システム工学 I
第 3 回
制御対象のモデリング

どこにでもある微分方程式モデル

今回の講義では, まず, 現象の微分方程式によるモデリングが, 工学だけでなく, 色々な分野で有用であることを見る. モデルおよびモデリングという言葉の詳しい説明は後回しにする.

以下の事例の出典は

- 野原, 応用微分方程式講義, 東京大学出版会, 2013
- D. N. Burges and M. S. Borrie (垣田, 大町訳), 微分方程式で数学モデルを作ろう, 日本評論社, 1990

人口モデル (1)

- 時刻 t におけるある国の人口を $N(t)$ とする.
- Malthus のモデル (1798) は, 出生率, 死亡率ともに人口に比例するというモデル： 微小時間 δt における出生数を $\alpha N(t)\delta t$, 死亡数を $\beta N(t)\delta t$ となる.
- $\delta N(t) = \alpha N(t)\delta t - \beta N(t)\delta t.$

人口モデル (2)

- 先の式を $\frac{\delta N(t)}{\delta t} = (\alpha - \beta)N(t)$ と書き直し, $\delta t \to 0$ となる極限を取ると, 微分方程式 $\frac{dN}{dt} = (\alpha - \beta)N$ が得られる. これを Malthus のモデルという.
- 以下では $\gamma = \alpha - \beta$ とする.
- Malthus モデルの解は指数関数,

人口モデル (3)

- Malthus モデルは単純ではあるが, その解は必ずしも実際の人口統計と適合しない.
- $\gamma > 0$ のとき, Malthus モデルは人口が無限大に発散することを許容するが, これは不合理である.

人口モデル (4)

- 人口には許容可能な上限 N_∞ が存在すると仮定し, $\dfrac{dN}{dt} = \gamma N \left(1 - \dfrac{N}{N_\infty}\right)$ とする修正モデル (Verhulst, 1838; **logistic モデル**) が提案されている.

  `http://scienceworld.wolfram.com/biography/Verhulst.html`

- logistic モデルは色々な場所で使われる.

人口モデル (5)

- 人口モデルは, 人口の推移を説明あるいは予想するために, 人口の増減という現象を単純化したものであり, 人口統計に全面的に一致するわけではない (logictic モデルは Malathus モデルより人口統計によく合う).
- さらに複雑なモデルも色々提案されている.

http://www.ms.u-tokyo.ac.jp/q̃inaba/inaba2002_jiten1.pdf

指数関数モデル (1)

- 微分方程式 $\dfrac{dx}{dt} = \alpha x$ の解は指数関数である.
  この形の微分方程式は至る所にあらわれる.
  $\alpha \geq 0$ なら解は時間とともに発散し, $\alpha < 0$ なら解は時間とともに零に収束する.

指数関数モデル (2)

- 電気関連では, コンデンサやコイルにおける電圧と電流の関係がこの形の微分方程式で与えられる.
- 他の例は, 先の Malthus モデル, 薬剤の血中濃度, 放射性物質の崩壊, 水の加熱 (あるいは冷却), 広告に対する売り上げの反応など.

Lotka-Volterra モデル (1)

- Lotka-Volterra モデルは, 生態系に被補食者 X と被捕食者 Y (それらの個体数を $x(t)$ および $y(t)$ とする) が存在するとき, その個体数の時間的な変動を説明するために考えられたモデル (1931). このモデルの考え方は次の通り (現象の単純化 (近似) であることに注意).

Lotka-Volterra モデル (2)

- 被補食者は放置すれば増加し, その増加率は $x(t)$ に比例する (Malthus モデル).
- 被捕食者は補食によって減少し, その減少率は $x(t)$ と $y(t)$ の積に比例する.
- 捕食者は放置すれば減少し, その減少率は $y(t)$ に比例する (Malthus モデル).
- 被捕食者は補食によって増加し, その増加率は $x(t)$ と $y(t)$ の積に比例する.

Lotka-Volterra モデル (3)

- 以上を式でまとめると次のようになる.

$$\frac{dx}{dt} = ax - bxy$$
$$\frac{dy}{dt} = -cy + dxy$$

これを Lotka-Volterra の捕食者-被補食者モデルという (ただし $a, b, c, d > 0$).

Lotka-Volterra モデル (4)

- Lotka-Volterra モデルの解は周期解で, これにより魚の漁獲量の周期的な変動などといった現象をある程度説明できるが, その適合度は必ずしも高くない.
- より実際の生態系に合うように改良された**一般化 Lotka-Volterra モデル**と呼ばれるモデルもある (Rosenzweig, MacArthur)

Lotka-Volterra モデル (5)

- 生物 X と生物 Y が捕食者と被捕食者ではなく競合する 2 種であるとき (たとえば在来種と外来種の競合), その個体数の変動の説明に, 似たようなモデルが使われる. これを Lotka-Volterra の **2 種間競合モデル** という.

Lotka-Volterra モデル (6)

- Lotka-Volterra の 2 種間競合モデルでは, X, Y とも個体数が自然に増加するものとし, その変動には Malthus モデルではなく logistic モデルが使われる. (x の上限を X_∞, y の上限を Y_∞ とする).
- X と Y の積は, X および Y 双方の個体数を減らす方向に働く (足のひっぱりあい).

Lotka-Volterra モデル (7)

- これらをまとめると, 次のようになる ($a, b, c, d > 0$; 係数に付けられた符号に注意).

$$\frac{dx}{dt} = ax\left(1 - \frac{x}{X_\infty}\right) - bxy$$
$$\frac{dy}{dt} = cy\left(1 - \frac{y}{Y_\infty}\right) - dxy$$

南雲の神経回路モデル (1)

- 生物の神経系 (脳を含む) は電気回路, この認識がニューロコンピューティングの基礎のひとつ
- 神経細胞などの活動電位を表現した数理モデルに, 南雲モデルと呼ばれるものがある (1962). これは次のような微分方程式モデル.

南雲の神経回路モデル (2)

- $u(t)$ を膜電位, $w(t)$ を細胞の不活性化に関する変数, I を外部刺激電流, a, b, μ を定数として,

$$\mu \frac{du}{dt} = w - \frac{u^3}{3} + u + I$$

$$\frac{dw}{dt} = a - u - bw$$

モデルとは (1)

- ある事象・現象に特徴的な物理量間の関係などを単純化して, 論理系に定式化したもの (旺文社 物理事典)
- 問題とする事象 (対象や諸関係) を模倣し, 類比・単純化したもの. また事象の構造を抽象して論理的に形式化したもの (大辞林 第3版)

モデルとは (2)

- (IT 用語) 現実世界のある対象を, 目的に応じた側面から捉えて抽象化・単純化し, 対象を理解したり操作したりしやすく形式化したもの

  http://itpro.nikkeibp.co.jp/article/lecture/20061204/255757/

この講義におけるモデルは上記 3 種類をすべて含む.

モデリングとは (1)

- 設計案に基づく模型の製作 (大辞林 第 3 版)
- 塑像に肉づけをしたり絵画に陰影をつけたりして立体感を出すこと (大辞林第 3 版)
- (IT 用語) モデルを作成する工程

http://itpro.nikkeibp.co.jp/article/lecture/20061204/255757/

モデリングとは (2)

- この講義におけるモデリングの意味はIT 用語と同じ.
- IT エンジニアは業務のモデリングに UML のようなツールを用いる. UML はフローチャートなどの図によって一連の業務を統一的に表現するツールだが …

http://objectclub.jp/technicaldoc/uml/umlintro

モデリングとは (3)

- 制御工学におけるモデルは物理現象をある程度正確に記述する必要があるため, UML では力不足. 制御工学が対象とするのは, 微分方程式によるモデル (あるいはそれを離散化したもの).
- モデリングのツールとしては, MATLAB やその GUI である simulink が標準的.

モデリングとは (4)

以下しばらく次の文献にもとづいて説明する.

- 大畠 (古田監修), モデルベース開発のために複合物理領域モデリング, TechShare, 2012
- 石川ほか (スキルマネジメント協会監修), モデルベース開発とエンジニア育成の最前線, TechShare, 2012
- 久保, 自動車業界 MBD エンジニアのための Simulink 入門, TechShare, 2012
- 片山, システム同定, 朝倉書店, 2004
- 相良ほか, システム同定, 第 3 版, コロナ社, 1987

何のためにモデリングをするのか (1)

- モデリングをおこなうと, 制御システムの挙動をコンピュータでシミュレ- ションできる
- 現実の制御対象に現実の制御装置を接続して試行錯誤により制御系設計をおこなうことには, 色々なデメリットがある.

実機による試行錯誤のデメリットは …

- 危険: 制御対象や制御用ハードウェアが破損する可能性, 重大事故にも
- 高コスト: 実験に費用と人手が必要
- 所要時間が長い : 物理現象と同じ時間軸でしか実験できない
- あまり多様な条件で実験できない

何のためにモデリングをするのか (3)

- モデルベースド開発 (Model Based Developement; MDB) では, 制御対象と制御装置双方のモデルを作成し, 実機を用いた実験に先立ってモデルを用いたコンピュータシミュレーションを繰り返す手法. 工程数が多いという問題はあるが, 次に述べるメリットがある.

モデルベースド開発のメリットは …

- 安全: テスト中の制御システムに重大な問題があっても被害は発生しない
- 低コスト: コンピュータだけで「実験」可能
- 所要時間が短い： 物理現象より速いタイムスケールで「実験」できる
- 多様な条件で「実験」できる

何のためにモデリングをするのか (5)

- とはいっても, モデルは実機そのものではないので, モデルではなく実機が要求仕様を満たすか否かを検証するために, 最後の段階で実運用環境でのテストが必要になる.

モデルベースド開発のプロセス (1)

モデルベースド開発のプロセスは以下の通り.

1. 制御対象と制御装置のモデリング
2. 制御系設計と シミュレ- ション
3. Rapid Control Prototyping (RCP): 後述
4. Automatic Code Generation (ACG): 後述
5. Hardware In the Loop Simulation (HILS): 後述
6. 実機を用いたテスト

モデルベースド開発のプロセス (2)

		制御装置	
		モデル	実機
制御対象	モデル	Simulation	HILS
	実機	RCP	実機実験

- 制御対象, 制御装置ともにモデルを使うのが制御理論の守備範囲 (表に Simulation と記載された部分)

モデルベースド開発のプロセス (3)

- これが終わったら, 次に制御対象は実機, 制御装置はモデルの組み合わせでテストおよび改良をおこなう. これを Rapid Control Prototyping (RCP) という. シミュレー ションで用いた制御装置を速やかにテストでき, 制御対象のモデルと実機のずれの影響を確認して修正できるのがメリット.

モデルベースド開発のプロセス (4)

- RCP に続いて, 制御用ハードウェア用のプログラムを自動生成する. これを Automatic Code Generation(ACG) という. コード生成に続いて, 制御器のモデルと実機の動作の相異の検証が必要になる (これを Back-to-Back テストと呼ぶ).

モデルベースド開発のプロセス (5)

- ACG に続いて，制御対象はモデル，制御用ハードウェアは実機の組み合わせでテストおよび改良をおこなう．これを Hardware In the Loop Simulation (HILS) という．HILS の利点は, 制御対象の実機が不要で, 網羅的な実験ができ, 再現性があり, 安全で, テストが自動化でき, 実験条件を再利用可能なことである.

モデルベースド開発のプロセス (5)

- 最後に実機を用いたテストおよび微調整をおこなう.
- 以上, 多様な工程があったが, これらを一人のエンジニアがおこなうわけではない. 色々な専門性を持つエンジニアが共同作業することが普通である.

制御対象のモデリング (1)

- 以下では制御対象のモデリングについて考える (制御器のモデリングには制御系設計を含めるが, 制御系設計はこの講義の守備範囲ではない).

制御対象のモデリングの工程

1. モデルを使用する目的を明確にする
2. システムの構成要素を定め, 構成要素間の相互作用を調べる
3. 構成要素のモデルを導く
4. モデルのパラメータを実験データから推定する
5. モデルを検証する
6. 問題があれば上記の事項を再検討する

制御対象のモデリング (3)

- 構成要素のモデルを導く手法には, 物理モデリング (物理法則に基いてモデルを構築する), black-box モデリング (先験的にモデルの構造を決め打ちし, そのパラメータのみを実験によって決める), grey-box モデリング (black-box モデリングにおいて部分的に物理法則を利用する) の 3 種類がある.

制御対象のモデリング (4)

- black-box モデリングおよび grey-box モデリングは統計的な信号処理との関連が深く, 狭義の制御理論とは別の分野として取り扱われることが多いため, この講義では対象とせず, 以下では物理モデリングについて述べる.

制御対象のモデリング (5)

- 制御システムは, 多くの場合, 機械的な構成要素, 電気的な構成要素, 化学的な構成要素, 情報処理に関する構成要素などが複雑に組み合わされたシステムであり, それらの (物理的な) 関係は複雑であるが, 各要素を数式で表現してしまえば, システム全体を統一的な見地で取り扱うことができる.

制御対象のモデリング (6)

- どんなに複雑なシステムでも, 各要素を微分方程式でモデリングすることができれば, 連立微分方程式系として統一的に表現できる (連続時間要素と離散時間要素が混在するシステムでは連立微差分分方程式系になる). このようにすると, 個々の構成要素の (物理的な) 特性をそれほど気にする必要がなくなる.

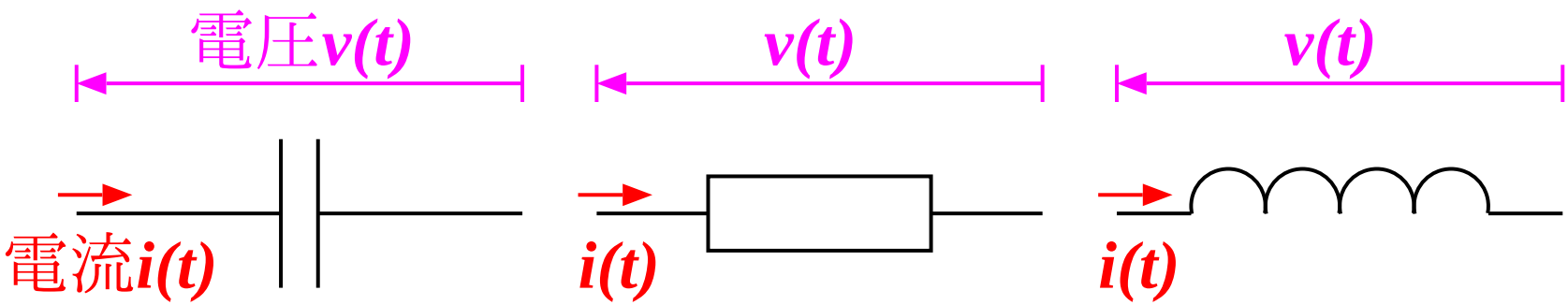
制御対象のモデリング (7)

- この「数式を通じて工学的な対象を統一的な見地から眺める」という見通しの良さが制御工学の特徴であり，「現実のハードウェア」への拘りを持つ工学の他分野 (の多く) と異なる点である.

制御対象のモデリング (8)

- 数式の世界で研究がおこなわれていることの副作用として, 制御理論の現実の問題への適用において問題が発生することがあり, 美しい理論と醜い現実と揶揄される. この問題は, H_∞ 制御理論の完成により緩和された.

電気系

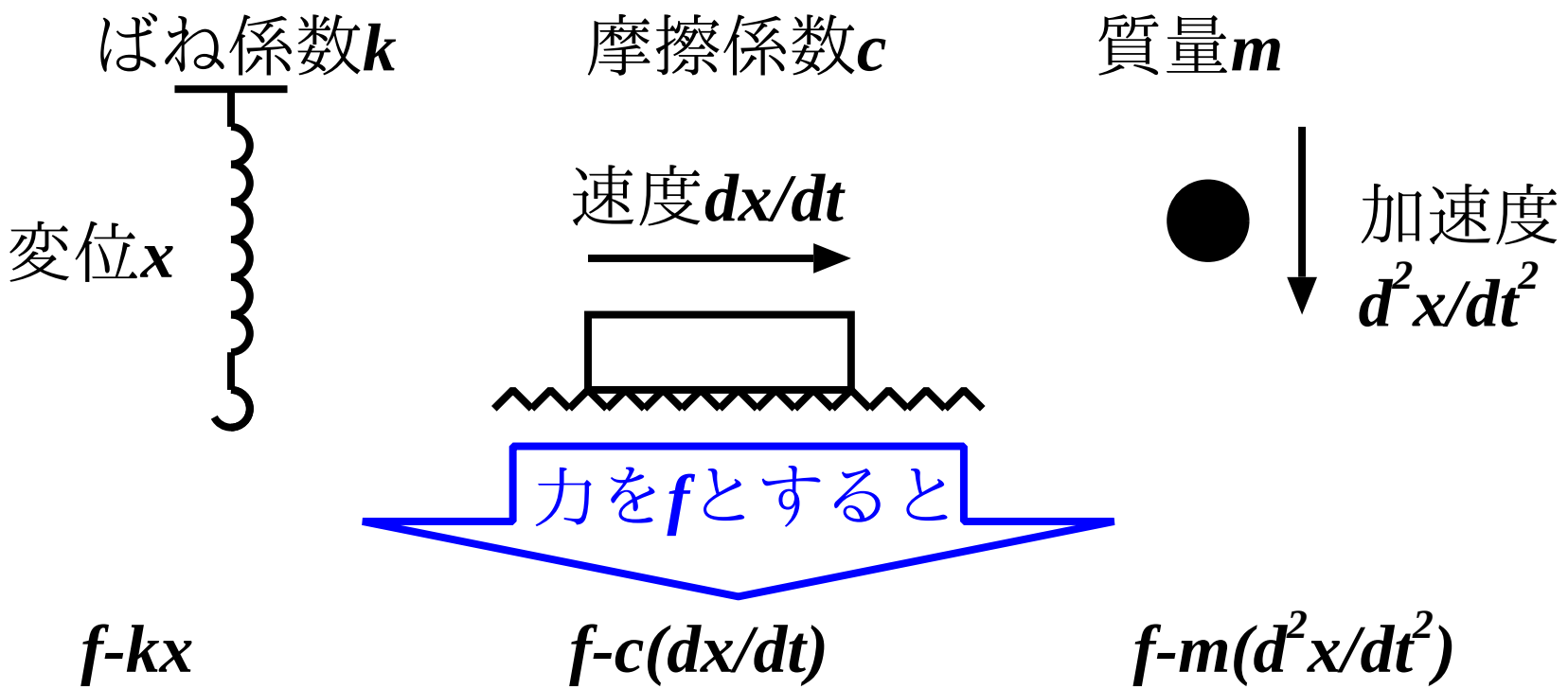


$i=C(dv/dt)$ $\quad v-Ri$ $\quad v=L(di/dt)$

$q(t)=\int i(\tau)d\tau$とすると

$v=(1/C)q$ $\quad v-R(dq/dt)$ $\quad v-L(d^2q/dt^2)$

機械系 (並進)


ばね係数k
摩擦係数c
質量m
変位x
速度dx/dt
加速度
d^2x/dt^2
力をfとすると
$f-kx$
$f-c(dx/dt)$
$f-m(d^2x/dt^2)$

電気系と機械系の対応 (1)

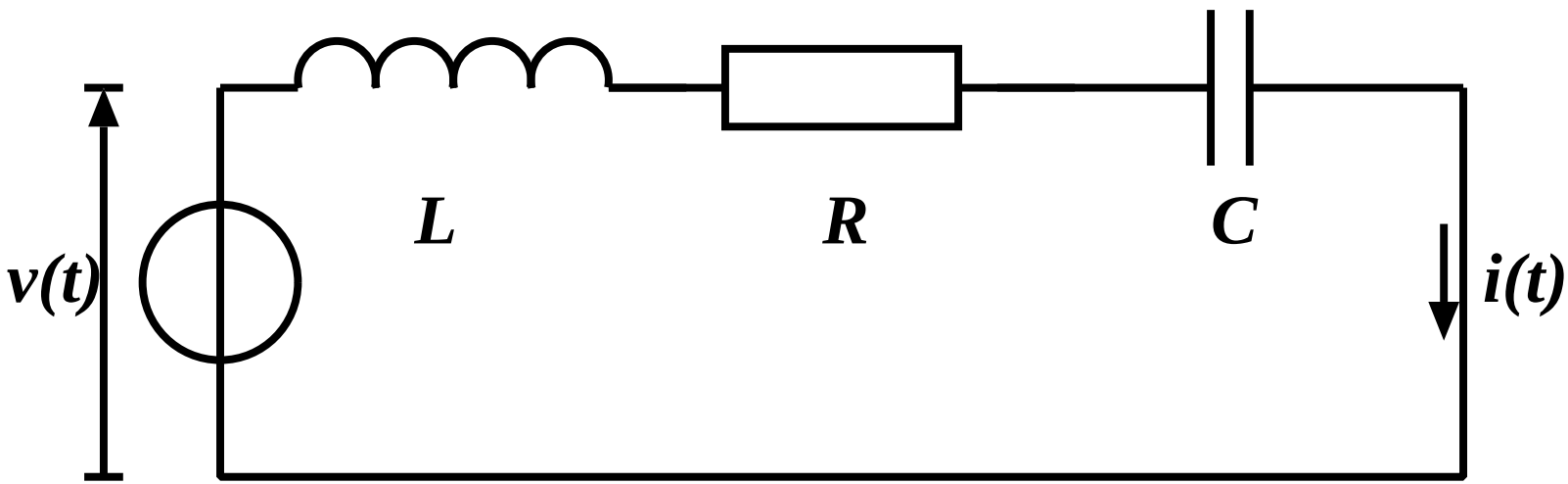
- 電気系では, 電圧を $v(t)$, 電流の積分値 (電荷) を $q(t) = \int_0^t i(\tau)d\tau$ とする.
- 機械系 (並進) では位置を $x(t)$, 力を $f(t)$ とする.

電気系と機械系の対応 (2)

	コンデンサ	抵抗	コイル
電気系	$v = \frac{1}{C}q$	$v = R\frac{dq}{dt}$	$v = L\frac{d^2q}{dt^2}$
機械系	$f = kx$	$f = c\frac{dx}{dt}$	$f = m\frac{d^2x}{dt^2}$
	ばね	摩擦	質量

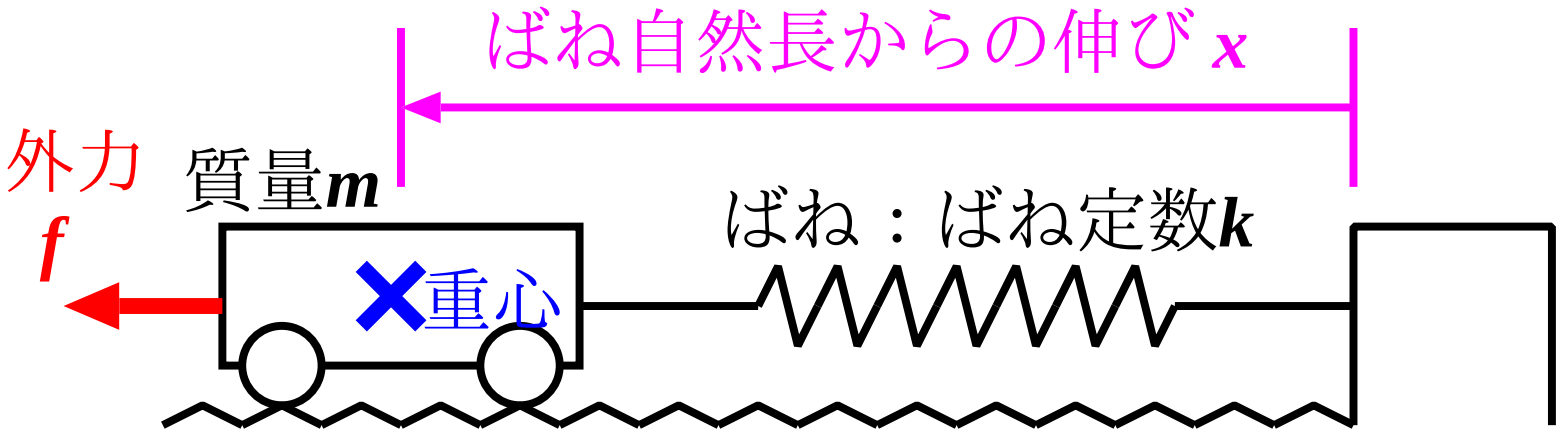
電気系と機械系の対応 (3)

- 機械系 (並進) と電気系は同じ形の微分方程式でモデリングできる
- 電気系の RLC 系回路に機械系のばね-質量-ダンパ (摩擦) 系が対応する
- よって, 上述の電気系と機械系 (並進) の振動などに関する特性はまったく同じ



$$v(t) = L\frac{di}{dt} + Ri(t) + \frac{1}{C}\int_0^t i(\tau)d\tau$$

$$= L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q \quad (\text{q:電荷})$$



$$f(t) = m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx$$

電気系と機械系の対応 (6)

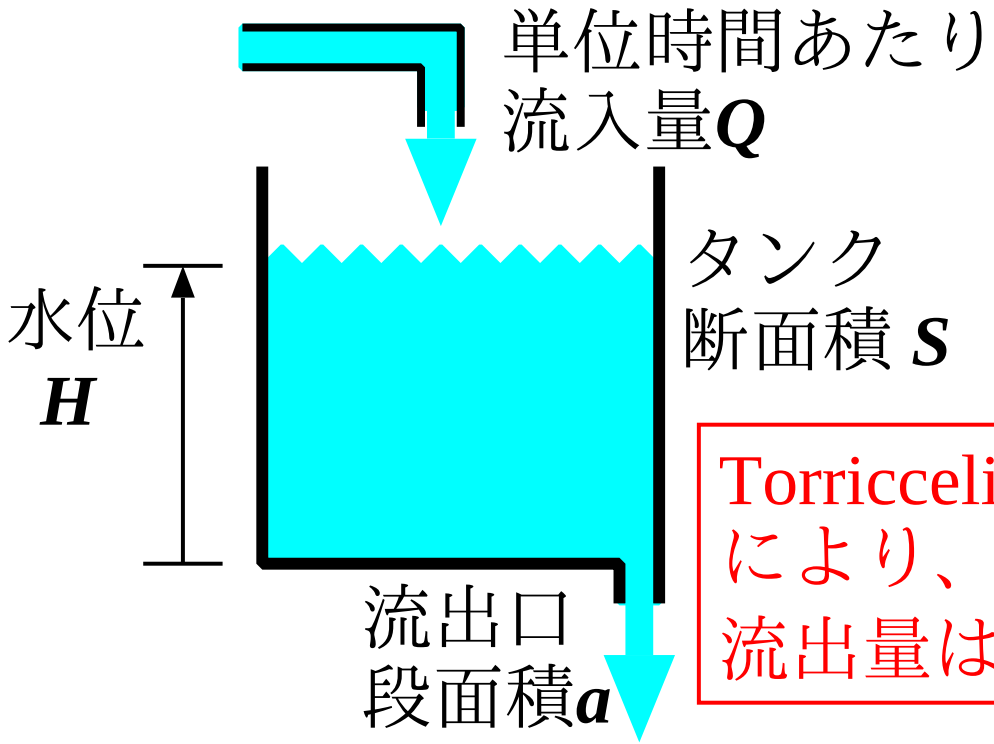
機械系 (回転) において角度を θ, 慣性モーメントを I として機械系 (並進) と比較すると …

	ばね	摩擦	質量
並進	$f = kx$	$f = c\dfrac{dx}{dt}$	$f = m\dfrac{d^2x}{dt^2}$
回転	$f = k\theta$	$f = c\dfrac{d\theta}{dt}$	$f = I\dfrac{d^2\theta}{dt^2}$
	ねじりばね	回転摩擦	慣性モーメント

電気系と機械系の対応 (7)

- 機械系における並進系と回転系も, 同じ形の微分方程式でモデリングできる
- ロボットの制御などでは, 電気系と機械系 (並進), 機械系 (回転) が組み合わされているが, 微分方程式としては, これらは同様の構造を持つサブシステムが結合されたものとして表現できる.

タンクシステム (1)



Torricceliの定理により、単位時間あたり流出量は $a(2gH)^{1/2}$

タンクシステム (2)

微分方程式モデルを作ると,

$$S\frac{dH}{dt} = Q - a\sqrt{2gH}$$

線形代数の復習 (3-1)

階段行列とは, 次のような形の行列をいう.

- 行を右に動いてゆくと, 以下のパターンのどれかになる:

$$
\begin{array}{lccccccc}
\text{i)} & 1 & * & & \cdots & & & * \\
\text{ii)} & 0 & \cdots & 0 & 1 & * & \cdots & * \\
\text{iii)} & 0 & & \cdots & & 0 & & 1 \\
\text{iv)} & 0 & & & \cdots & & & 0
\end{array}
$$

- 1 の左はすべて零, 1 の右側は任意, 1 を含まない行は全要素が零, 1 の系列は右斜め下に進む

線形代数の復習 (3-2)

階段行列の例
空白は零をあらわす
$*$ の部分は何でもよい

$$\begin{pmatrix} 1 & * & * & * \\ & & 1 & * \\ & & & \end{pmatrix}$$

- 行列は (正方であるか否かにかかわらず) 行基本変形によって階段行列に変形できる
- 行列 A を行基本変形によって階段行列に変形したとき, 零でない行を, 行列 A の階数 (ランク) という.

線形代数の復習 (3-3)

帰納法により, なぜ階段行列に変形できるかを見る.

- 行列の行が 1 個のときには行の最左端の非零要素で行全体を割り, 全要素が零のときにはそのままにすると, 主張の形が得られる.
- 行列の行が $m-1$ までのときには主張が正しいものと仮定し, A を m 行 n 列の行列とする.

線形代数の復習 (3-4)

A の列を左から順に調べ, 第 j 列に初めて零でない要素が含まれていたものとし ($j = 1$ なら左端の零のみのブロックは無い), j 列の零でない要素をひとつ選ぶ (a_{ij} とする); 他の数値 ($*$) は任意

$$
\begin{array}{c}
\\
i)
\end{array}
\begin{array}{c}
\overbrace{\phantom{0\ \cdots\ 0\ a_{ij}\ *\ \cdots\ *}}^{j} \\
\begin{pmatrix}
0 & \cdots & 0 & * & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
0 & \cdots & 0 & a_{ij} & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
0 & \cdots & 0 & * & * & \cdots & *
\end{pmatrix}
\end{array}
$$

線形代数の復習 (3-5)

行基本変形により第 i と第 1 行を入れ換える

$$
\begin{pmatrix}
 & & & \overbrace{\phantom{a_{ij}}}^{j} & & & \\
0 & \cdots & 0 & a_{ij} & * & \cdots & * \\
0 & \cdots & 0 & * & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
0 & \cdots & 0 & * & * & \cdots & *
\end{pmatrix}
$$

線形代数の復習 (3-6)

第 1 行を $a_{ij}(\neq 0)$ で割る

$$
\begin{pmatrix}
 & & & \overset{j}{\smile} & & & \\
0 & \cdots & 0 & 1 & * & \cdots & * \\
0 & \cdots & 0 & * & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
\vdots & & \vdots & * & * & \cdots & * \\
0 & \cdots & 0 & * & * & \cdots & *
\end{pmatrix}
$$

線形代数の復習 (3-7)

第 2 行以降に第 1 行の定数倍 ($*$ の数値に -1 をかけたもの) を加える (零のときは何もしない)

$$
\begin{pmatrix}
 & & & \overbrace{}^{j} & & & \\
0 & \cdots & 0 & 1 & * & \cdots & * \\
0 & \cdots & 0 & 0 & * & \cdots & * \\
\vdots & & \vdots & \vdots & * & \cdots & * \\
\vdots & & \vdots & \vdots & * & \cdots & * \\
0 & \cdots & 0 & 0 & * & \cdots & *
\end{pmatrix}
$$

線形代数の復習 (3-8)

第 1 行を除去すれば帰納法の仮定が使える.

$$\begin{pmatrix} 0 & \cdots & 0 & \overbrace{1}^{j} & * & \cdots & * \end{pmatrix} \qquad \text{第 1 行 (除去)}$$

$$\begin{pmatrix} 0 & \cdots & 0 & \overbrace{0}^{j} & * & \cdots & * \\ \vdots & & \vdots & \vdots & * & \cdots & * \\ \vdots & & \vdots & \vdots & * & \cdots & * \\ 0 & \cdots & 0 & 0 & * & \cdots & * \end{pmatrix} \qquad \text{残りの行}$$

線形代数の復習 (3-9)

- 行列 A を列基本変形によって階段行列の転置に変形したとき, 零でない列の数は, 行列 A の階数に一致することが示せる.