

数理工学への招待

2024年1月17日

阿瀬 始

0 . 目次と自己紹介

1 本講演の動機

2 数理工学とは

3 (応用) 数学

4 制御

5 最適化

6 まとめ

自己
紹介

1968年3月 大阪府立北野高等学校卒業 (80期)

在学中は卓球部に所属

1973年3月 京都大学工学部数理工学科卒業

1975年3月 京都大学大学院工学研究科数理工学専攻修士課程修了

1975年4月 日本鋼管入社 「制御と最適化」に関する研究開発に従事

2015年3月 JFEグループ退職

～現在

最適化に関するコンサルタント

資格

京都大学工学博士

趣味

(鉄道) 旅行 日課として「詰め将棋」と「数独」を解く

1. 本講演の動機

ドイツにおける例

右の本：シュプリンガー(2009)

オーバーヴォルフアッハ数学研究所
がまとめた本

アネッテ・シャヴァン
ドイツ連邦教育研究省大臣

- ・ 数学は科学の基礎。
- ・ 数学は革命的イノベーションのための鍵を届けてくれる。
- ・ ハイテクは数学なしにはありえない。

G.-M.グロイエル/R.レンメルト
G.ルツプレヒト…編
戸瀬信之/丸山文綱…訳



..21..

数学が経済を動かす

ドイツ企業篇

数学が経済を動かす

Mathematik
Motor der Wirtschaft

ドイツ企業篇

シュプリンガー・ジャパン

グロイエル/レンメルト
ルツプレヒト…編
戸瀬信之/丸山文綱…訳



1 . 本講演の動機 (2)

前スライドの本の分析： 掲載された企業と使われている数学

| 分野 | 企業 | 数値解析 | 確率統計 | 最適化 | 制御 | その他 |
|----------|----------------|------|------|-----|----|-----|
| 保険 | アリアンツ・ドイツ | | ○ | ○ | | |
| 製薬 | ドイツ・バイエル | ○ | ○ | ○ | | ○ |
| 製薬 | ベーリンガーインゲルハイム | | ○ | ○ | | |
| 政府 | ドイツ連邦労働エージェンシー | | ○ | ○ | ○ | |
| 自動車 | ダイムラー | ○ | | | ○ | |
| 鉄道 | ドイツ鉄道 | ○ | | ○ | ○ | |
| 金融 | ドイツ銀行 | | ○ | | | |
| 金融 | ドイツ証券取引所 | | ○ | | | |
| 製造 | デュル | ○ | | ○ | ○ | |
| 保険 | HUK- COBURUG | | ○ | | | |
| コンピュータ | IBM | | | | | ○ |
| 半導体 | インフィニオンテクノロジーズ | | | | | ○ |
| エンジニアリング | リンデ | ○ | | ○ | | |
| 航空 | ルフトハンザ航空 | ○ | | ○ | | |
| 保険 | ミュンヘン再保険 | | ○ | | | |
| エネルギー | RWE | | ○ | ○ | | |
| ソフトウェア | SAP | | | ○ | | ○ |
| エネルギー | シエル | ○ | | | ○ | ○ |
| 電気 | シーメンス | ○ | ○ | ○ | | |
| 旅行 | TUI | | ○ | ○ | | |

1 . 本講演の動機 (3) この本を読んだ感想

日本でも数学は重要な役割を果たしているはず
ドイツに決してひけをとっていない！

日本の実情を分析して明らかに
した本はない

使われている数学はすべて
数理工学がカバーしている

日本の実情を一人で
分析するのは難しい

数理工学ができることを紹介し
アピールする本や資料を作成する

取りかかったものの、途中でほったらかし

本講演がよい機会と考え、
再度取り組む

パソコンを使った講演可視
化できるわかりやすい例を
多用する

2 . 数理工学とは (1)

数理工学

一般的には Mathematical Engineering

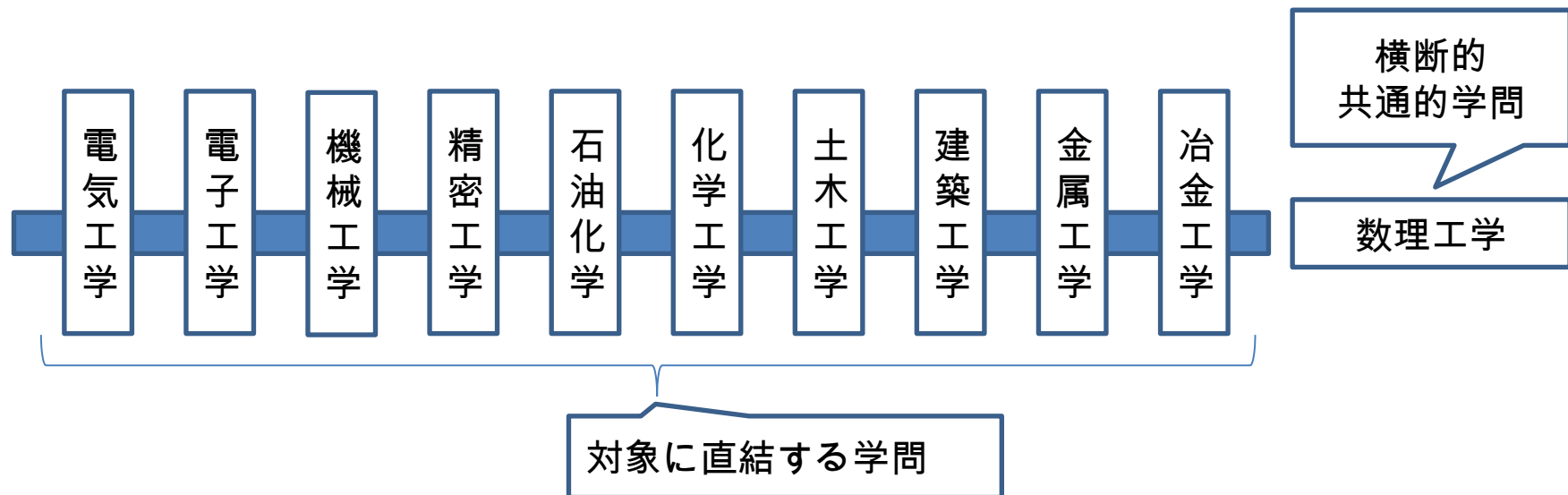
数学を道具に用いて現実の諸問題に取り組む学問

Department of Applied Mathematics and Physics

数理工学科の設立の狙い

1959年 (昭和34年) 4月に京都大学工学部数理工学科創設

工学部に数学や物理学などの刺激を入れて、より新しい工学へ脱皮を図る。



2 . 数理工学とは (2)

- 数理工学の役割 -

数理工学の役割

数学に基づいて現象を説明したり、(数学的な) 問題を数学を応用して解決する。
。

一般的手順

1 現象や問題を数学的に定式化する

変数を考え、満たすべき方程式や不等式を作成する。

2 必要な数学ツールを開発する

数値解析法、解法アルゴリズム、制御手法、など

3 数学ツールを使って数学的解析をする

解析的に解く。

計算機を使って解く。

← 計算機プログラムの開発が必要

4 結果を検証・実証する


シミュレーションによるケーススタディを行い有効性を確かめる。

最適化問題では数学が応用できるかどうか不明な場合も多い。

2 . 数理工学とは (3)

- 具体例 -

ムービーによって動きを
目で捉えることができる



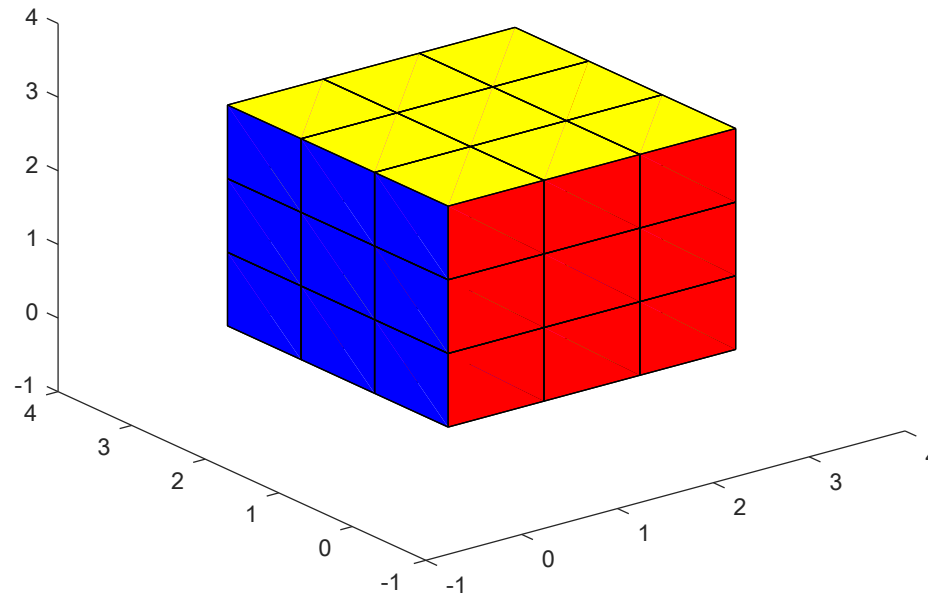
| 対象分野 | わかりやすい例 | その他の実応用例 |
|-----------|--------------------|----------|
| (応用) 数学 | 代数： ルービック・キューブ | |
| | 解析： ソリトン | |
| 制御 | 振り上げ倒立振子 | 鉄鋼など |
| | 建物の制振制御 | |
| 最適化 | 巡回セールスマン問題 (TSP) | 鉄鋼など |
| | 箱詰め問題 | |

3 応用数学

3.1 代数の応用例：ルービックキューブ (1)

パズル

1974年にハンガリーの建築学者Ernö Rubikが発明した。



世界標準色
白、黄、青、赤、緑、橙

MATLABでは橙をマ
ゼンタに変更

パズルとしての遊び方

崩されて各面の色が不揃いの状態から、 x 、 y 、 z 方向の回転操作により、全部の面の色を揃える。

3.1 代数の応用例：ルービックキューブ (2)

数学的なたらえ方

状態 26個の小キューブの面の色

状態の数 $M = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times \underbrace{12! \times 2^{12}}_{\text{端の中小キューブの配置}} \times \underbrace{8! \times 3^8}_{\text{角の小キューブの配置}} = 43,252,003,274,489,856,000$

回転操作の表現

N個の点の座標 $U = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & & \vdots \\ x_N & y_N & z_N \end{bmatrix}$

$$T_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \quad T_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad T_z = \begin{bmatrix} \cos \xi & -\sin \xi & 0 \\ -\sin \xi & \cos \xi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

x 軸を軸とする回転の回転後の座標 U T x

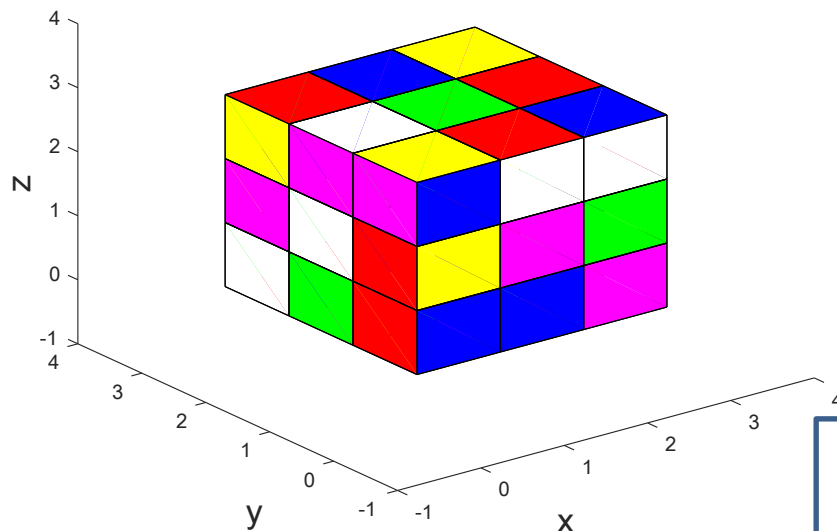
y
z

U T y
U T z

群

回転操作はある状態を別の状態に写す。
回転操作を群演算と見ると、状態集合上で作用する群である。

3.1 代数の応用例：ルービックキューブ（3）



2023年に6面を揃えるまでの時間
3.13秒の世界記録

解き方

面の色が揃った最初の状態からこの状態にするまでの一連の回転操作を逆にたどる。

表示方法

$\pi / 2$ 回転を10等分して表示。

1. 回転する9個の小キューブの中心が原点になるように座標を平行移動する。
2. 行列計算で回転後の座標を計算する。
3. 中心が元の座標になるように平行移動する。
4. グラフィック表示する。

動画

3.2 解析への応用例 1 : K d V 方程式

Korteweg and de Vries方程式

$$\frac{\partial u}{\partial t} + 6u \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0$$
$$u_t + 6uu_x + u_{xxx} = 0$$

3階の偏微分方程式
変数は x と t
未知関数 u は波の高さ

解の挙動に関する興味深い性質

- 1) ソリトンは形を変えずに一定の速度で進む。その速度は振幅に比例する。
- 2) 振幅の大きなソリトンが小さなソリトンの後ろにある場合、時間が進むと大きなソリトンが小さなソリトンを追い抜き、追い抜いた後は元の形、元の速度で進む。

数学的に興味深い性質

- 1) 非線形方程式であるが厳密解がある。
- 2) N 個のソリトンに対しても厳密解がある。
- 3) 無限個の保存量を持つ。

解 u は孤立した波
Solitary Wave

ソリトン

3.2.1 歴史的経緯

1834-1844年 John Scott Russel (英エンジンバラの造船技師)

運河を進む孤立波の発見

1845-1880年 Airy, Boussinesq, Stokesの研究

1895年 Korteweg and de Vries (オランダ)

浅水波の理論。KdV方程式の誕生。

1950年代 Fermi-Pasta-Ulamの非線形バネシステム
に対する数値実験

1965年 Zabusky and Kruskal

Fermi-Pasta-UlamのシステムがKdV方程式で
記述できる。

Miura 保存量解析：初めに保存量を13個求め、最終
的に無限個あることを示した。

Gardner, Greene, Kruskal, and Miura 逆散乱法

1)KdV方程式の初期値問題には厳密解が存在する。

2)N個のソリトンの相互作用を記述する厳密解が存
在する。それらは指数関数を用いて表される。

1967年 戸田盛和 戸田格子

1968年 Lax Laxペア

1970年代 広田良吾 厳密解を求める広田の方法

1981年 佐藤幹夫 佐藤理論 普遍Grassman多様体

Boussinesqは後で再登場

エンジンバラの運河



3.2.2 K d V 方程式の導出

浅水波の理論から

$$\frac{\partial v}{\partial t} + \frac{3c_0}{2h} v \frac{\partial v}{\partial x} + \frac{c_0 h^2}{6} \frac{\partial^3 v}{\partial x^3} = 0$$

$v(x, t)$: (x, t) における平均水面からの水の高さ

h : 平均水面の高さ

$$c_0 = \sqrt{gh}$$

変数変換

$$u(x, t) = \left(\frac{3c_0^2}{32h^5} \right)^{1/3} v \left(\left(\frac{c_0 h^2}{6} \right)^{1/3} x, t \right)$$

を行うと

$$\frac{\partial u}{\partial t} + 6u \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0$$

第二項の係数6はスケール変換 $a u$ により、任意の値に変えることができる。

3.2.3 KdV方程式の顕著な性質 1 (1)

性質 1

ソリトンは一定の速度で形を変えずに進む。速度は振幅の1/3。

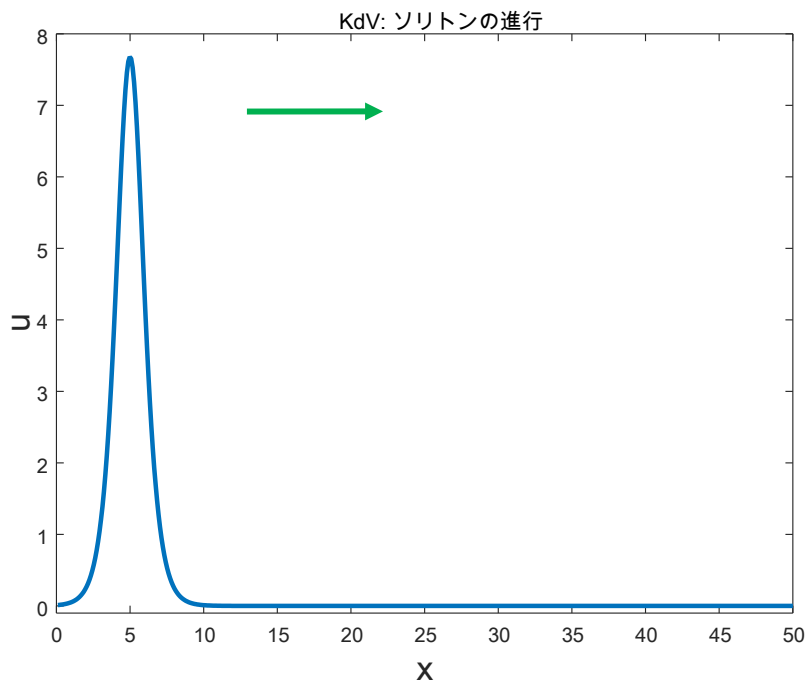
厳密解

$$u(x, t) = A \operatorname{sech}^2 \left(\sqrt{\frac{A}{12}} \left(x - \frac{A}{3} t \right) \right)$$

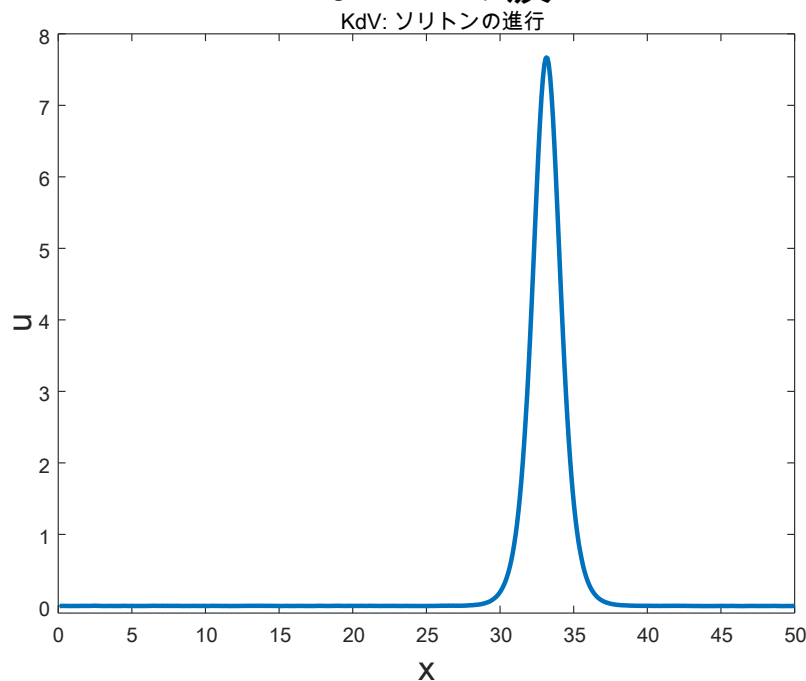
$$\operatorname{sech} x = \frac{1}{\cosh x} = \frac{2}{e^x + e^{-x}}$$

非線形偏微分方程式の厳密解を求めるのは難しい。
過去30年ほどで研究が進んだ。Lie群法など。
KdV方程式はそれより前にすごく研究された。

初期条件



t = 11の波

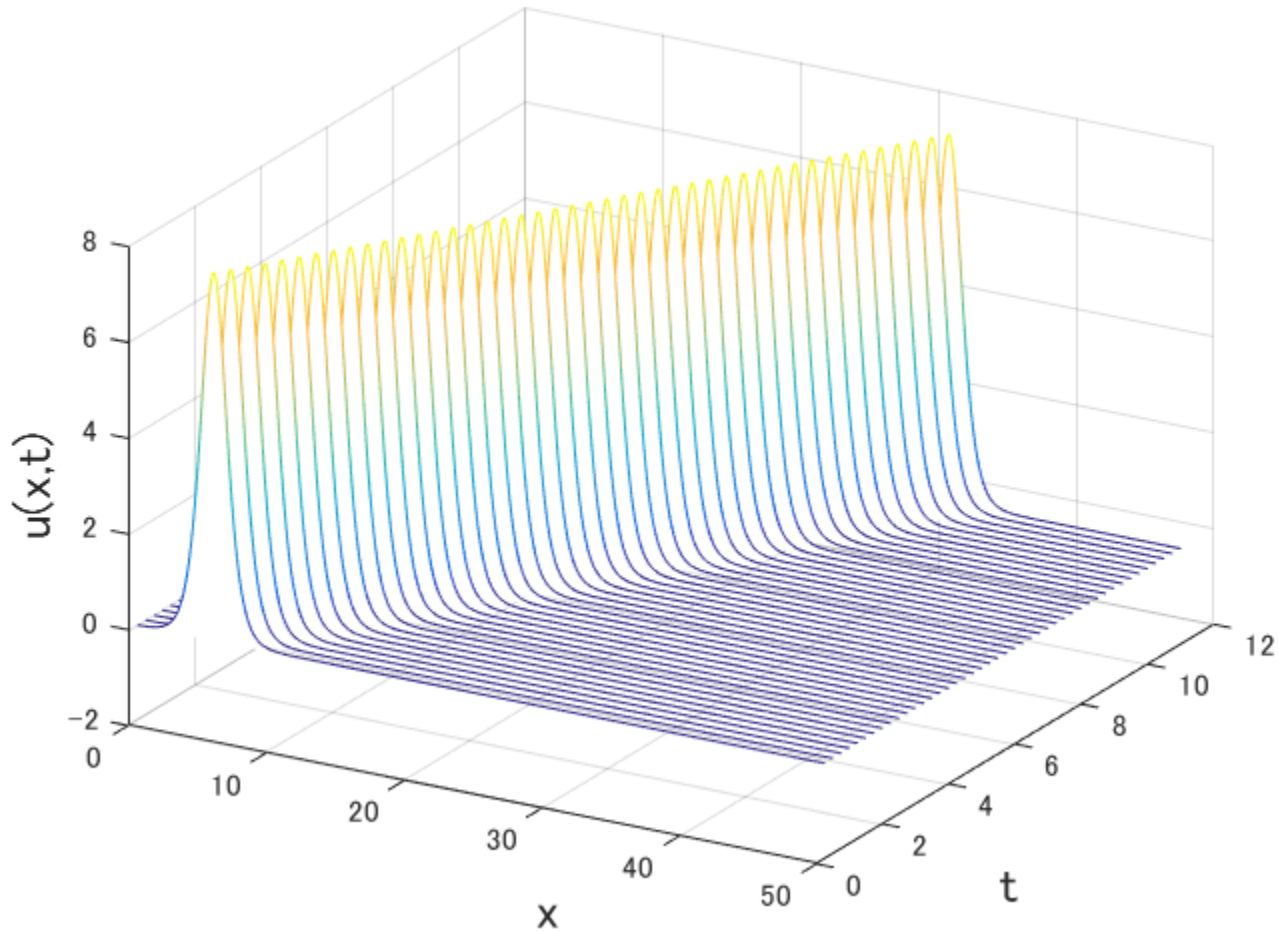


$A \approx 7.8$ 、速度 ≈ 2.6 。∴山の位置は5から $5 + 2.6 \times 11 \approx 33.6$ に移動。

動画

3.2.3 K d V 方程式の顕著な性質 1 (2)

前のスライド (ムービー) の結果を 3 次元表示したもの

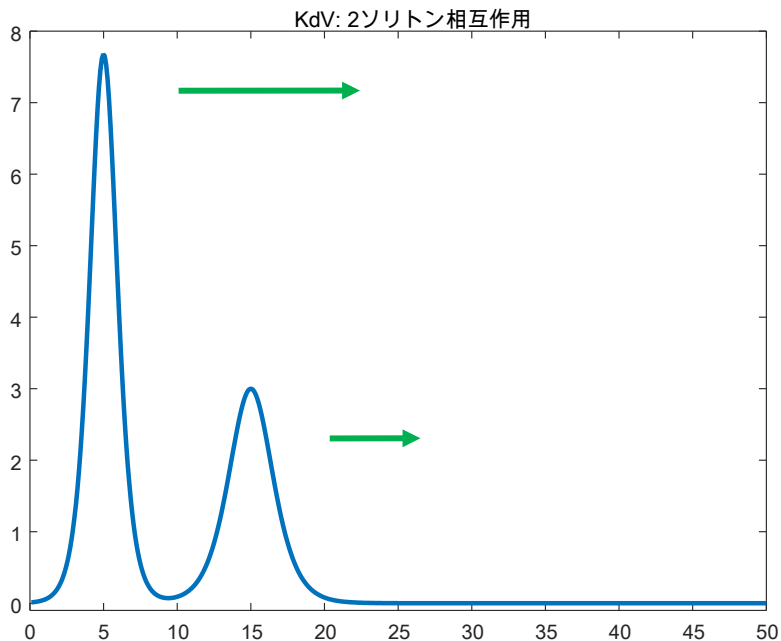


3.2.3 KdV方程式の顕著な性質2 (1)

性質2

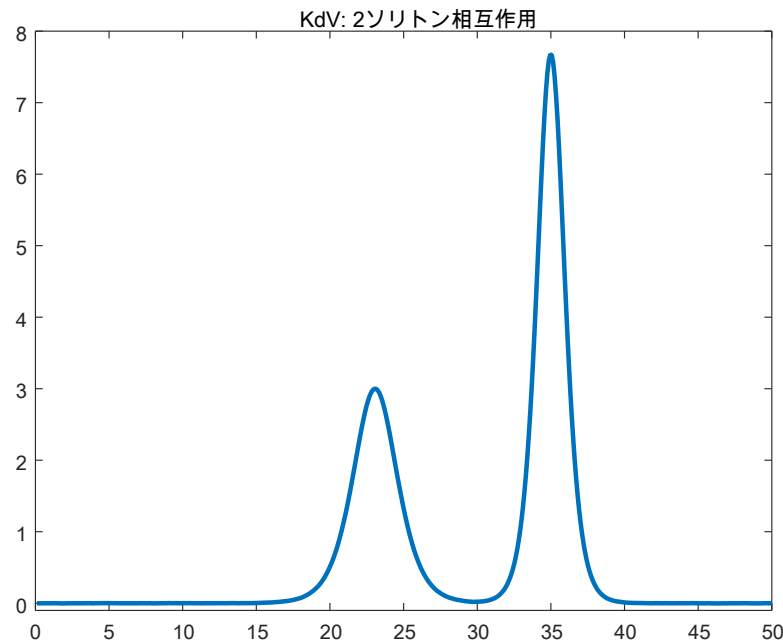
波の速度は振幅に比例するので、振幅の大きな波が後ろにあると前の波を追い抜く。
速い波が遅い波を追い越した後は、それぞれ元の振幅と元の速度で進む。

初期条件



振幅 $A \approx 7.8, 3.0$
速度 $\approx 2.6, 1.0$
山の位置は 5, 15

t = 11の波



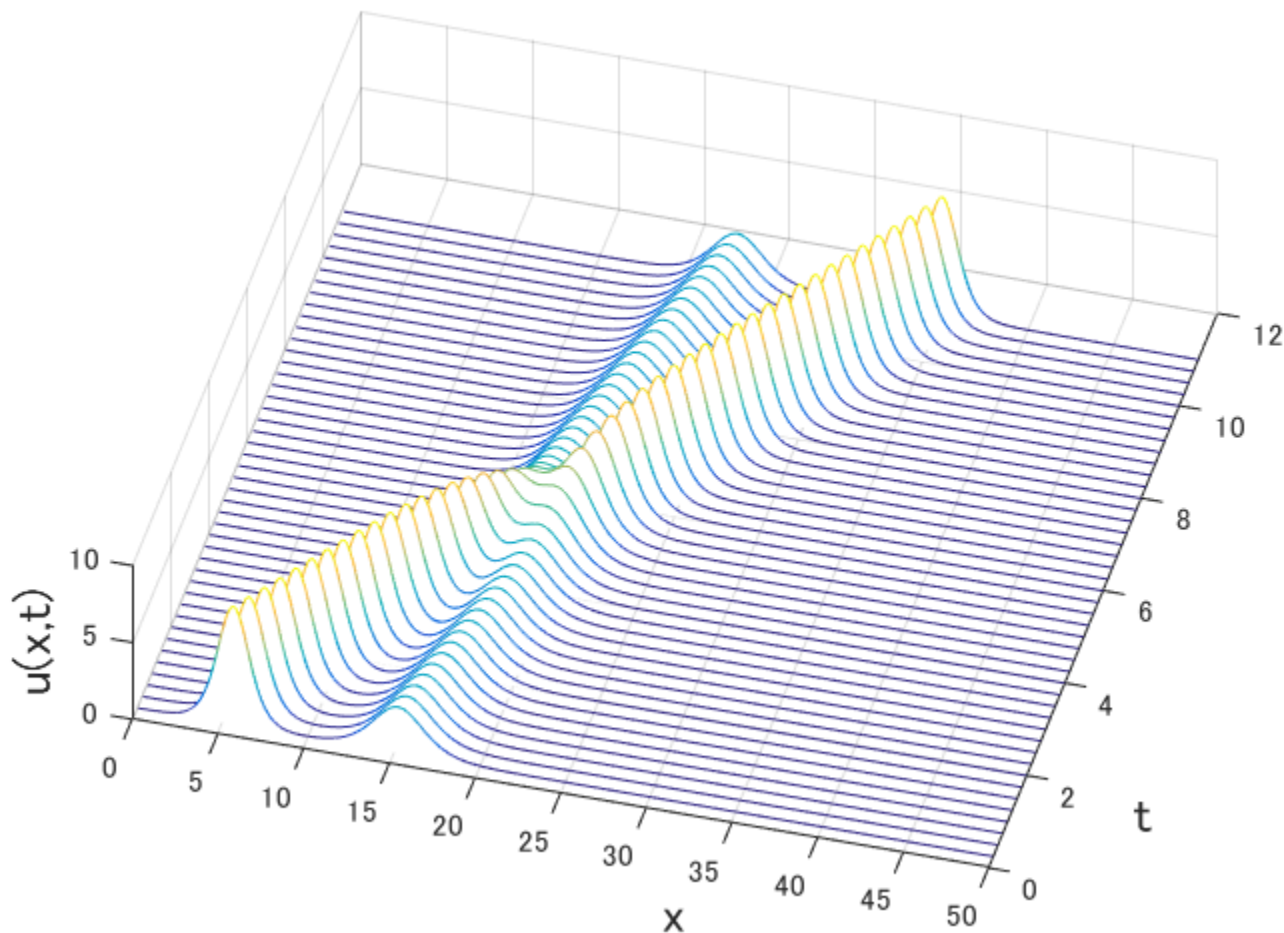
山の位置
 $35 > 33.6 \approx 5 + 2.6 \times 11$
 $24 < 26 \approx 15 + 1.0 \times 11$

追い抜くとき、波
が入れ替わる

動画

3.2.3 K d V 方程式の顕著な性質 2 (2)

前のスライド (ムービー) の結果を 3次元表示したもの



- ・ 相互作用をしているとき、速い波の振幅が一時的に下がる。
- ・ 追い越した後、速い波は位相が進み、遅い波は位相が遅れる。

3.3 解析への応用例 2 : Boussinesq方程式

KdV方程式

ソリトンは一方向にしか進まない



反射や正面衝突は確認できない

Boussinesq方程式

$$\underbrace{u_{tt} - u_{xx}}_{\text{波動方程式}} - \underbrace{(u^2)_{xx}}_{\text{非線形性}} - \underbrace{u_{xxxx}}_{\text{散逸性}} = 0$$

4階の
偏微分方程式

u_t : u の t による偏微分
 u_{xx} : u の x による2階偏微分

Boussinesq方程式の特徴

- ・ ソリトン解を持つ
- ・ 左右両方向に進む

Boussinesqはフランスの流体力学者

1877年の論文でKdV方程式も発見していたようであるが、論文が680ページという長さだったため、きちんと読んでもらえなかったようである。

反射に対する着目点

- ・ 反射のとき波の高さはほぼ倍になる。
- ・ 反射後は元の大きさ元の速度で進む。

正面衝突に対する着目点

- ・ 正面衝突時は波の大きさはほぼ2つの和になる。
- ・ 正面衝突後はどちらも元の大きさ元の速度で進む。

3.3 解析への応用例 2 : Boussinesq方程式

Boussinesq方程式の計算には数値解析法を用いる。

数値解析法

大別すると2つの方法がある。

(1) 有限差分法

区間 $[a, b]$ を N 等分し

$$x_n = a + (n - 1)h, \quad h = (b - a)/N, \quad x_1 = a, \quad x_{N+1} = b$$

微分を有限差分で近似する。

$$U_n(t) = u(x_n, t)$$

$$u_x(x_n, t) \approx \frac{u(x_{n+1/2}, t) - u(x_{n-1/2}, t)}{h} = \frac{U_{n+1/2}(t) - U_{n-1/2}(t)}{h}$$

※差分の間隔 h 、を小さくしていくと微分になるのであるから

、
hが小さければ良い近似になっているはず。

(2) 有限要素法

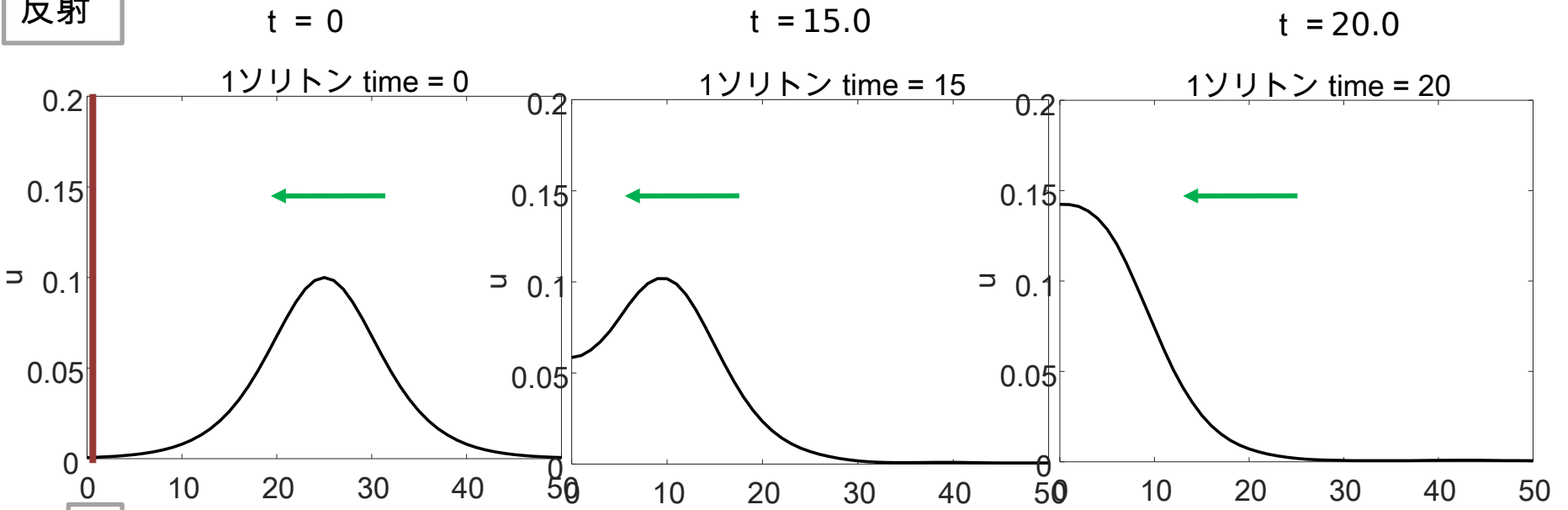
解関数を基底関数展開で近似する。誤差を何らかの意味で小さくする。

$$\hat{u}(x, t) = \sum_{i=1}^{N+1} \xi_i(t) \phi_i(x) \quad \phi_i(x) : \text{基底関数}$$

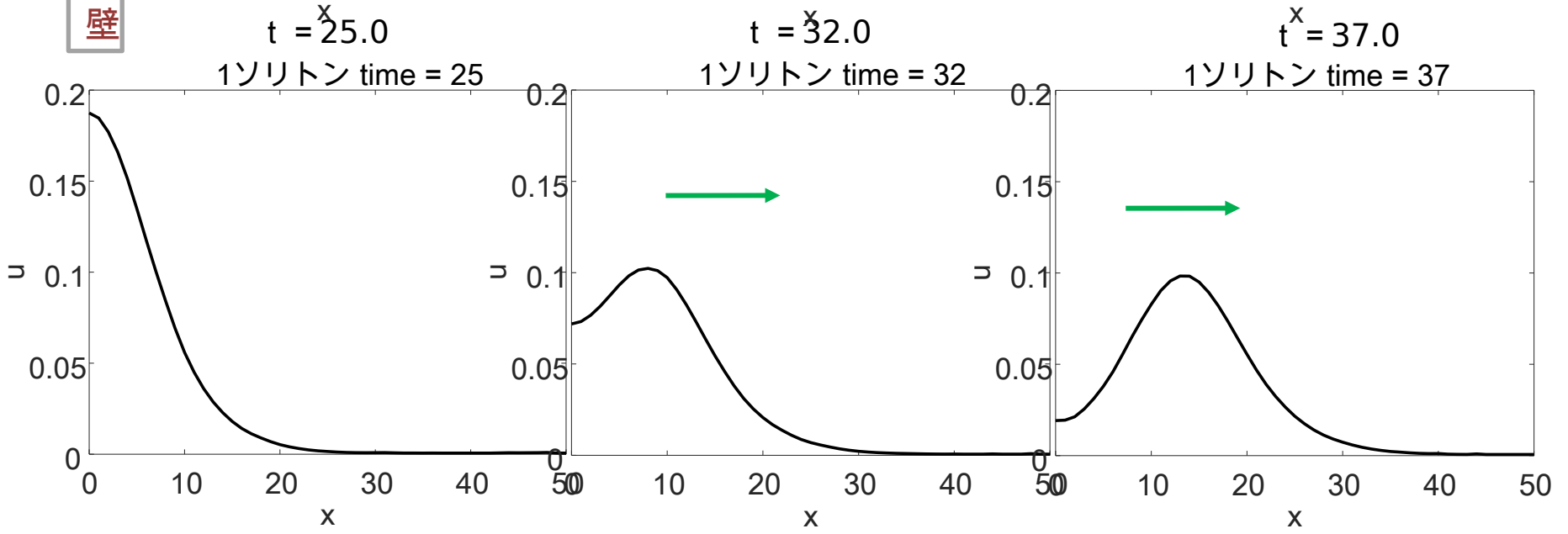
※先に建築や土木の分野で構造解析法として発展した。数学的研究は後追い。
数学の後追い例 : Diracの δ 関数、Mikshinskyの演算子法など

3.3 解析への応用例 2 : Boussinesq方程式

反射



壁



動画

3.3 解析への応用例 2 : Boussinesq方程式

衝突

t = 0

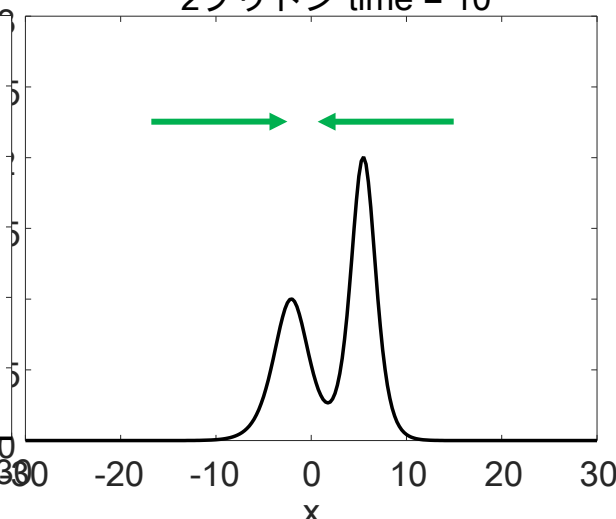
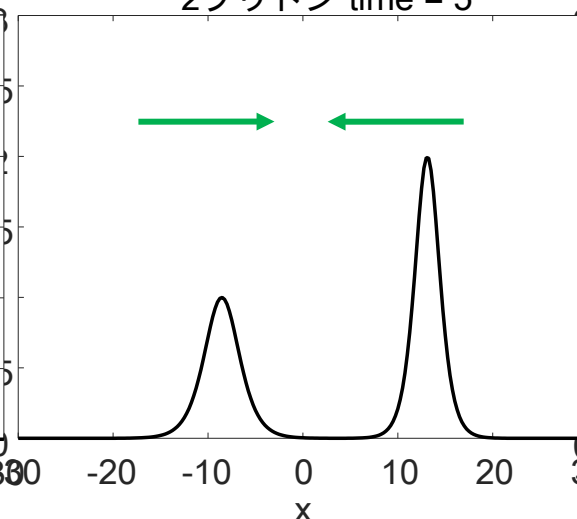
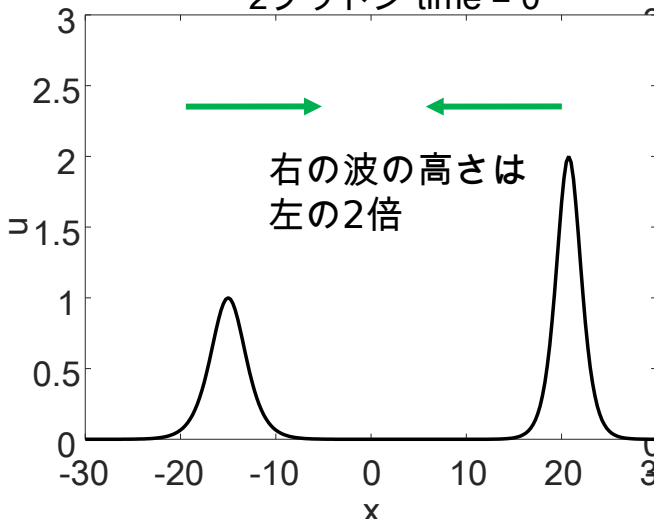
t = 5.0

t = 10.0

2ソリトン time = 0

2ソリトン time = 5

2ソリトン time = 10



t = 12.5

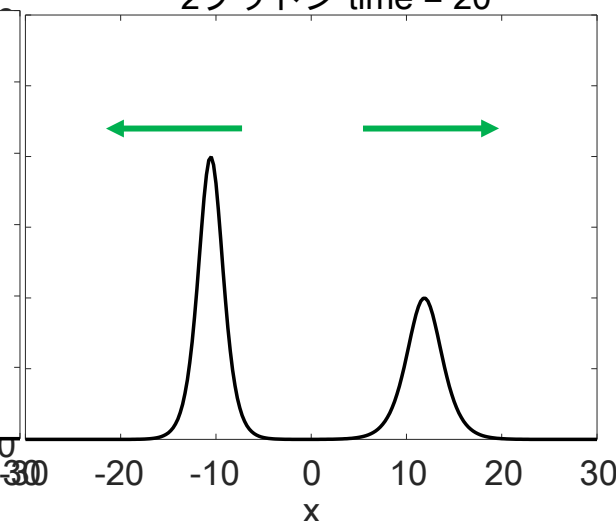
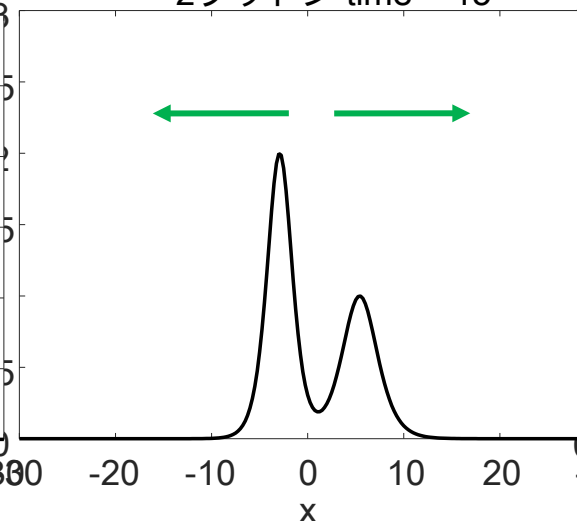
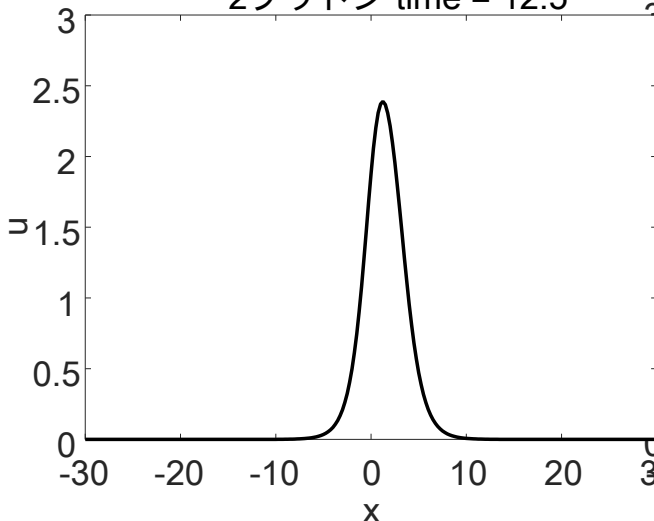
t = 15.0

t = 20.0

2ソリトン time = 12.5

2ソリトン time = 15

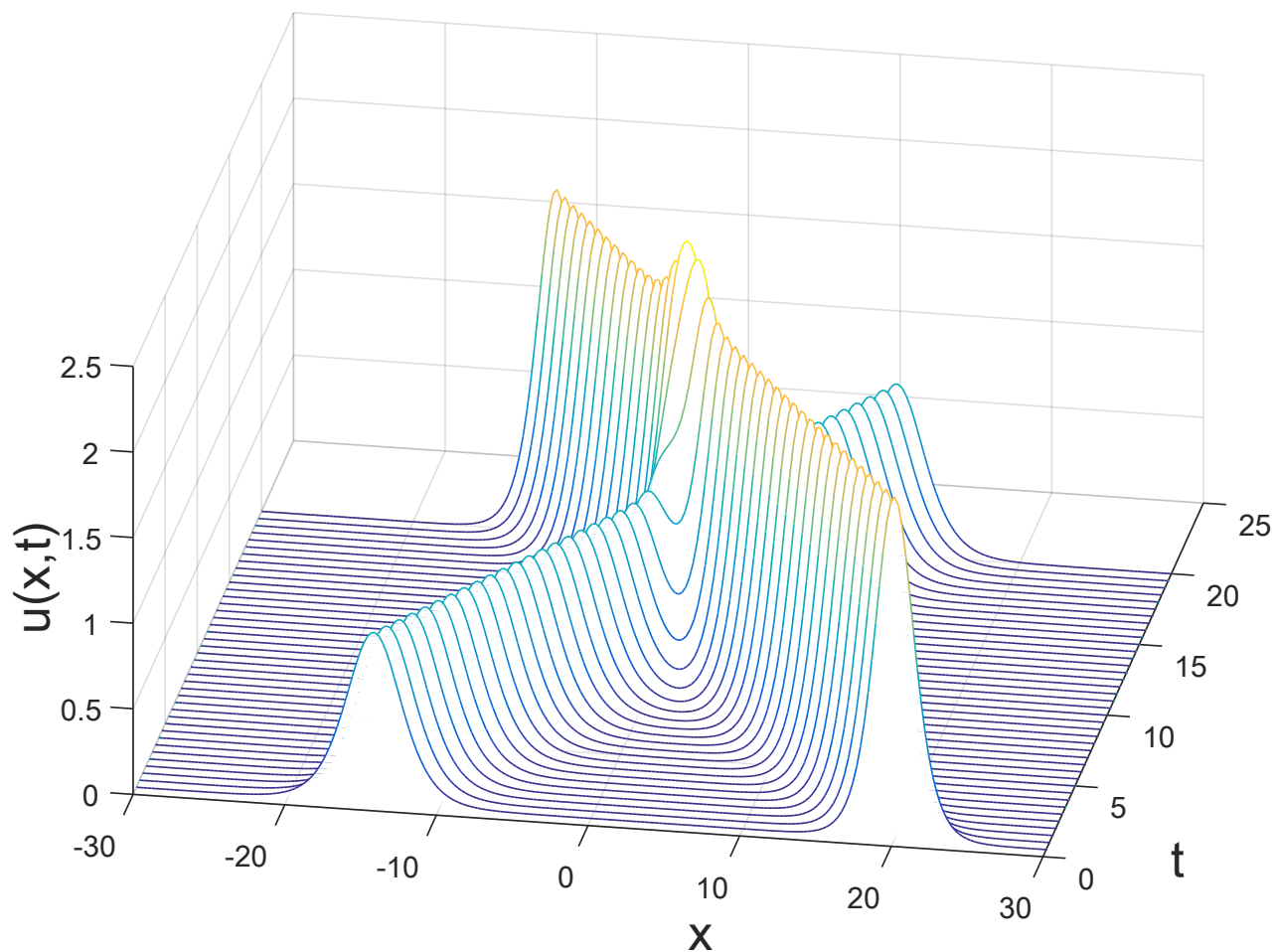
2ソリトン time = 20



動画

3.3 解析への応用例 2 : Boussinesq方程式

前のスライド (ムービー) の結果を 3次元表示したもの



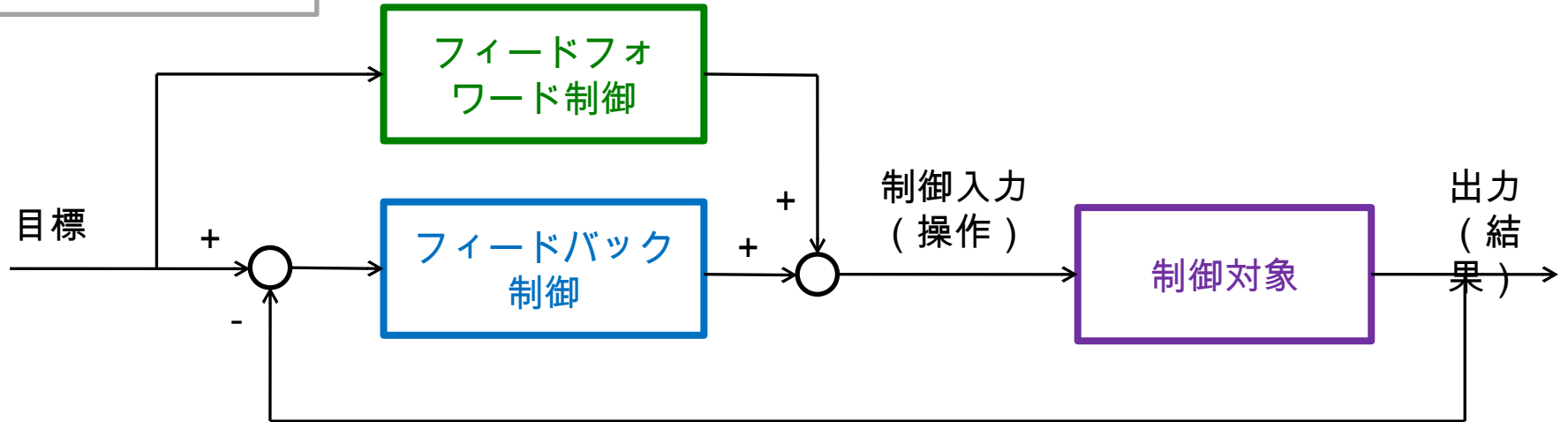
衝突後はどちらの波も衝突前の形と速度で進む

4 . 制御

制御とは

ある対象にある操作を行うことによって、目標とする状態に持って行くこと

制御方式の構造



フィードフォワード制御

目標値に対しあらかじめ決められた手順で行う制御。速い、が目標からずれる恐れがある。

フィードバック制御

目標値に対するずれを補正する制御。目標に対して正確、が時間がかかる。

人間の運動機能

初期の段階はフィードバック制御。練習と学習が進むとフィードフォワード制御に移行する。

フィードフォワード制御は制御理論によって求めることができる。

振り上げ倒立振り子

4.1 制御の代表的な種類と数理工学の役割

1入力1出力制御

例：家庭用エアコンによる暖房の制御

設定温度に対し室温の方が低ければ熱風を出し、室温の方が高ければ熱風を止める。

オンオフ制御
最もシンプルな制御

↓ レベルの高い制御

室温と温度の差に応じて熱風の量を調整し、室温を設定温度にキープする。

PID制御
P：比例、I：積分、D：微分

多入力多出力制御

例：プラントの制御 原子炉、蒸留塔など

一つの操作が複数の変数に影響を与える。



複数の入力（操作量）と複数の出力（変数）の間の関係を表現する数式モデルを作成し、適切な制御理論を適用する。

最適制御

初期状態から最終状態に最適に移行させる制御。

ムービーの例
振り上げ倒立振り子

システム同定

入出力データから制御対象の数式モデルを作る。

数学の逆問題に対応する

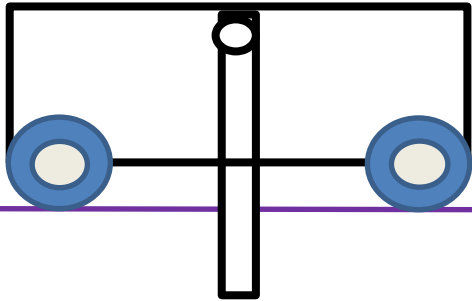
4.2 振り上げ倒立振り子への応用

振り上げ倒立振り子の制御

台車にジョイントで真下に垂れ下がった振子を、台車を左右に動かすことによって振り上げ、真上で静止させる。

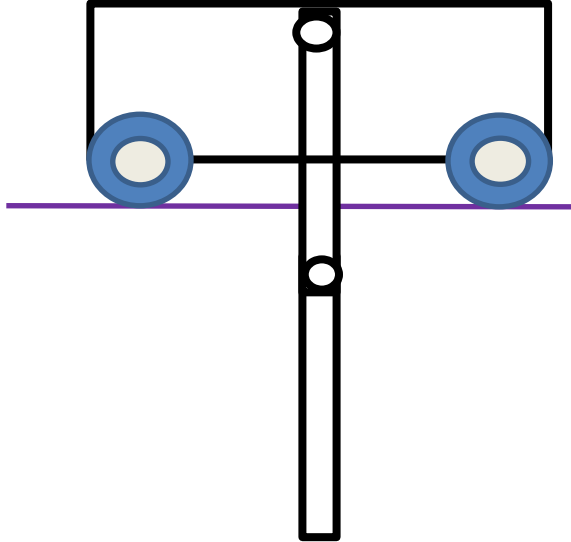
振子のバリエーション

一重

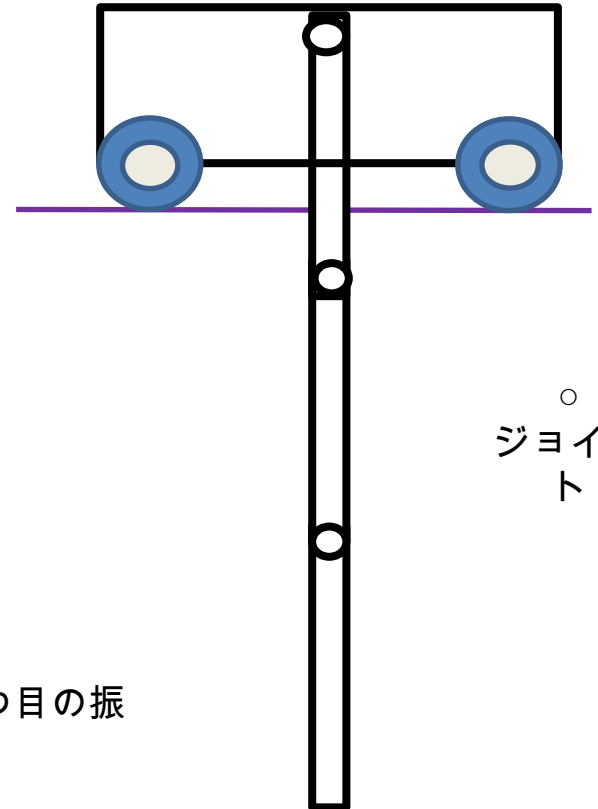


操作量 (制御量) は台車を左右に引っ張る力 (加速度)

二重



三重

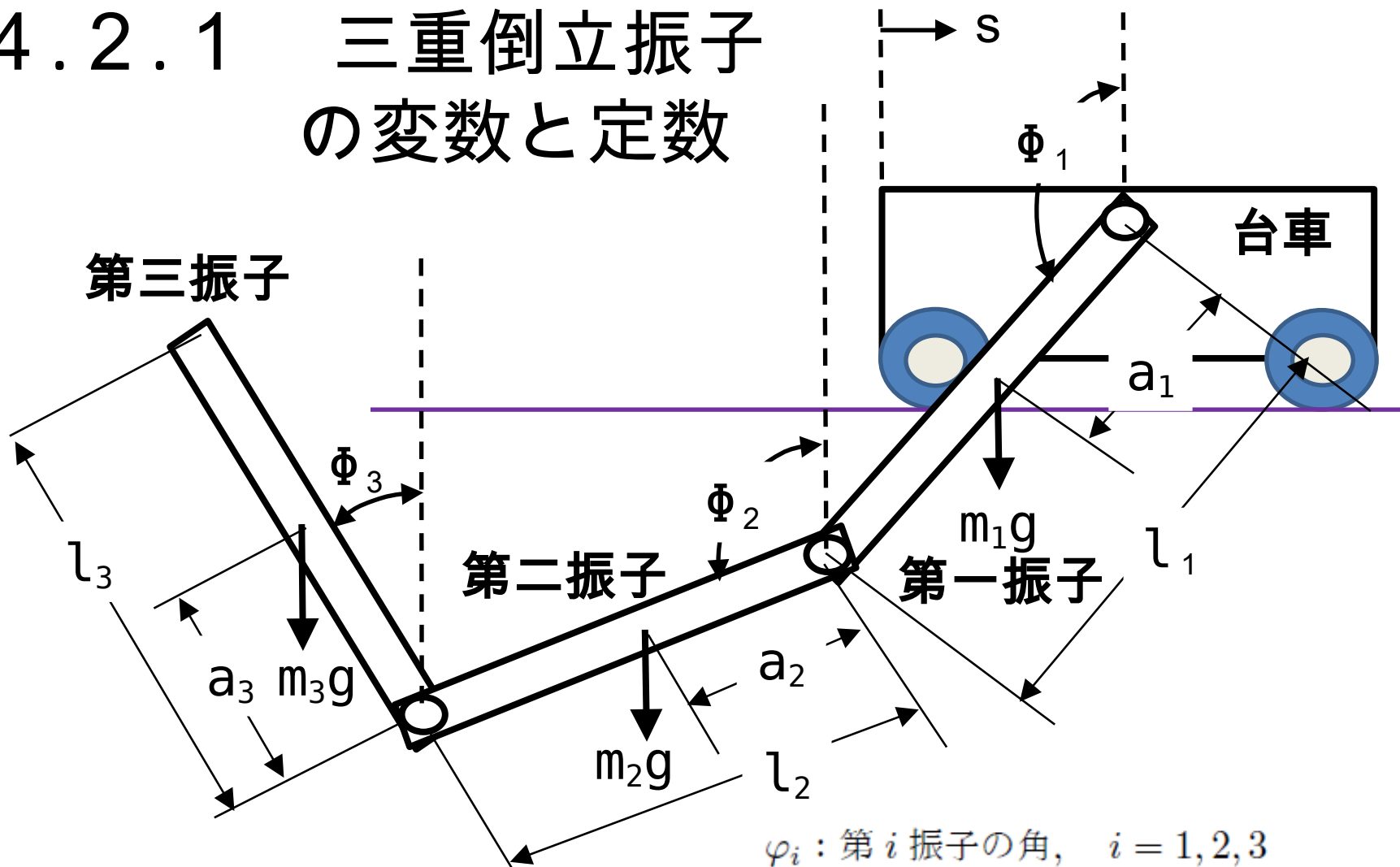


○
ジョイント

振子が一つの場合は台車ではなく人でも比較的簡単。

振子が二つ以上になると、二つ目、三つ目の振子がジョイントの回りで回転する。

4.2.1 三重倒立振子の 変数と定数



m_c : 台車の質量,

m_i : 第 i 振り子の質量, $i = 1, 2, 3$

l_i : 第 i 振り子の長さ, $i = 1, 2, 3$

a_i : 第 i 振り子の重心-ジョイント間長さ, $i = 1, 2, 3$

φ_i : 第 i 振り子の角, $i = 1, 2, 3$

J_i : 第 i 振り子の回転モーメント, $i = 1, 2, 3$

s : 台車の重心位置

g : 重力加速度

4.2.2 三重倒立振子の運動方程式

Lagrangeの運動方程式

國井修二郎： 力学II、丸善(1958), p.281

$$\frac{d}{dt} \frac{\partial}{\partial \dot{q}_k} \mathcal{L} - \frac{\partial}{\partial q_k} \mathcal{L} + \frac{\partial}{\partial \dot{q}_k} \mathcal{R} = \tau \delta_{k4}, \quad k = 1, \dots, 4$$

$$q = [\varphi_1 \quad \varphi_2 \quad \varphi_3 \quad s]^t$$

$\mathcal{L} = \mathcal{T} - \mathcal{V}$: Lagrange 関数

\mathcal{T} : 運動エネルギー

\mathcal{V} : ポテンシャルエネルギー

\mathcal{R} : Rayleigh 消散関数 (粘性摩擦などの抵抗力)

振子の重心位置 $p_{c1} = \begin{bmatrix} s - a_1 \sin \varphi_1 \\ a_1 \cos \varphi_1 \end{bmatrix}$, $p_{c2} = \begin{bmatrix} s - l_1 \sin \varphi_1 - a_2 \sin \varphi_2 \\ l_1 \cos \varphi_1 + a_2 \cos \varphi_2 \end{bmatrix}$, $p_{c3} = \begin{bmatrix} s - l_1 \sin \varphi_1 - l_2 \sin \varphi_2 - a_3 \sin \varphi_3 \\ l_1 \cos \varphi_1 + l_2 \cos \varphi_2 + a_3 \cos \varphi_3 \end{bmatrix}$

運動エネルギー $\mathcal{T} = \mathcal{T}_c + \mathcal{T}_p$

台車の運動エネルギー $\mathcal{T}_c = \frac{1}{2} m_c \dot{s}^2$

振子の運動エネルギー $\mathcal{T}_p = \frac{1}{2} \sum_{j=1}^3 m_j \dot{p}_{cj}^T \dot{p}_{cj} + \frac{1}{2} \sum_{j=1}^3 J_j \omega_j^2$ 角速度 $\omega_j = \dot{\varphi}_j, j = 1, 2, 3$

ポテンシャルエネルギー $\mathcal{V} = g(m_1 p_{c1,2} + m_2 p_{c2,2} + m_3 p_{c3,2})$

Rayleigh 消散関数 $\mathcal{R} = \frac{1}{2} d_1 \omega_1^2 + \frac{1}{2} d_2 (\omega_2 - \omega_1)^2 + \frac{1}{2} d_3 (\omega_3 - \omega_2)^2$

4.2.3 最適制御問題

倒立振子の最適制御問題

評価 (目的) 関数 最終状態の真上での静止状態からのずれ → 最小

制約条件

運動方程式

初期条件 (最初は真下で停止)

最終状態 (最終は真上で停止)

台車の位置と速度はある範囲内

台車にかけられる力はある範囲内



ポントリャーギンの最大原理

ポントリャーギン
旧ソ連の盲目の数学者

計算機を使って解く手法

- ・ 離散型最大原理
- ・ 非線形計画法



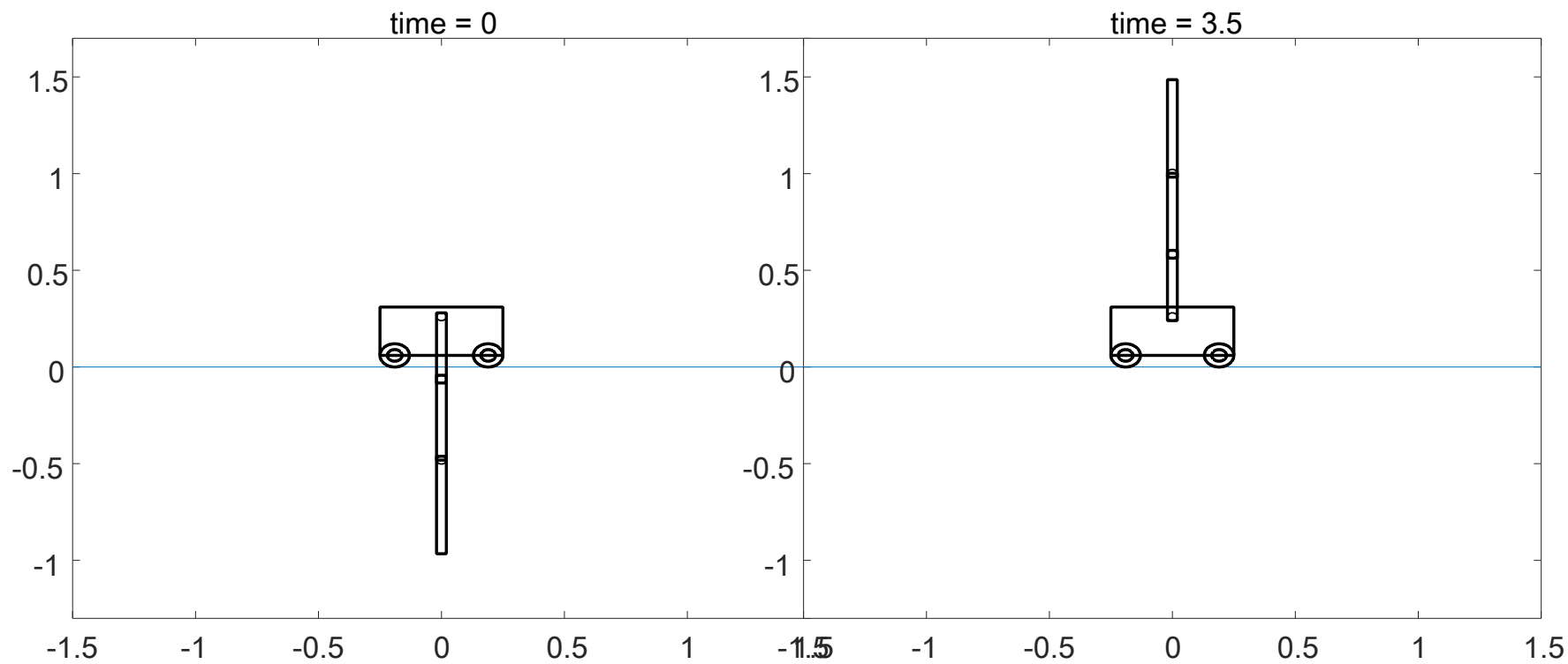
次のテーマ「最適化」の理論の一つ

4.2.5 例 (3) 三重倒立振り子

初期状態
真下で静止



最終状態
真上で静止

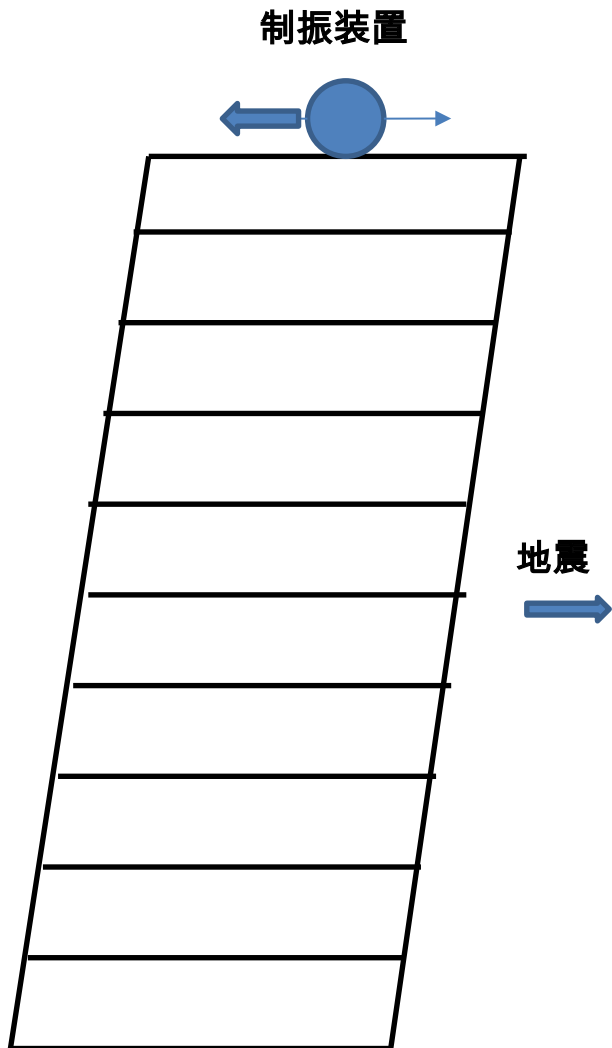


動画

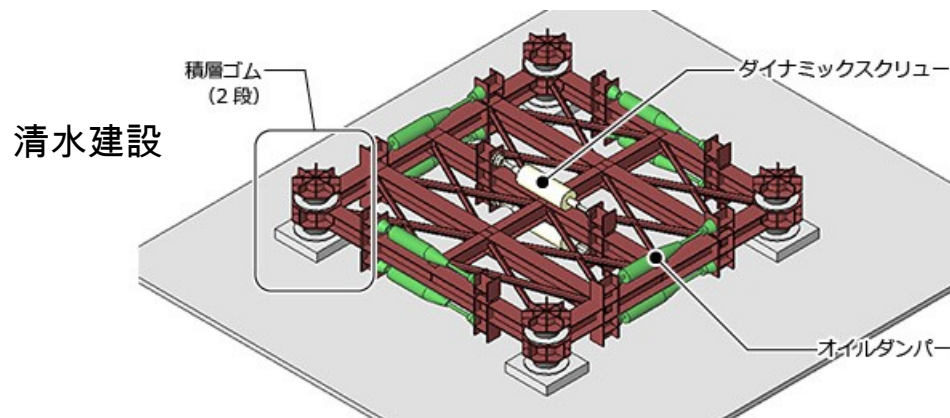
4.3 建物の制振制御への応用

地震で建物が揺れたとき、屋上に設置された制振装置のおもりを反対方向に動かし、ゆれを小さくする。

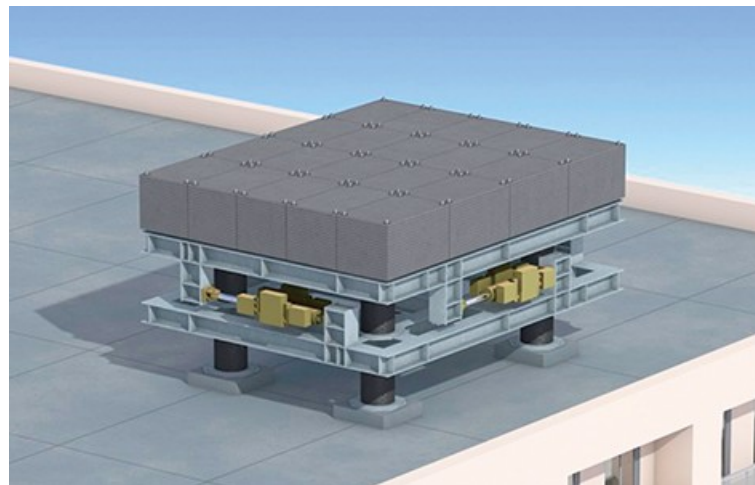
フィードバック制御の代表例の一つ



実際の制振装置



大林組

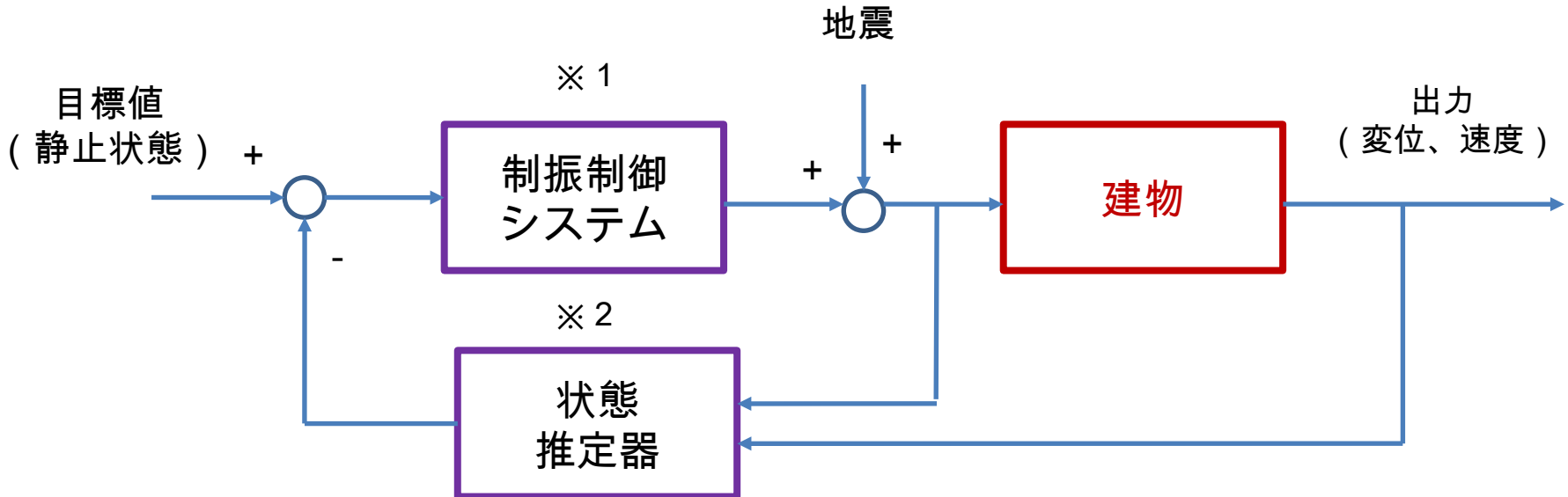


4.3.1 制振制御システム

制御システムの構造

フィードバック制御

□部分は実際の物
制御系を設計するときは
数式モデルで置き換える
。



- ※1 制御方式としては、
最適レギュレータ、モデル予測制御、極配置など
- ※2 出力がすべて計測できる場合は状態推定器は不要。

4.3.2 地震による建物の振動モデル

n階ビルの地震と制振制御の数式モデル

地震による加速度

制御入力

$$M\ddot{q}(t) + \tilde{C}\dot{q}(t) + Kq(t) = -M[1, \dots, 1, 1]^T \ddot{x}_g(t) - [0, \dots, 0, 1]^T F(t)$$

$$F = m\ddot{\delta}$$

$q \in \mathbb{R}^n$: 各階の変位, m : AMD の質量, δ : AMD のストローク
Active Mass Damper

M:質量行列、C:減衰行列、K:剛性行列

$$M = \begin{bmatrix} m_1 & & & 0 \\ & m_2 & & \\ & & \ddots & \\ 0 & & & m_n \end{bmatrix}, \quad K = \begin{bmatrix} k_1 + k_2 & -k_2 & & 0 \\ -k_2 & k_1 + k_2 & -k_2 & \\ & & \ddots & \\ 0 & & -k_n & k_n \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} c_1 + c_2 & -c_2 & & 0 \\ -c_2 & c_1 + c_2 & -c_2 & \\ & & \ddots & \\ 0 & & -c_n & c_n \end{bmatrix}$$

建物データ

H.Naderpour, A.Kiani, and A.Kheyroddin, Structural control of RC buildings subjected to near-fault ground motions in terms of tuned mass dampers. Scientia Iranica A (2020) 27(1), 122-133

地震波形データ

一般社団法人建築性能基準推進協会 (7 都市、12データ)

4.3.2 地震による建物の振動モデル (2)

状態空間表現

$$\dot{x}(t) = Ax(t) + Bu(t) + Hv(t)$$

$$y(t) = Cx(t) + Du(t)$$

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} \in \mathbb{R}^{2n}, \quad A = \begin{bmatrix} 0_{n \times n} & I_n \\ -M^{-1}K & -M^{-1}\tilde{C} \end{bmatrix} \in \mathbb{R}^{2n \times 2n},$$

$$B = \begin{bmatrix} 0_{n \times 1} \\ M^{-1}e \end{bmatrix} \in \mathbb{R}^{2n \times 1}, \quad e = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^n, \quad H = \begin{bmatrix} 0_{n \times 1} \\ 1_{n \times 1} \end{bmatrix} \in \mathbb{R}^{2n \times 1}$$

$$u(t) = -m\ddot{\delta}(t), \quad v(t) = -\ddot{x}_g(t)$$

離散時間表現

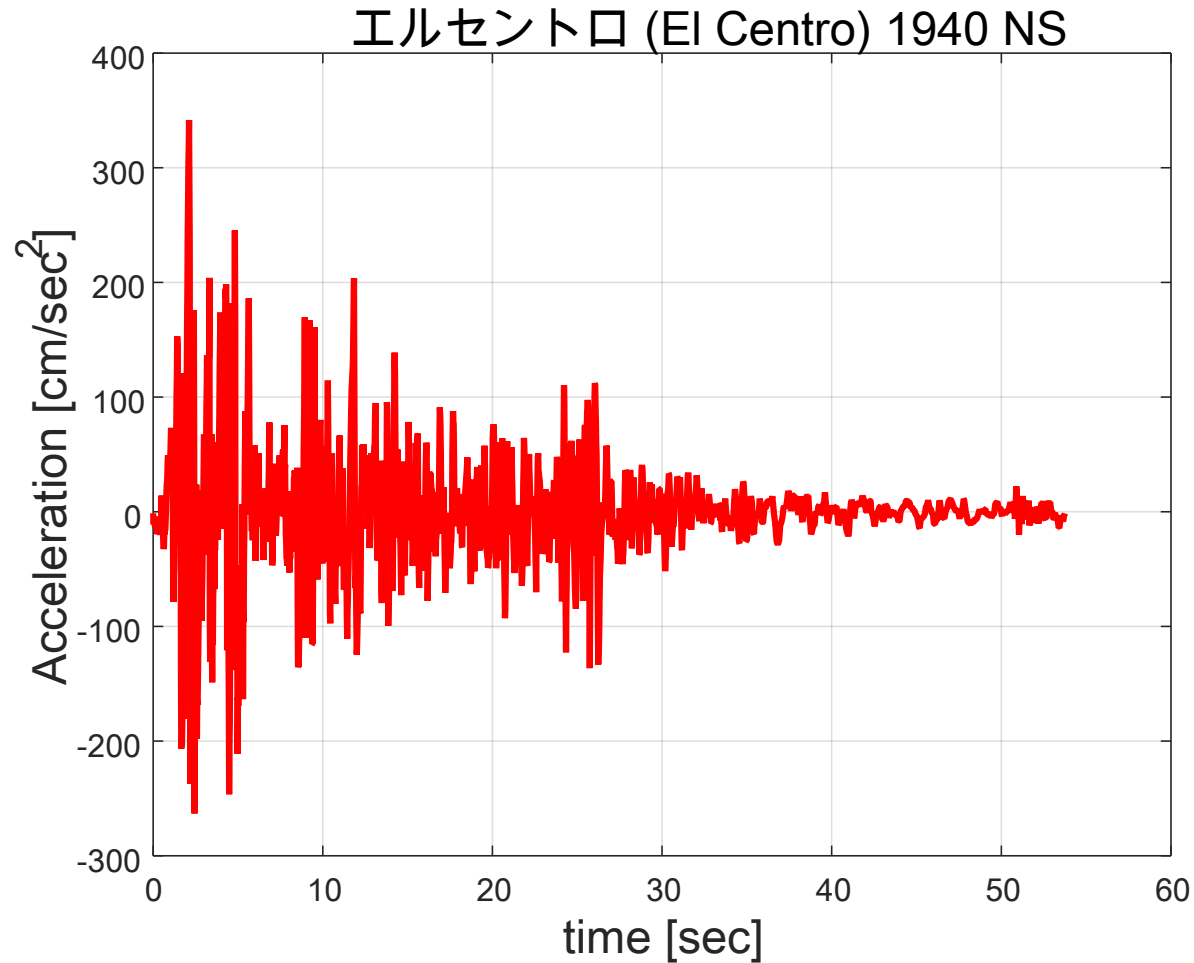
$$x(k+1) = A_d x(k) + B_d u(k) + H_d v(k)$$

$$y(k) = C_d x(k)$$

地震データ
0.01秒、0.02秒刻み

4.3.4 エルセントロ (El Centro) 1940 NS

地震波形



最大加速度 = 341.70 cm/sec²

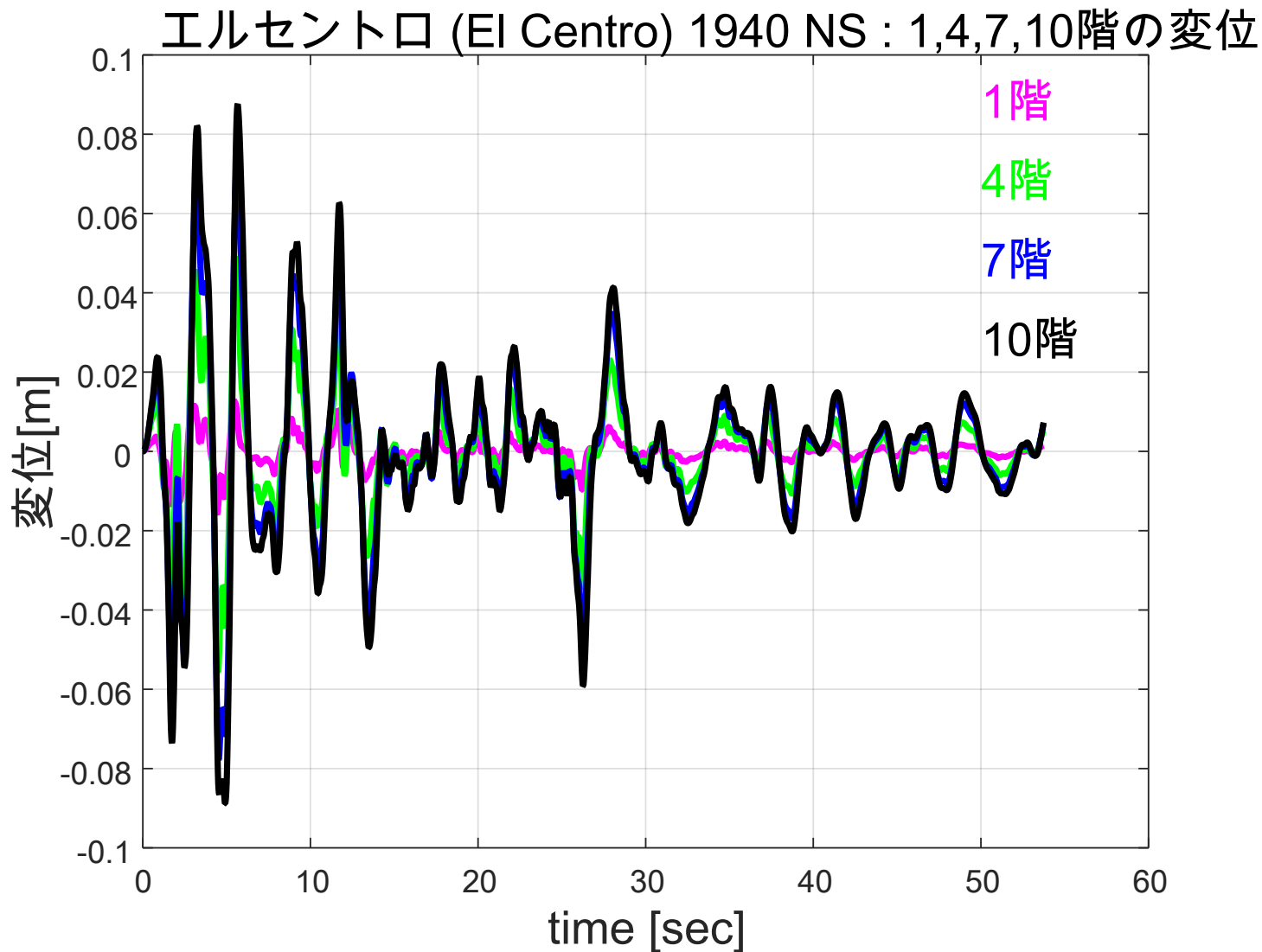
変位：制御なし

動画

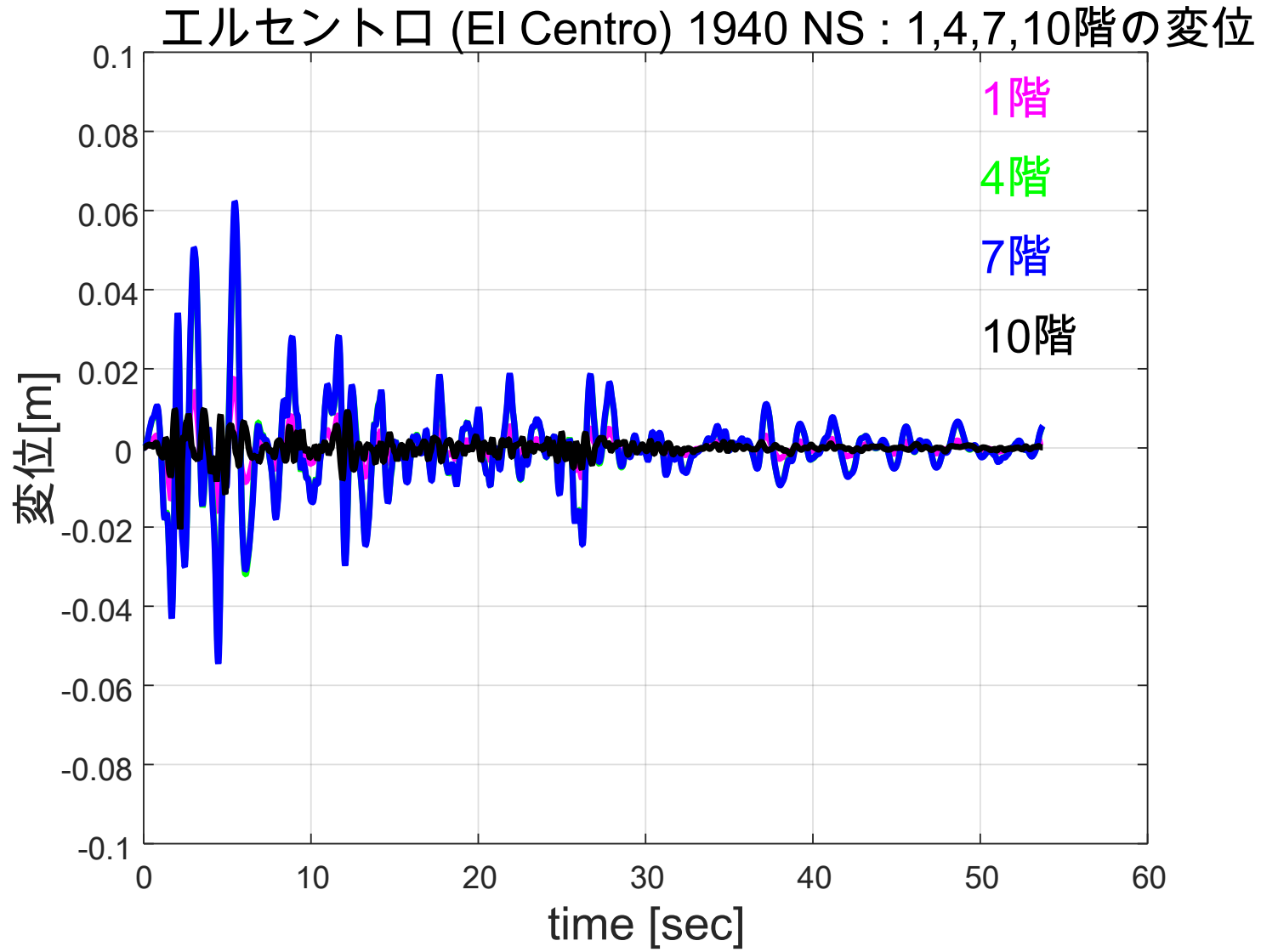
変位：制御あり（最適レギュレータ）

動画

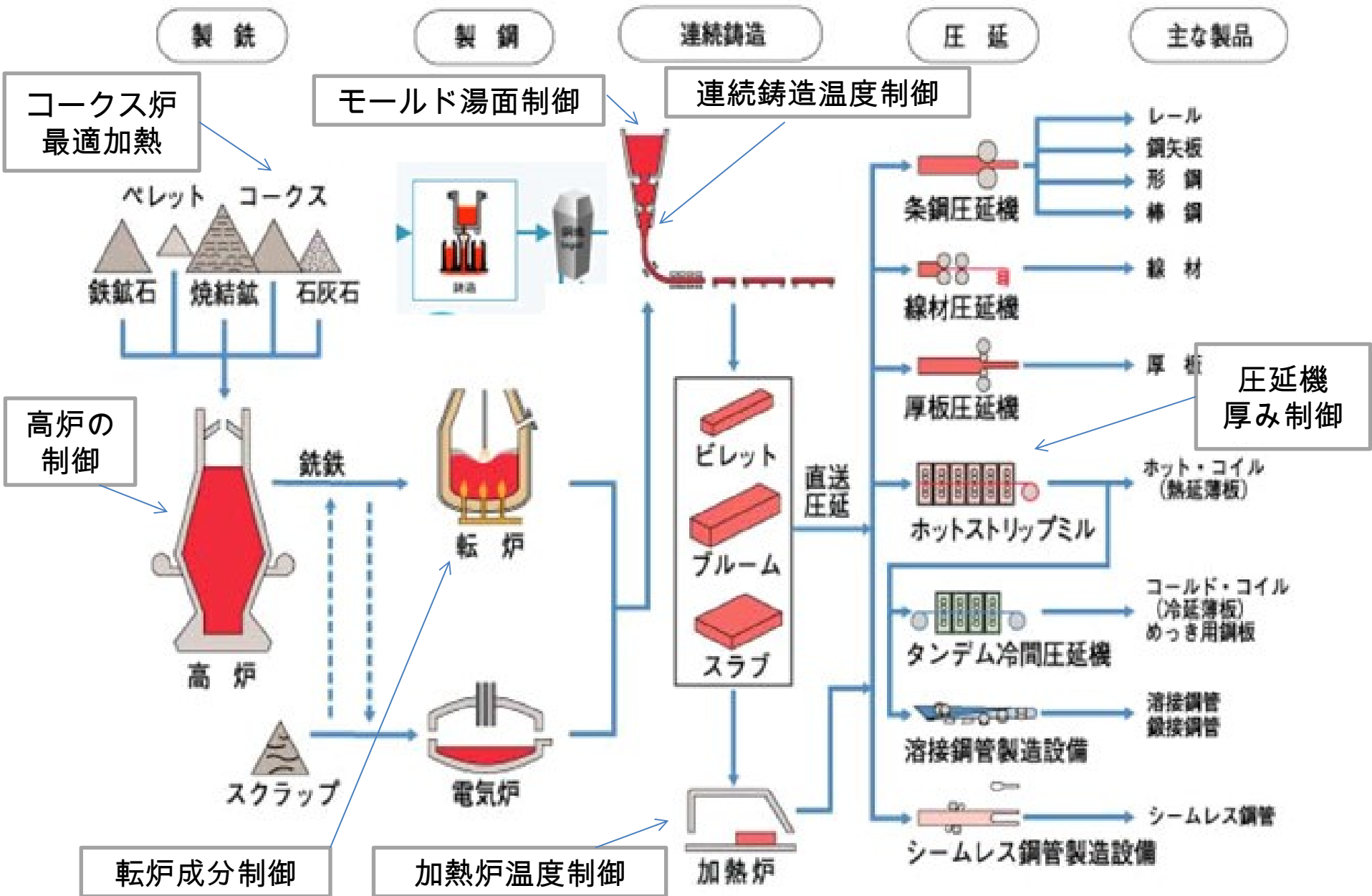
変位のプロット：制御なし



変位のプロット：最適レギュレータ



4.4 鉄鋼業における主な制御



連続鋳造プロセス

モールド湯面
自由境界

連続鋳造を支える三大技術

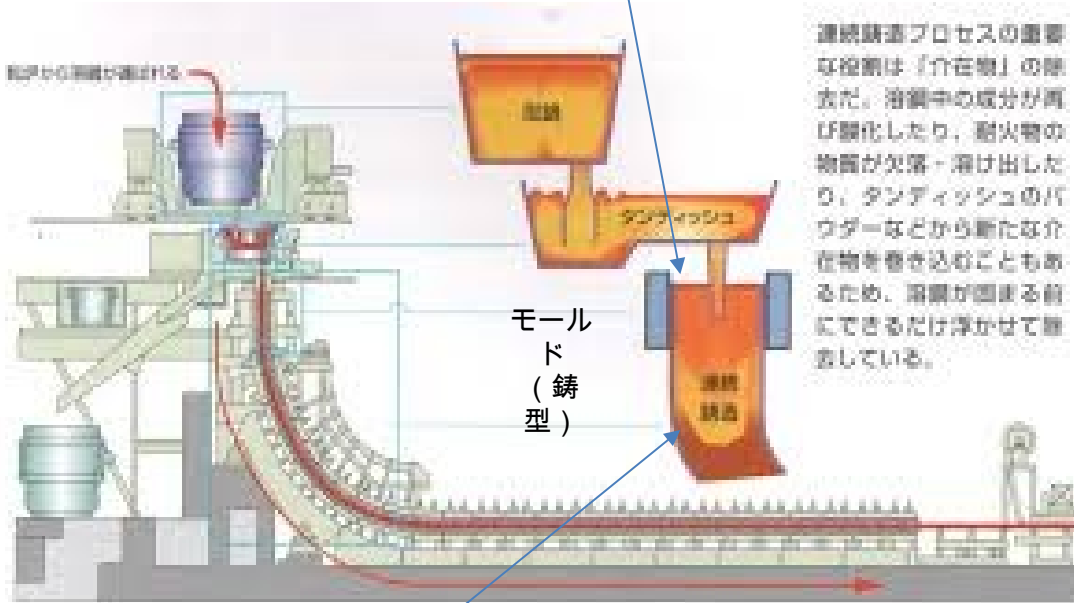
- ・ 浸漬（しんせき）ノズル
- ・ モールドパウダー
- ・ モールド振動

モールドパウダー（粉末状潤滑剤）

- ・ 保温、酸化防止、不純物除去

バルジング（鋳片のふくらみ）

- ・ これが発生すると湯面が周期的に変動する。
- ・ 湯面が激しく変動するとパウダーや不純物を巻き込む。
- ・ その結果スラブ表面にきずができる。



連続鋳造プロセスの重要な役割は「介在物」の除去だ。溶鋼中の成分が再び酸化したり、耐火物の物質が欠落・溶け出したり、タンディッシュのパウダーなどから新たな介在物を巻き込むこともあるため、湯面が固まる前にできるだけ浮かせて除去している。

出口で裁断されスラブができる



相変化による自由境界

連続鋳造プロセスの自由境界問題 1

熱伝導方程式

$$\frac{\partial T}{\partial t} = \frac{\lambda}{\rho C} \frac{\partial^2 T}{\partial z^2}$$

T : 温度, λ : 凝固シエルの熱伝導率, ρ : 密度, C 比熱

モールドと凝固シエルの間 ($z = 0$) での境界条件

$$T = T_0(t)$$

$$\lambda \frac{\partial T}{\partial z} = q_0(t)$$

$T_0(t)$: モールド温度, $q_0(t)$: モールドでの抜熱量

凝固シエルと溶鋼の間 ($z = \delta$) での境界条件

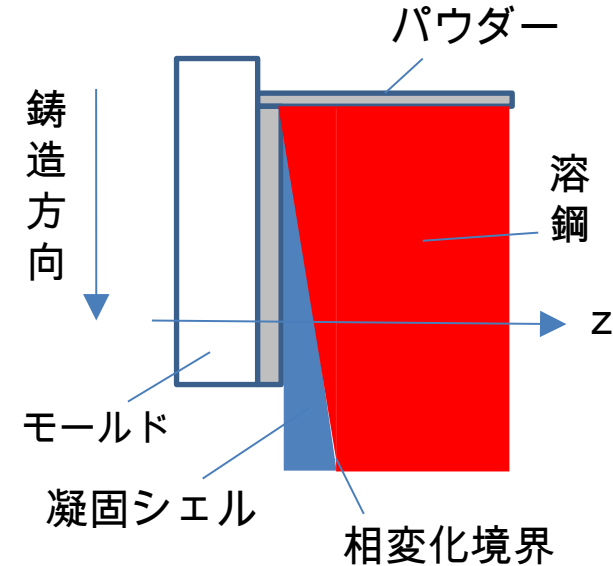
$$T = T_s$$

$$\lambda \frac{\partial T}{\partial z} = q_\delta(t) + \rho L \frac{d\delta}{dt}$$

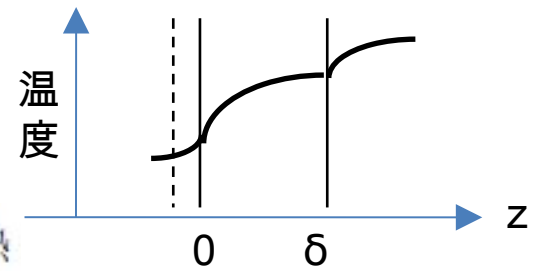
← ステファアン条件に相当

T_s : 固相線温度, $q_\delta(t)$: 溶鋼から凝固シエルへの乳熱量, L : 凝固潜熱

モールド内部

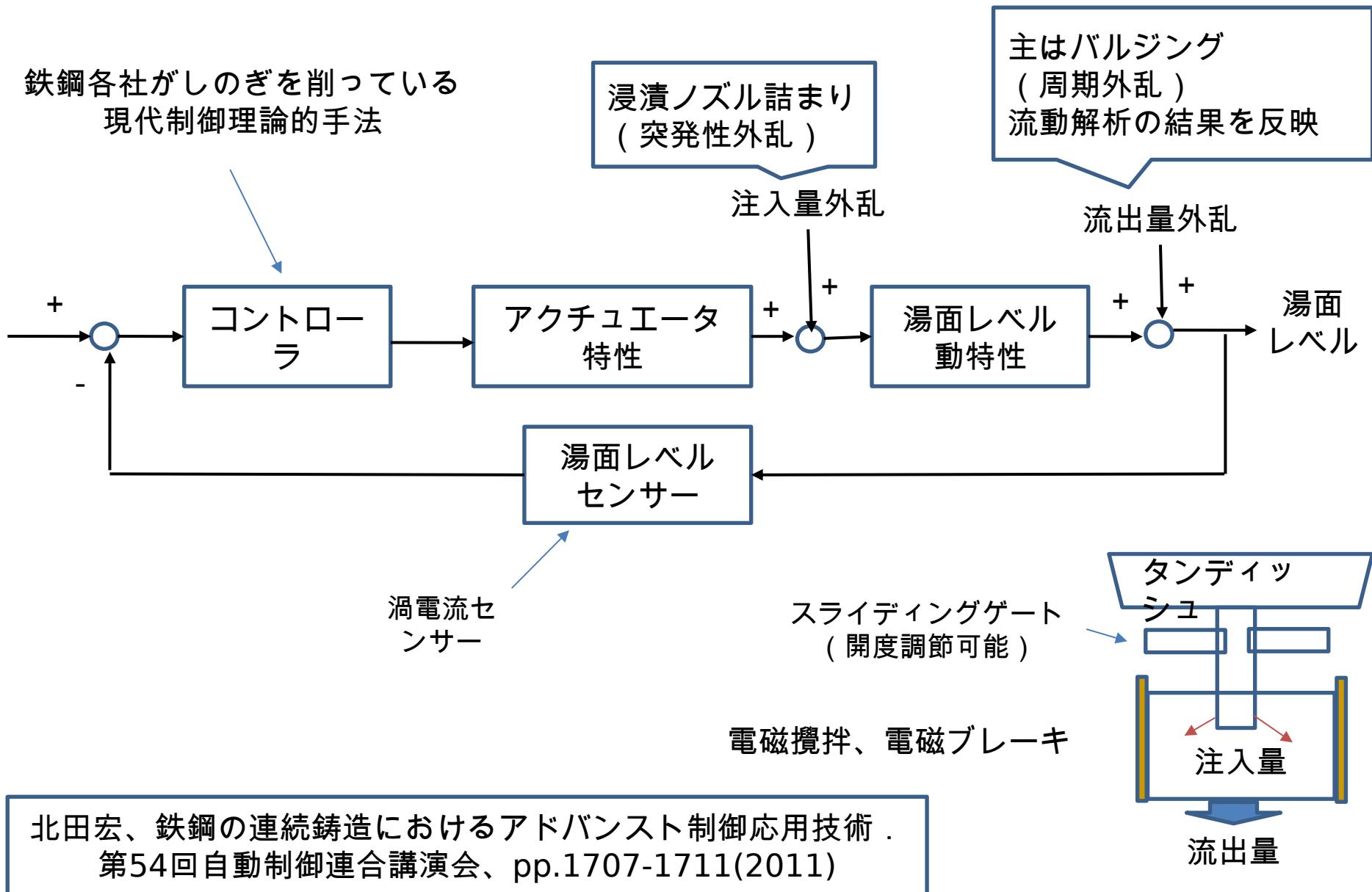


温度分布のイメージ



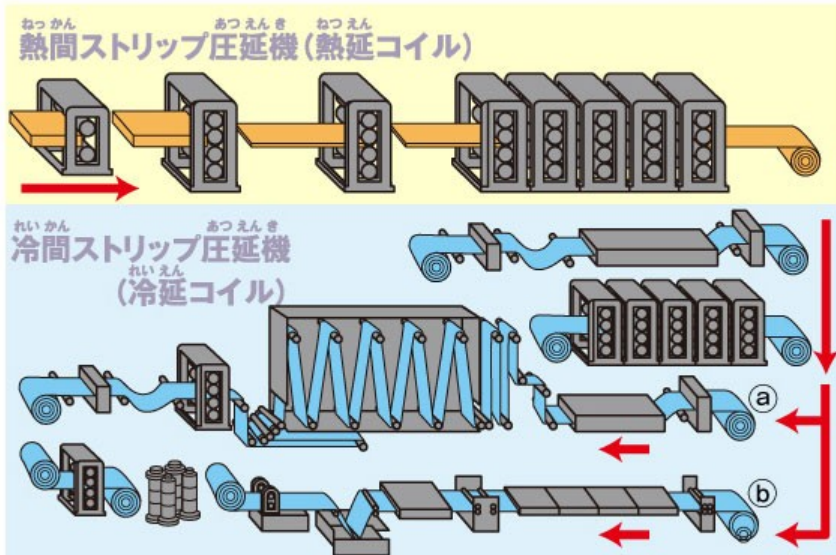
山崎伯公、計算機シミュレーションを用いた連続鋳造における冷却過程での凝固不均一の改善に関する研究．東京大学学位論文(2017)

連続鋳造プロセスの湯面制御

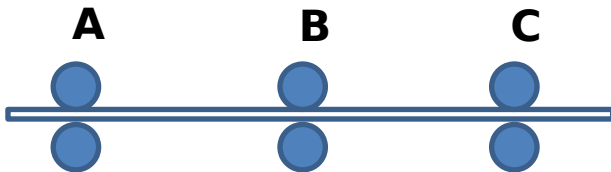


北田宏、鉄鋼の連続鋳造におけるアドバンスト制御応用技術。
 第54回自動制御連合講演会、pp.1707-1711(2011)

圧延の制御



多変数制御

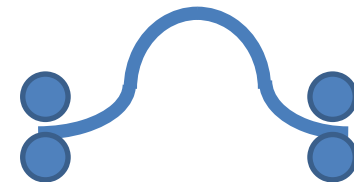


- ・ Bの圧下力が強すぎる
- ・ AB間では詰まり、BC間は引っ張られる
- ・ Bの回転が速すぎる
- ・ AB間では引っ張られ、BC間は詰まる



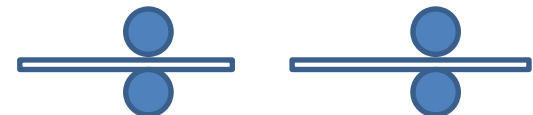
個別ではなく全ロールを同時に制御する

詰まった場合の極端例



たわむ！

引っ張られた場合の極端例



切れる！

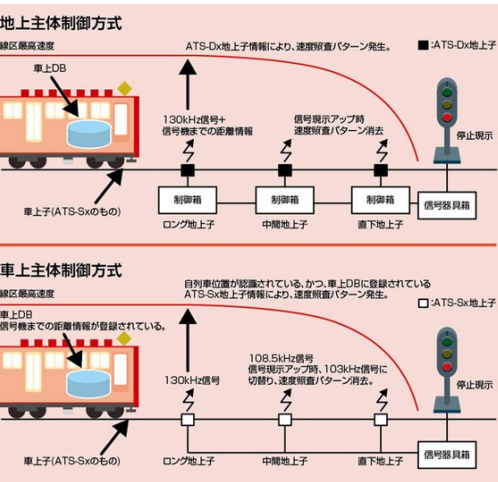
4.5 他の産業における制御

化学プラント

蒸留塔や反応器などの制御。



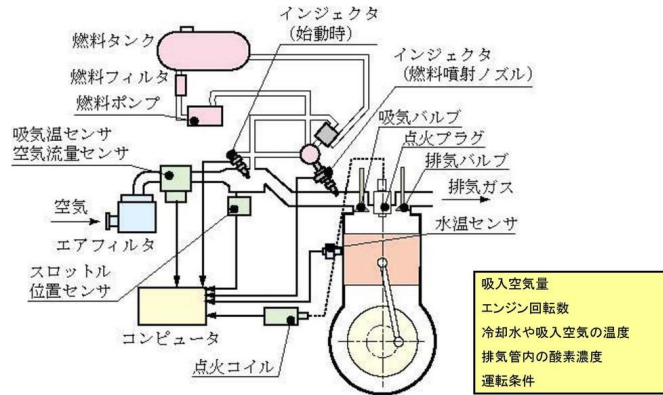
列車のATC Automatic Train Control



自動車

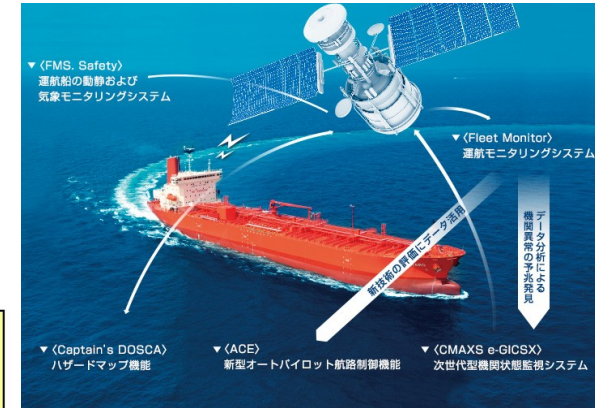
エンジンの制御、自動運転

●実際の自動車用エンジンの制御システム



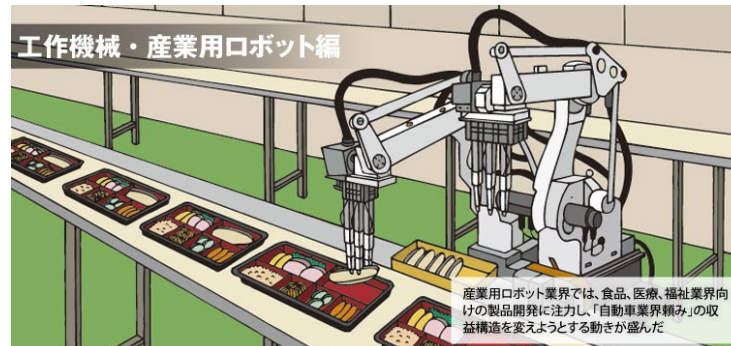
船

GPSを用いたオートパイロット



社会インフラ

食品産業におけるロボット



河川における可動堰



5 最適化

最適化とは

制約条件を満たす多くの選択肢（解の候補）の中から最も良いものを求めること。

数理工学の役割

1. 実際の問題を数式に表現する。
2. 得られた最適化モデルの解法を開発し解く。

最適化（数式）モデル

目的関数 ← どういう解が望ましいかを表す関数： 最大または最小にする

制約条件 ← 例

- ・ 1万円以下の資金で
 - ・ 製品Aは3日以内に仕上げなければならない
 - ・ 1つのラインでは同時に2つ以上の処理はできない
 - ・ 処理Bは処理Aが終わってからでないとできない
- など

最適制御との違い

最適制御

- ・ 制約条件は物理や化学の法則に基づいた式
- ・ 制約条件に微分方程式がある。

微分方程式制約は最適化理論の枠外

数理工学者の出番

最適化

- ・ 制約条件は生産活動や日常生活に関係した文章表現されている。
- ・ 数式表現するには変数をどう選ぶかから悩むことも多い。
- ・ 論理条件を含む制約条件も多く、習熟していないとその定式化は簡単ではない。

5.1 TSP (巡回セールスマン問題)

TSPとは

N個の都市すべてを訪問して最初の都市に戻って来るルートの中で、最短のルートを求めよ。

問題のバリエーション

- 標準： 都市Aから都市Bに距離が与えられている
- (1) 距離に代わりに、時間、金額など
 - (2) 都市Aから都市Bに行く場合と、都市Bから都市Aに行く場合は異なる → 非対称TSP

計算時間がかかる代表的な問題

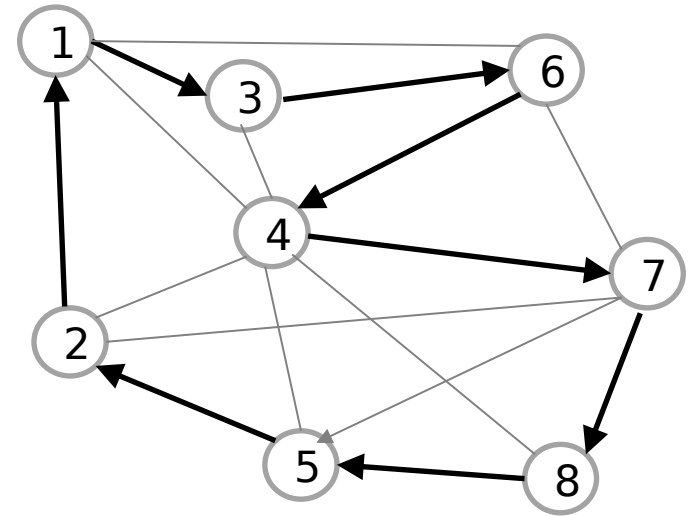
N個の都市を回る回り方 = $(N - 1)!$ 通り

$10! = 3628800$

$20! = 2.4329 \times 10^{18}$

量子コンピュータの性能比較に取り上げられる

太線は最適解のイメージ



- : 都市
- : 都市間の移動 (の道)

混合整数計画問題を分枝限定法で解くため時間がかかる

TSPの特徴

TSPは問題の意味は簡単にわかるが、最適化モデルを作成するのは意外に難しい問題

5.1.1 TSPのモデル (定式化)

Dantzig, Fulkerson, Johnson (1954)

$$\min \sum_{a \in E} c_a x_a \quad \leftarrow \text{目的関数: ルートの長さ最小}$$

$$\text{s.t.} \quad \sum_{a \in \delta^-(i)} x_a = 1, \quad \forall i \in V$$

$$\sum_{a \in \delta^+(i)} x_a = 1, \quad \forall i \in V$$

$$\sum_{a \in \delta^+(S)} x_a \geq 1, \quad \emptyset \subset S \subset V$$

$$x_a \in \{0, 1\}, \quad \forall a \in E$$

制約条件の表現は意外に難しい

- ・ 各都市に入って来るルートは一つだけ
- ・ 各都市から出て行くルートは一つだけ

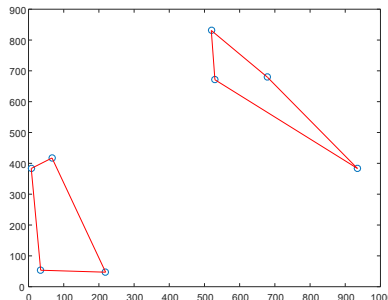
上の2つの制約だけでは閉路が複数発生し得る。

- ・ 閉路は一つだけ

部分閉路除去制約

の制約条件式は $2^N - 2$ 個ある。

この制約がないと部分閉路が発生し得る。



大きな N に対しこの実装は難しい

5.1.1 TSPのモデル(定式化)(2)

Gavish-Graves (1978)

$$\min \sum_{a \in E} c_a x_a$$

$$\text{s.t.} \quad \sum_{a \in \delta^-(i)} x_a = 1, \quad \forall i \in V$$

$$\sum_{a \in \delta^+(i)} x_a = 1, \quad \forall i \in V$$

$$y_{ij} \leq (N-1)x_{ij}, \quad i, j = 1, \dots, N, i \neq j$$

$$\sum_{j=1, j \neq i}^N y_{ij} - \sum_{k=1, j \neq i}^N y_{ki} = 1, \quad i = 2, \dots, N$$

$$\sum_{j=2}^N y_{1j} - \sum_{k=2}^N y_{k1} = -(N-1)$$

$$\sum_{j=2}^N y_{1j} = 0$$

$$x_a \in \{0, 1\}, \quad \forall a \in E, y_{ij} \in \mathbb{R}^+, \quad i, j = 1, \dots, N, i \neq j$$

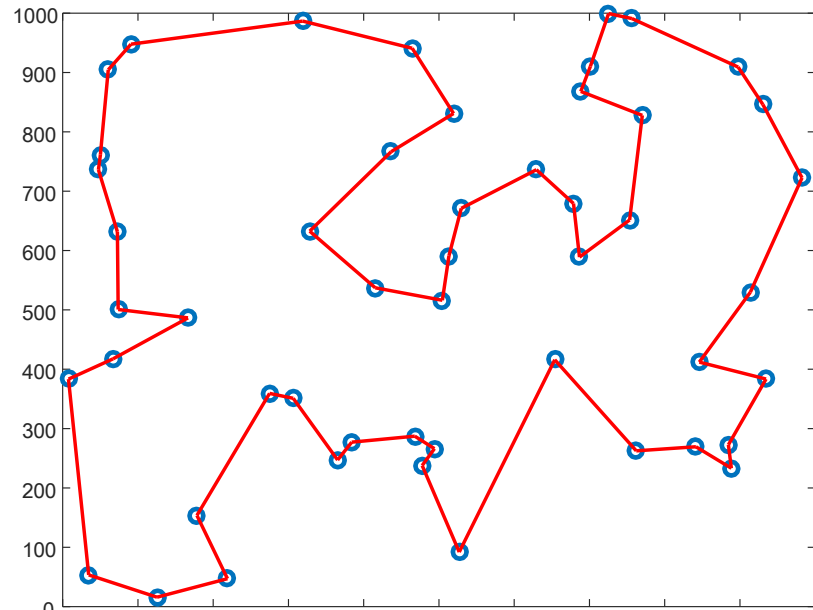
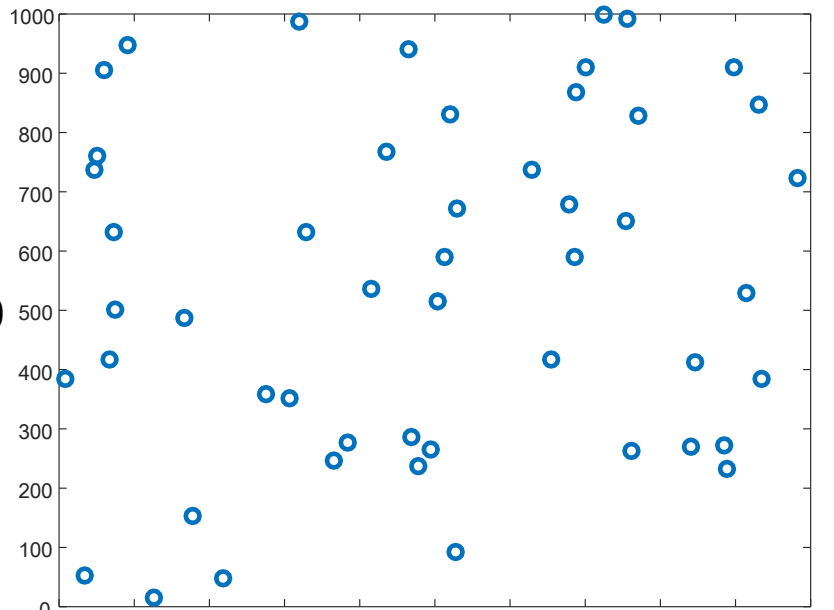
前スライドのモデルと同じ

部分閉路除去制約の改良

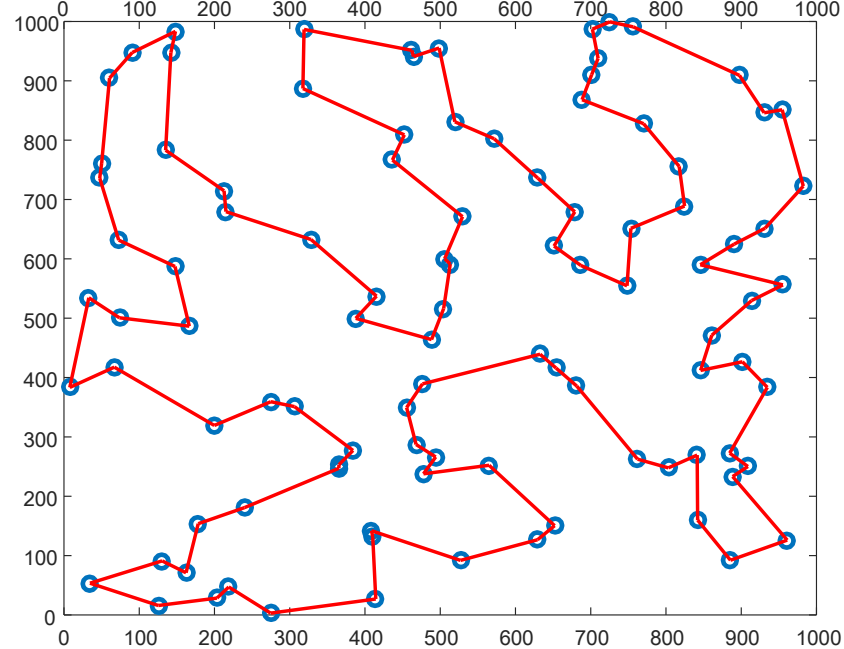
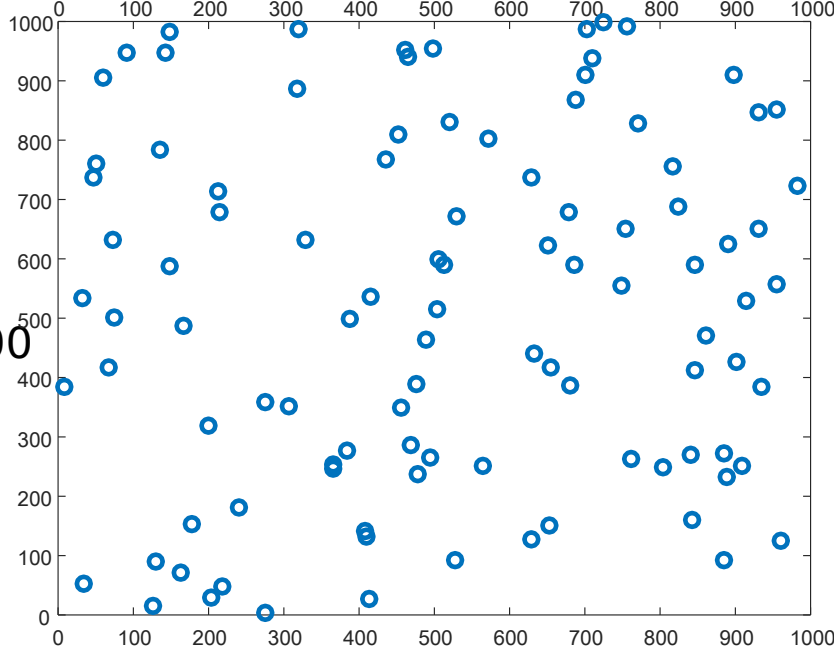
このモデルにより計算時間がかなり速くなる。

5.1.2 TSPの計算結果(2)

N = 50

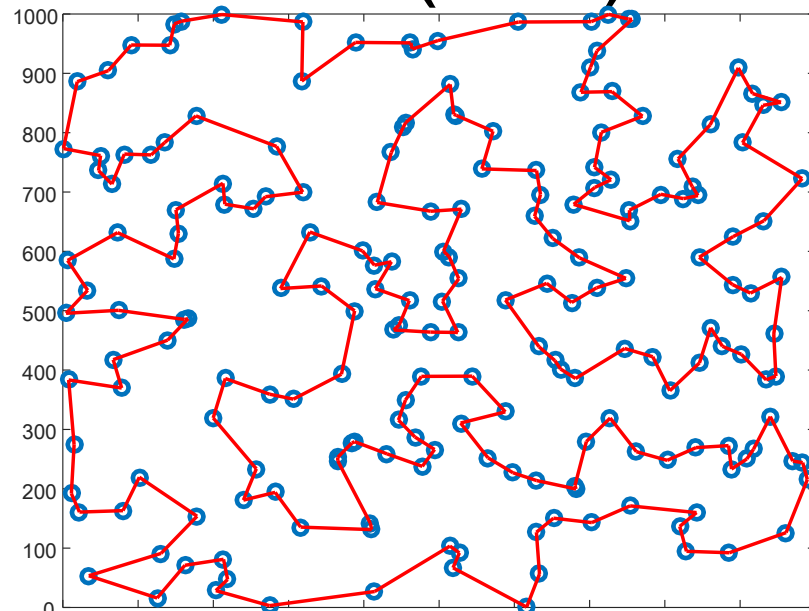
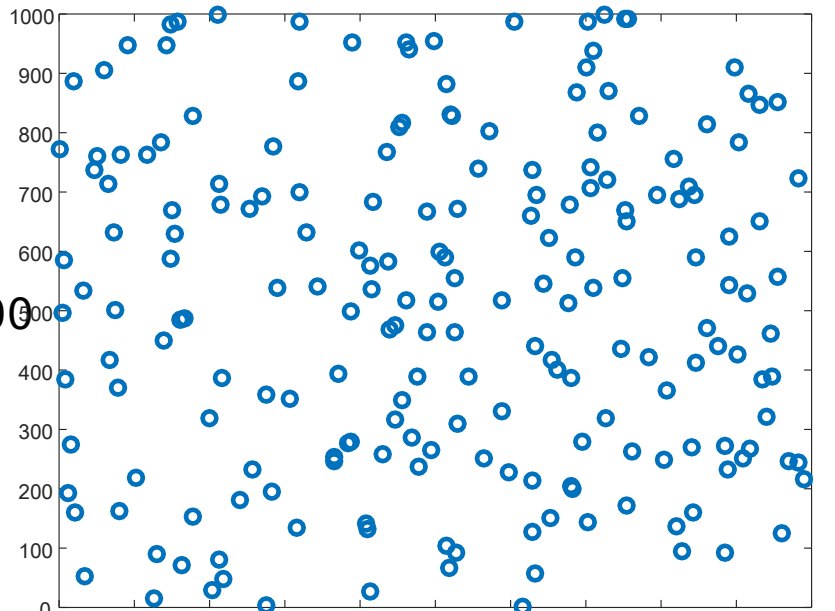


N = 100

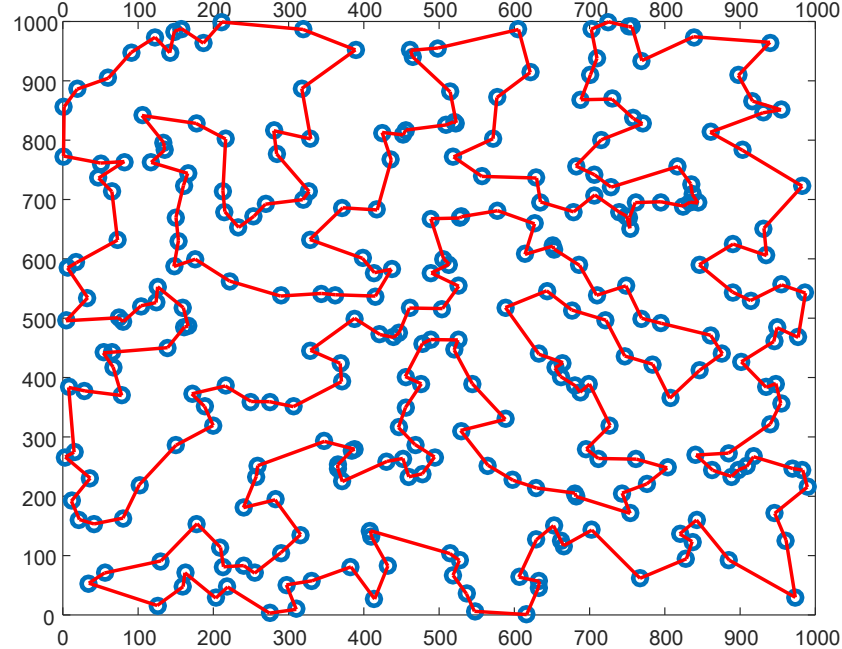
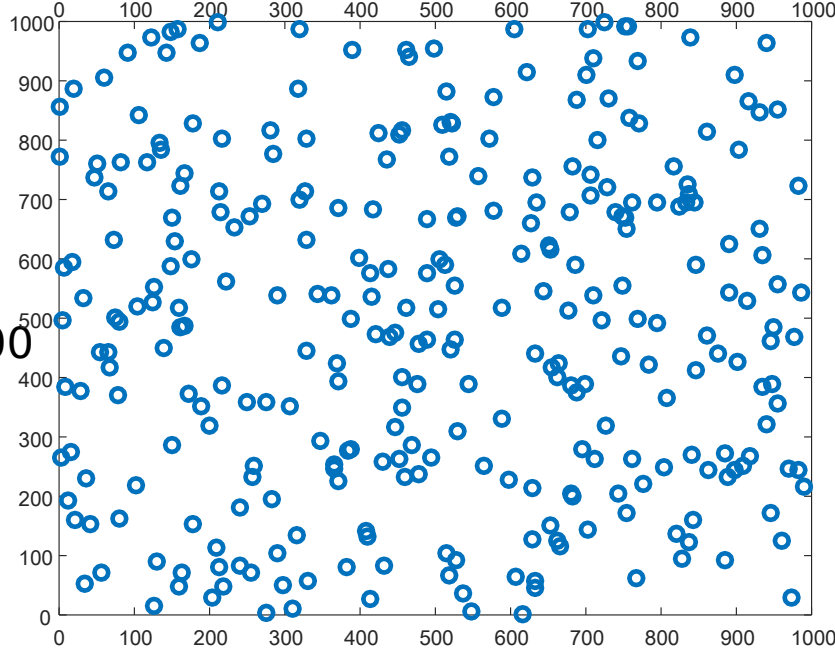


5.1.2 TSPの計算結果 (3)

N = 200



N = 300



5.2 長方形パッキング問題

N個の小長方形を横長さ固定の大長方形の中に、最も密に充填するにはどのように詰めれば良いか？ 小長方形は90度回転させてもよい。

最適化モデル

$$\min L_L(2)$$

← 縦長さ最小
(充填率最大と等価)

$$\text{s.t. } \sum_{k=1}^{K_S(i)} y(i, k) = 1, \quad i = 1, \dots, N$$

※小長方形は互いに重なってはいけない

横方向に離れている または
縦方向に離れている

$$x(i, n) + \sum_{k=1}^{K_S(i)} L_S(i, k, n) y(i, k) \leq L_L(n), \quad n = 1, 2; i = 1, \dots, N$$

← 90度回転するかしないか2つに1つ

← どの小長方形も
大長方形の中になければならない

$$\sum_{n=1}^2 z(i, j, n) \leq 1, \quad i, j = 1, \dots, N, i < j$$

$$2z(i, j, n) \leq w(i, j, n) + w(j, i, n) \leq 1 + z(i, j, n), \quad i, j = 1, \dots, N, i < j; n = 1, 2$$

$$-Q(1 - w(i, j, n)) + \varepsilon \leq x(i, n) + \sum_{k=1}^{K_S(i)} L_S(i, k, n) y(i, k) - x(j, n) \leq Qw(i, j, n)$$

$$i, j = 1, \dots, N, i \neq j; n = 1, 2$$

$$x(i, n) \geq 0, i = 1, \dots, N; n = 1, 2; y(i, k) \in \{0, 1\} i = 1, \dots, N; k = 1, \dots, K_S(i)$$

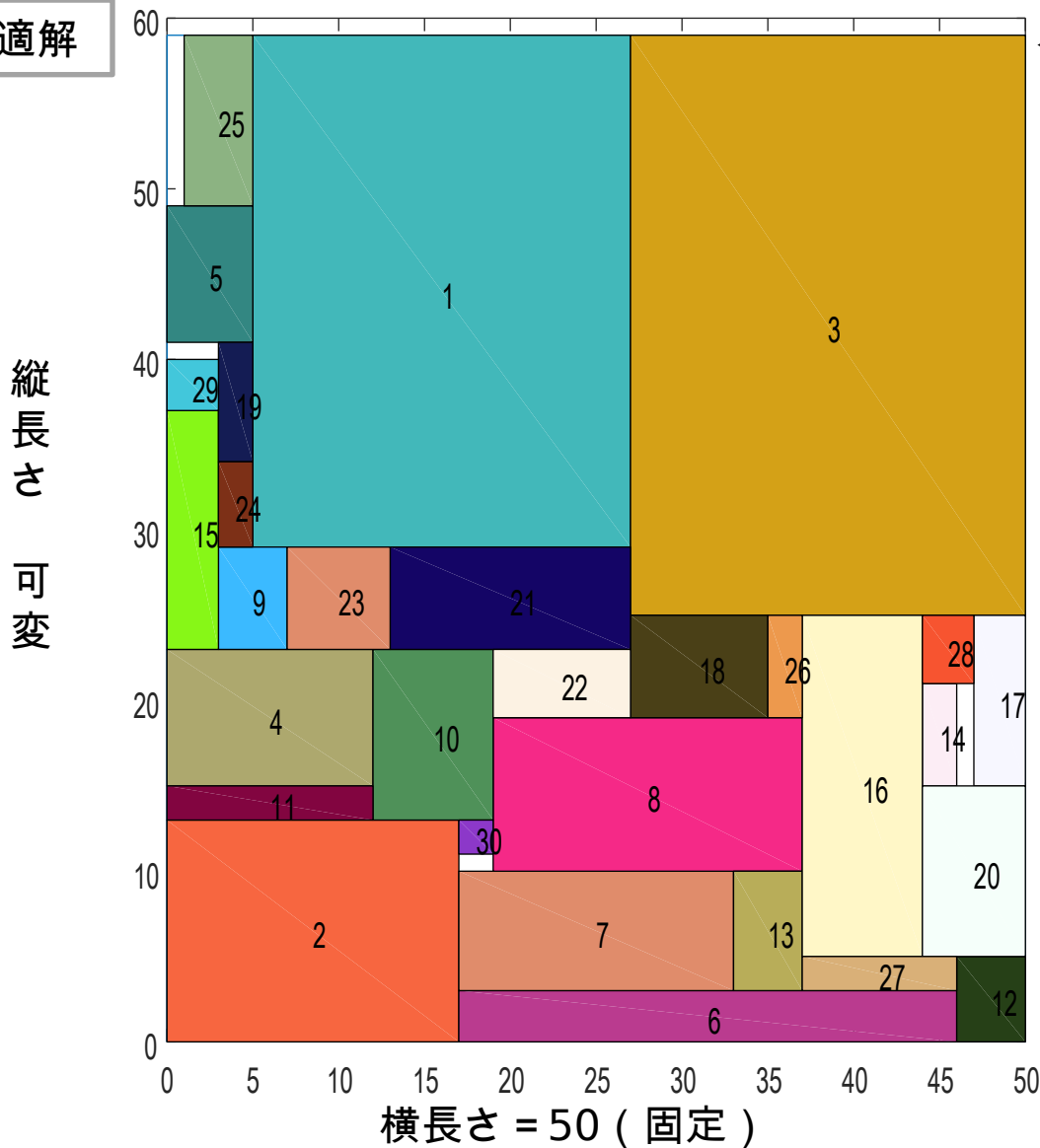
$$z(i, j, n) \in \{0, 1\}, i, j = 1, \dots, N, i < j; n = 1, 2$$

$$w(i, j, n) \in \{0, 1\}, i, j = 1, \dots, N, i \neq j; n = 1, 2; L_L(2) \geq 0$$

※

5.2 長方形パッキング問題 N=30

最適解



最適 (最小) 縦長さ
(= 59)

番号が付いていない
白い部分が詰め
られなかった所

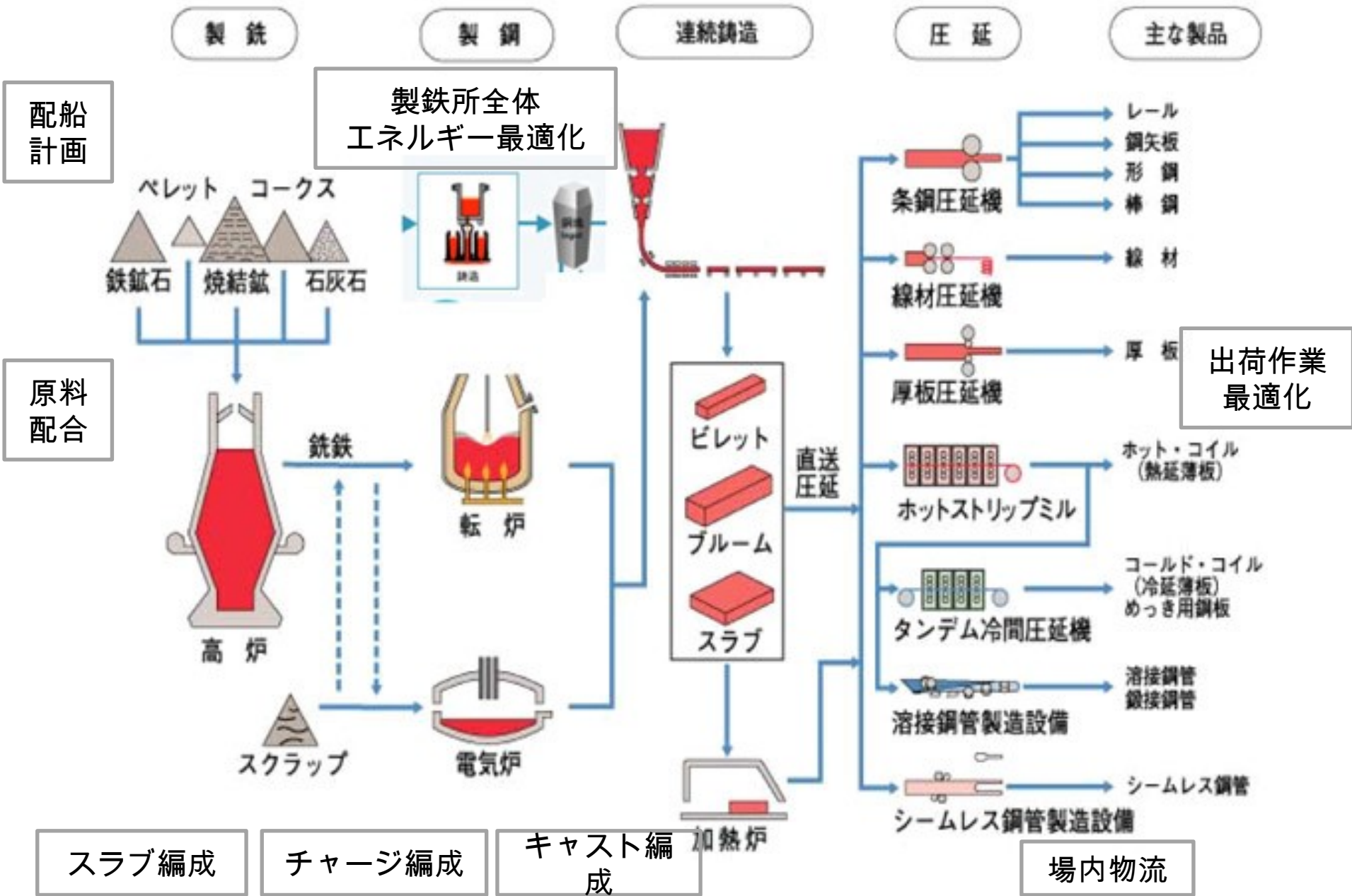
白い部分 (4カ所) の面積
 $1 \times 10 + 3 \times 1 + 2 \times 1 + 1 \times 6 = 21$
 これは $50 \times 1 = 50$ より小さいので
 縦を1縮めることはできない。



この解は最適である！

小長方形のサイズは整数値

製鉄プロセスと主な最適化問題

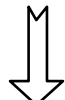


(1) 厚板スラブ編成

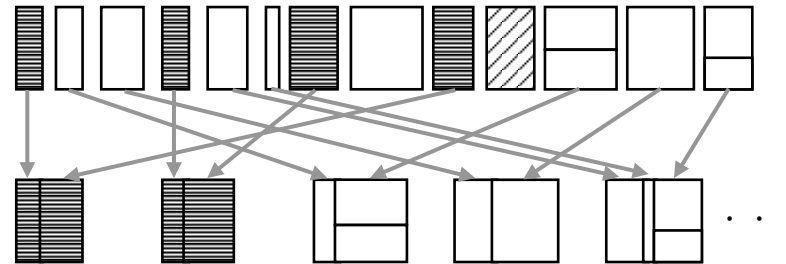
各オーダー部材を制約条件 (20個) に基づき製造ロット (スラブ) にまとめる



構造物部材のオーダー



スラブ



制約条件

- ・ 同じ厚みのオーダーでないと同じスラブにできない。
- ・ 成分が同じ
- ・ 組み合わせたときにスラブの許容サイズになるなど

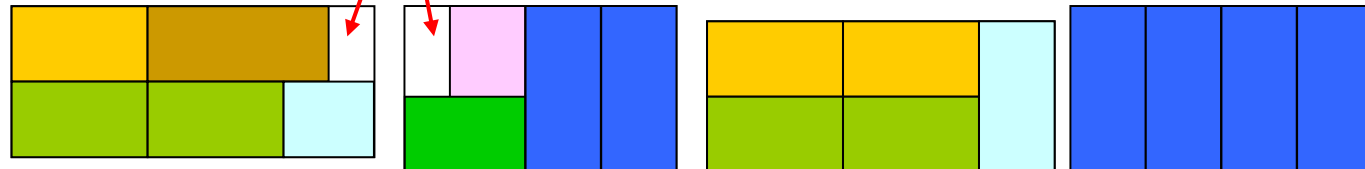
1日のオーダー数6千~12千
200項目以上の条件

同ースラブに組み込み可能なもの
10~1000オーダー

目的関数

1. 歩留まりロス最小
2. 圧延設定時間最小

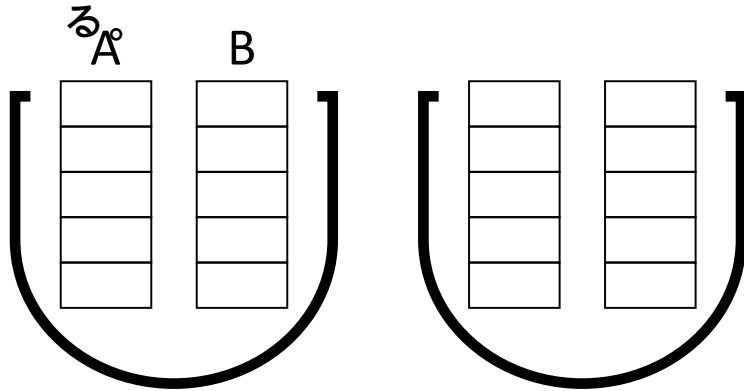
ロス



仮に100オーダーとしても組合せ
 1.27×10^{30} !

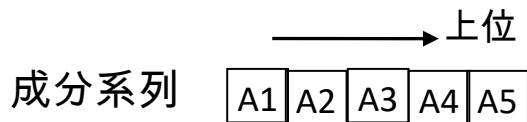
(2) チャージ編成

(1) のスラブ編成で計画したスラブを組合せ、転炉の 1 回ごとのチャージにまとめる
1チャージ：約300トン



制約条件

- ・ 同じ厚みのスラブを組み合わせる。
- ・ 成分系列が同じ。
- ・ 組み合わせたときのチャージ重量が許容範囲以内など



目的関数

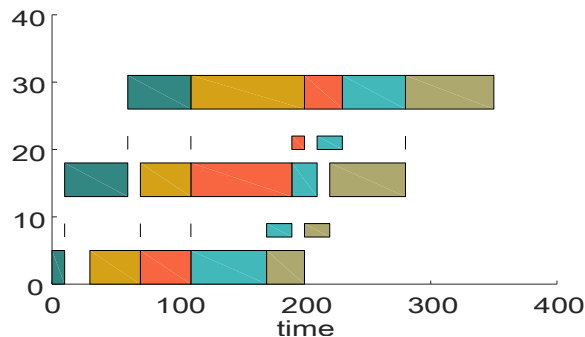
1. チャージ数最小
2. 成分変換コスト最小など



5.4 他産業における最適化

生産ラインのスケジューリング

各ラインにおいて製品をどのような順序で製造すると、最も早くできるか。最も納期遅れが少なくなるか。



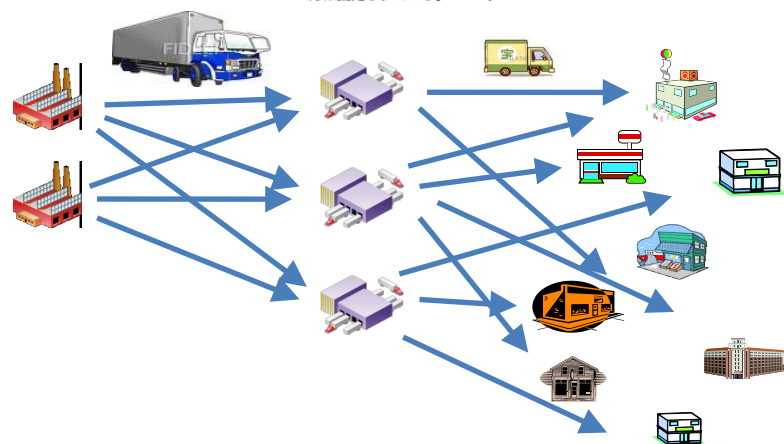
製紙、合成フィルム、薄膜、テープなどにおける取り合わせ

最終製品の小さなロールやシートを原材料の大ロールからどのように切り取ると、最もロスを少なくできるか。



流通業

どこに倉庫を新設したりどこの倉庫を廃止すると流通コストを最小にすることができるか。



非製造業

- ・ 金融のポートフォリオ最適化
- ・ 看護師スケジューリング
- ・ 航空機や鉄道の乗務員スケジューリング
- ・ 列車のダイヤ編成
- ・ Jリーグの試合日程の作成

6 まとめ

私見 1

「数学が経済を動かす」とまでは言わないにしても、数学が現在の社会を様々な分野でかなり支えていることは確かである。

私見 2

日本における数学の貢献度は、決して欧米に見劣りするものではない。

残念な点

量子コンピュータが普及するようになれば、ソルバーの性能は問題にならなくなるかも知れない。

最適化において、性能の良い（高速に解ける）混合整数計画ソルバー（問題を解くソフトウェア）が外国（とくにアメリカ）製である。日本に高性能のソルバーがほしい。

希望

社会における数学の重要性の認知度が上がること。