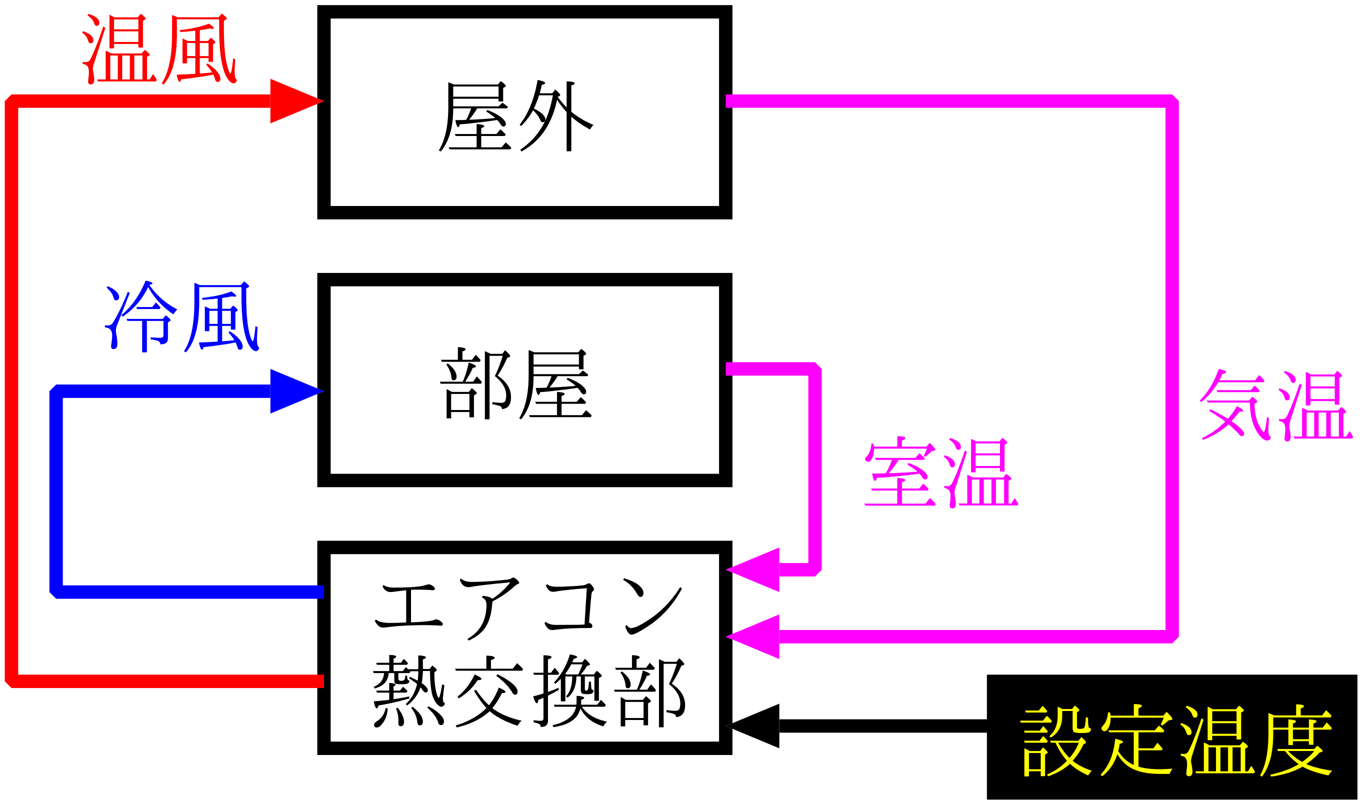
- 入力: 設定温度
- 出力: 室温
- 制御対象: 部屋の空気
- 制御装置: エアコン

▷ 冷房のしくみは…

1. 室外機のコンプレッサーで冷媒を圧縮し, 熱交換器で冷却してから減圧して室内器に送る.
2. 室内器の熱交換器に冷媒を送り, そこで気化させる． 気化熱で冷やされた熱交換器からファンによって冷風を送る.
3. 気化した冷媒を室外器に送る (1 に戻る).

▷ 室温の制御

- 室温が設定温度に近付くようフィードバックをかける．


温風
屋外
冷風
部屋
気温
室温
エアコン
熱交換部
設定温度

- 室温とエアコンの冷風がフィードバックループを構成している
- 気温とエアコンの温風にもフィードバックループがある
- エアコンの運転は室温と設定温度の比較値に応じて調整される

制御システムの分類

- 制御システムを適切に設計するためには, 設計者が制御対象と制御装置を「良く知っている」必要があるが, この知識は, 多くの場合, 数学モデルという形で集約される. 数学モデルとは, システムの入力, 出力および内部の変数 (あれば) の関係を記述する式などである.

- 数学モデルを構築することを**モデル化**, 数学モデルがパラメータを含むときにそのパラメータを制御系などに適合するように定めることを**システム同定**と呼ぶ.
- 数学モデルは制御系そのものではなく, その近似表現であり, モデル化およびシステム同定はつねに誤差を伴う.
- 以下, 数学モデルの分類について述べる.

- システムの出力がその時点での入力のみで決まれば静的システム, 過去の入力の履歴に依存すれば動的システム
- 数学モデルが常微分方程式なら集中定数システム, 偏微分方程式なら分布定数システム

- 制御システムの出力が入力に戻る経路があればフィードバック制御システム, そうでなければフィードフォワード制御システム
- 1入力1出力システム (single-input single-output(SISO) System), 多入力多出力システム (multi-input multi-output (MIMO))

- SIMO, MISO という構成もあり得るが, この言葉はあまり用いられない.

- システムの挙動が初期時刻に依存しないシステムは**時不変システム (Time-Invariant System;TI システム)**, 依存するシステムは**時変システム (Time-Varying System;TV システム)**
- 非線形要素を含まなければ**線形システム (Linear System(LS))**, 含めば**非線形システム (Nonlinear System(NLS))**

- 時間が連続値のシステムは連続時間システム, 離散値のシステムは離散時間システムあるいはディジタルシステム

- 離散時間システムは**ディジタルシステム**とも呼ばれる．離散時間システムでは, 信号の値が連続的なことも離散的なこともあるが, 離散的な値を取る離散時間のみをディジタルシステムと呼ぶ流儀もある (この講義では採用しない).

- 連続値要素と離散値要素が混在したシステムはハイブリッドシステム (hybrid system)
- 確率的な要素を含まないシステムは確定システム (deterministic system), そうでないシステムは確率システム (stochastic system)

- この講義では集中定数の動的システムのみを取り扱う. 分布定数システムは取り扱わない. 離散時間線形時不変システムを中心に議論するが, それ以外についても言及する.
- 信号の値が離散的なシステムは連続的なシステムより数学的な取り扱いが繁雑なため, この講義では積極的には取り扱わない.

ディジタル制御

- ディジタル制御系とは, 制御にディジタルハードウェア (多くはコンピュータ) を用いた制御系のことをいう (Levine, The Control Handbook, 2/e, 2011). これに対応する制御をディジタル制御と呼ぶ. ディジタル制御器, ディジタル補償器という言葉も用いられる.

- 多くの場合, 制御対象はアナログ (連続時間, 連続値)
- ディジタルコンピュータが処理できるのはディジタル信号 (有限ビット長の数列)
- これらを組み合わせるためには, 観測量 (アナログ) の数列への変換と, ディジタル制御器による操作量 (数列) のアナログ信号への変換が必要になる.

アナログ信号から数列を得る手順

1. 飛び飛びの時刻 (一定の間隔のことが多い) でアナログ信号の標本を取得する (**標本化**)
2. 有限個の**量子化代表値**と呼ばれる値を設定しておき, 信号の値を量子化代表値で近似する (**量子化**)

- 上記を **A/D 変換**と呼ぶ.

数列からアナログ信号を得る処理

1. 数列の数値を物理的な次元を持つ量に変換 **(D/A 変換)**
2. 物理量に変換された数値のあいだを連続信号で補間 (ホールド)

- D/A 変換とホールドについては後で改めて述べる.

標本化

- 最も単純には, 標本化は一定の時間間隔でおこなわれ, この周期を標本化周期あるいはサンプリング周期という.
- 標本化周期の逆数を標本化周波数あるいはサンプリング周波数という. 標本化周波数に 2π を乗じたものを標本化角周波数あるいはサンプリング角周波数という.

量子化

- 信号が取りうる値の集合 (実数の部分集合とする) を N 個の区間 $[x_i, x_{i+1}]$ に分割し ($0 \leq i \leq N-1$, $x_i < x_{i+1}$), 各区間から, その区間を代表する値 y_i を選ぶ. y_i を量子化代表値と呼ぶ. 信号が区間 $[x_i, x_{i+1}]$ に値を取る時には, 信号の値を y_i で近似することを量子化と呼ぶ. 近似に伴う誤差を量子化雑音と呼ぶ.

標本化に関する注意

- 標本化は一定周期でおこなわれることも, そうでないこともあるが, この講義では前者のみを取り扱う.

- 信号 $x(t)$ を標本化周期 T_s で標本化すると, 数列 $(x(0), x(T_s), x(2T_s), \ldots)$ が得られるが, T_s を略して $(x(0), x(1), x(2), \ldots)$(講義資料), $(x[0], x[1], x[2], \ldots)$, $(x_0, x_1, x_2, \ldots)$(教科書) などと書く (色々な流儀がある).

D/A 変換

- 回路などを用いて, 数値を物理的な次元を持つ量 (電圧など) に変換する操作を **D/A 変換**と呼ぶ.

ホールド

- 物理量に変換された数列の数値のあいだ (一定値の時間間隔とするのが一般的) を連続信号で補間する装置をホールドと呼ぶ.
- よく用いられるのは, 零次ホールドと呼ばれる, 次の数値 (物理量) が得られるまで前の数値を保持する装置. これによって得られた信号は時間軸に対して階段状に変化する.

ディジタル制御の利点

1. 柔軟: コンピュータのプログラムで書けることは何でもでき, システムの拡張が容易
2. 高品質: 量子化雑音, 演算雑音のみを考慮すればよくディジタル部に想定外の雑音が混入するリスクが少ない
3. 長期記憶が可能で, 経年変化や環境の変動の影響を受けにくい

4. LSI 化および大量生産によって小型化, 低コスト化が可能
5. 仕様変更に柔軟に対応可能で開発者の熟練を要さず製品のばらつきが少ない

- 長所が多いため, 今日ではディジタル制御が一般的で, 電気的な要素を含む工業製品には何らかのディジタル制御が用いられていることが多い

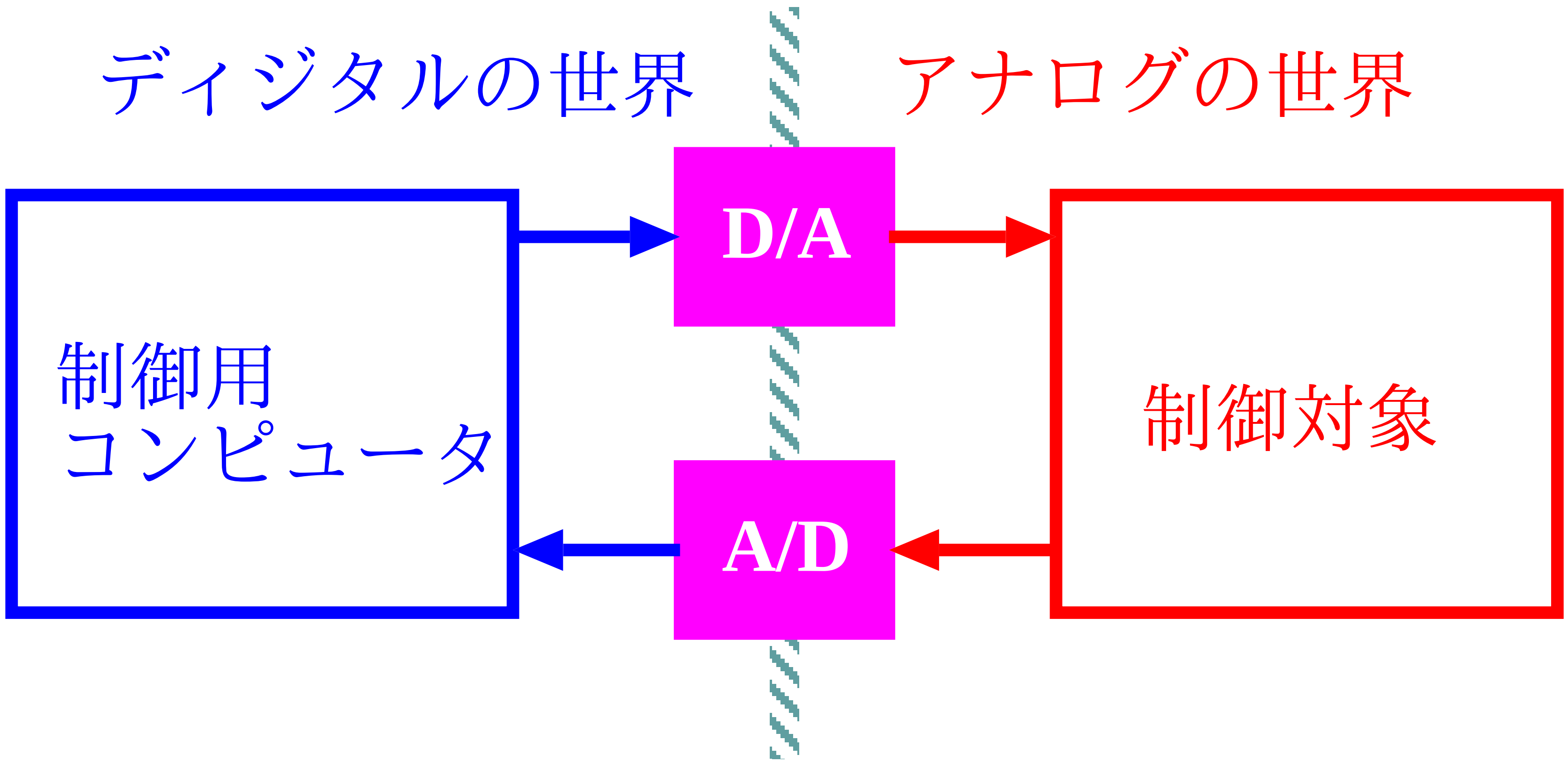
ディジタル制御の欠点

- A/D 変換, D/A 変換用ハードウェアが必要
- 量子化雑音, 演算雑音の影響がある
- 理論的にアナログ制御より複雑
- 処理内容が極めて簡単なときには回路規模的にアナログ処理と比べて不利
- 処理速度に限界がある (CPU, I/O 等の能力)

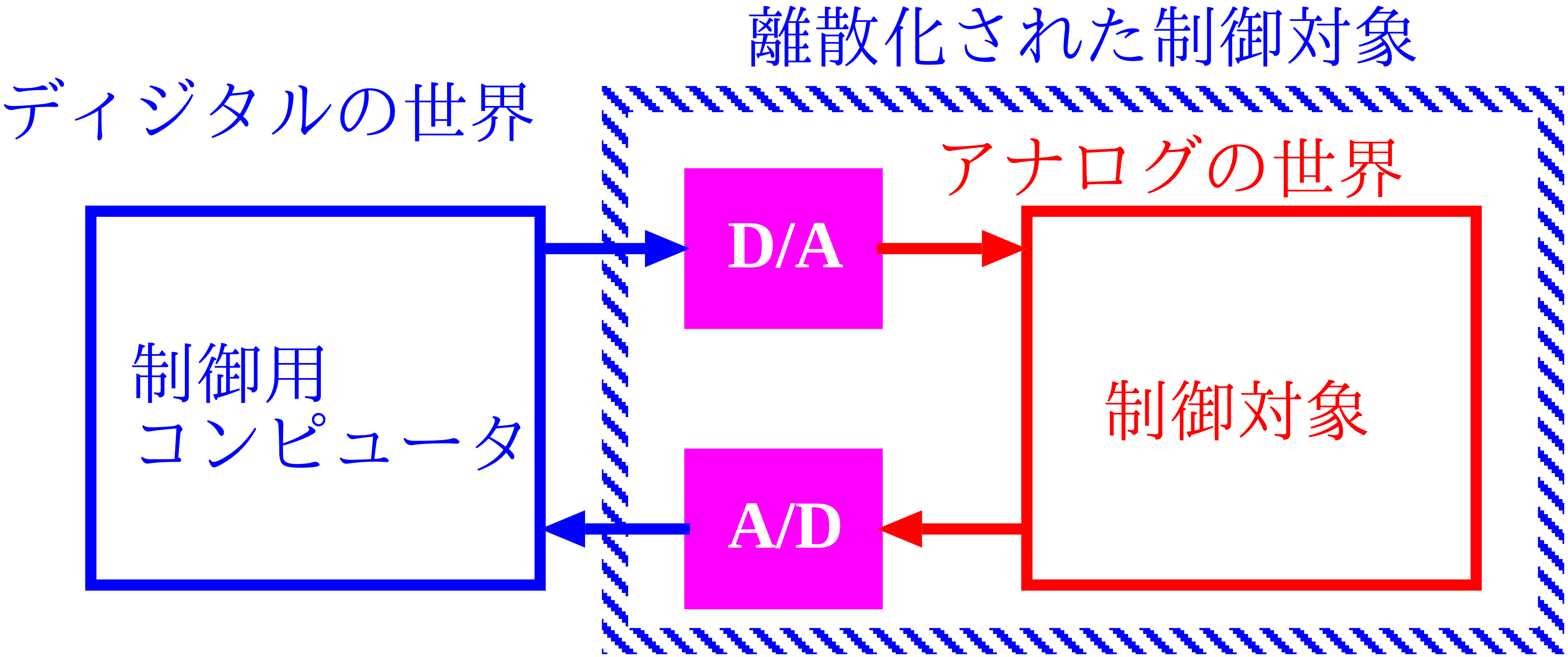
ディジタル制御系の見方

- ディジタル制御系の見方には, 以下の 2 種類がある.
  - ▷ 制御対象, A/D, D/A を統合してディジタルシステムとして取り扱う.
  - ▷ 制御用コンピュータ, A/D, D/A を統合してアナログシステムとして取り扱う.

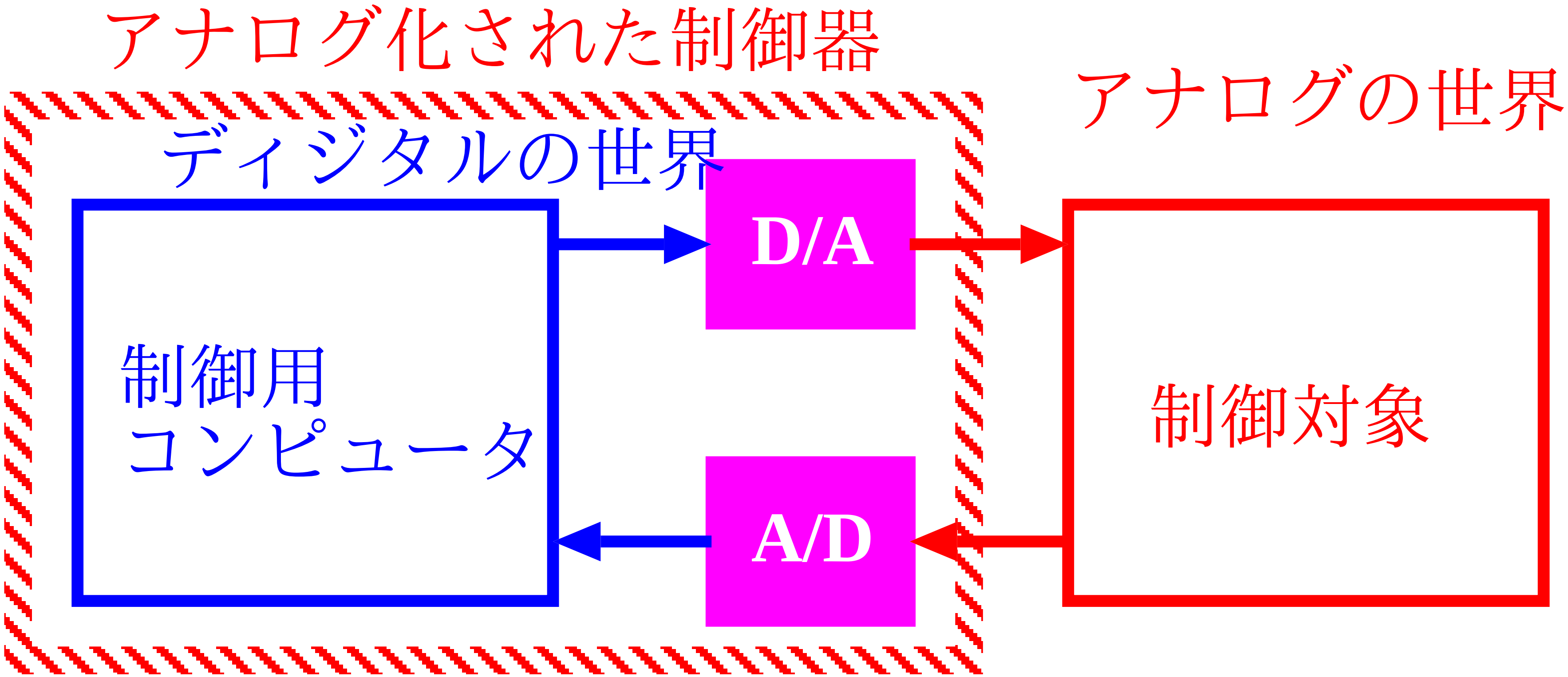
- これらの概念を図で示す.
- 次ページのような制御システムが与えられているものとする.


ディジタルの世界
アナログの世界
制御用
コンピュータ
D/A
A/D
制御対象

- 制御対象, A/D, D/A を統合してディジタルシステムとして取り扱うと次ページのような解釈になる.


離散化された制御対象
ディジタルの世界
アナログの世界
D/A
制御用
コンピュータ
制御対象
A/D

- 制御用コンピュータ, A/D, D/A を統合してアナログシステムとして取り扱うと, 次ページのような解釈になる.


アナログ化された制御器
アナログの世界
ディジタルの世界
D/A
制御用
コンピュータ
A/D
制御対象

- より一般的な連続リフティングという考え方もあるが, この講義では取り扱わない.

線形時不変システム

- 入力が $u = (u(n))_{n\in\mathbb{Z}}$ で出力が $y = (y(n))_{n\in\mathbb{Z}}$ のシステム $\mathcal{F}$ を考える ($\mathbb{Z}$ は整数全体の集合).

$$\overset{\text{入力}}{u} \longrightarrow \boxed{\mathcal{F}} \longrightarrow \overset{\text{出力}}{y}$$

- システム $\mathcal{F}$ が

$$\mathcal{F}[a_1u_1 + a_2u_2] = a_1\mathcal{F}[u_1] + a_2\mathcal{F}[u_2]$$

という等式を満たすとき，これを線形システムという．

- 初期時刻によってシステムの挙動が変わることがないシステムを時不変システムと呼ぶ.

- 線形かつ時不変のシステムを, 線形時不変システムと呼ぶ.

- 線形時不変システムは, その解析および設計がしやすいという点において, 理論および応用の双方の観点から重要である.

- 単位インパルス (δ と書く) を, $\delta(0) = 1$, $\delta(n) = 0 (n \neq 0)$ によって定義する.
- $\mathcal{F}[\delta]$ をシステムの**インパルス応答**と呼ぶ. インパルス応答は線形時不変システムの性質を調べる際に重要な役割を果たす.
- $h = \mathcal{F}[\delta]$ とする.

- 信号を時間軸に関して m シフトさせる作用素を τ_m とおく. 信号 u に対し, $\tau_m u$ は,

$$(\tau_m u)(n) = u(n-m)$$

を満たす信号として定義される.

- システムが時不変であれば, 入力を時間軸に関して m シフトすると, 出力も同様に m シフトするから, $\mathcal{F}[\tau_m u] = \tau_m(\mathcal{F}[u])$ となる.
- $\tau_m \delta$ を δ_m と書く.

- $u = \sum_{p\in\mathbb{Z}} u(p)\delta_p$ だから, $\mathcal{F}$ が線形時不変ならば,

$$\left(\mathcal{F}[u]\right)(n) = \left(\mathcal{F}[\sum_{p\in\mathbb{Z}} u(p)\delta_p]\right)(n)$$
$$= \sum_{p\in\mathbb{Z}} u(p)\left(\left(\mathcal{F}[\delta_p]\right)(n)\right)$$

である.

- 一方,

$$\begin{aligned}(\mathcal{F}[\delta_p])(n) &= (\mathcal{F}[\tau_p \delta])(n) \\ &= (\tau_p \mathcal{F}[\delta])(n) \\ &= (\tau_p h)(n) = h(n-p)\end{aligned}$$

である.

- よって, 次の無限和が得られる (暗黙のうちに和が絶対収束することを仮定する).

$$(\mathcal{F}[u])\,(n) = \sum_{p\in\mathbb{Z}} u(p)h(n-p)$$

- 信号 u と h の畳み込み，重畳 ($h * u$ と書く) を，$(h * u)(n) = \sum_{p \in \mathbb{Z}} u(p)h(n-p)$ によって定義する．
- 先の式は，$\mathcal{F}[u] = h * u$ となることを示している．すなわち，線形時不変システム $\mathcal{F}$ の入力 u に対する出力 (応答) は，入力と $\mathcal{F}$ のインパルス応答の畳み込みになる．

時間軸に関する注意

- 信号処理の一般論では時間軸を $-\infty$ から $+\infty$ までに取るが，制御では，時刻零で因果的なシステムが動作を開始するものと考え，時間軸を 0 から $+\infty$ までに取ることが多い．

- 前者では信号 u は $(u(n))_{n\in\mathbb{Z}}$ であり，後者では信号 u は $(u(n))_{n\in\mathbb{N}}$ である.

- $\mathbb{N}$ は自然数全体の集合, $\mathbb{Z}$ は整数全体の集合をあらわす.
- この講義では零を自然数に含めるが, 自然数を 1 から始める流儀もある. 後者の場合, 零以上の整数全体を $\mathbb{Z}_{\geq 0}$, $\mathbb{Z}_{+}$ などといった記号で表すことが多い.

フィードバック制御の歴史

古典制御以前

- 紀元前の文献に既に浮きを使った流量調整システムの記述がある.
- フィードバックによる温度調整を用いた孵化器が 1620 年に作られた.
- ワットの蒸気機関 (1789 年) には遠心調整速機が使われた.

- 蒸気機関の振動現象の解析が制御理論が始まる契機．重要な発見は, J. C. Maxwell による安定性条件 (1868), Routh の安定性条件 (1877), Heaviside の演算子法 (Laplace 変換; 1893) など.

- 1890 年に 非線形システムの 安定性が Lyapunov によって 研究された (時代に先行し過ぎており数十年間あまり注目されなかった).

古典制御の時代 (20 世紀前半)

- 1 入力 1 出力の線形時不変システムを対象とし, 主に周波数領域において制御系の解析と設計をおこなう手法を古典制御と呼ぶ.
- この手法による制御理論は 20 世紀前半におもに研究された. 理論的には完成しており, 応用でよく用いられる理論である.

- 古典制御の時代には，制御に関連して，以下のような装置や手法が発明されている．

- ▷ ジャイロスコープと自動航行システム (1910)
- ▷ フィードバック増幅器 (1920-1940)
- ▷ Nyquist の安定条件 (1932)
- ▷ PID 補償器 (1936)
- ▷ Bode の周波数応答法 (1938),
- ▷ Nichols 図 (1947)
- ▷ Evans の根軌跡法 (1948)

現代制御の時代 (20 世紀中盤)

- 現代制御とは, 多入力多出力システムの線形システムを対象とし, 時間領域および周波数領域で制御系の解析と設計をおこなう手法をいう. 状態空間法 (Kalman ら, 1960 年頃) と伝達関数行列 (1950 年代後半) はこの時代に開発された.

- **最適制御 (1950 頃–) :** 制約条件を満たす範囲内で制御目的に対応する評価関数を最小 (あるいは最大) とする制御器を数学的に求める手法. それ以前の最適化法と異なり, 不連続関数を取り扱う必要があったため, 新たにいろいろな技法が開発された. Pontryagin による最大値原理と Bellman の動的計画法が有名.

- 最適制御自体は線形系も非線形系も区別せずに扱える手法であるが, 線形時不変システムに対する最適制御は, 行列に関する Riccati 方程式という方程式と解けば補償器が求まるという意味で比較的取り扱いやすく, 大流行した.

- **サイバネティクスの流行 (1950 年代):**
  - ▷ サイバネティクスとは動物および機械の通信および制御に関する科学 (N. Wiener) であり, この時代に流行した.
  - ▷ この考え方は通信・制御・情報工学に大きな影響を与え, 今日の AI ブームにも繋がっている.

現代制御以降 (20 世紀後半–)

- 1970 年の前後から, 我々が制御対象に関して持っている知識は不完全なので, 不完全な知識が制御系に与える影響を小さくするべきである, という考え方が出て来た. これがロバスト制御である. ロバスト制御には色々な手法があるが, H_∞ 制御が有名. H_∞ 制御は 1990 年代に完成した.

- H_∞ 制御理論の完成をもって, 線形時不変システムの制御理論は一応完成.
- 制御理論の主要研究テーマは, 非線形システム, ネットワークシステム, ハイブリッドシステム, 確率システム, 無限次元システムなど. 線形時不変システムは理論研究の対象にはなりにくい.

制御とコンピュータ

- 制御にコンピュータを用いるというアイデアの提案は 1948 年.
- z 変換の提案 (Ragazzini and Zadeh, 1952)
- ディジタル制御の最初の例は Hughes Aircraft Company(1954).

- 産業用ディジタル制御システムの最初の例は Texaco Company(1959).
- ディジタル制御の普及は 1970 年代以降.
- 今日では制御には何らかのコンピュータを用いることが普通.

参考文献

[1] S. Bennett, A history of control engineering 1800-1930, The Institute of Engineering and Technology, 1979

[2] S. Bennett, A history of control engineering 1930-1955, Pater Peregrinus, 1993

[3] N. S. Nise, Control Systems Engineering, 7/e, Wiley, 2015

[4] G. F. Franklin, J D. Powell and A. Emami-Naeini, Feedback Control of Dynamic Systems, 4/e, 2002

[5] T. Kailath, Linear Systems, Prentice-Hall, 1980.

[6] M. S. Safonv, Origins of robust control: Early history and future speculations, Annual Reviews in Control, Vol. 36, No. 2, pp. 173-181, 2012.

[7] S. Benett, Control and digital computers: early history, IFAC 15th Triennial World Congress, Barcelona, Spain, 2002.

[8] 大須賀，足立，システム制御へのアプローチ，コロナ社, 1999.

[9] 木村, 制御工学の考え方, 講談社, 2002.

[10] 古田，ディジタルコントロール，コロナ社, 1989.

[11] http://www.asc-cybernetics.org/foundations/history.htm (Viewed: November 22, 2019).