

自動車エンジンの制御開発

2007年10月9日

大畠 明

トヨタ自動車株式会社

講義の目的

複雑な制御対象をどのように扱うか？

伝えたいメッセージ

- 企業に行く人へ：企業でどのように仕事をするか？
- 大学に残る人へ：大学と企業のギャップを埋めるには？

それぞれの人にヒントが提供できれば幸い！

制御の複雑さの進展

世界初のハイブリッド車



PRIUS

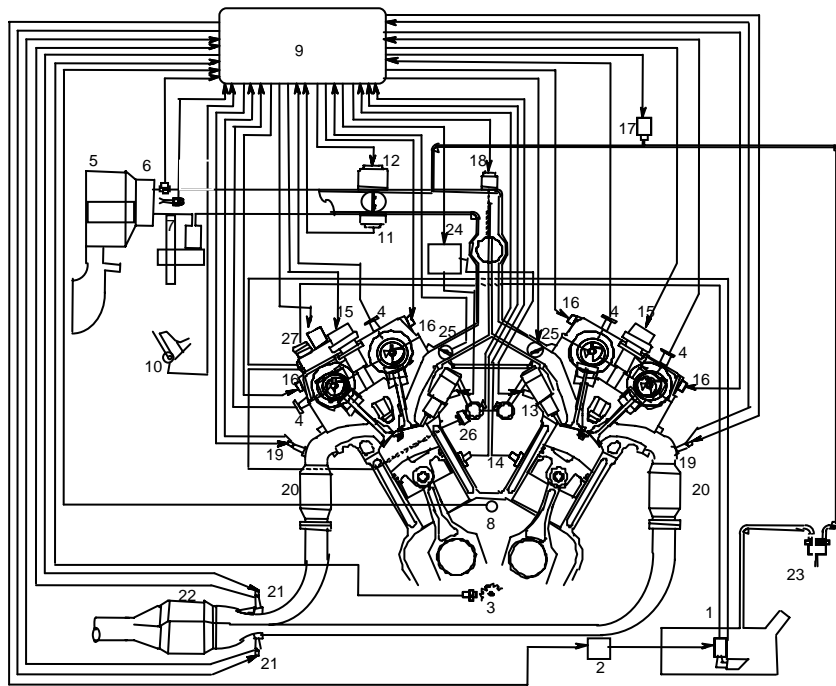
統合制御の進展



LS460

車両制御側からエンジン制御への要求が複雑化！

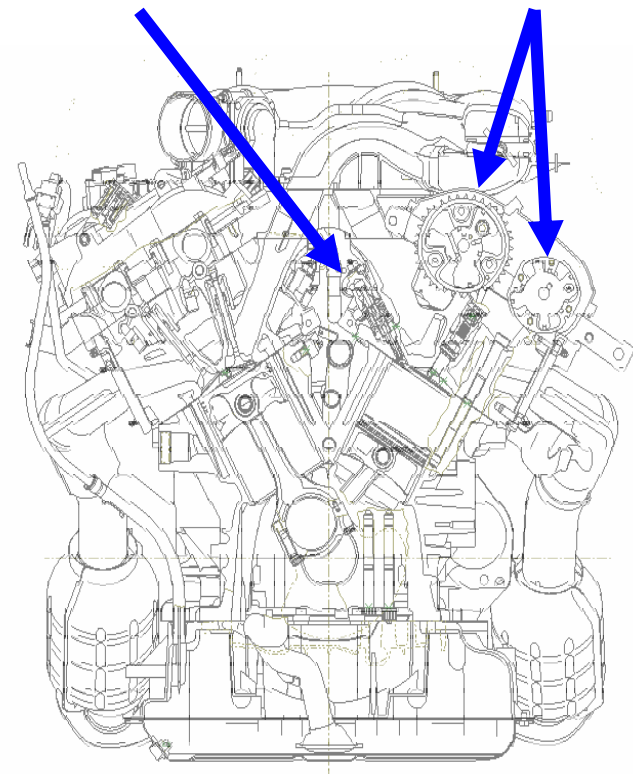
エンジン制御の複雑さの進展



ECUのコネクタピン数は200以上

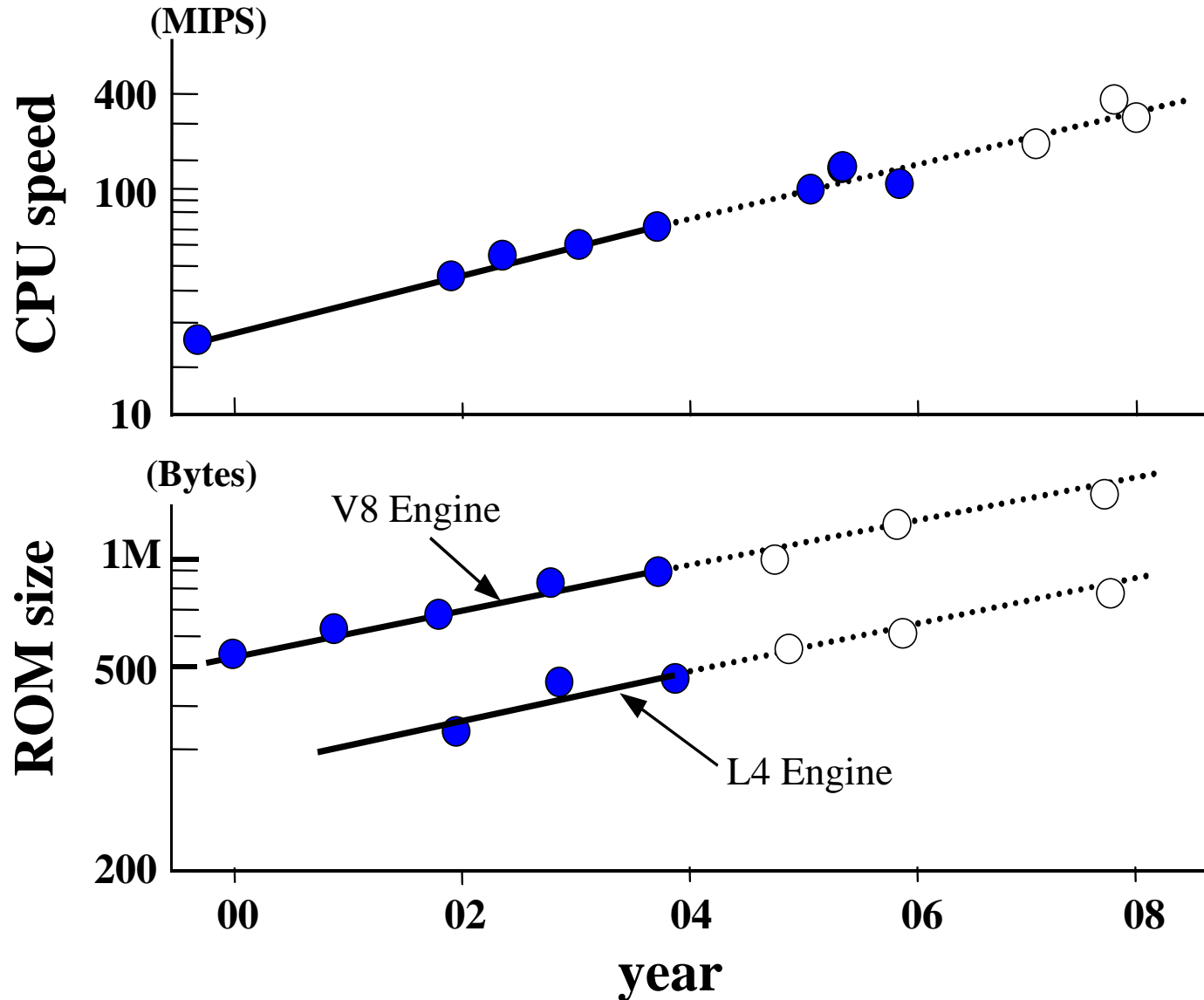
Direct Injection

In Ex VVT-i

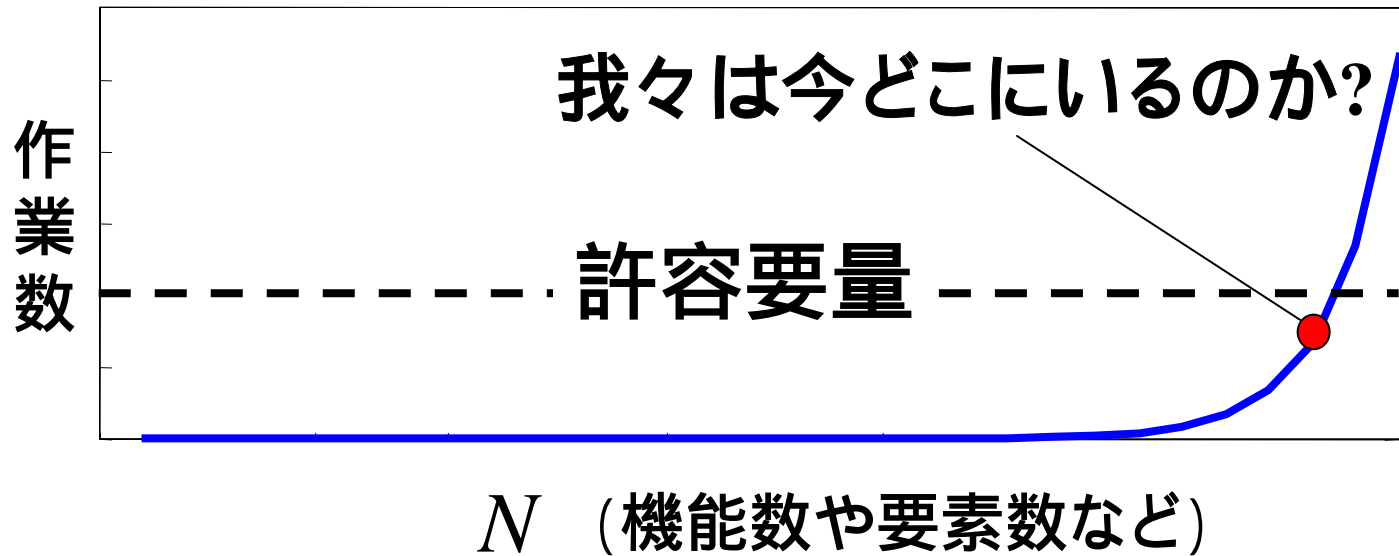


V6 3.0L 3GR-FSE

エンジン制御の複雑さの傾向

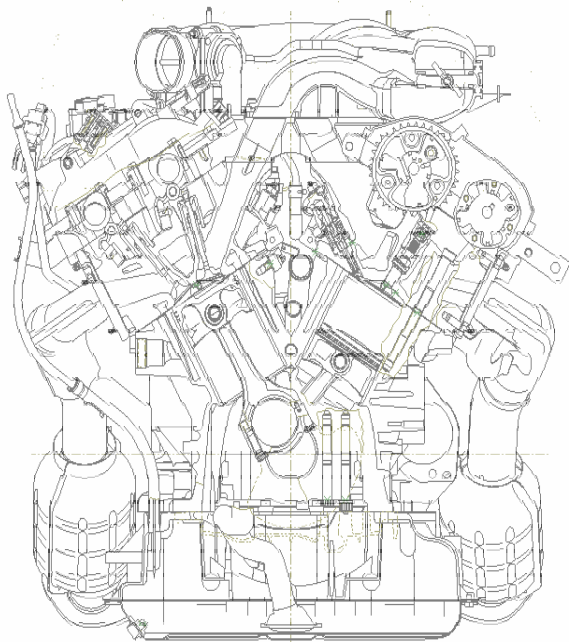


背景：自動車制御が直面する問題



これまでやったという思い込み、投資への躊躇が対応を遅らせる。対応の遅れがより深刻な事態を招く。

複雑の定義



エンジンが与えられ、
制御システムを開発
することが求められる



- 複雑な対象が与えられる。
 - モデルは与えられない。
- 企業の技術者が直面する状況

複雑：一つの理論や方法の枠組みでは解けない

背景：大学と企業のギャップ？

- 大学教育：やさしい むずかしい

簡単 複雑

複雑さを避けて問題が作られる。

- 企業活動：複雑 簡単・やさしい 統合

ここができないと仕事にならない！

統合過程が極めて重要

講義の構成

第1回講義：問題提起と

自動車制御の基礎

第2回講義：制御対象のモデリング

第3回講義：エンジンの制御系設計

第4回講義：モデルベース開発

第1回講義

問題提起と自動車制御の基礎

第1回講義の構成

- 問題提起と自動車制御の基礎 -

1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. 制御設計における問題の明確化
5. 自動車制御における制御設計
6. 第1回講義のまとめ

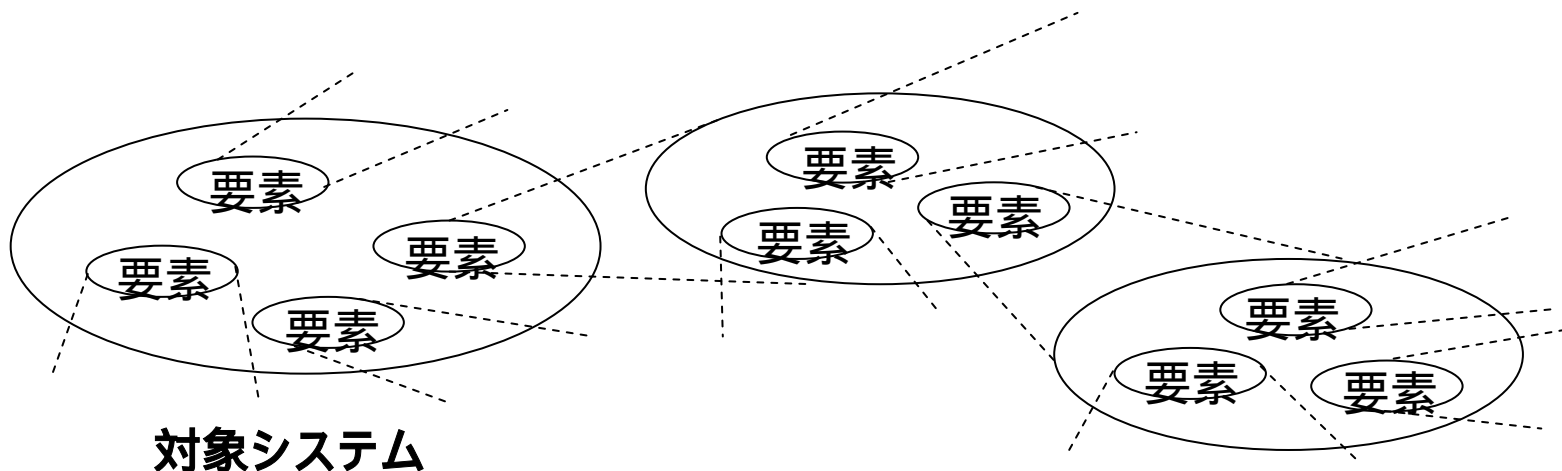
第1回講義の構成

1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. 制御設計における問題の明確化
5. 自動車制御における制御設計
6. 第1回講義のまとめ

還元主義

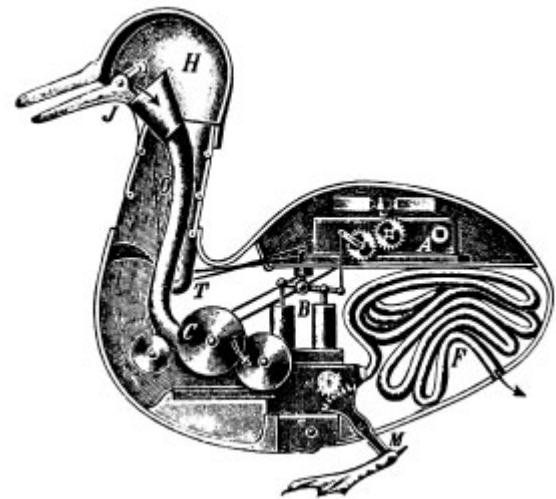
Wikipediaより

還元主義(かんげんしゅぎ、英: Reductionism、独: Reduktionismus)とは、複雑な物事はそれを構成する要素に分解し、それらの個別の要素だけを理解すれば、元の複雑な物事全体の性質や振る舞いもすべて理解できるはずだと想定する考え方。それに対する否定的な呼称。



歴史

デカルトは、動物は(人間とは異なり)還元的に、からくり人形(ある種の自動機械)として説明できるかも知れないと述べた。



そのイメージ図
(*De homines* 1622年)

歴史

還元主義の考え方は、デカルトにより1637年に刊行された『方法序説』の第5部において提示された。

デカルトは、世界を機械に譬え、世界は時計仕掛けのようであり、部品をひとつひとつ個別に研究した上で最後に全体を大きな構図で見れば、機械が理解できるように世界も分かるだろうという主旨のことを述べた。

「**分解の後に統合する**」という考え方であり、この前半の「分解」が強調され後に還元主義となってゆくことになった。

成果

還元主義は、無機的な物事を対象とする物理学や化学においては有効であった。それらの分野での17-20世紀における発展に大きく寄与した。

例えば、“物質の性質は、物質を小さく切り刻んだ要素を調べれば判るだろう”と考えることで、分子や原子の発見へとつながり、やがて素粒子研究へとつながった。

還元主義により、膨大なデータから偏差を整理しながら近づいてゆき、ひとつの推察にたどりつくという手法が生まれた。

難点

統合的分析が見落とされる原因となった。デカルトは「**分解の後に統合**」を目指していたにもかかわらず、還元主義により分解の後に分解、できるだけ分解して統合しないという傾向が生まれていった。

還元主義はそのプロセスの過程で、比較的目立たない要素の研究は放置した上で目立つ要素をばかりを深掘りしてゆく傾向を生みがちで、結果としていつまでも元の「全体」を理解するためのデータが出揃わないということになりがちだった。

還元主義による手法では理解し難い対象も存在するという考え方は、デカルトの時代以前からすでにあった。例えば、アリストテレスは「全体とは、部分の総和以上のなにかである」と述べている。全体性を見失わない考え方は「ホーリズム (Holism)」と呼ばれている。

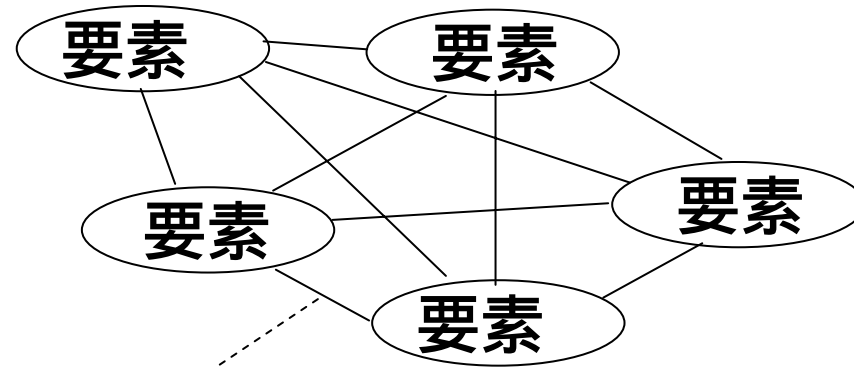
難点

様々なものは階層構造を持ち、(還元主義で得られる)下層要素の情報だけでは上層や全体の振る舞いが予想できないことが後にはっきりと認識されるようになった。

そのような現象は「創発」と呼ばれている。自然科学に携わる者の間でも化学や物理学分野で分解された要素の目覚ましい発見が一巡し、関心の中心が生体、集団、複合体といった複雑性や複合性を持つより複雑なものへと移り変わっていった。

すると、還元主義の問題点は理解されはじめ、否定的な論調で扱われる事が多くなった。その為、"複雑系の科学"の考え方や他の考え方が生まれ、現在も様々な試みが生まれている。

創発



相互作用

多数の要素間の相互作用により大規模な構造が形成



構造主義

構造主義(こうぞうしゅぎ)とは、数学、言語学、精神分析学、文芸批評、生物学、文化人類学などにおいて何らかの形で構造を重視する立場である。一般的には、研究対象を構成要素に分解して、その要素間の関係を整理統合することでその対象を理解しようとする点に特徴がある。

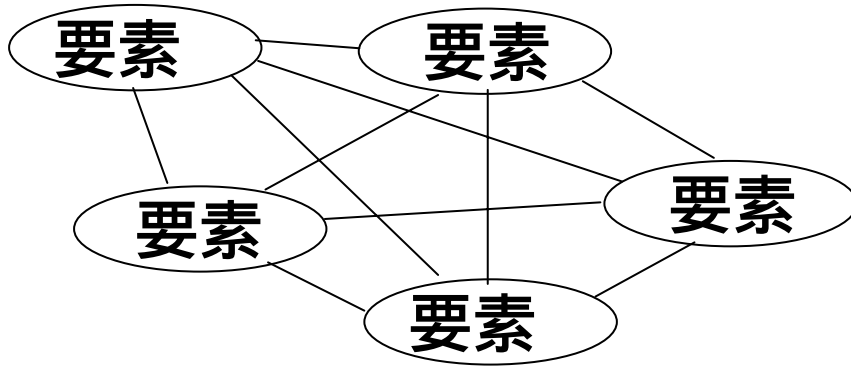
例えば、言語を研究する際、構造主義では特定の言語、例えば日本語だけに注目するのではなく、英語、フランス語など他言語との共通点を探り出していくメタ的なアプローチをとり、さらに、数学、社会学、心理学、人類学など他の対象との構造の共通性、非共通性などを論じる。

構造

要素が何であるかは問わず、要素間の関係に注目



点、線などは無定義用語。
机、紙と置き換えても数学的構造は変わらない。



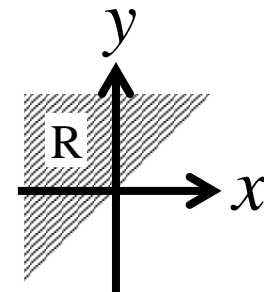
関係の数学的定義)

関係: 直積集合の部分集合

$$(x, y) \in R$$

$$y > x$$

大小関係



構造について

数学において、ブルバキというグループが公理主義的な数学の体系化を進めているが、その中心人物であるアンドレ・ヴェイユは言語学者エミール・バンヴェニストからの影響を認めている。ブルバキはしばしば「構造主義」と呼ばれるため、「構造」の起源を求めると循環論になってしまう恐れがある。

しかし少なくとも文化人類学においては、婚姻体系の「構造」は数学の群論 (group theory) と直接の関係がある。群論は代数学のひとつで、クロード・レヴィ＝ストロースによるムルンギン族の婚姻体系の研究を聞いたアンドレ・ヴェイユが群論を活用して体系を解明した。

現代思想としての構造主義

言語、文学作品、神話などを対象として分析するにあたって、語や表現などが形作っている構造に注目することで対象についての重要な理解を得ようとするアプローチがなされている。構造を見出すことができる対象は商品や映像作品などを含み、言語作品に限らない。

こうした象徴表現一般を扱う学問は記号論と呼ばれる。言語学や記号論に起源を持つ構造主義にとっての構造とは、単に相互に関係をもつ要素からなる体系というだけではなく、レヴィ＝ストロースの婚姻体系の研究にみられるように、顕在的な現象として何が可能であるかを規定する必ずしも意識されているわけではない潜在的な規定条件としての関係性を意味する。

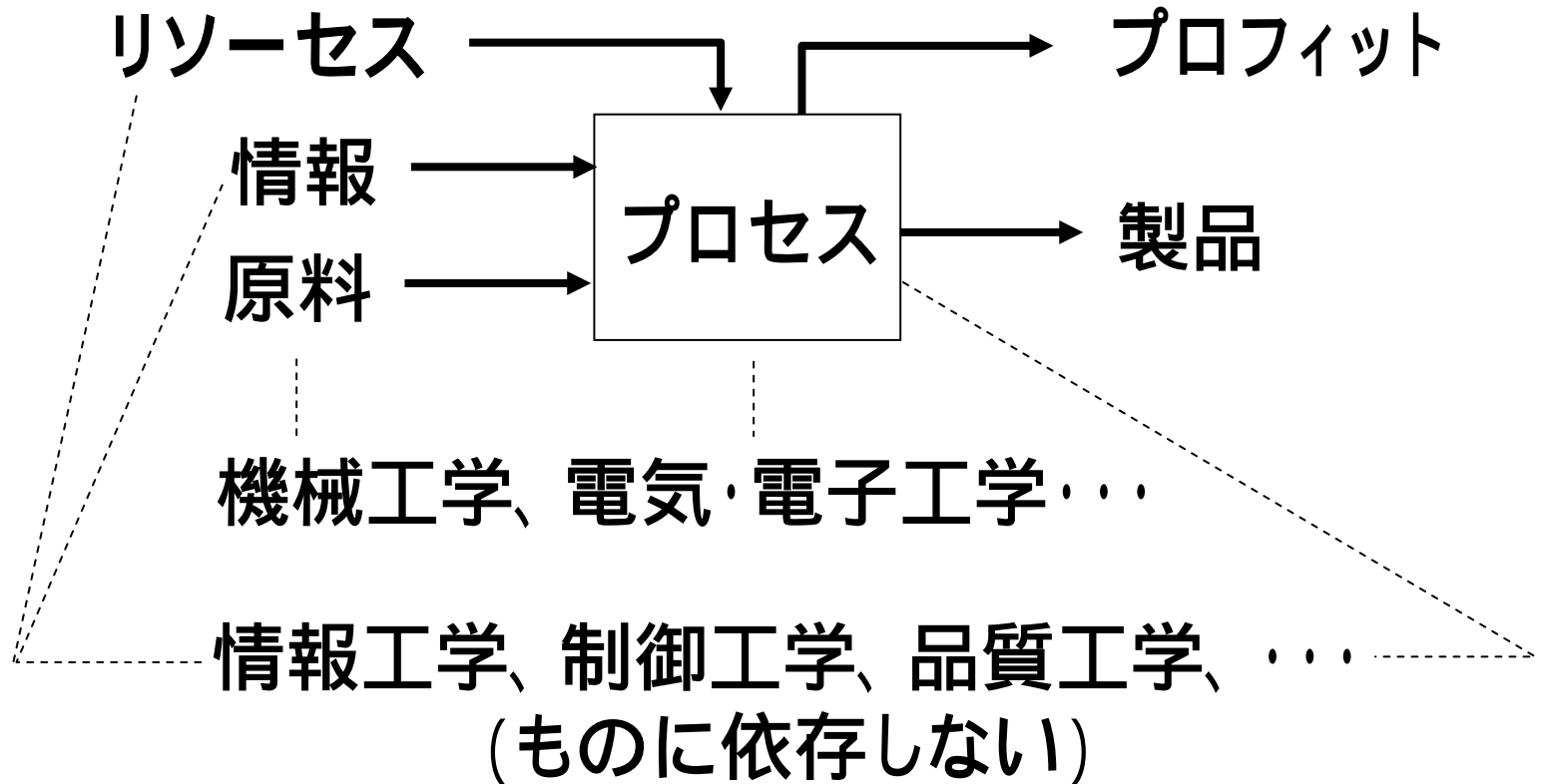
また原則として要素還元主義を批判し、関係論的構造理解が為される。ソシュールが言語には差異しかないと述べたと伝えられているように、まず構造は一挙に一つの要素が他のすべての要素との関係において初めて相互依存的に決定されるものとして与えられる。このような構造主義の構造理解においては、構造を構成する要素は、原則として構造を離れた独立性を持たない。

厳密に数学の群論にモデルを仰ぐものから、もう少し緩く、多様なバリエーションを持つ現象において、それぞれのバリエーションがその構成要素の間の組み換えによって生成されたものだと見なしうる時、その顕在的な一連の変換を規定する潜在的な構造に重心をおいて分析するようなもの全般を包含していわれることもある。

第1回講義の構成

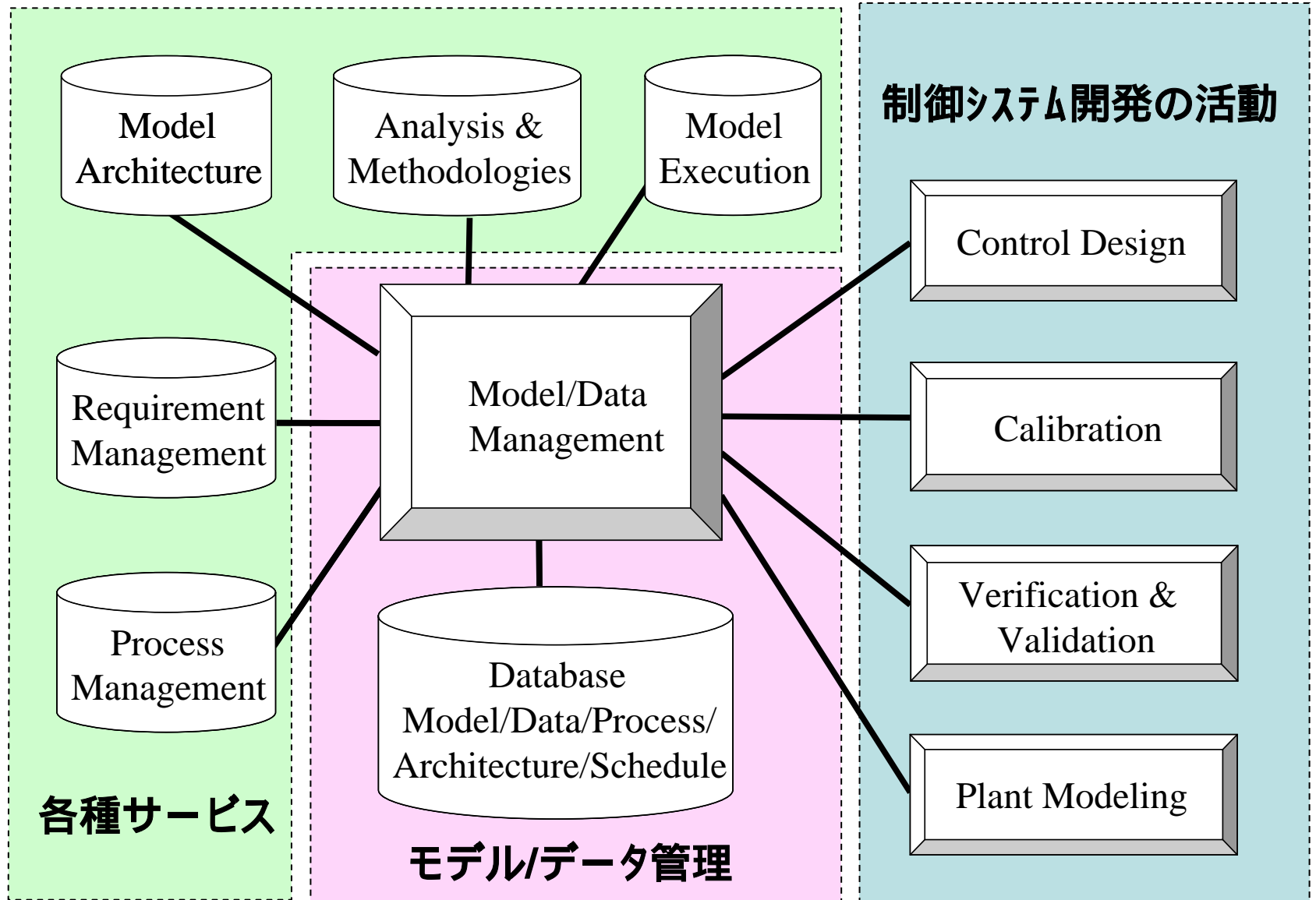
1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. 制御設計における問題の明確化
5. 自動車制御における制御設計
6. 第1回講義のまとめ

製品、プロセスと工学



現在の“製品”作りは、高度な“プロセス”が必要!!

制御システム開発



第1回講義の構成

1. 近代科学の方法論
2. 制御システム開発の構造
3. **制御設計のレビュー**
4. 制御設計における問題の明確化
5. 自動車制御における制御設計
6. 第1回講義のまとめ

制御設計

$$\begin{aligned}\frac{dx}{dt} &= A x + B u & x &\in R^n \\ y &= C x & u &\in R^m \\ & & y &\in R^p\end{aligned}$$

制御対象

$$J = \int_0^{\infty} (y^T Q y + u^T R u) dt$$

$$Q \geq 0, R > 0$$

制御目標: J を最小化

$$H = \frac{1}{2} (x^T C^T Q C x + u^T R u) + \lambda^T (A x + B u)$$

$$\frac{dx}{dt} = \frac{\partial H}{\partial \lambda^T}, \frac{d\lambda^T}{dt} = -\frac{\partial H}{\partial x}, \frac{\partial H}{\partial u} = 0, \lambda = P x, P > 0$$

$$A^T P + P A + C^T Q C - P B R^{-1} B^T P = 0$$

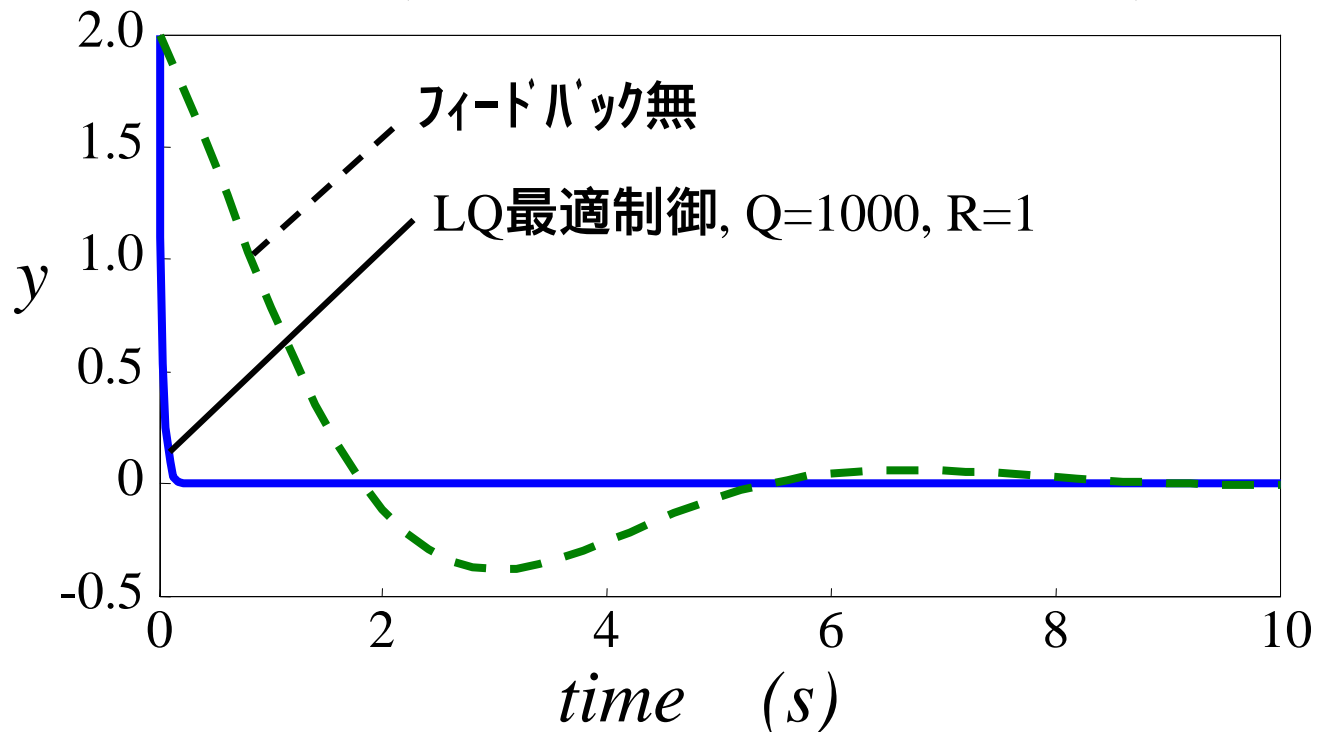
$$u = -R^{-1} B^T P x$$

最大原理によるLQ最適制御設計

LQ最適制御

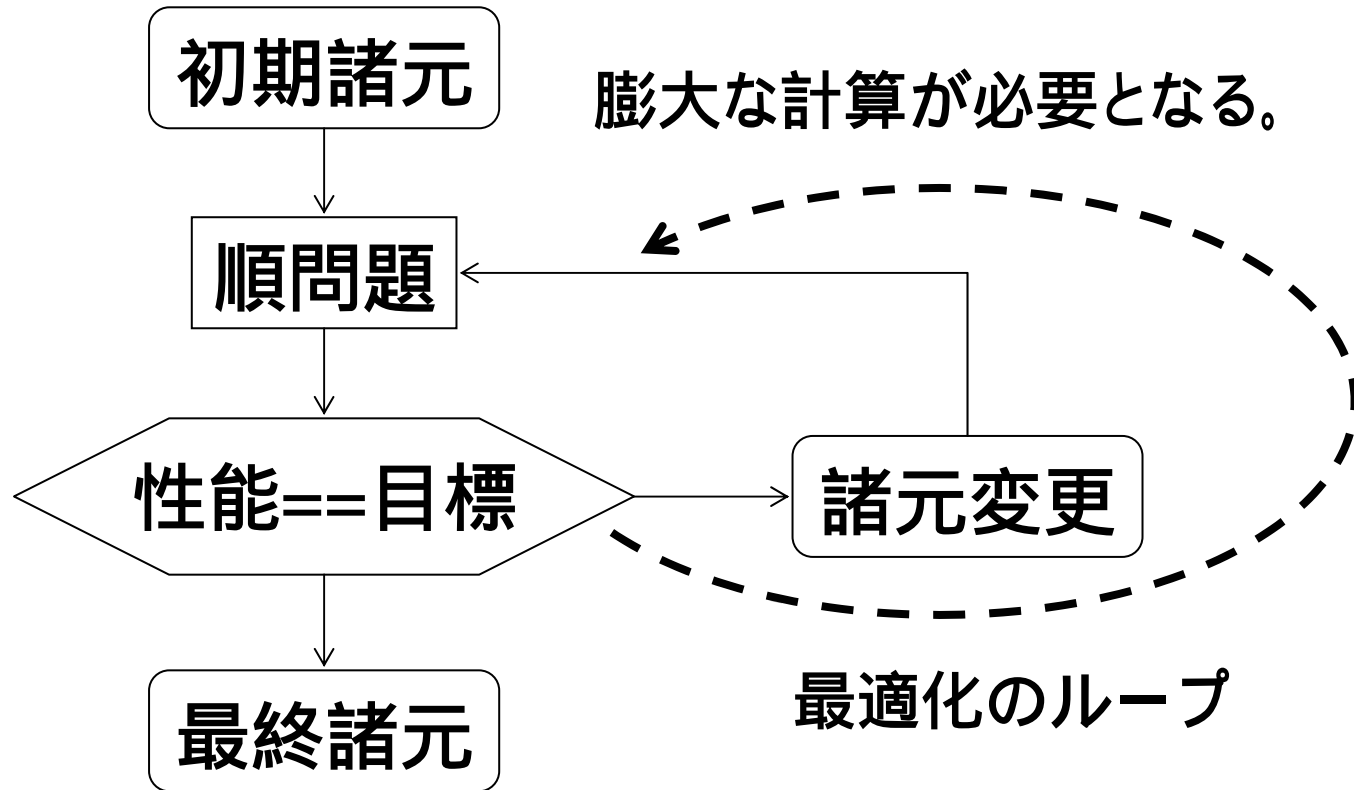
$$\frac{dx}{dt} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 1 \end{bmatrix} x,$$

$$J = \int_0^{\infty} \left(x^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} 1000 \begin{bmatrix} 1 & 1 \end{bmatrix} x + u^T u \right) dt$$



設計の構造

目標性能 設計諸元



ここを上手くやることがポイント！

順問題と逆問題

逆問題(ぎゃくもんだい、Inverse problem)とは、応用数学の一分野であり、順問題(じゅんもんだい、Direct problem)と対になる。

入力(原因)から出力(結果、観測)が求められる問題を順問題といい、その逆に出力から入力を推定する問題を逆問題という。

研究の発端は、弾道計算やレーダー探査など、軍事上の目的が主であり、第二次世界大戦中に急速に発展した。現在では、非破壊検査や医療を目的とした利用も盛んに研究されている。

制御設計の要素

逆問題：目標性能 制御仕様

前スライドの要素：

- 制御対象モデル
- 制御目標
- 最適化手法

制御は、逆問題を正面から取り組んでいる

第1回講義の構成

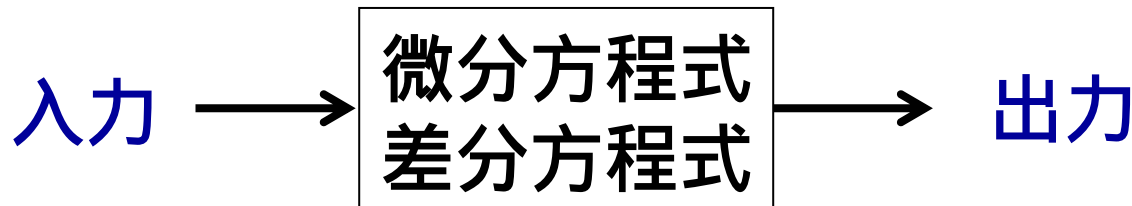
1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. **制御設計における問題の明確化**
5. 自動車制御における制御設計
6. 第1回講義のまとめ

制御対象モデル

モデル: 対象の簡易表現

(制御対象に対する知識の表現)

本来は、この表現法に限定されないはず！とは言え、万人に精度よく理解される方法が必要。プログラム = モデルでは？



- 疑問:
1. 何を記述するのか？
 2. 誰がモデルを作るのか？
 3. いつモデルを作るのか？それは可能か？
 4. どうすればモデルが作れるのか？

評価関数

$$J = \int_0^{\infty} (y^T Q y + u^T R u) dt$$
$$Q \geq 0, R > 0$$

{ Q が大きいほど、早く0に収束！
 R が大きいほど、収束が遅くなる！

- 疑問:
1. あまりにも簡単すぎるのでは？
 2. 制約もあるはず！
 3. 誰が評価関数を作るのか？
 4. いつ評価関数を作るのか？
 5. どうすれば評価関数が作れるのか？

最適制御問題

1. 最短時間制御問題:

- 初期状態から望ましい状態とする時間を最小化する

2. 終端制御問題:

- 与えられた時間で初期状態から望ましい状態にする

3. 最小積分制御問題:

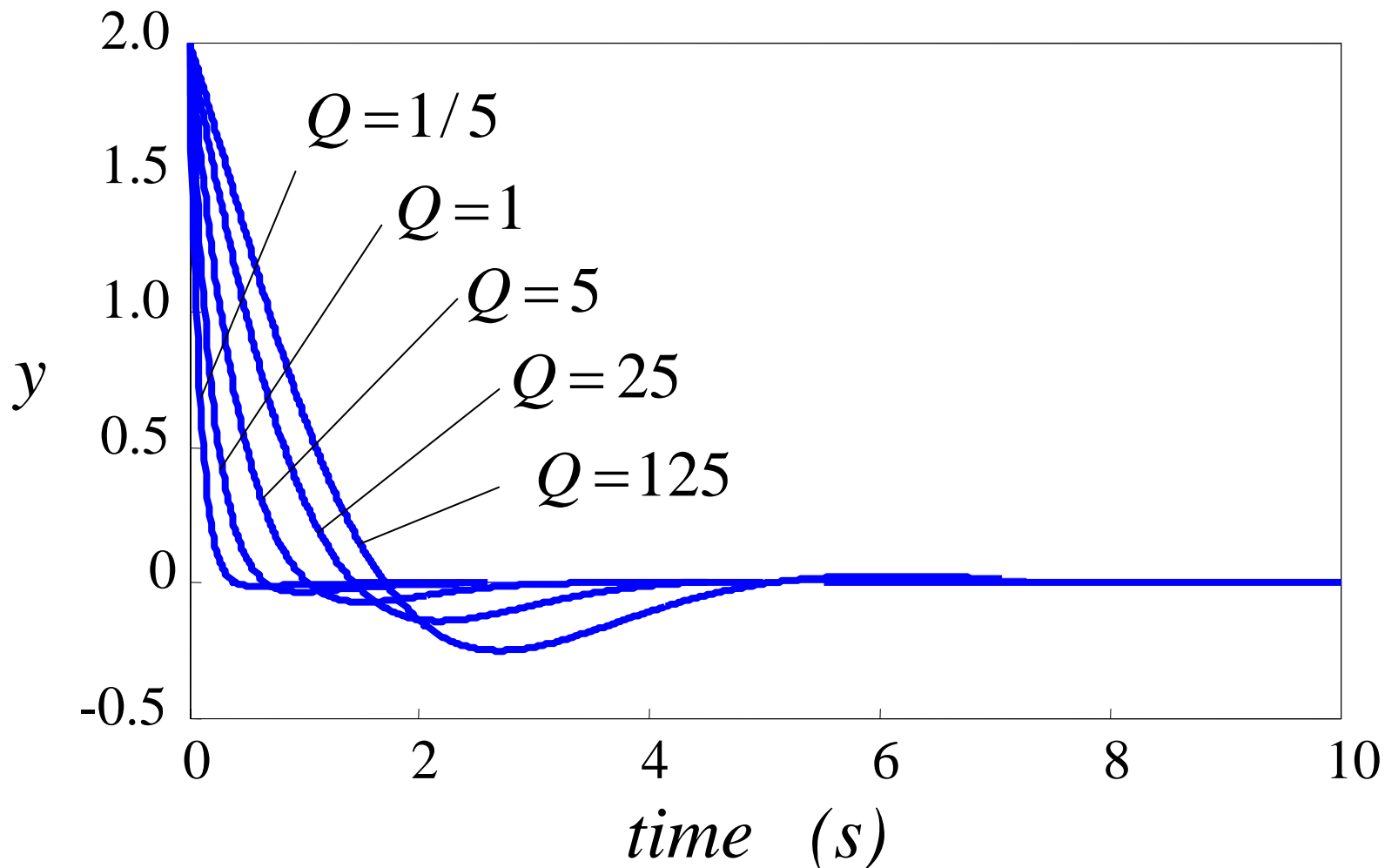
- 与えられた時間で初期状態から望ましい状態に遷移する際にある時間に対する積分量を最小化する。



うまく使い分けたり、組み合わせたりすれば、かなりのことができるかも？でもそれで十分か？

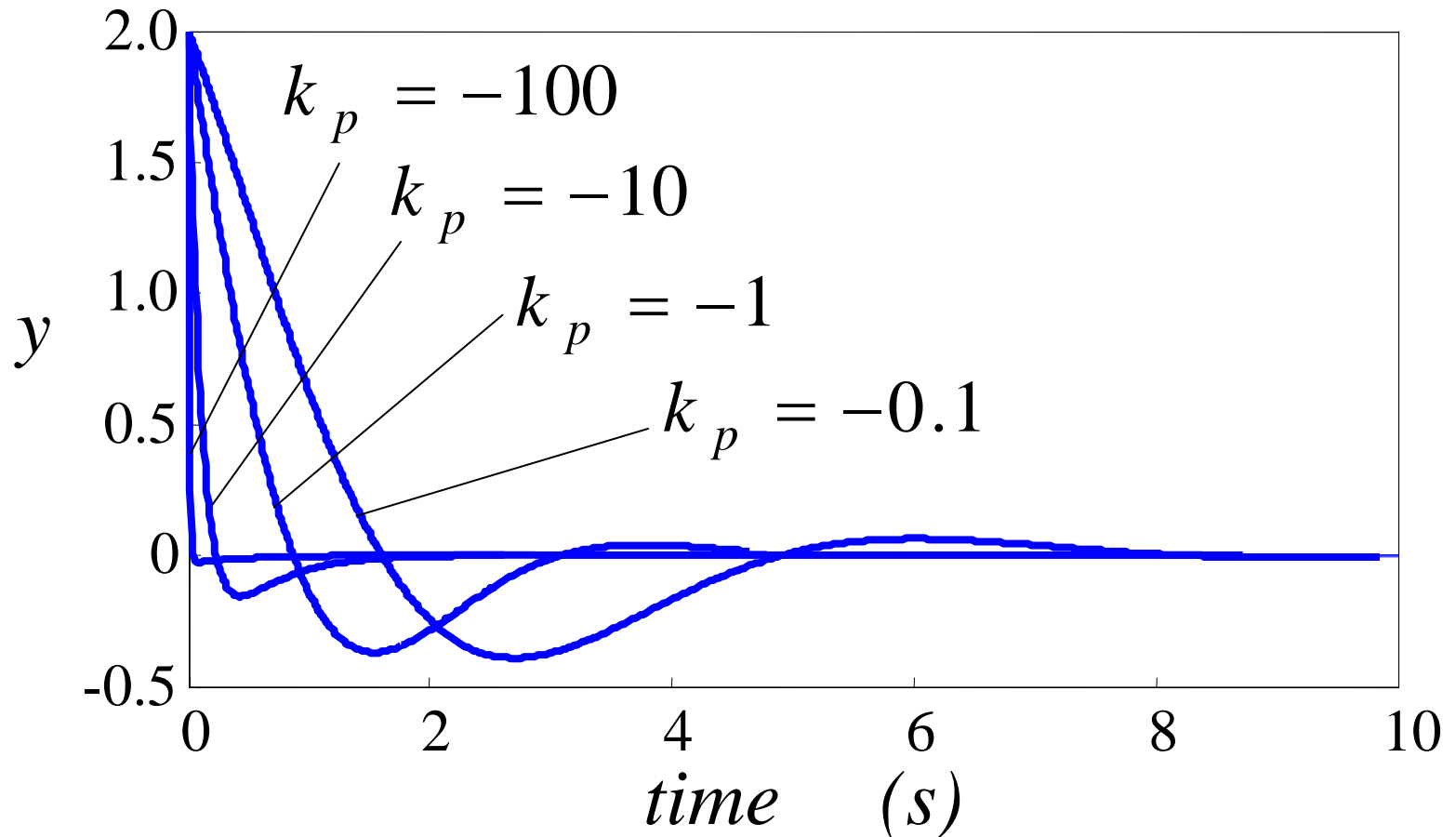
評価関数と制御性能

評価関数を変えればいくらでも性能は上がる。本当か？



比例制御

PID制御でも似たようなもの？自動車に例えると100km/hで走行している車が一瞬で止まるようなもの。ありえない！



実際の問題のギャップ

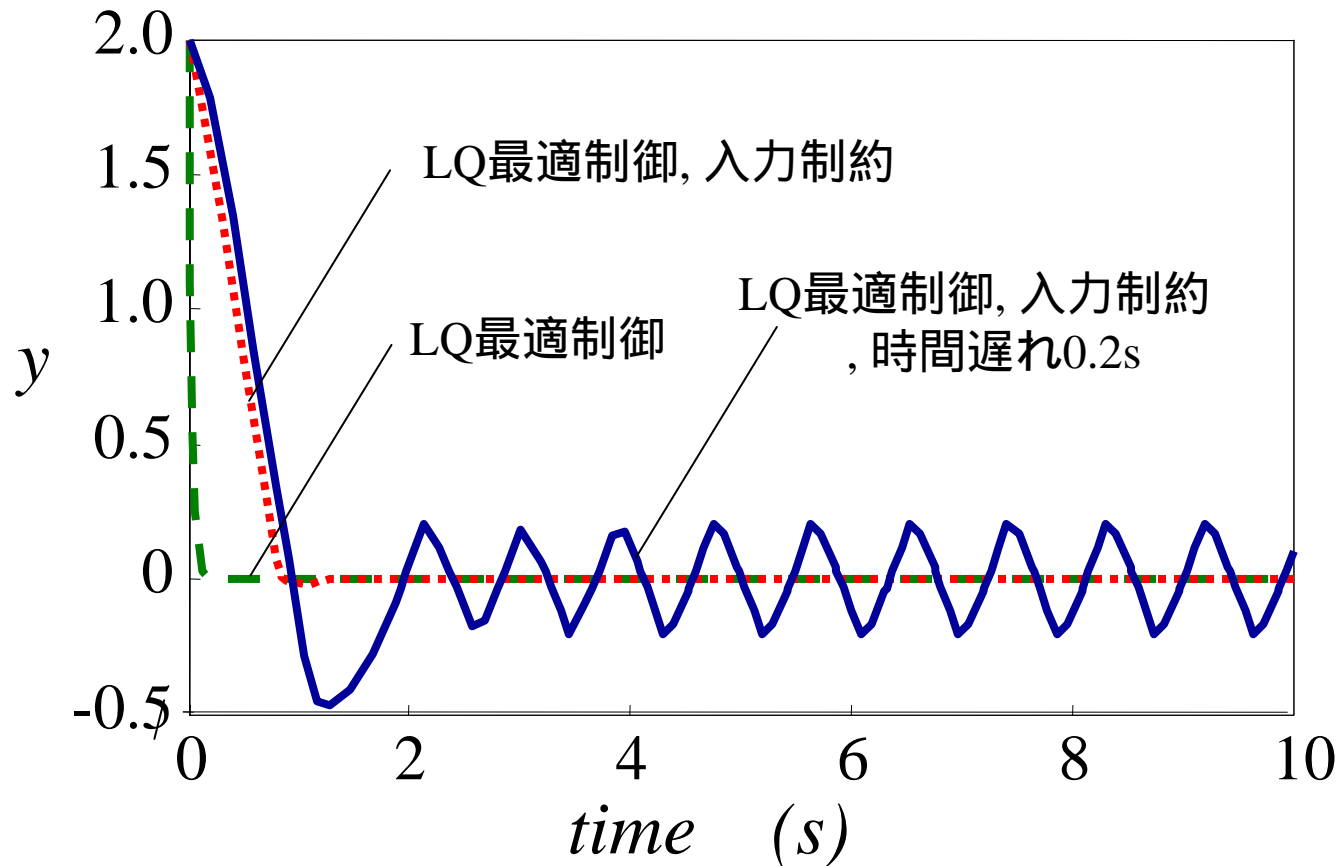
- 制御対象モデルをどう作る？
- 制御目標が単純過ぎる？
- 最適化理論は難しい
 - 制御対象モデルが単純
(実用上の各種制約があるはずだが)
 - 制御目標が単純
(それも 0 に収束するのでは...)



実際に使えるの？ { 制御対象や制御目標が複雑であるべきで最適化は簡単でない！

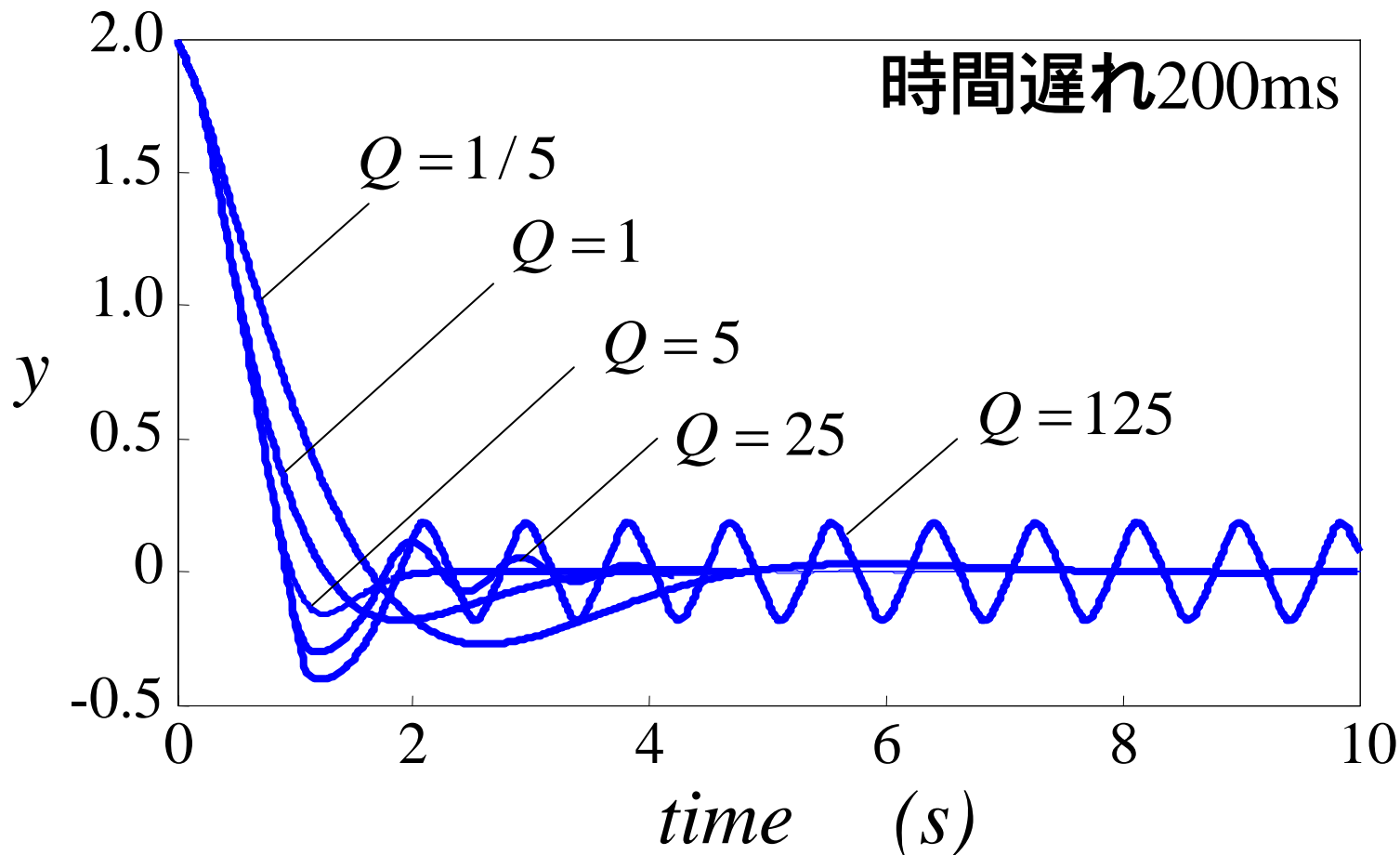
時間遅れと入力範囲制約

時間遅れと入力制約範囲は不可欠な要素



時間遅れとLQ制御特性

時間遅れで、制御特性が大きく変化

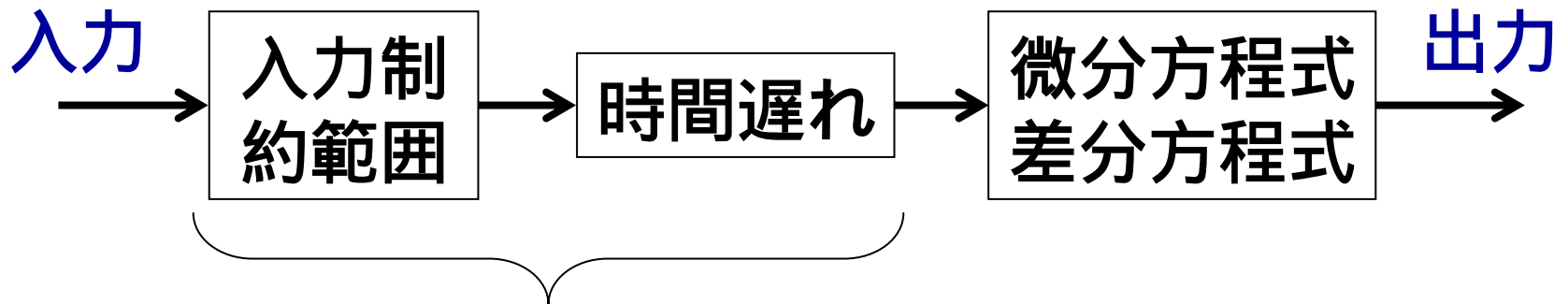


制御対象モデル

モデル: 対象の簡易表現

(制御対象に対する知識)

制御性能を決める重要な要素を記述する必要がある



制御性能を決める重要な要素

第1回講義の構成

1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. 制御設計における問題の明確化
5. **自動車制御における制御設計**
6. 第1回講義のまとめ

目標追従制御

制御対象とモデルを下記の通りとする。

$$\begin{array}{ll} \text{制御対象:} & \frac{dx}{dt} = f(x, u) \\ & y = g(x) \\ \text{モデル:} & \frac{dx_m}{dt} = f(x_m, u_m) \\ & y_m = g(x_m) \end{array}$$

$e_x = x - x_m, e_u = u - u_m, e_y = y - y_m$ とし、誤差システムを作る。

$$\frac{de_x}{dt} = f(x, u) - f(x_m, u_m) \cong \frac{\partial f}{\partial x} \Big|_{\substack{x=x_0 \\ u=u_0}} e_x + \frac{\partial f}{\partial u} \Big|_{\substack{x=x_0 \\ u=u_0}} e_u$$

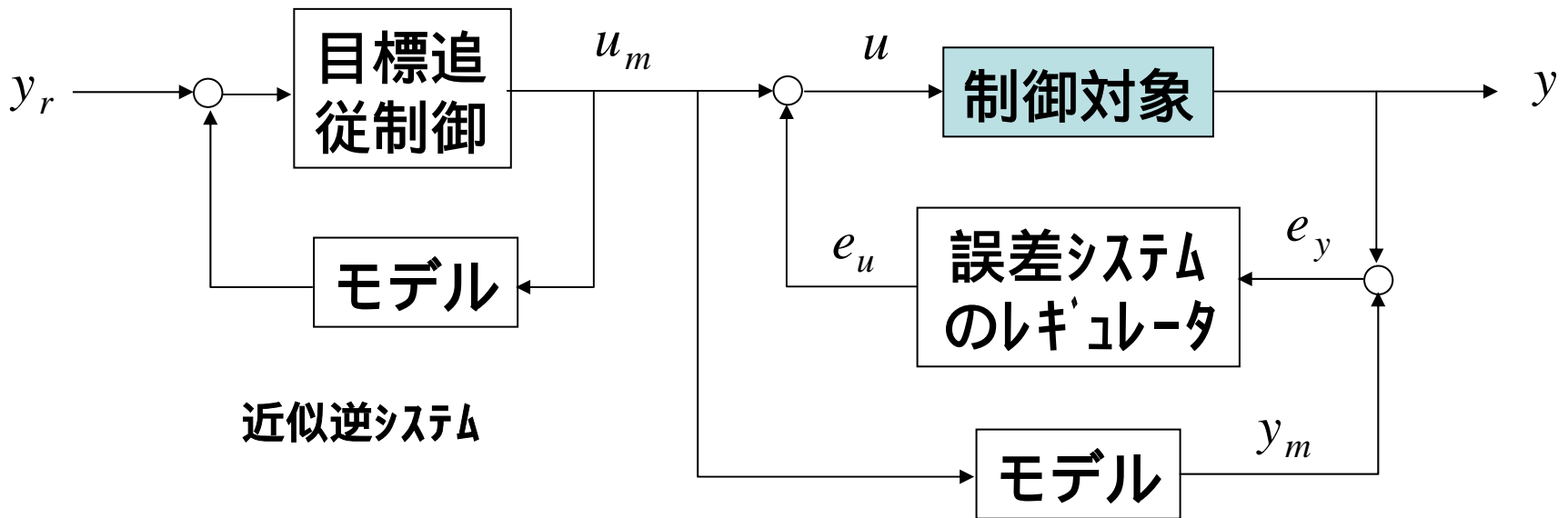
$$e_y \cong \frac{\partial g}{\partial x} \Big|_{x=x_0} e_x$$



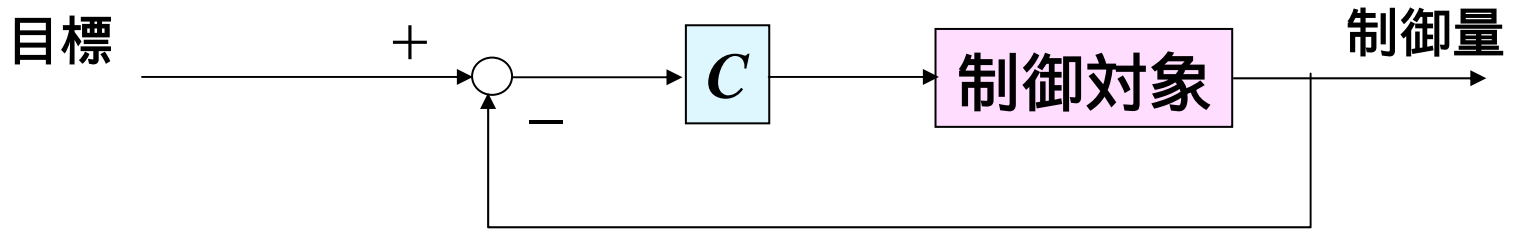
$$\begin{array}{l} \frac{de_x}{dt} = A e_x + B e_u \\ e_y = C e_x \end{array} \quad A = \frac{\partial f}{\partial x} \Big|_{\substack{x=x_0 \\ u=u_0}}, B = \frac{\partial f}{\partial u}, C = \frac{\partial g}{\partial x} \Big|_{x=x_0}$$

目標追従制御

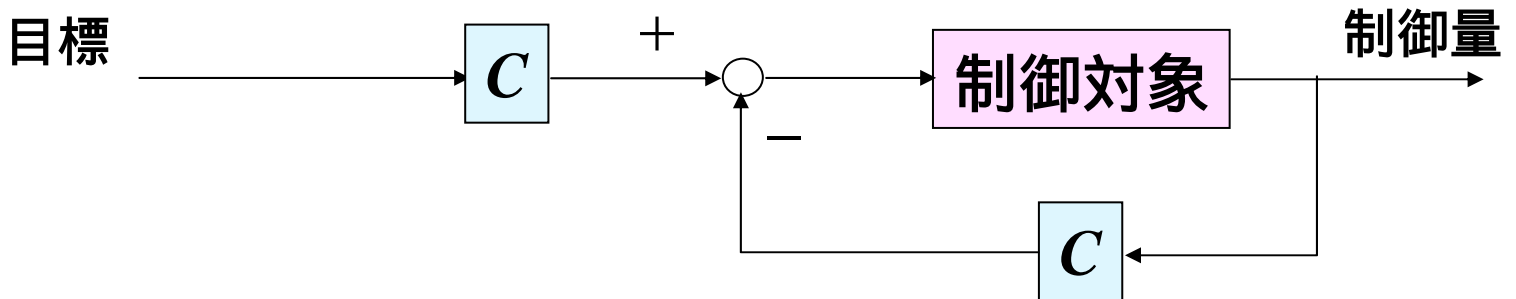
$$\frac{de_x}{dt} = A e_x + B e_u$$
$$e_y = C e_x$$
$$A = \frac{\partial f}{\partial x} \Big|_{x=x_0, u=u_0}, B = \frac{\partial f}{\partial u} \Big|_{x=x_0, u=u_0}, C = \frac{\partial g}{\partial x} \Big|_{x=x_0}$$



よく見る目標追従制御系

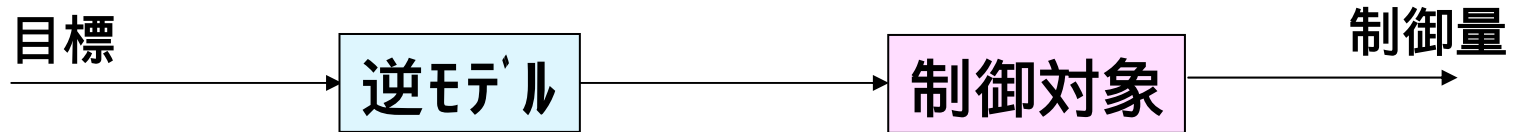


↓ 等価変換

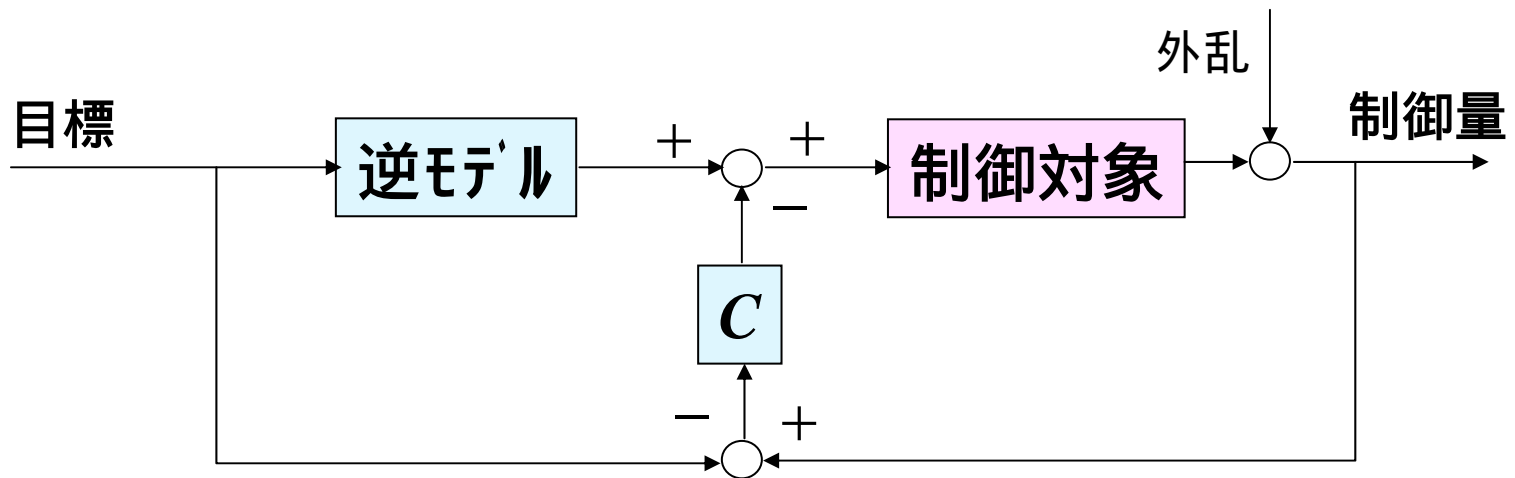


フィードバックとフィードフォワードが同じ！

自動車制御における 目標追従制御系の基本考え方

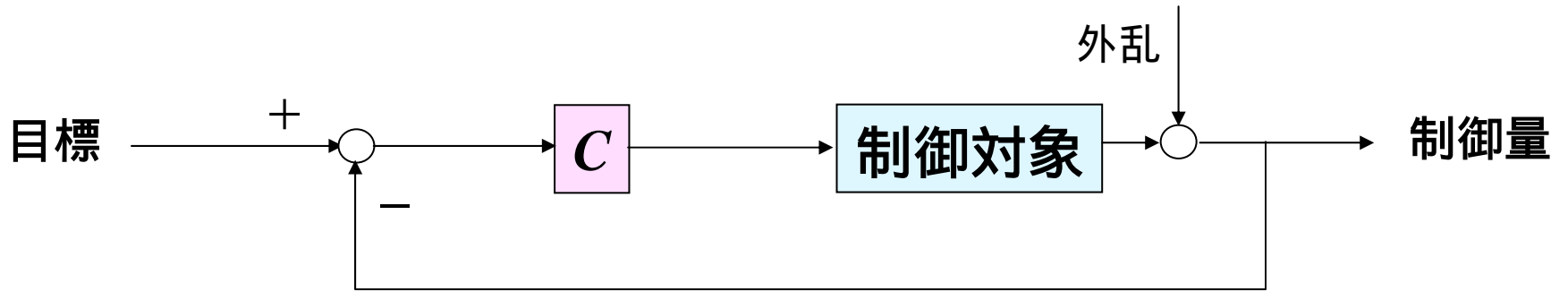


フィードフォワードによる目標追従制御系

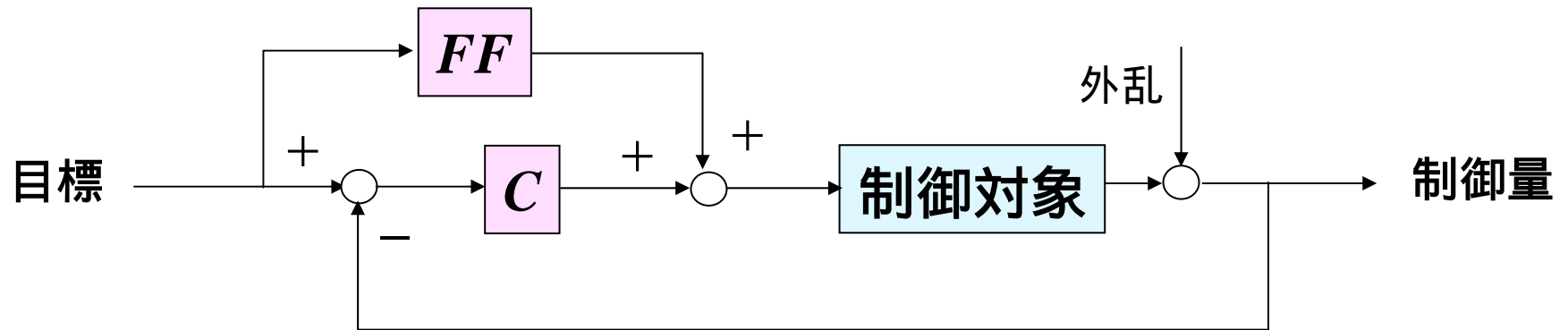


外乱(モデル誤差を含む)を補償する制御系

2自由度制御系

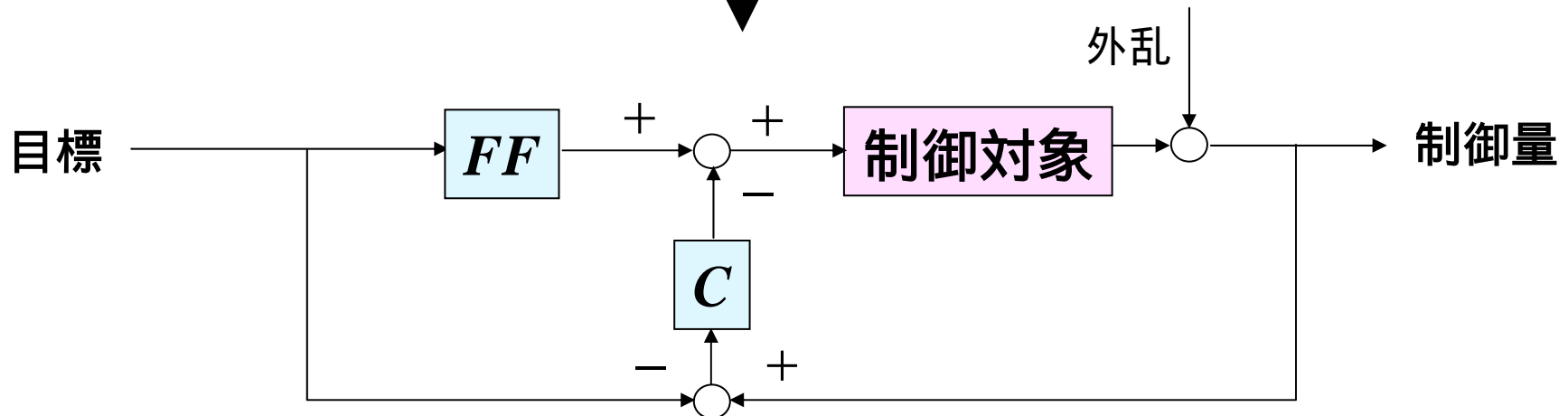
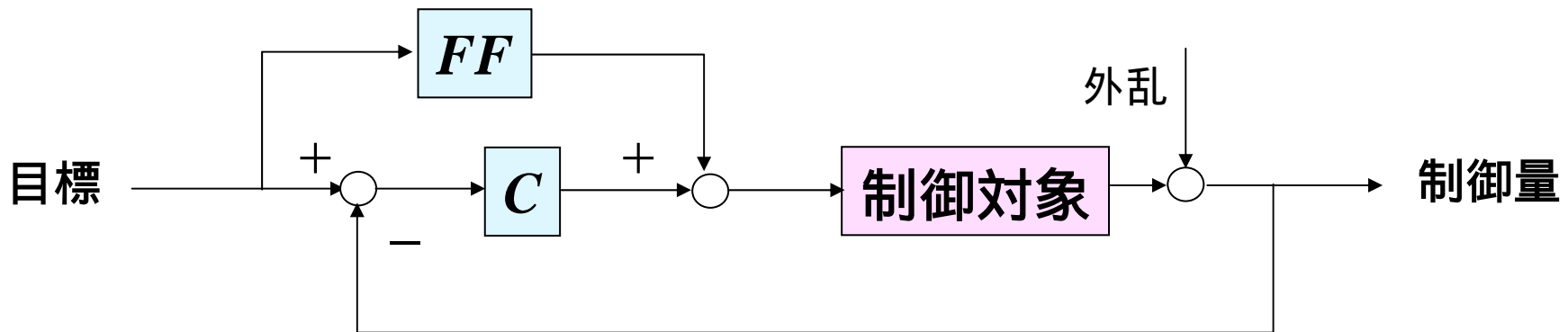


よく使われる目標追従制御系

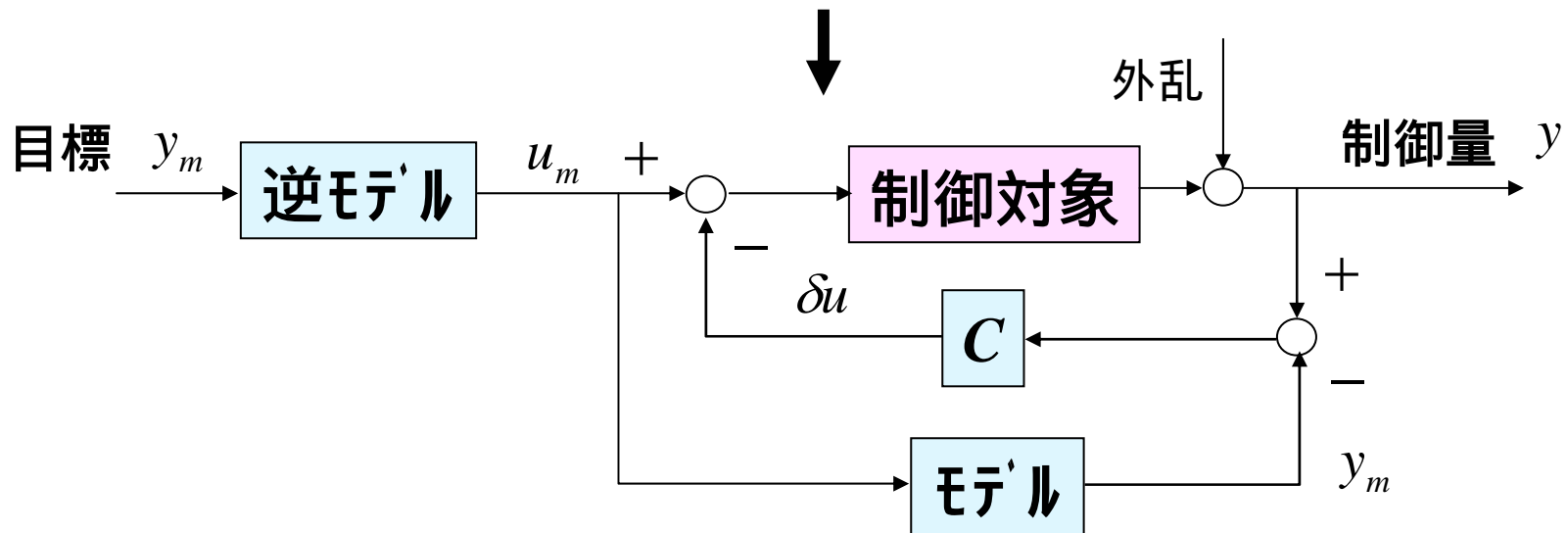
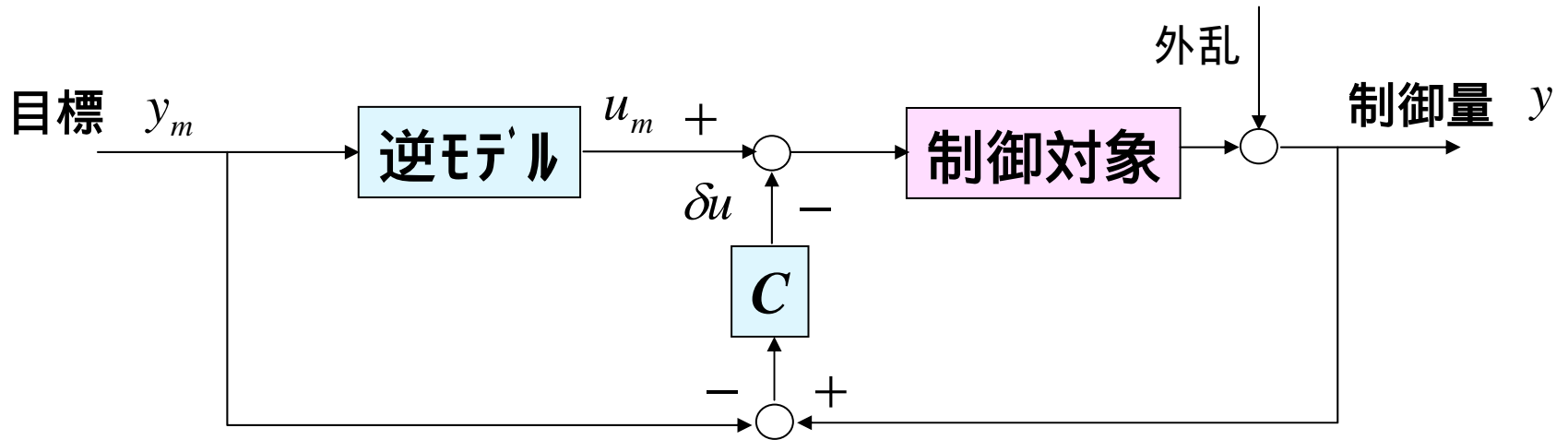


2自由度制御系

2自由度制御系の等価変換

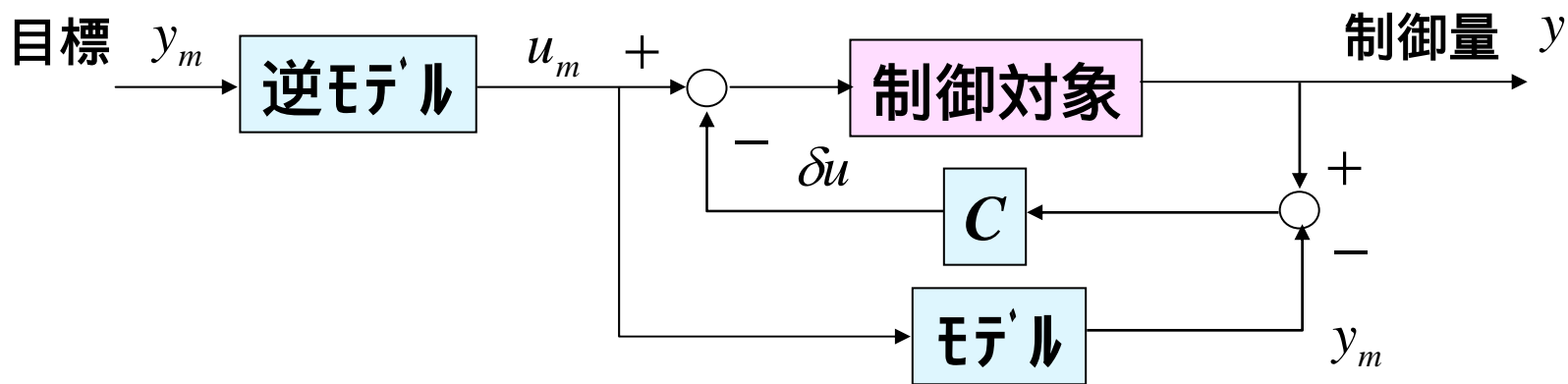
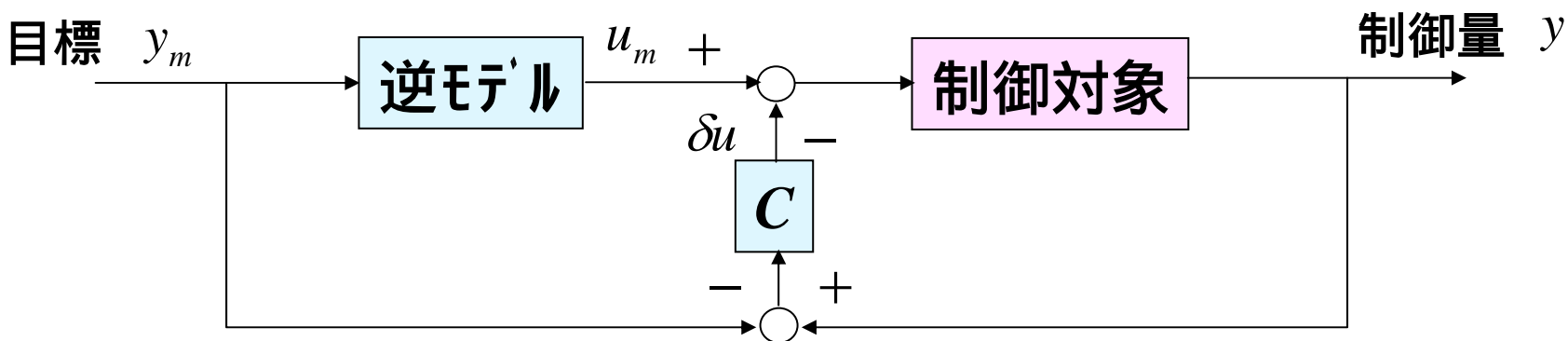
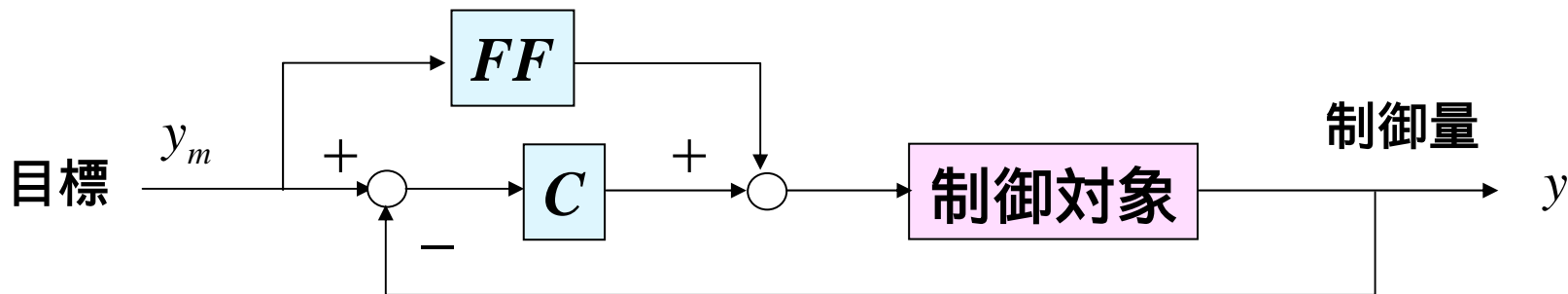


目標追従制御系の等価変換



この部分のまとめ

1. 自動車制御では、目標追従制御が多用される。
2. 目標追従制御には、2自由度制御が有効である。
3. 逆システムをベースとするフィードフォワードが使われる
(機械で実現していた機能を電子制御にした歴史的理、センサが活性化しない時期の存在、フェールセーフ、高応答性などの要求による)
4. 考え方などで導かれるブロック線図が異なる。
5. 等価なブロック線図が各種あり、それぞれ名前を付けられることがある。
(2自由度制御、Internal Model Control、外乱推定オブザーバなど)



スミスの時間遅れ補償系

$G e^{-sL}$: 制御対象

G_m : G のモデル

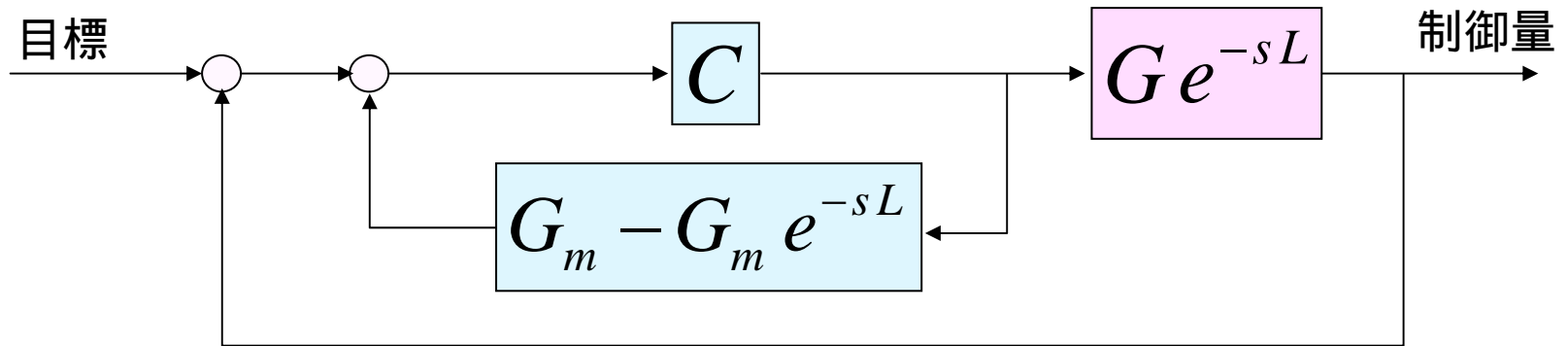
L : 時間遅れ

C : コントローラ

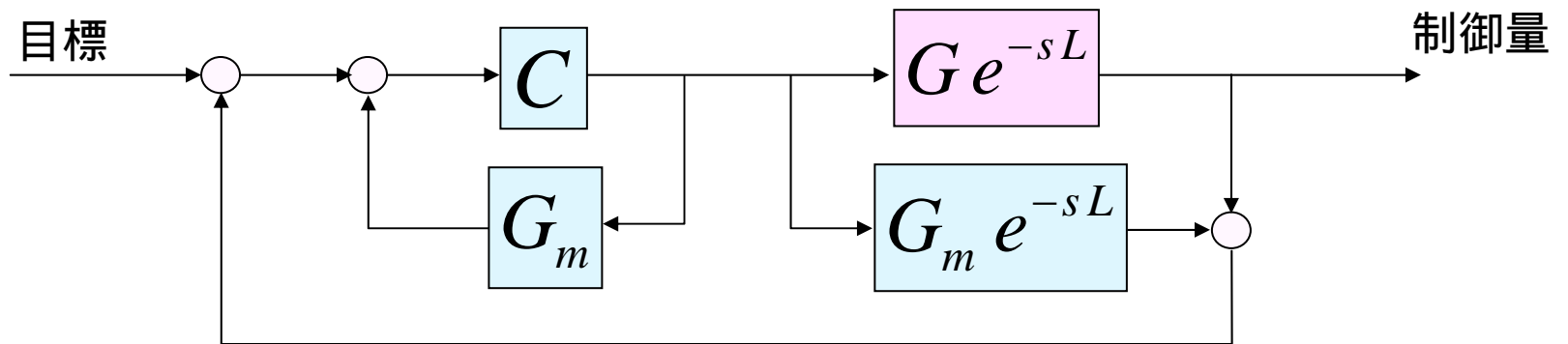
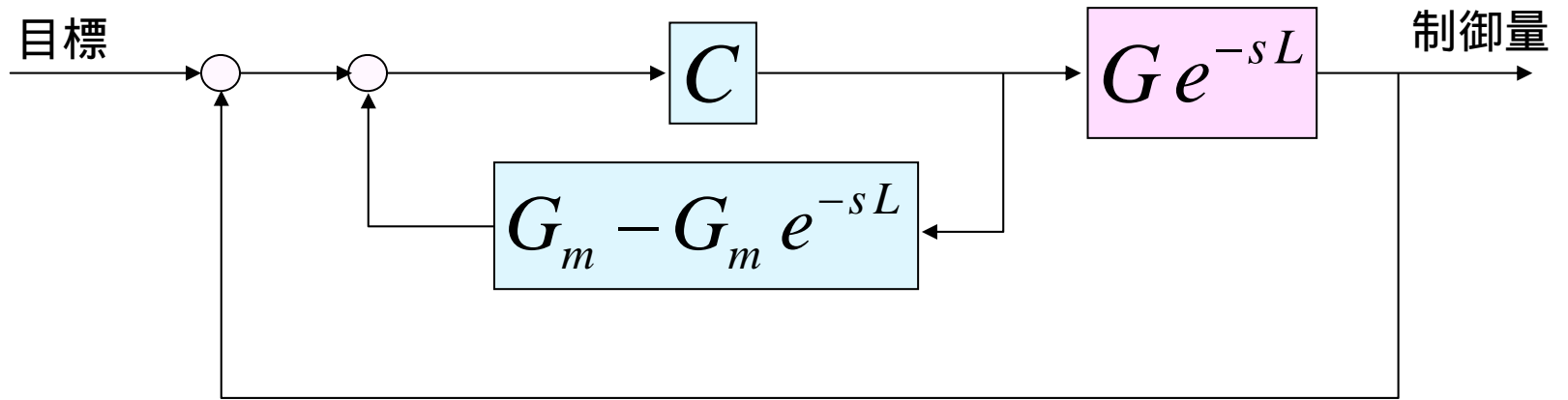
$$\frac{G e^{-sL} C_d}{1 + G e^{-sL} C_d} = \frac{G C}{1 + G C} e^{-sL}$$

とする C_d を求めると下式

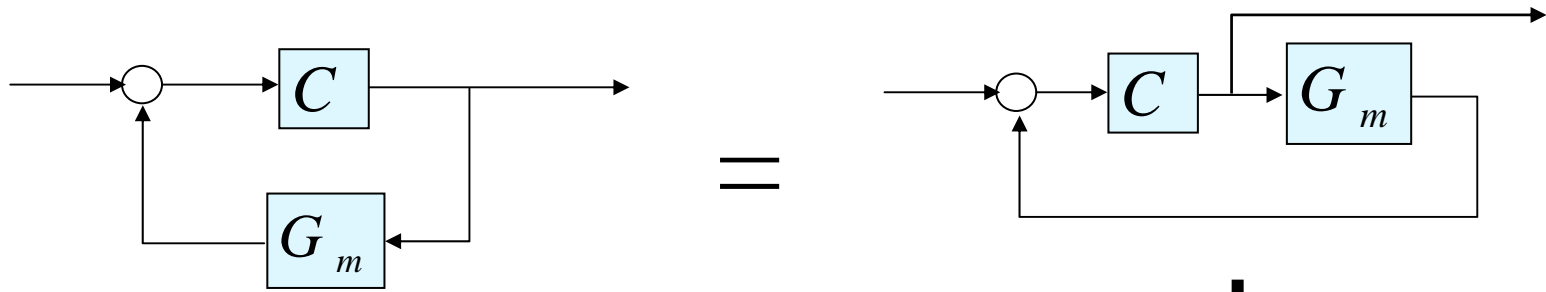
$$C_d = \frac{C}{1 + G C - G e^{-sL} C}$$



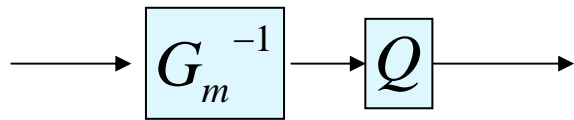
スミスの時間遅れ補償の等価変換



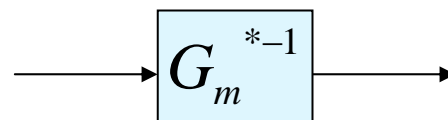
近似逆モデル



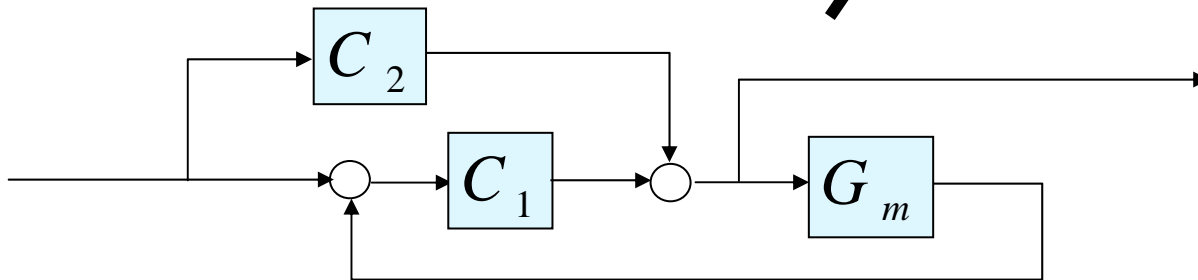
時間遅れ無し近似逆モデルと解釈する



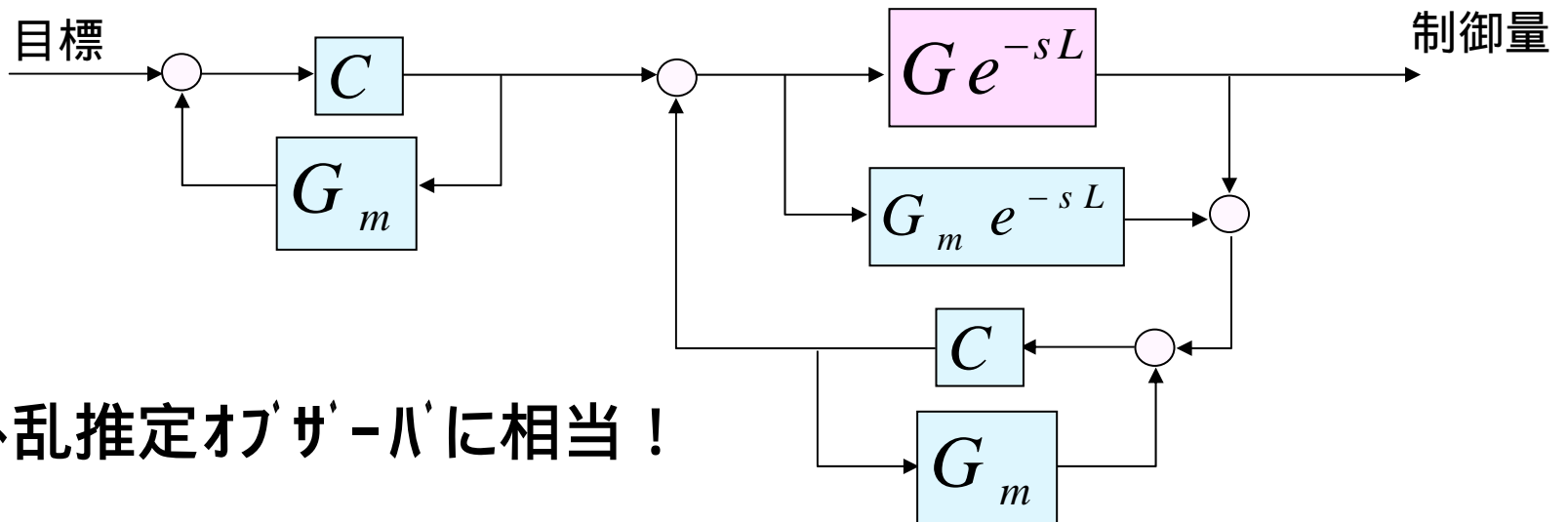
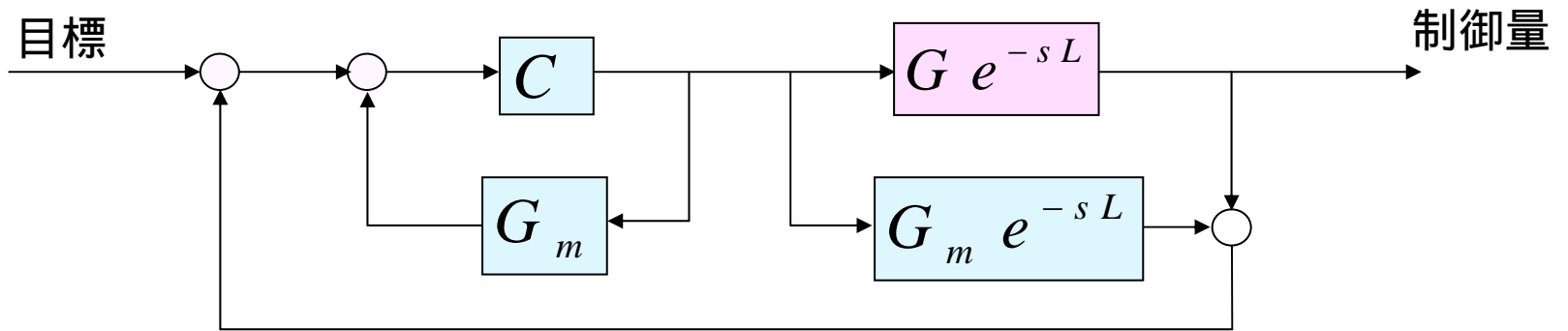
目標に遅れを持って追従する



(必要があれば拡張)

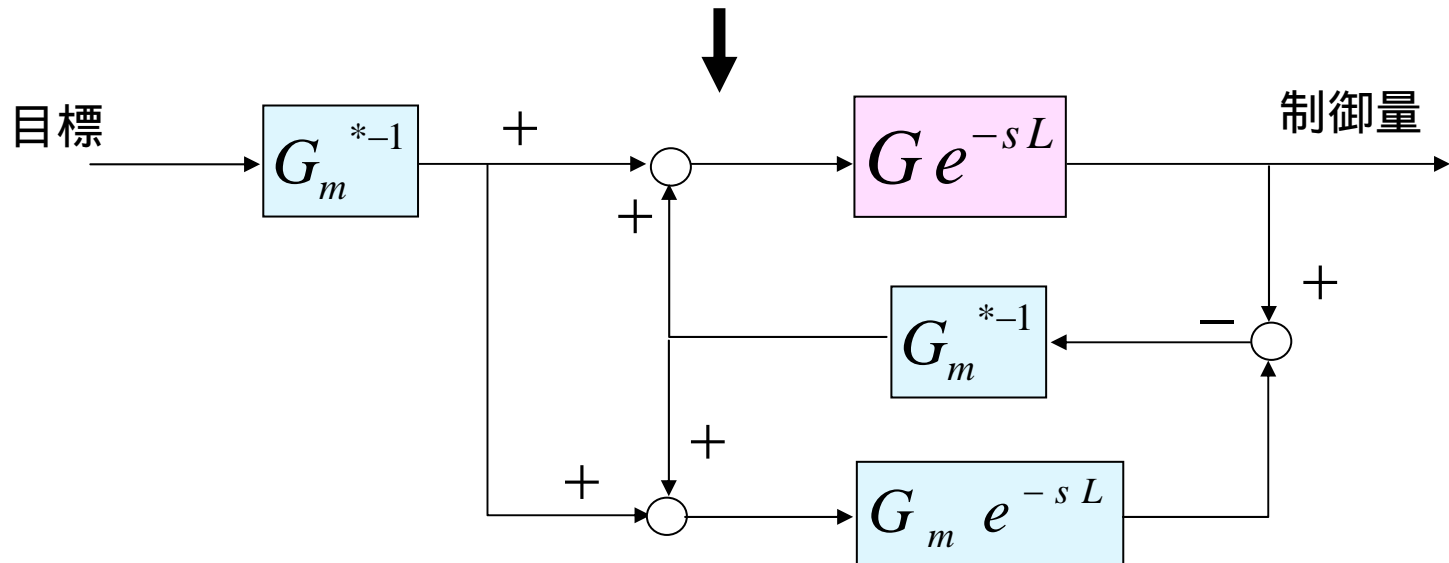
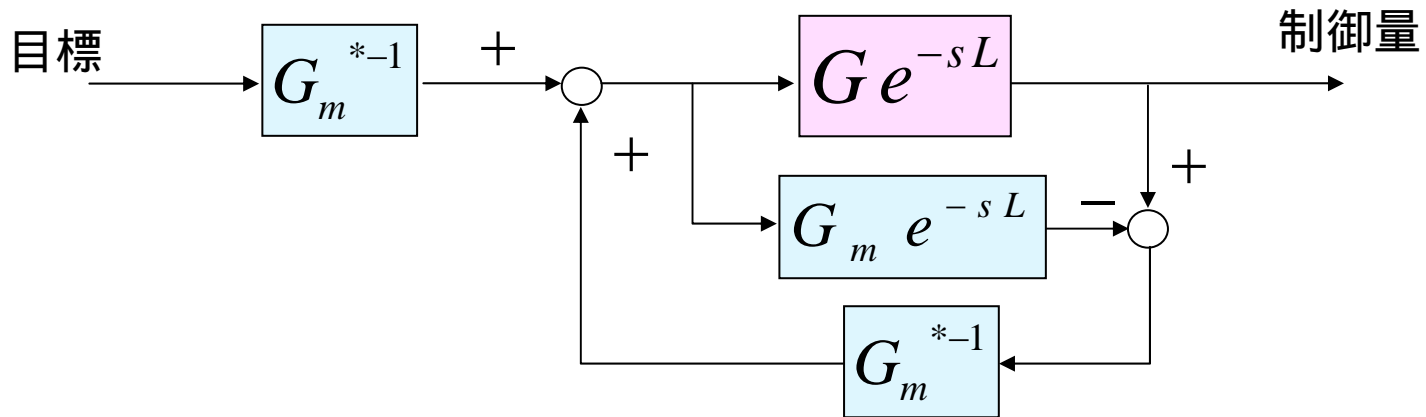


スミスの時間遅れ補償の等価変換 (2)

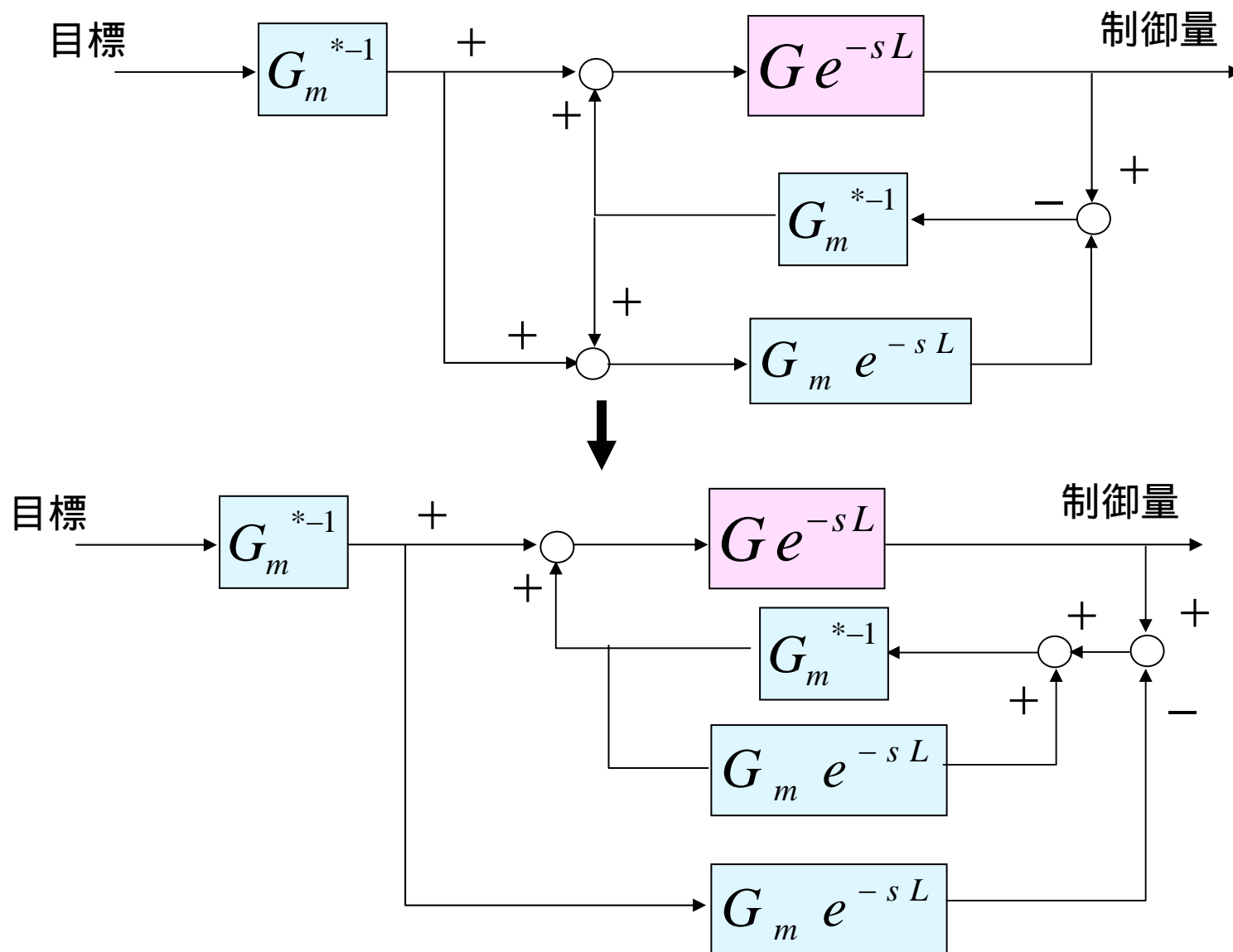


外乱推定オブザーバに相当！

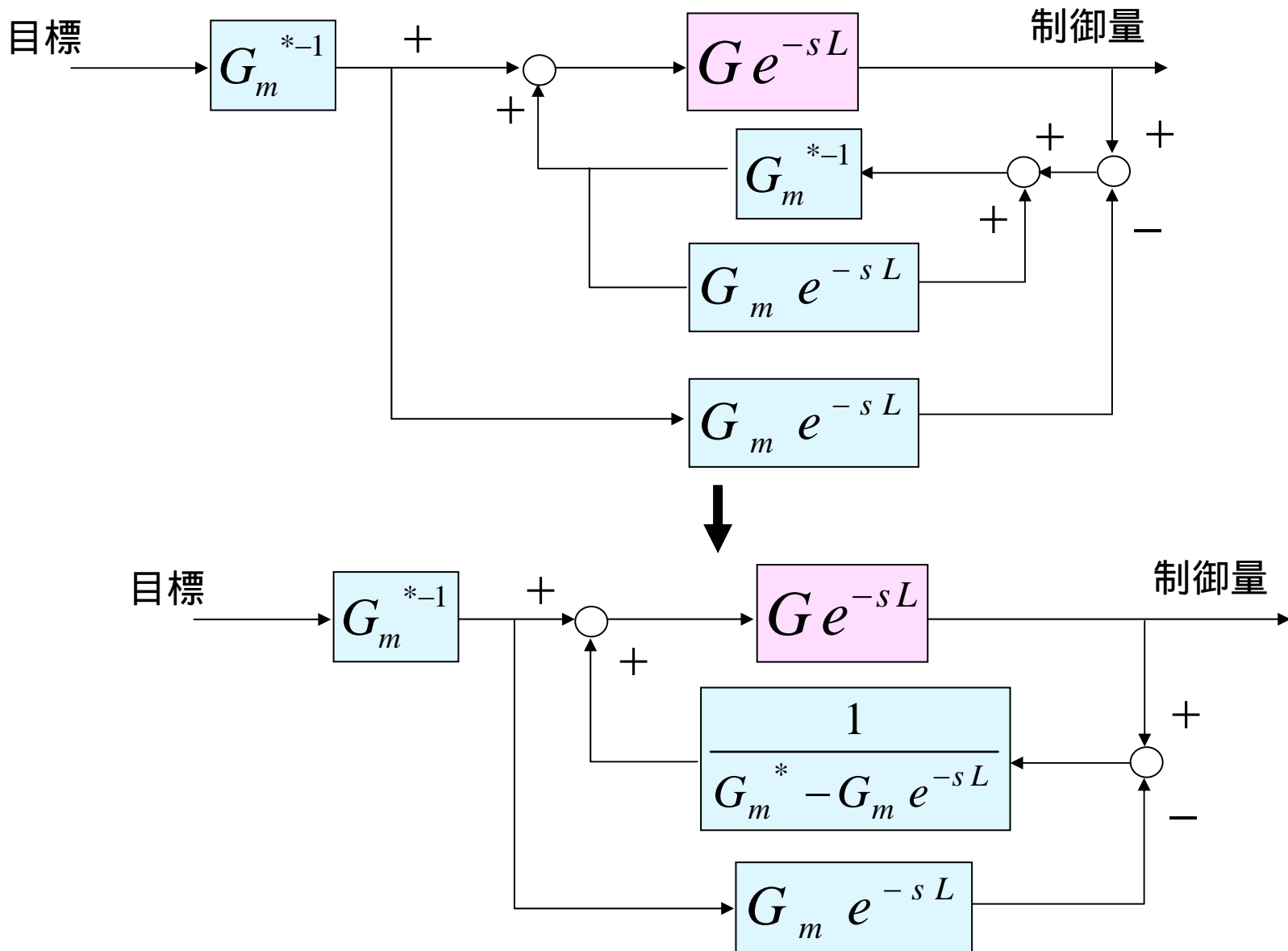
スミスの時間遅れ補償の等価変換 (3)



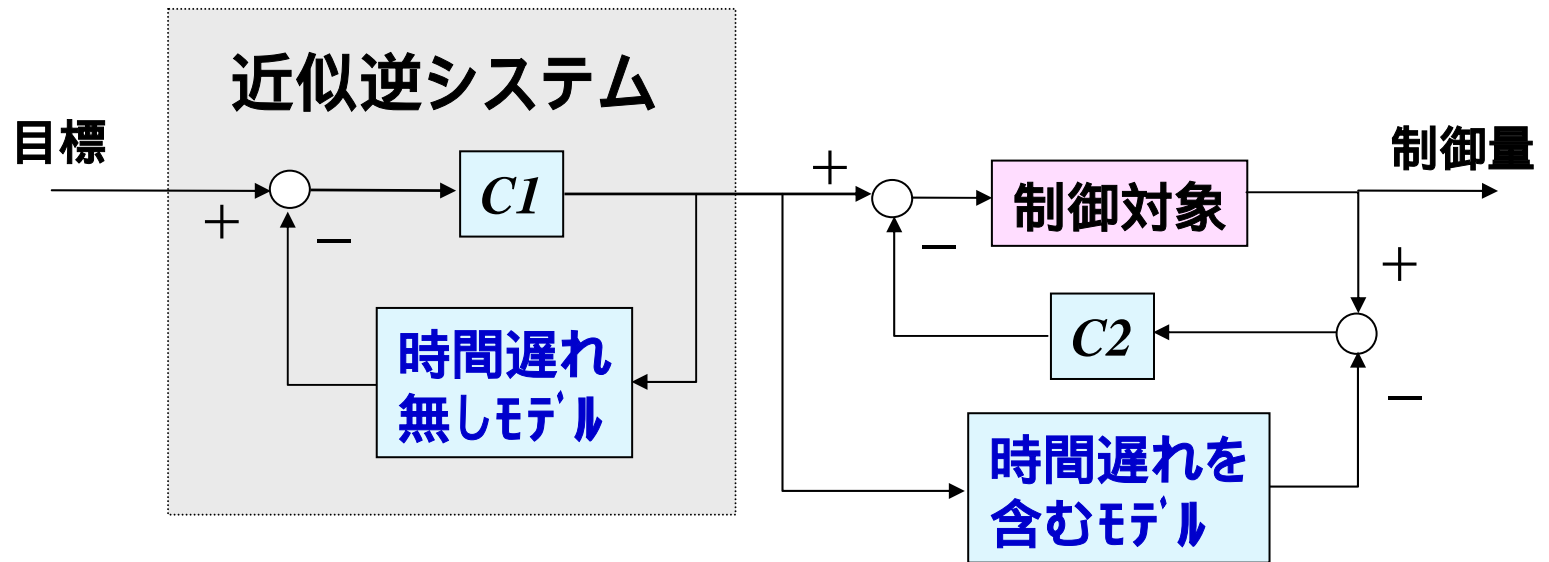
スミスの時間遅れ補償の等価変換 (4)



スミスの時間遅れ補償の等価変換 (5)



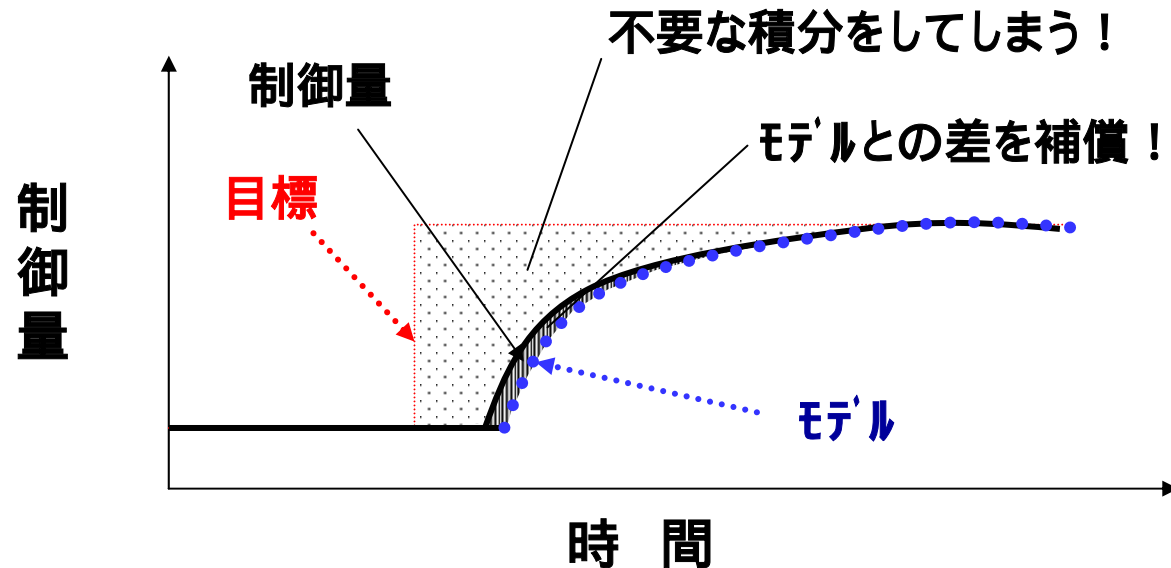
標準目標追従制御系



非線形性、操作量制約、時間遅れがあってもよく、テンプレートにシミュレーションモデルをはめ込み、 $C1$ と $C2$ を調整する

制御系の評価

- (1) 近似逆システム：時間遅れ無しモデルの目標追従制御
(時間遅れ無し、外乱無しの場合は易しい)
- (2) 時間遅れ補償：制御対象とモデルの偏差補償



組込み制御としての評価基準

自動車の制御装置(ECU): { ROMサイズ 50MB程度
CPU性能 50MHz

評価基準

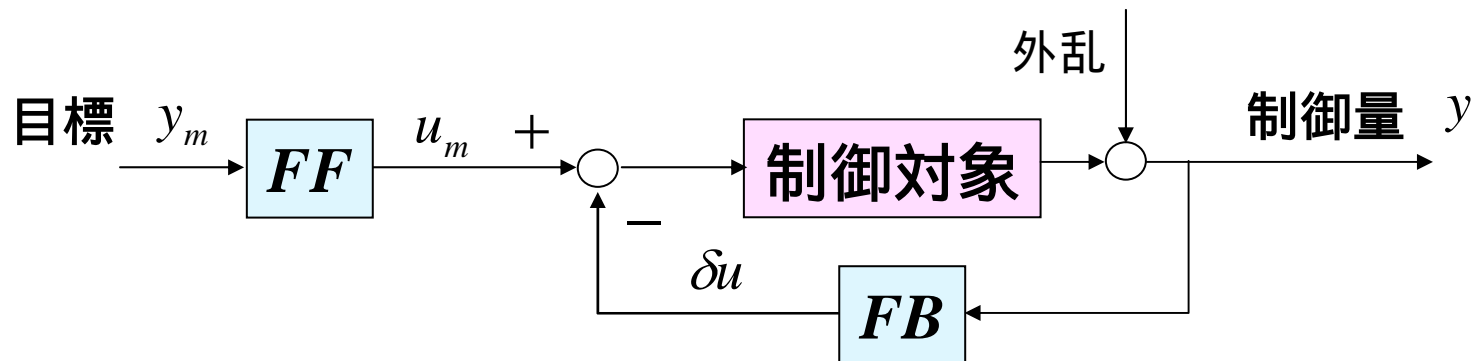
1. ROM/RAMメモリーサイズ
2. 実行時間
3. 性能(制御精度、ロバスト性)
4. 制御定数決定に要す時間(実験時間)
5. 展開性(各種の車両に応じた調整に要す時間)
6. 調整の難易度(自動化可能性含む)

制御の簡素化

- 標準目標追従系は、モデルの複雑度に応じて複雑になる
- 制御装置のリソースに応じた簡素化が不可欠

可能性のある簡素化

- 低次元化(定常値、無くすことを含む)
- 線形化(非線系のままの方が簡素な場合もある)
- $FF + FB$ としての整理



第1回講義の構成

1. 近代科学の方法論
2. 制御システム開発の構造
3. 制御設計のレビュー
4. 制御設計における問題の明確化
5. 自動車制御における制御設計
6. **第1回講義のまとめ**

第1回講義のまとめ

1. 制御システム開発プロセスの重要性

- 複雑性に対処するか？

2. 制御設計プロセスの見直し

- 制御対象モデルをどう作るか？

(時間遅れ、操作量範囲は必須の要素)

- 実際の制御設計はもっと複雑！

3. 自動車制御では目標追従制御が頻出

- 2自由度制御系