ディジタル制御 第 **14** 回

ハイブリッドシステム

今回の講義について

- 今回の講義では, ディジタル制御に関連がある分野で比較的最近流行した (している) ハイブリッド制御と, それと関連が深いネットワーク化制御制御の話題を紹介する. また, 流行という観点から, モデル予測制御にも触れる.

- 研究には流行がある．流行の研究テーマには，研究費が獲得しやすい，資料が多い，研究グループを作りやすい，研究成果が評価されやすいなどの理由から，研究者が殺到する傾向がある．

- 一方で, 流行の研究テーマでは, 関連論文等をきちんとリサーチした上で研究の新規性や有用性を説得力のある形で述べることの手間も激増するし, リサーチの結果研究成果に新規性がないことが判明するリスクも高まる. あまりに出版数が多い分野では, 著者・査読者・編集者のすべてにおいて, 関連研究のリサーチが不十分になることがある.

- 流行していない研究テーマには, すでに廃れたもの, 重要でないもの, 重要だが地味なもの, ニッチなもの, これから流行する可能性があるものなど, 色々なものがある.

ハイブリッドシステムとは

- ハイブリッドシステムとは, その内部に連続値システムと離散値システムが混在した動的システムのこと. それが内在する連続値システムを連続ダイナミクス, 離散値システムを離散ダイナミクスと呼ぶ.

- たとえば，2 足歩行ロボットの動作は，遊脚 (地面から離れた足) が前に移動しているあいだは連続ダイナミクスで記述されるが，遊脚と支持脚 (地面に接触して体重を支える足) が交代する瞬間は離散ダイナミクスで記述される．オペレータや監視システムが連続時間システムの動作を運転状況に応じて切り換える場合，そのシステムは全体としてはハイブリッドシステムである．

- ハイブリッドシステムという考え方自体の提案は比較的古く, 1966 年 (Witsenhausen) だが, 研究が本格的に始まったのは 1990 年代から.

- 制御対象の状態に応じて補償器を切り換えるゲインスケジューリングはある種のハイブリッドシステムと見倣せるが，この種のシステムは第2次世界大戦で既に使われていたという指摘もある．

- シーケンス制御もある種のハイブリッドシステムと見倣せるが, これについてはいつごろ導入されたものか不明. あるシステムの振舞いが離散的な時刻で発生する事象によって特徴付けられるシステムを離散事象システムと呼び, シーケンス制御は離散事象システムの一種であるが, 離散事象システムの代表的なモデルであるペトリネットが提案されたのが1962年.

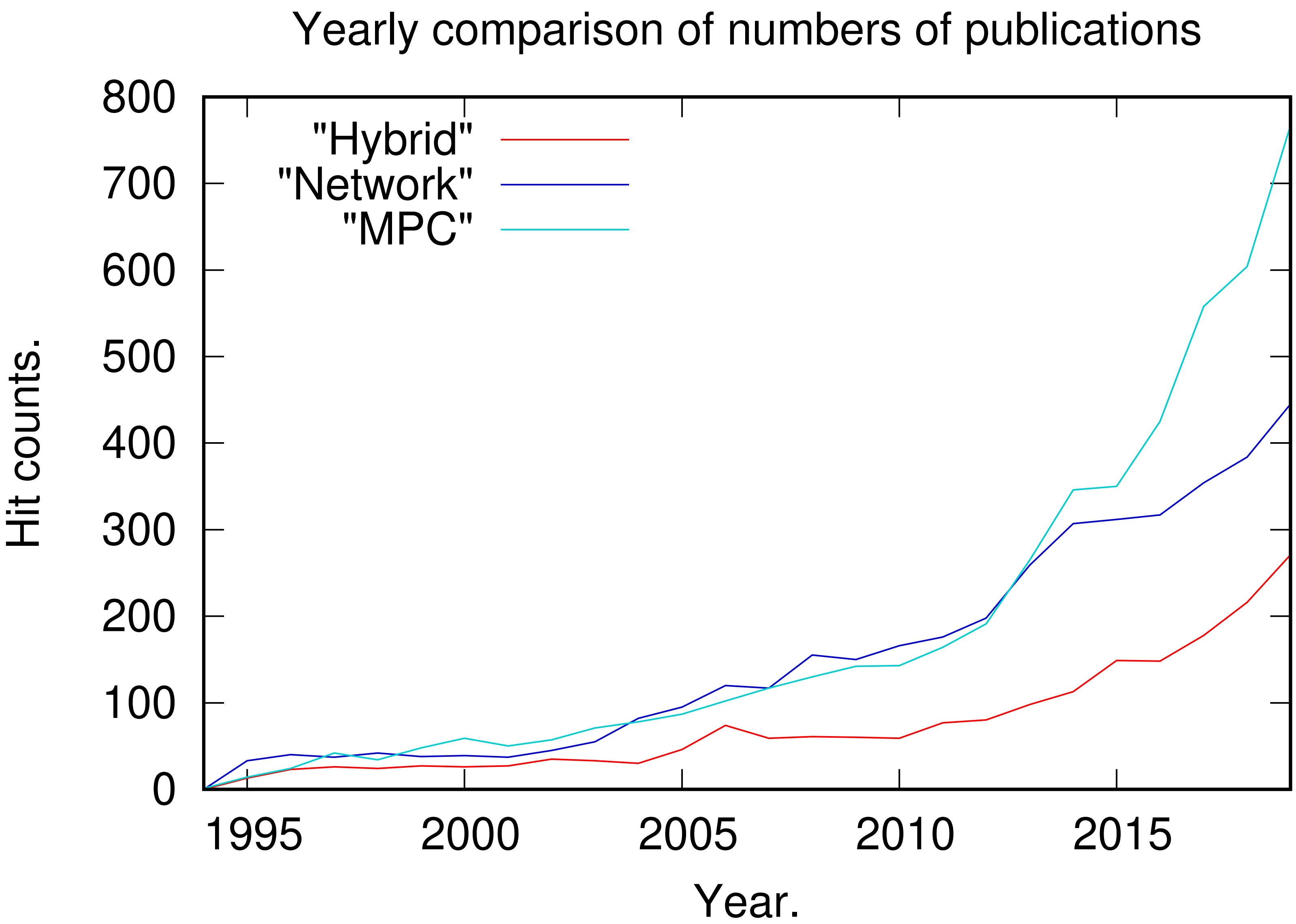
- サンプラとホールドを介して連続時間制御対象を制御するディジタル制御システムは, それ自体がハイブリッドシステムの一種であるという解釈も可能.

- 状態空間に切り換え面と呼ばれる超平面を設定し, 状態が切り換え面を通過すると入力を切り換えるスライディングモード制御と呼ばれる制御方式をハイブリッド制御の一種と解釈することがある.

- ファジィ論理に基づくファジイ制御や制御対象をネットワークを介して制御するネットワーク化制御もハイブリッド制御の一種と解釈することが可能.

- 近年の研究の動向を見るために, Web of Science において hybrid control systems (Hybrid と略す), Networked control systems (Network と略す), Model predictive control (MPC と略す) というキーワード群をタイトルに含む文献を検索した結果を次ページに示す (期間は 1994-2019 年).

- ハイブリッドシステムの研究も伸びてはいるが, 2000 年前後にはネットワーク化制御の方が分野として大きくなっている. モデル予測制御に関する研究は急増している.


Yearly comparison of numbers of publications
"Hybrid"
"Network"
"MPC"
Hit counts.
Year.
0
100
200
300
400
500
600
700
800
1995
2000
2005
2010
2015

- 数式で書くと, ハイブリッドシステムとは, 次式のようなシステムのことをいう.

$$
\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{I}(t), \boldsymbol{u}(t)) \\
\boldsymbol{I}(t) &= \boldsymbol{g}_1(\boldsymbol{x}(t_-), \boldsymbol{I}(t_-), \boldsymbol{u}(t_-)) \\
\boldsymbol{x}(t) &= \boldsymbol{g}_2(\boldsymbol{x}(t_-), \boldsymbol{I}(t_-), \boldsymbol{u}(t_-)) \\
\boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{I}(t), \boldsymbol{u}(t))
\end{aligned}
$$

- 前ページの式において, $\boldsymbol{x}(t)$ と $\boldsymbol{u}(t)$ は実ベクトルで, $\boldsymbol{I}(t)$ は離散的な集合に値を取るベクトルとする．次元は明示しない．関数 $\boldsymbol{f}(\cdot)$, $\boldsymbol{g}_1(\cdot)$, $\boldsymbol{g}_2(\cdot)$, $\boldsymbol{h}(\cdot)$ はこれらの次元が適合するように定義されているものとする. $\boldsymbol{x}(t_-)$ は,

$$\lim_{\tau \to t, \tau < t} \boldsymbol{x}(\tau)$$

という意味である. $\boldsymbol{x}$ 以外についても同様である.

- $\boldsymbol{x}(t)$ を連続状態, $\boldsymbol{I}(t)$ を離散状態と呼び，これらを纏めたものをハイブリッド状態と呼ぶ.

- 第１式

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{I}(t), \boldsymbol{u}(t))$$

は，離散状態が $\boldsymbol{I}(t)$ のときの状態方程式である．

- 第 2 式

$$\boldsymbol{I}(t) = \boldsymbol{g}_1(\boldsymbol{x}(t_-), \boldsymbol{I}(t_-), \boldsymbol{u}(t_-))$$

を離散遷移あるいはスイッチと呼ぶ．これは，離散状態の遷移を記述する式である．

- 第 3 式

$$\boldsymbol{x}(t) = \boldsymbol{g}_2(\boldsymbol{x}(t_-), \boldsymbol{I}(t_-), \boldsymbol{u}(t_-))$$

をジャンプと呼ぶ．これは，連続値状態の時間に関して不連続な遷移を記述する式である．

- 離散遷移とジャンプをまとめて事象と呼ぶ.

- 第 4 式

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{I}(t), \boldsymbol{u}(t))$$

は出力方程式である.

- あるハイブリッドシステムをハイブリッドシステムの一般形で書き表した場合, 必ずしもすべての式が必要となるわけではない. 第 1 式から第 3 式のいずれかが存在しないこともある.

- ハイブリッドシステムの一般形の第 3 式は, $\boldsymbol{x}(t)$ などが $[n-1, n)$ で一定値であると仮定すれば,

$$\boldsymbol{x}(n) = \boldsymbol{g}_2(\boldsymbol{x}(n-1), \boldsymbol{I}(n-1), \boldsymbol{u}(n-1))$$

となる. これは離散時間の状態方程式とも解釈できる. 一方, 時間軸を始めから離散的にしておくこともある.

- ハイブリッドシステムの一般形では, 時間の通常の関数としての解が存在することは必ずしも保証されない. 以下のような状況が発生し得る:

デッドロック　解が存在しない, あるいは解が複数存在する.

複数事象 複数の事象が発生する可能性があり時間軸を進めることができない.

Zeno 解 事象が生起する時刻の列が集積点を持ち, その先に時間が進まない.

スライディング動作 ある時刻で事象が無限回発生する.

- スライディング動作は, スライディングモード制御と呼ばれる応用上重要な制御方式で発生する動作. この動作が発生した場合, 微分方程式は通常の意味の解を持たないが, **Fillipov** 解と呼ばれる拡張された意味の解を持ち, スライディングモード制御系の解析ではこの解が用いられる (後述).

- 一般には, ハイブリッドシステムの解としては, 零集合を除いて (ルベーグ測度の意味で) 上述の第 1 式を満たす $\boldsymbol{x}(t)$ などが採用される. 上述の Zeno 解を排除することもある.

- 上述の一般形は漠然とし過ぎていて, これだけでは大したことができないので, ハイブリッドシステムの研究では, もう少し具体的な形のシステムを取り扱うことが一般的.

- その代表的なものは以下の通り: ハイブリッドオートマトン, 区分的アフィンシステム, 線形相補性システム, 混合論理動的システム, スイッチドシステム. **Max-plus** 代数システム, ハイブリッドペトリネット.
- ハイブリッドシステムに確率構造を追加したものが取り扱われることもある.

- ハイブリッドオートマトンはオートマトンに時間に係る構造を追加した時間オートマトンがベースになっており, ハイブリッドペトリネットはペトリネットに時間構造を追加した時間ペトリネットがベースになっている.

- 区分的アフィンシステムは, システムの状態方程式が線形時不変の微分方程式に定数項を追加したもので記述されていて, その状態方程式が状態と入力の値に応じて何種類かに切り換わるというもの. 似たような文脈で Linear parameter-varying system (LPV システム) という言葉が使われることがある.

- ゲインスケジューリングやファジイ制御は, 複数の状態方程式を状態や入力の値に応じて切り換えるという点で区分的アフィンシステムと類似しているが, 非線形系も取り扱うことができるという点で, 区分的アフィンシステムに基づく制御理論よりも適用範囲が広い.

- スイッチドシステムは微分方程式

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \cdots)$$

において関数 $\boldsymbol{f}$ が不連続なもの．応用上よく用いられるスライディングモード制御はこの典型例．

- 線形相補性システム, 混合論理動的システム, Max-plus 代数システムについては, この講義では名前を紹介するのみとする.

- 先に挙げた具体的なハイブリッドシステムに関し, これらを相互に変換することの可否が研究されている.

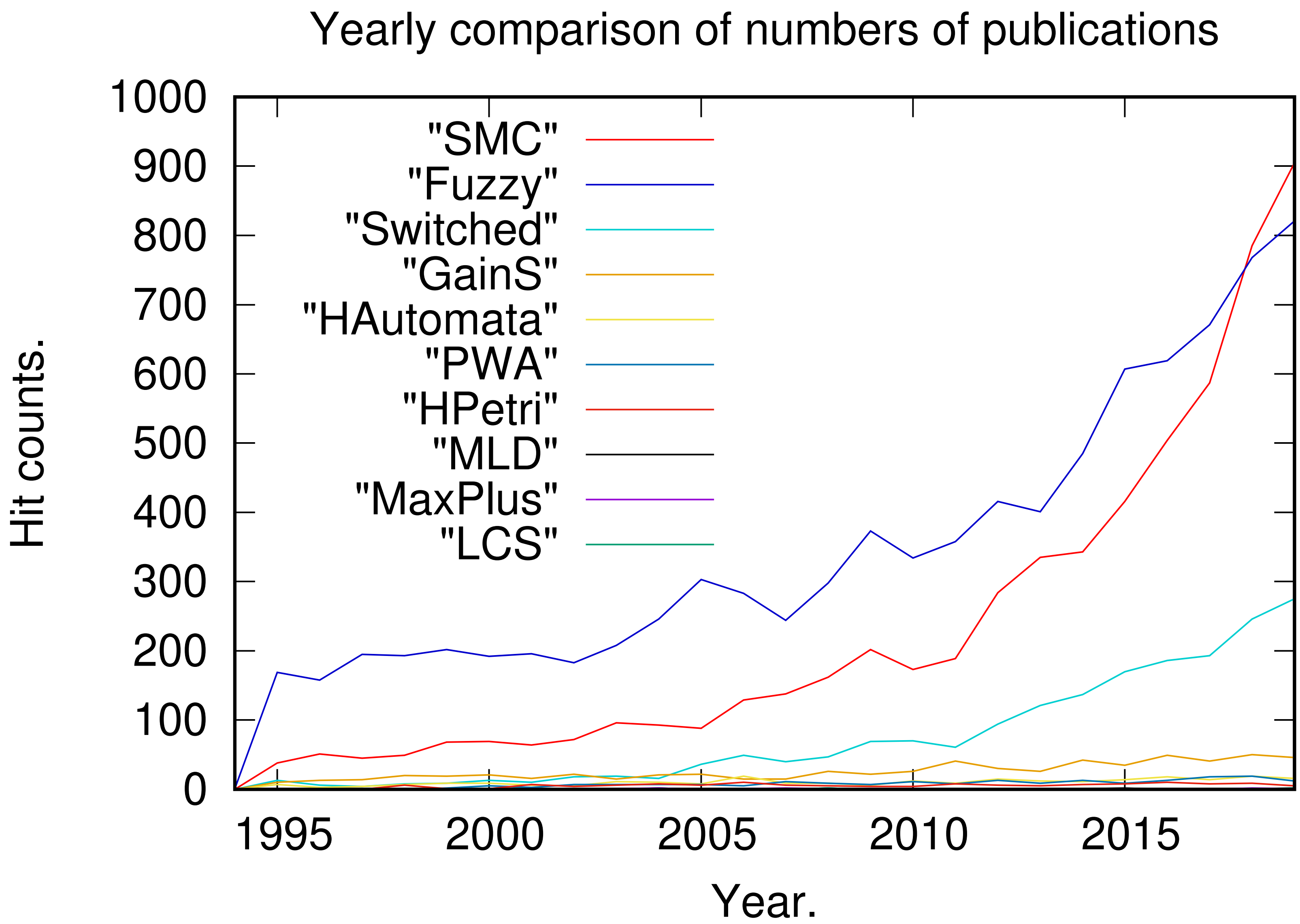
- これとは別に，Lyapunov の方法に基づいてハイブリッドシステムの安定性が調べられている．

- とはいうものの, 上述のハイブリッドシステムの一般形に基づいて調べることができる性質は, 解の定義, 相互変換, 安定性ぐらいしかない.

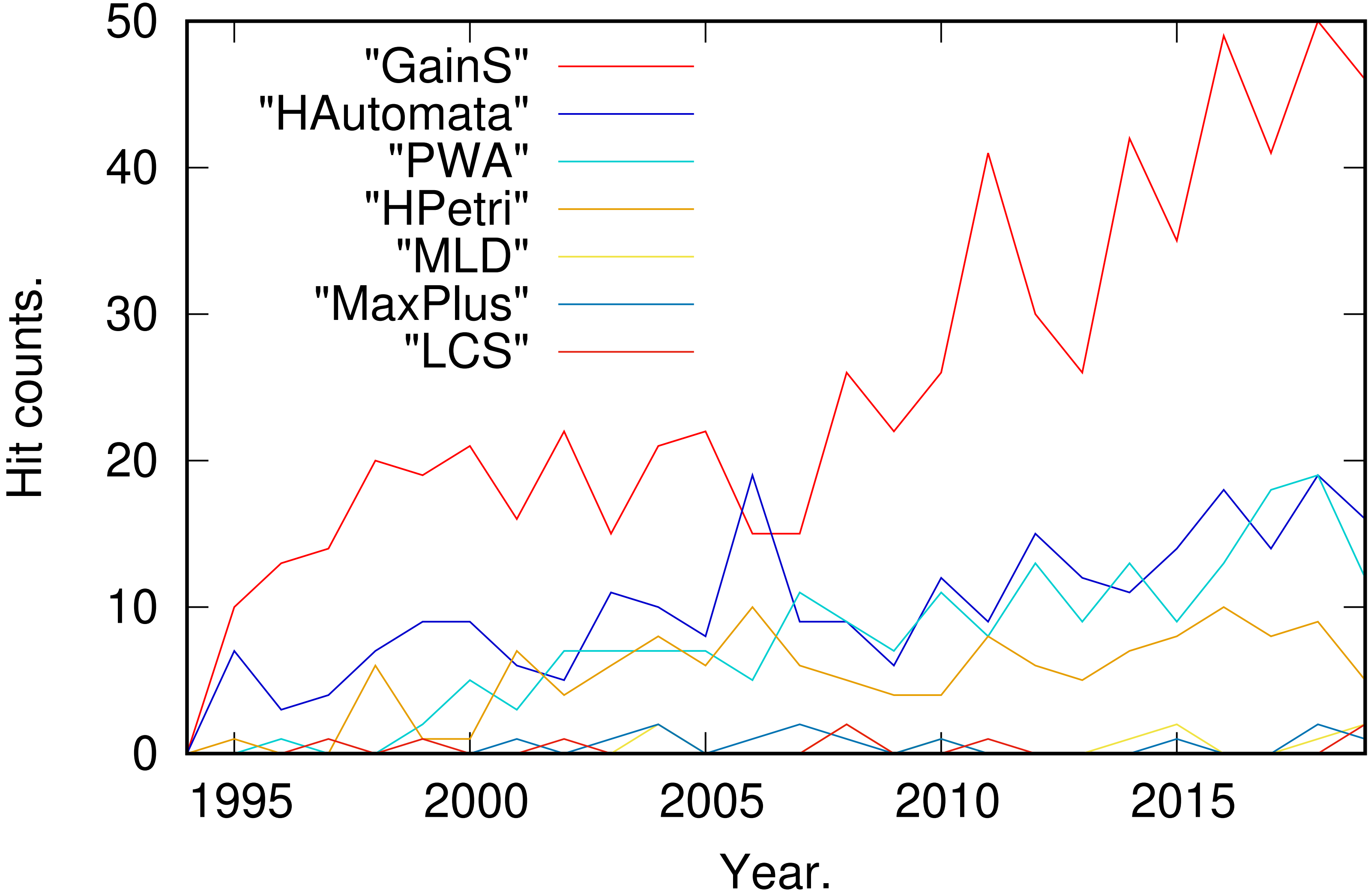
- 先に挙げた個々の具体的なハイブリッドシステムについては, もっと踏み込んだ研究がある.

- Web of Science で 1994 年から 2019 年までの各年における上述の分野に係る文献数を調べたものをグラフにして示す.

項目	検索語	グラフ中のキー
ハイブリッドオートマトン	“Hybrid automata”	HAutomata
区分的アフィンシステム	“Piecewise affine systems”	PWA
線形相補性システム	“Linear complementary systems”	LCS
混合論理動的システム	“Mixed logical dynamical systems”	MLD
スイッチドシステム	“Switched systems”	Switched
ハイブリッドペトリネット	“Hybrid Petri Nets”	HPetri
Max-plus 代数システム	“max-plus” “discrete event systems”	MaxPlus
スライディングモード制御	“Sliding mode control”	SMC
ゲインスケジューリング	“Gain scheduling”	GainS
ファジイ制御	“Fuzzy control”	Fuzzy


Yearly comparison of numbers of publications
Hit counts.
Year.
1000
900
800
700
600
500
400
300
200
100
0
1995
2000
2005
2010
2015
"SMC"
"Fuzzy"
"Switched"
"GainS"
"HAutomata"
"PWA"
"HPetri"
"MLD"
"MaxPlus"
"LCS"

By omitting SMC, Fuzzy and Switched...



- 論文数が多ければその分野の研究が急激に進展しているとは必ずしも言えないのだが (試行錯誤の要素が多い分野では, 大量に論文が出版されても, 研究自体はあまり進んでいないことがある), それはそれとして, スライディングモード制御とファジィ制御の論文数が他を圧倒している.

- スイッチドシステムに関する論文数はこれらの 1/3 程度である.

- この 3 分野を除くと, ゲインスケジューリングでは相対的に多くの論文が出版されているが, それ以外ではほとんど論文が出ていない.

- 以下では, スイッチドシステム, スライディングモード制御, ゲインスケジューリング, ファジィ制御について基礎的な部分を紹介する.

スイッチドシステム

- スイッチドシステム (スイッチングシステムとも呼ばれる) とは, 状態空間表現された (非線形) 連続時間システムで, その状態方程式が

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_{\sigma(\cdot)}(\boldsymbol{x})$$

によって与えられるもののことをいう.

- 前ページの式で, σ は時間 t ないし状態の関数. 非線形関数として $\{\boldsymbol{f}_\lambda(\cdot) : \lambda \in \Lambda\}$ が用意されていて (Λ は添字の集合で, 自然数の部分集合など, 何でもよい), $\sigma(\cdot) = \lambda$ のときには $\boldsymbol{x}(t)$ が微分方程式

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_\lambda(\boldsymbol{x})$$

の解となるというのが上記の意味.

- $\sigma(\cdot)$ は の独立変数は, 時間のこともあれば, 状態のこともある. $\sigma(\cdot)$ の変数が時間 t であるもの を時間駆動スイッチング, $\sigma(\cdot)$ の変数が状態 $\boldsymbol{x}$ であるものを状態駆動スイッチングと呼ぶ.

- 線形スイッチドシステムについては, 安定性の条件が詳しく調べられているが, この講義では詳細を略す.

- スイッチドシステムの例としては, スライディングモード制御 (後述), ゲインスケジューリング (後述), ファジイ制御 (後述), スーパーバイザ制御 (複数の補償器を上位の監視機構が切り換えて使う制御方式; 詳細略), 量子化制御 (制御入力が取りうる値が有限個で, それを切り換えて使うもの; 詳細略) が挙げられる.

スライディングモード制御

- **スライディングモード制御 (Sliding Mode Control;SMC)** は, 設計者が状態空間に設定した**切り換え面**と呼ばれる超平面 (あるいは超曲面) を使って制御入力を切り換える制御方式. 切り換え面は, 状態がそこに束縛されたときに, 制御系が望ましい性質 (たとえば漸近安定性) を持つように設計される.

- スライディングモード制御系は**可変構造系 Variable Structue;VSS** とも呼ばれる．歴史的には可変構造系はスライディングモード制御系より一般的な枠組だったが，今日では同一視されることも多い． 提案されたのは 1950 年代と古く，過去何回か流行し，今日でも多くの論文が出版されている．連続時間版と離散時間版があるが，この講義では前者のみ紹介する．

- スライディングモード制御系の設計は, 以下の 2 段階から成る.
  - ▷ 切り換え関数と呼ばれる関数 $\sigma(\boldsymbol{x})$ を設定する ($\{\boldsymbol{x} : \sigma(\boldsymbol{x}) = 0\}$ を切り換え面と呼ぶ) ;
  - ▷ 状態を切り換え面に到達させる制御入力 $u(t)$ を定める

- 多くの場合, 制御入力は状態の不連続関数で, 状態が有限時間で切り換え面に到達するように設定される. このようにした場合, 状態は, 切り換え面に到達した後は, 切り換え面から逸脱すると不連続な入力によって瞬間的に切り換え面に引き戻されるため, 切り換え面に束縛されたまま動く. この動作を, 切り換え面に沿って滑るという意味で, スライディングモードと呼ぶ.

- スライディングモードにおける状態の挙動 (Fillipov 解の意味で) は, 状態方程式を切り換え面に射影したものになる. 切り換え面に直交する外乱は抑制される.

- 実用上は入力の切り換えは有限の遅延を伴うので, 状態は切り換え面の近傍で振動する. この現象をチャタリングと呼び, この現象を抑制するために様々な手法が提案されている.

- スライディングモード制御は線形システムにも非線形システムにも適用できるが, 以下では, N 次の 1 入力連続時間線形システム

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u$$

を考える.

- 切り換え関数は線形でも非線形でもよいが, ここでは

$$\sigma(\boldsymbol{x}) = \boldsymbol{f}^T \boldsymbol{x}$$

とする.

- 切り換え関数を前ページのように取った場合,

$$\frac{d}{dt}\sigma^2(\boldsymbol{x}(t)) = 2\sigma(\boldsymbol{x}(t))\boldsymbol{f}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u)$$

となる.

- $\operatorname{sgn}(\cdot)$ を, 引数の符号を返す関数とする. ただし, 0 については, $\operatorname{sgn} 0 = 0$ と定義する.

- このとき, $\boldsymbol{f}^T\boldsymbol{B} \neq 0$ であれば,

$$u(t) = -(k_0 + k_1\|\boldsymbol{x}(t)\|) \cdot \operatorname{sgn}\left(\boldsymbol{f}^T\boldsymbol{B}\right)\operatorname{sgn}\left(\sigma(\boldsymbol{x}(t))\right)$$

としたうえで k_0 を正の定数, k_1 を $k_1 \geq \frac{\|\boldsymbol{f}^T\boldsymbol{A}\|}{|\boldsymbol{f}^T\boldsymbol{B}|}$ を満たす定数とすれば, $\frac{d}{dt}\sigma^2(\boldsymbol{x}(t)) < 0$ なので, 状態は切り換え面に到達する.

- $k_0 > 0$ だから到達に要する時間は有限で, 到達以降はスライディングモードになる.

- Fillipov 解の下で, 不連続な制御入力は, 実質的には, 状態を切り換え面に射影する連続な制御入力として作用していることになる．この実質的な制御入力を等価制御入力と呼び, $u_{\mathrm{eq}}(\boldsymbol{x})$ と書く．これを求める.

- $\boldsymbol{f}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u_{\mathrm{eq}}(\boldsymbol{x})) = 0$ より,

$$u_{\mathrm{eq}}(\boldsymbol{x}) = -\frac{\boldsymbol{f}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{f}^T\boldsymbol{B}}.$$

- 等価制御入力のもとで, スライディングモードにおける状態方程式は,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u_{\mathrm{eq}} \\ &= \left(\boldsymbol{I} - \frac{1}{\boldsymbol{f}^T\boldsymbol{B}}\boldsymbol{B}\boldsymbol{f}^T\right)\boldsymbol{A}\boldsymbol{x} \qquad (☆)\end{aligned}$$

となる.

- 切り換え関数は, (☆) が望ましい性質 (少なくとも安定) を持つように選ばれる. 非線形系の場合も (☆) に相当する動的システム (非線形) が望ましい性質を持つようにするという点は共通である.

- スライディングモードにおける制御入力の取り方としては，上に述べたもの以外に，以下のように様々なものがある:

- ▷ $u = \boldsymbol{k}^T(\boldsymbol{x})\boldsymbol{x}$ として $\boldsymbol{k}(\boldsymbol{x})$ の各成分を状態に応じて不連続に切り換えるもの,

- $u(\boldsymbol{x}) = -k_0 \mathrm{sgn}\,(\sigma(\boldsymbol{x})) + \boldsymbol{k}^T \boldsymbol{x}$ のように線形状態フィードバックと組み合わせるもの

- $u(\boldsymbol{x}) = -k(\boldsymbol{x})\mathrm{sgn}\,(\sigma(\boldsymbol{x}))$ のようにして $k(\boldsymbol{x})$ を非線形関数にするものなど, 様々なバリエーションがある.

- 多入力システムへの拡張は容易で, 1 入力の場合と同様に等価制御入力による解析がなされる.
- $\boldsymbol{f}^T\boldsymbol{B} = 0$ の場合に対する拡張法も知られている.
- スライディングモード制御と類似した構造がオブザーバに用いられることもある.

- 高階の微分

$$\frac{d\sigma}{dt}, \quad \frac{d^2\sigma}{dt^2}, \quad \frac{d^3\sigma}{dt^3}, \ldots$$

を考慮する High Order Sliding Mode (HOSM) と呼ばれる手法も近年流行している.

- 切り換え面を非線形にしてスライディングモードにおける状態を有限時間で原点に収束させる手法も近年流行している.

- チャタリングを抑止する手法としては, 切り換え面を複数用いる手法, 切り換え面を動的にする手法, 符号関数を連続関数で近似する手法など, 様々な手法が知られている.
- 近年, fractional order システム (微分の階数が有理数のシステムで, ガンマ関数を使って定義される) に対するスライディングモード制御が研究されている.

ゲインスケジューリング

- ゲインスケジューリングは, 時間や状態などに応じて線形補償器を切り換えることで非線形システムを制御する手法. 歴史は長く, V2 ロケットで既に使われていたという指摘もあり, 応用上は, 航空宇宙の分野では使われ続けている. 一方, 軍事機密の関係で, 1990 年代までは学術的な出版物は少なかった.

- ゲインスケジューリングのコンセプトは以下のように集約される.
    - ▷ 複数の動作領域の設定と近似線形モデルの導出
    - ▷ 近似線形モデルに対する制御系設計
    - ▷ 各補償器の補間 (スケジューリング)

具体的な設計法は様々(文献参照).

- ゲインスケジューリングのメリットは, 各動作点では線形モデルを取り扱うので, 線形システムに関する豊富な知見が利用可能で, 設計者の経験やノウハウが生かしやすいこと. 一方で, 補間の仕方等にアドホックな要素があり, 補償器が本質的に局所的であるなどの欠点がある. 連続時間システム, 離散時間システムの双方に適用可能.

- 歴史的にはゲインスケジューリングは実用寄りの手法であったが, 1990 年代以降は, **Linear Parameter-Varying(LPV) システム**として定式化され, 線形行列不等式 (説明略) に基づく解析などがなされるようになっている. これに伴い, 上述の欠点も緩和される傾向がある. 後述のファジイ制御などとの組み合わせも多い.

ファジイ制御

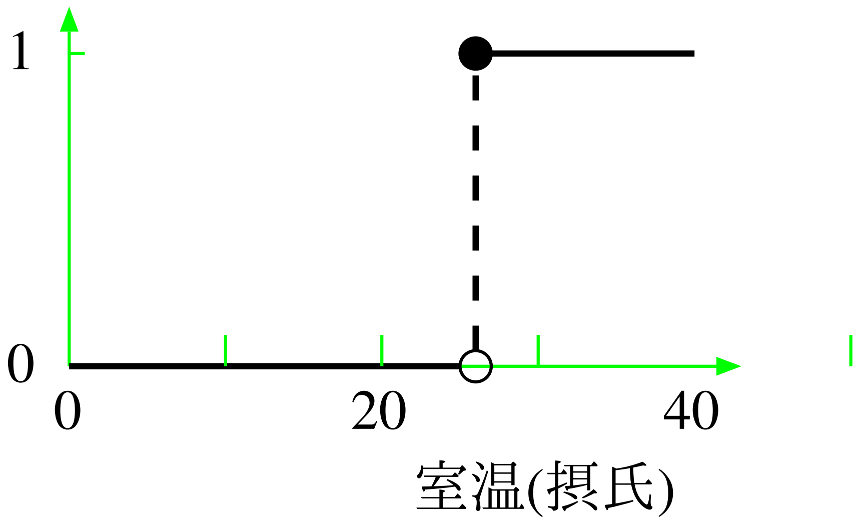
- ファジイ制御はファジイ集合やファジイ論理を部分的に導入した制御方式. ファジイ集合は1965年に提案されたもので, ファジイ制御に係る論文は1970年代に既に出版され始めている. ファジイ制御は1980年代以降何回か流行し, 今日も多数の論文が出版されている.

- ファジイ制御は, ハイブリッドシステムが流行する前に流行していたインテリジェントコントロールという枠組で, 人工ニューラルネットワーク, 遺伝的アルゴリズム, エキスパートシステムと同列に論じられることもあるが, その考え方はゲインスケジューリングと類似しており, スイッチドシステムの一種と見做すことも可能である.

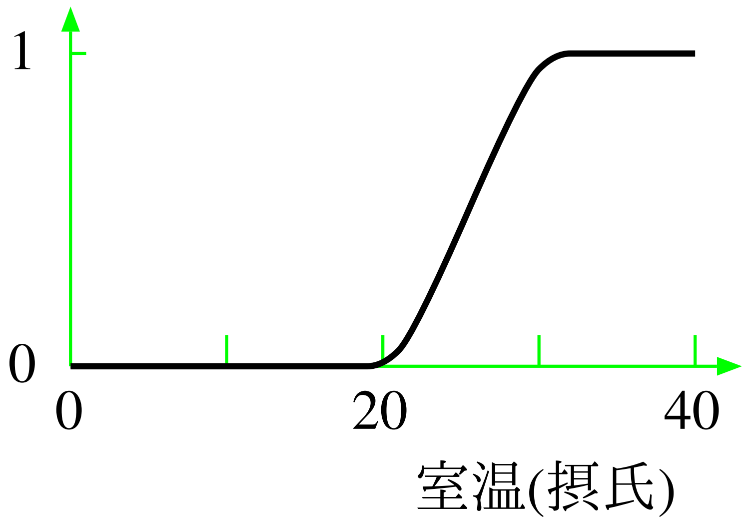
- ファジイ集合とファジイ論理は，「人間の思考の持つ曖昧さをコンピュータに取り込む」というアイデアに基づいている．

- ファジイの分野では, ファジイによらない伝統的な集合あるいは論理を **crisp** な論理と呼ぶ.

- たとえば,「この部屋は暑い」という主張は真か偽かを考えた場合 (真を値 1, 偽を値 0 で表現する), crisp な論理では,「部屋が暑い」という関数が室温摂氏 26 度未満で 0, 摂氏 26 度以上で 1 を取ると考えるのに対し, ファジイ論理では,「部屋が暑い」という関数の値は室温摂氏 26 度の前後で 0 から 1 に緩やかに変わると考える (次ページの図を参照).



crispな「この部屋は暑い」



ファジイの「この部屋は暑い」

- ファジイ集合の考え方は, 集合の特性関数を上記のような考えにしたがって連続関数 (等) で置き換えたもの.

- 集合 A が与えられているとき, その特性関数 $1_A(x)$ は, 多くの場合,

$$1_A(x) = \begin{cases} 1, & x \in A, \\ 0, & x \notin A \end{cases}$$

によって定義され, この特性関数 $1_A(x)$ を集合 A と同一視することができる.

- 上記に対応するファジイ集合とは, $\mu_A(x)$ という $[0,1]$ に値を取る関数であって, $x \in A$ のときに 1 に近い値を取り, $x \notin A$ のとき 0 に近い値を取る関数として定義される．この関数の取り方には大きな自由度がある．このように, ファジイ集合とは関数そのものであるが, この関数をメンバシップ関数と呼ぶことがある.

- crisp な論理演算 **AND** , **OR** , **NOT** は, crisp な集合に係る, 積集合, 和集合, 補集合を取る演算に基づいている. A, B を集合, $\mathrm{C}A$ を A の補集合としたとき, A と B の積集合および和集合 $A \cap B$, $A \cap B$, A の補集合 $\mathrm{C}A$ の特性関数は以下のようになる.

$$
\begin{aligned}
1_{A\cap B}(x) &= \min\{1_A(x), 1_B(x)\} \\
1_{A\cup B}(x) &= \max\{1_A(x), 1_B(x)\} \\
1_{\complement A}(x) &= 1 - 1_A(x)
\end{aligned}
$$

- ファジイ論理では, 典型的には, 上記の min, max および 1 からの減算を **AND** , **OR** , **NOT** の定義として採用し, 以下のように定義する.

$$(\mu_A \,\mathbf{AND}\, \mu_B)(x) = \min\{\mu_A(x), \mu_B(x)\}$$
$$(\mu_A \,\mathbf{OR}\, \mu_B)(x) = \max\{\mu_A(x), \mu_B(x)\}$$
$$(\mathbf{NOT}\, \mu_A)(x) = 1 - \mu_A(x)$$

高木・菅野モデル

- ファジイ制御には色々なやり方があるが, 比較的よく用いられるのは, 高木・菅野モデルと呼ばれるモデルに基づくファジィ制御 (英語では **TSK model** と呼ばれることが多い).
- 制御対象を N 次のシステムとし, その状態を $\boldsymbol{x} \in \mathbb{R}^N$, 各成分を x_i とする $(i = 1, \ldots, N)$.

- 高木・菅野モデルでは, 事前に NK 個のファジィ集合 $\{\mu_j^i : 1 \le j \le N, 1 \le i \le L\}$ を用意しておき, 制御対象を次のようにモデリングする (**IF**, **THEN** の意味は後述).

$$\mathbf{IF}\, x_1 \in \mu_1^i \mathbf{AND} \ \cdots \mathbf{AND}\ x_N \in \mu_N^i$$
$$\mathbf{THEN}\ \dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{B}_i \boldsymbol{u},$$

- 前ページの式は,

$$\{\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i\boldsymbol{u} : 1 \le i \le L\}$$

をファジィ論理で切り換えるという意味.

- **IF** , **THEN** の部分では，第 i 番目のモデル $\boldsymbol{A}_i\boldsymbol{x}+\boldsymbol{B}_i\boldsymbol{u}$ に対する確信の度合い $w_i(\boldsymbol{x})$ を

$$w_i(\boldsymbol{x}) = \mu_1^i(x_1)\mu_2^i(x_2)\cdots\mu_N^i(x_N)$$

によって定義し，

$$\dot{\boldsymbol{x}} = \frac{\sum_{i=1}^{L} w_i(\boldsymbol{x})\left(\boldsymbol{A}_i\boldsymbol{x}+\boldsymbol{B}_i\boldsymbol{u}\right)}{\sum_{i=1}^{L} w_i(\boldsymbol{x})}$$

というモデルを用いる．

- 慣例に従い **IF** , **THEN** という書き方をしたが, これは上記のような 重み付き線形結合によってモデルを求める演算であり, 論理演算ではない. なお, よりファジイ論理を前面に出した手法もある.

- 上記は線形モデルの補間であって, ゲインスケジューリングと類似しており, LPV システムの範疇に入る．また, ハイブリッドシステムの一種と見做すことも可能.

- 高木·菅野モデルに基づく制御系設計では, 上記の各線形モデルに対して制御系設計をおこない, 上記と類似した方法で補間する. LPV システムの枠組で安定性解析などができる.

- 制御対象が離散時間システムの場合には, 状態方程式が微分方程式から差分方程式に変わるが, それ以外の変更点はない.

- ファジイ制御には無数と思えるほどのバリエーションがある.

ネットワーク化制御

- ネットワーク化制御は, 制御システムがデータ通信にインターネットや無線 LAN などの汎用ネットワークを用いることを想定した上で, 通信回線の特性や制約を考慮に入れて作られた制御理論. ネットワーク化されたシステムはハイブリッドシステムとも見倣せる.

- ネットワーク化制御は この 20 年あまりで急激に成長した分野で, 今日の論文出版数はハイブリッドシステムより多く, 専門誌が立ち上げられるなど, 活発な研究が続いている.

- 考慮すべき通信回線の制約には, 遅延, 通信路容量の制約, パケットロスがある.

- この分野でどのような研究がなされているかというと…
  - ▷ 通信量を低減する量子化の手法
  - ▷ 通信路容量の制約と関連し, 安定化制御をおこなうために必要な通信路容量の計算

- ▷ センサが一定の条件を満たすときのみに状態を送信する**イベントトリガ制御**, 補償器が一定の条件を満たすときのみセンサからの信号を受信する**セルフトリガ制御**およびその安定性条件 (常時通信は通信路容量をよび消費電力の双方から好ましくないという観点).

- 制御システムをネットワークに接続された多数の装置が局所的にデータをやり取りしながら全体として一定の目的を達成するシステムと見倣し(このようなシステムを**マルチエージェントシステム**と呼ぶ), 合意形成, 情報の伝播, 分散最適化などについて調べる

- モデル予測制御 (後述) のネットワーク化制御への応用
- 端末等をクラウドに接続する前に端末に近い側で分散処理をおこなうことでクラウドへの負荷を低減するフォグコンピューティング

モデル予測制御

- モデル予測制御 (後退ホライゾン制御ともよばれる) は, 最適制御の実時間実装とでも呼ぶべき制御方式であり, 1960 年代から研究が始まっている. 初期には応用研究が多く理論的な研究はあまりなされなかったが, 1980 年代以降に安定性や最適性に関する研究がなされるようになり, 理論的な研究も盛んになった.

- 最適制御に基づいているため, 連続時間システム, 離散時間システム, ハイブリッドシステム, ネットワーク化システムのすべてに適用でき, 入力や状態に関する制約条件を含んだ形で解くことができるため (解くべき最適化問題の性質が変わるだけ), 適用範囲が広い. ただし, 一般に多くの計算機資源を必要とする. 近年大きく流行している研究分野である.

- この講義では, 離散時間非線形システムのモデル予測制御の導入部分を紹介する.

- 以下の非線形システムを考える:

$$\boldsymbol{x}(n+1) = \boldsymbol{f}(\boldsymbol{x}(n), \boldsymbol{u}(n)) \qquad (\text{☆})$$

ただし, $\boldsymbol{x}(t) \in \mathbb{R}^N$, $\boldsymbol{u}(t) \in \mathbb{R}^M$ とする.

- 次の最適制御問題を考える:

$$\min l_T(\boldsymbol{x}(K)) + \sum_{n=0}^{K-1} l_S(\boldsymbol{x}(n), \boldsymbol{u}(n))$$

$$\text{s.t.}\boldsymbol{x}(n+1) = \boldsymbol{f}(\boldsymbol{x}(n), \boldsymbol{u}(n)), 1 \le n \le K$$

$l_T(\cdot)$ は終端コスト, $l_S(\cdot)$ はステージコストと呼ばれる関数で, 一般には非負.

- 線形系では

$$l_S(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x}^T \boldsymbol{R} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u}$$

とすることが一般的だが ($\boldsymbol{R}$ と $\boldsymbol{Q}$ は正定 (半) 対称行列), 必ずそうしなければならないわけではない). 制御系の性能はこれらの決め方で決まる.

- 上記の最適制御問題を時刻 0 で解き, 得られた最適な入力を

$$\boldsymbol{u}(0; 0), \boldsymbol{u}(1; 0), \ldots, \boldsymbol{u}(K-1; 0)$$

とする．通常の最適制御では, この入力をそのまま対象に適用するのだが, モデル予測制御では次のようなことをおこなう．

▷ **時刻 0:**　最小化問題

$$\min\{l_T(\boldsymbol{x}(K)) + \sum_{n=0}^{K-1} l_S(\boldsymbol{x}(n), \boldsymbol{u}(n))\}$$

を (☆) のもとで解き, 得られた最適な入力 $(\boldsymbol{u}(0; 0), \ldots, \boldsymbol{u}(K-1; 0))$ の最初の成分 $\boldsymbol{u}(0; 0)$ を対象に適用する．残りは捨てる．

▷ 時刻 k $(k \geq 1)$**:** 最小化問題

$$\min l_T(\boldsymbol{x}(K+k)) + \sum_{n=k}^{K+k-1} l_S(\boldsymbol{x}(n), \boldsymbol{u}(n))$$

を (☆) のもとで解き, 得られた最適な入力 $(\boldsymbol{u}(k;k), \ldots, \boldsymbol{u}(K+k-1;k))$ の最初の成分 $\boldsymbol{u}(k;k)$ を対象に適用する. 残りは捨てる.

- 上記を繰り返すことで制御入力を生成するのが, モデル予測制御の基本系である.
- 毎回最適化問題を解くことの狙いは, 本来は開ループの制御である最適制御入力に状態フィードバックと類似した特性を持たせることである.

- なお, モデル予測制御には連続時間版もある. こちらは, ある時刻 t において

$$\min l_T(\boldsymbol{x}(t+K)) + \int_t^{K+K} l_S(\boldsymbol{x}(t), \boldsymbol{u}(t))dt$$

を状態方程式の制約のもとで解き, 得られた入力 $\{\boldsymbol{u}(\tau) : \tau \in [t, t+K]\}$ の設計者が決めた切片 $\{\boldsymbol{u}(\tau) : \tau \in [t, t+K_1]\}$, $K_1 \leq K$ を対象に適用する. これ以外は上記と同じ.

- ステージコストや終端コストが一定の条件を満たすとき非線形モデル予測制御によって安定化補償器が得られることや, 最適制御との関係は1990年頃に確立されている (外乱がない場合).

- 2000 年前後に 外乱を考慮しない非線形モデル予測制御がロバスト性を持たないことが判明し, 2005 年頃からロバストな非線形モデル予測制御の研究が始まり, 現在も続いている. ロバスト化の主要な方法はロバスト最適化と確率的最適化の 2 種類であるが, どちらも計算量が多いことが問題視されている. これ以外に, tube model predictive control という方法もある.

- 他にも, 最適な補償器を状態などの関数として求める explicit model predictive control と呼ばれる手法, モデル予測制御と類似した手法で状態推定をおこなう moving horizon オブザーバと呼ばれる手法など, 様々な方法が研究されている.

- モデル予測制御は, 伝統的には, 大規模な化学プロセスで多く使われていた, 近年になって, ビークル, ロボット, 電力系, 空調システム, 金融システムなど, 様々な分野への応用が増えている.

制御理論の動向

- 確定的な線形時不変システムの制御に係る理論的な研究は H_∞ 制御理論の完成をもっておおむね終息し，その後の研究は，応用へ向かうか，確定的な線形時不変システム以外に向かうか，といった方向性になっている．
- 一方で，非線形システムの制御理論は完成には程遠く，今日でも活発な研究が続いている．

- H_∞ 制御理論は，連続時間システム，離散時間システムの双方で完成しており，離散時間システムについてはサンプル値ではなくサンプル点間の波形を検討の対象とする連続リフティングと呼ばれる手法も完成している．連続時間の H_∞ 制御の応用例は非常に多いが，離散時間の H_∞ 制御は (理論的には完成しているにもかかわらず) あまり使われていない．

参考文献

- 井村, 東, 増淵, ハイブリッドシステムの制御, コロナ社, 2014.
- 潮, ハイブリッドシステムへの誘い, 計測と制御, Vol. 44, No. 7, pp. 429–433, 2005.
- 児玉, 熊谷, 離散事象システム, 計測と制御, Vol. 24, No. 7, pp. 623–632, 1985.

- W. Michiels and S. -I. Niculescu, Stability and Stabilization of Time-Delay Systems, Society of Applied and Industrial Mathematics, 2007.
- R. Goebel, R. G. Sanfelice and A. R. Teel, Hybrid Dynamical Systems, Princeton University Press, 2012.

- W. P. M. H. Heemels, B. De Schutter and A. Bemporad, Equivalence of hybrid dynamical models, Automatica, Vol. 37, pp. 1085–1091, 2001.
- W. S. Levine (ed.), The Control Handbook: Control System Advanced Methods, CRC Press, 2011.

- Y. Shtessel, C. Edwards, L. Friedman and A. Levant, Sliding Mode Control and Observation, Birkhäuser, 2014.
- R. A. DeCarlo, S. H. Zak and G. P. Matthews, Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial, Proceedings of the IEEE, Vol. 76, No. 3, pp. 212–232, 1988.

- W. J. Rugh and J. S. Shamma, Research on gain scheduling, Automatica, Vol. 36, No. 10, pp. 1401–1425, 2000.
- D. J. Leith and W. E. Leithead, Survey of gain-scheduling analysis and design, International Journal of Control, Vol. 73, No. 11, pp. 1001-1025, 2000.

- A. Ilka, Gain-scheduled controller design, Summary of doctoral dissertation, Instutute of Robotics and Cybernetics, 2015, http://www.fei.stuba.sk/docs//2015/autoreferaty/Autoreferat_Ilka.pdf (viewed: Mun 14, 2017)

- G. Feng, A survey of analysis and design of model-based fuzzy control systems, IEEE Transactions on Fuzzy Systems, Vol. 14, No. 5, pp. 676–697, 2006.