

# システム制御工学特論

Lecture 1

## 今日の講義予定

1. 自己紹介
2. 講義の注意と講義予定
3. 「システム制御の基礎」との違い
4. 物理現象のモデル化の紹介

## 「制御工学」について

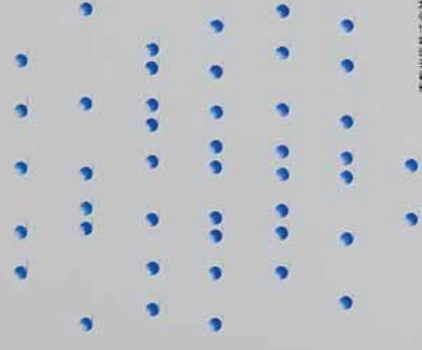
- 担当教員： クルモフ バレリー教授
- 開講学期： 3年春、単位数： 2単位
- 系列： 専門科目、選択
- 参考書： 小郷、美多「システム制御理論入門」実教出版
- 参考書： 野波、西村「MATLABによる制御理論の基礎」東京電機大学出版社

## 参考書

小郷、美多：システム制御理論入門  
実教出版(株)

システム制御理論入門

小郷 美多  
著



実教出版株式会社

## 基本的注意

1. 遅刻しないこと、欠席しないこと。無断欠席は大きく減点する。

2. 宿題の提出遅れは受け取らない。

( $\times$ 切後、解答をホームページに公開するため)

3. 成績評価：中間レポート(40%)、期末試験(60%)

4. 宿題を自力でとくことが重要である。

• URL：<http://shiwasu.ee.ous.ac.jp/control/>

[http://shiwasu.ee.ous.ac.jp/matweb\\_cs/](http://shiwasu.ee.ous.ac.jp/matweb_cs/)

• E-mail [val@ous.ac.jp](mailto:val@ous.ac.jp)

## 「制御工学」科目の受講条件

- 講義中の録画・静止画・録音等は禁止  
本講義資料には不特定多数者には配布が禁止されている著作権で保護された映像等を許された範囲で引用している。
- 本講義の受講者、および、特にクルマモフが許可した者以外への配布は犯罪に問われるので固く禁止する。
- 講義後に質問・相談は、メールまたは対面により

## 2024年度 講義予定 -シラバス通り-

回	開講	内容
1	4/16	制御工学の概要、講義の予定
2	4/23	動的システムと状態方程式
3	5/07	状態方程式の解とシステムの安定性理論
4	5/14	リャプノフ安定性理論
5	5/21	可制御性・可観測性と線形システムの等価変換
6	5/28	レギュレータと同一次元オブザーバの設計
7	6/04	最小次元オブザーバの設計 (その1)
8	6/11	最小次元オブザーバの設計 (その2)
9	6/18	演習
10	6/25	定常誤差と開ループシステムの型およびサーボ系の設計法
11	7/02	サーボ系の設計法Ⅱ
12	7/09	最適レギュレータの設計
13	7/16	非線形制御の導入
14	7/23	現代制御理論の新しい話題
15	7/30	まとめ
16	8/06	期末試験

## 制御理論

	古典制御理論	現代制御理論	新しい制御理論
理論構成	1940年頃	1960年頃	1980年頃から
制御系表現	伝達関数	状態方程式	既約分解表現
モデリング	周波数応答	システム同定	展開： ロバスト制御 適応制御 非線形制御 モデル予測制御
制御系の性質	—	可観測 可制御	
安定性判別	特性方程式、ラウスフルビッツ、根軌跡法、ナイキスト	固有値、リャプノフ関数	
フィードバック制御	補償器：PID制御、位相進み要素、位相遅れ要素	状態フィードバック、最適レギュレータ、状態観測器	
講義	システム制御の基礎	制御工学	

# 制御理論

状態方程式： R.E.Kalman  
 1930年、ハンガリー生、  
 1960年、制御の数学的理論、  
 Kalman フィルター



既約分解： M. Vidyasagar  
 1948年、インド生、  
 1969年、21歳で博士、  
 1972年、24歳、



カナダ Concordia 大学教授

2005年8月 岡山で特別講演

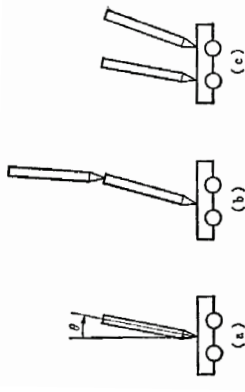
# 制御の必要な例：倒立振り子

倒立振り子：教科書18頁

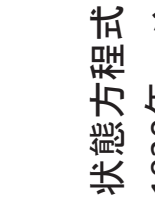
制御目的：安定に倒立

制御系の特徴

1. もともとは不安定
2. 不確かさ
3. 非線形
4. 劣駆動(少ない制御入力でも多くの出力を制御)



様々な倒立振り子



## 不安定系、安定系

不安定系

ほっておくと元の状態から大きく動く



安定系

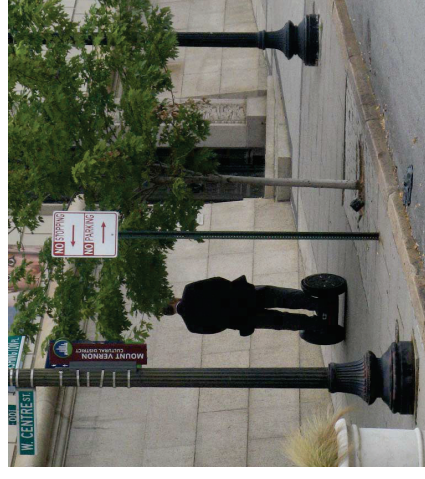
ほっといても元の状態へ戻る



## 制御工学の目的

1. 不安定系を安定系にする
2. 特性の改善

# Segway



# Pythonの特徴

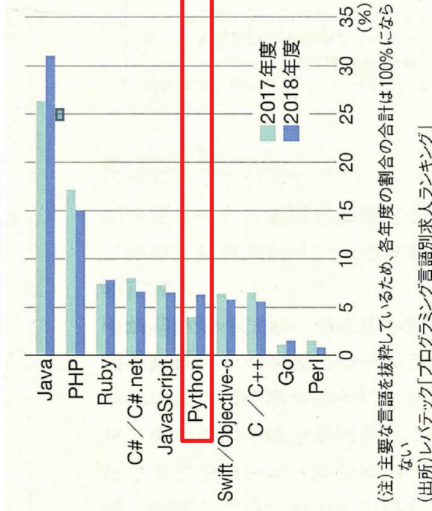
最も新しいプログラミング言語：1991年に最初にリリース  
現在、Version 3.8, 人工知能の開発は主にPython

—プログラミング言語別年収ランキング—

順位	言語	年収中央値 (万円)	最大提示年収 (万円)
1	Go	600	1,600
2	Scala	600	1,300
3	Python	575	1,499
4	Kotlin	575	1,200
5	TypeScript	575	1,200
6	R	574	1,000
7	Ruby	550	1,200
8	Swift	550	1,200
9	Perl	525	1,200
10	C	525	1,000

(注)2018年。公表は10位まで。年収中央値が同じ場合は最大提示年収が高く、求人数が多いものを上位とする  
(出所)求人検索エンジン「スタンプ」

—プログラミング言語別求人案件の割合—



(注) 主要な言語を抜粋しているため、各年度の割合の合計は100%にならない  
(出所)レバテック「プログラミング言語別求人ランキング」

週刊東洋経済2020年1月18日号

# プログラミング言語ランキング (年収)

順位	言語	平均年収 (2022年実績)
1位	Scala	682万9,000円
2位	TypeScript	667万1,000円
3位	Go	659万円
4位	Python3	644万7,000円
5位	Kotlin	644万1,000円
6位	Ruby	638万3,000円
7位	Swift	615万7,000円
8位	C++	609万6,000円
9位	JavaScript	599万4,000円
10位	Java	589万8,000円

(引用元:「プログラミング言語に関する調査 (2022年版)」の結果を発表。平均年収が高い言語、転職で企業ニーズが高い言語など | PR TIMES)

# プログラミング言語ランキング (人気)

順位	言語	評価率 (2023年6月実績)
1位	Python	12.46%
2位	C	12.37%
3位	C++	11.36%
4位	Java	11.28%
5位	C#	6.71%
6位	VisualBasic	3.34%
7位	JavaScript	2.82%
8位	PHP	1.74%
9位	SQL	1.47%
10位	Assembly language	1.29%

(引用元: 2023年6月の TIOBE 指数 | TIOBE)

# Pythonのライブラリ

ライブラリ (Module) の一覧: PyPIで見ることが出来る  
PyPi → 一覧表、開発中も含めて  
> 350,000個 (2022年1月17日) 全て **タダ**

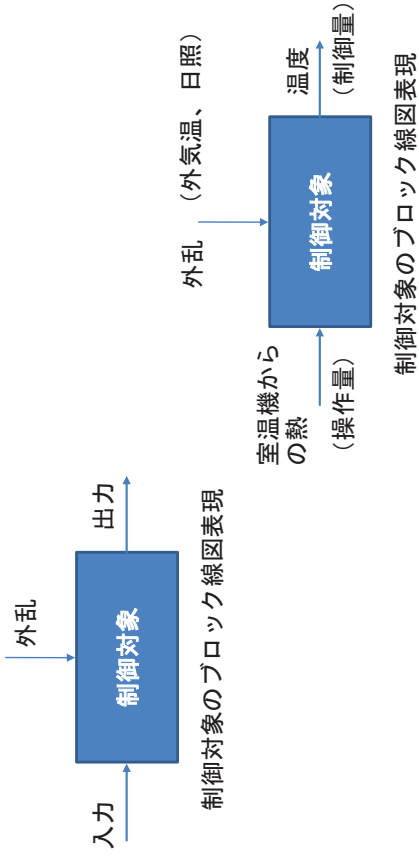
## ライブラリ (Module) の例

- control, control.matlab, slycot 制御工学
- numpy/scipy 数値計算の基本になるライブラリ
- matplotlib 図を描く、`import matplotlib.pyplot` 数式処理
- scikit-learn 機械学習ライブラリ
- TensorFlow 開発した Deep Learning
- jmri, dccpi 鉄道模型の制御



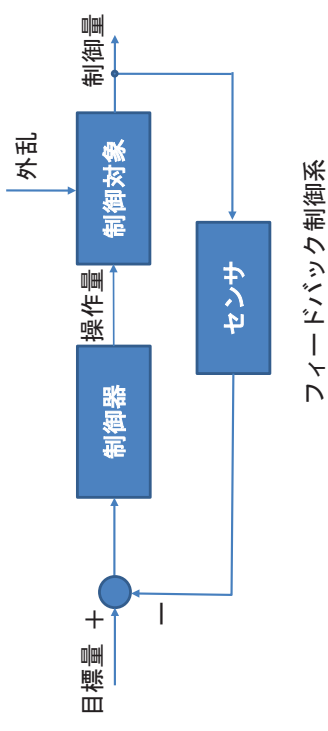
# 制御系の構成および目的

システム表し方：ブロック線図



# フィードバック制御（閉ループ制御）

制御器：目標値と制御量との比較偏差により処理、操作



# フェードフォワード制御（開ループ制御）

コントローラ（制御器）：制御対象の特製が分かっているならば、逆算

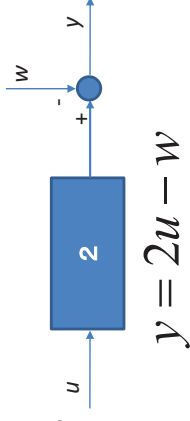


# フィードバック制御の利点

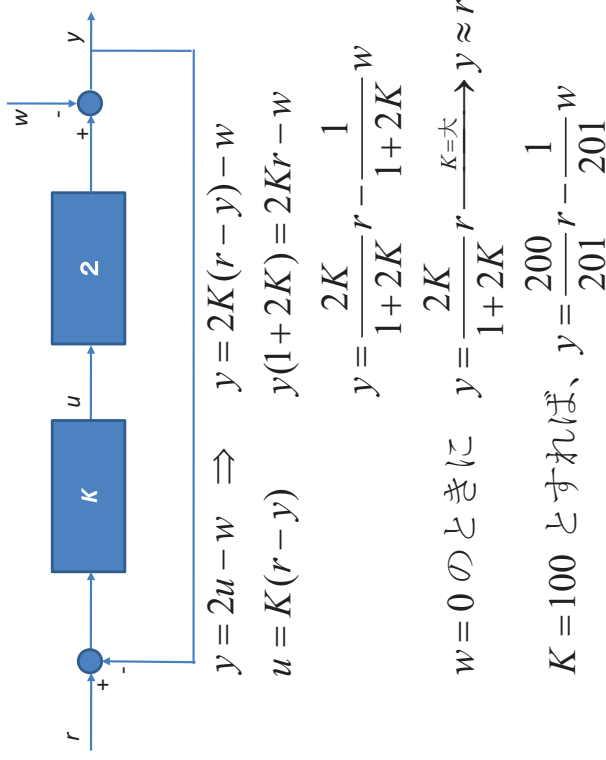
- 制御対象の安定化
- 目標値追従
- 外乱の影響の抑制
- 特性変動による影響の抑制

## フィードバック制御の例題

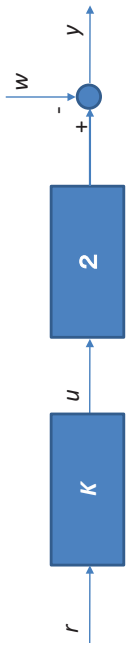
- モータで駆動される小型ボート：  
モータに加わる電流 $u$ [A]の大きさに比例する進行速度 $y$ [m/s]  
(電流 1 Aに対して速度 2 m/sが出るとする)
- 進行方向に対して逆方向に速度 $w$ [m/s]の水が流れるとする。
- 目標速度 $r$ [m/s]とする。



## 例題：フィードバック (FB)



## 例題：フィードフォワード (FF)



$$y = 2u - w$$

$$u = Kr$$

$$K = \frac{1}{2} \text{ (2の逆数)} \Rightarrow y \text{ に } u \text{ 代入する}$$

$$y = 2Kr - w = 2 \times \frac{1}{2}r - w = r - w$$

$$\Rightarrow w = 0 \text{ のときに } y = r$$

## 例題：外乱

目標値： $r = 5 \text{ m/s}$ 、外乱： $w = 3 \text{ m/s}$



$$K = \frac{1}{2} \text{ のとき}$$

$$y = r - w = 5 - 3$$

$$= 2 \text{ m/s (60%の誤差)}$$

$K = 100$  のとき

$$y = \frac{200}{201}r - \frac{1}{200}w$$

$$= \frac{200}{201} \times 5 - \frac{1}{200} \times 3$$

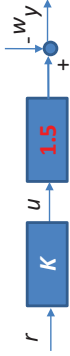
$$= 4.960124$$

$$\approx 4.96 \text{ m/s}$$

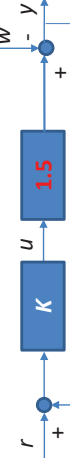
目標値に対する誤差は1%以内

## 例題：特性変動

目標値：  $r = 5 \text{ m/s}$ 、



外乱：  $w=0\text{m/s}$



$K = \frac{1}{2}$  のとき

$$y = 1.5 \frac{1}{2} r = 0.75r$$

$= 3.75 \text{ m/s}$  (25%の誤差)

$K = 100$  のとき

$$y = 1.5u = 1.5K(r - y)$$

$$151y = 150r$$

$$y = \frac{150}{151} r = \frac{150 \times 5}{151}$$

$$= 4.966887$$

$$\approx 4.97 \text{ m/s}$$

目標値に対する誤差は1%以内

## フィードバック制御の利点

- 制御対象の安定化
- 目標値追従
- 外乱の影響の抑制
- 特性変動による影響の抑制

## 制御とは（まとめ）

- 制御とは
- 制御系の構成と制御目的
- フィードバック制御の利点
- 学習目標：
  - 「制御」の重要性を理解する。
  - 「フィードバック制御の」利点を理解する。

## 定常応答(外乱のある場合)

図8.2で示される制御系がある。  
各部の伝達関数は

$$G_1(s) = \frac{K_1}{s(1+sT_1)}, \quad G_2(s) = \frac{K_2}{1+sT_2}$$

であるとし、 $R(s)=R_r/s$ ,  $D(s)=R_d/s$ の入力と外乱が加わるとき定常偏差を求めよ。

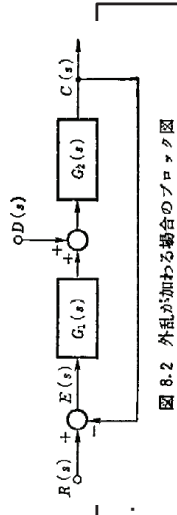


図 8.2 外乱が加わる場合のプロック図

$C(s)=G_2(s)(G_1(s)E(s)+D(s))$ ,  $E(s)=R(s)-C(s)$ の2つの式から $C(s)$ を消去  
 $G_1(s)$ ,  $G_2(s)$ ,  $R(s)$ ,  $D(s)$ を代入して $E(s)$ を求める

最終値の定理を使う  $\rightarrow$  定常偏差  $= e(\infty) = \lim_{s \rightarrow 0} sE(s)$



# 定常応答(外乱のある場合)

## 演習課題

$$e(\infty) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{R_r - \frac{K_2 R_d}{1 + sT_2}}{1 + \frac{K_1 K_2}{s(1 + sT_1)(1 + sT_2)}}$$

$$e(\infty) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{R_r - \frac{K_2 R_d}{1 + sT_2}}{1 + \frac{K_1 K_2}{s(1 + sT_1)(1 + sT_2)}}$$

Diagram illustrating the limit calculation for  $e(\infty)$  as  $s \rightarrow 0$ . The expression is shown as a fraction of two terms. The numerator is  $R_r - \frac{K_2 R_d}{1 + sT_2}$  and the denominator is  $1 + \frac{K_1 K_2}{s(1 + sT_1)(1 + sT_2)}$ . As  $s \rightarrow 0$ , the numerator approaches a finite value (有限), and the denominator approaches a finite value (有限). The overall result is finite (有限).