

フレキシブルリンク システム工学実験2019

氏名： _____

教員：大木健太郎 (ohki@i.Kyoto-u.ac.jp) ,

TA: 上田夏菜, 青木雅奈, 原圭祐 (ske.ta@bode.amp.i.Kyoto-u.ac.jp)

1. 共通課題

1. システム同定 ※ システム同定と制御系設計は独立に行えるので、
分業すること
2. 制御系設計
3. プレゼンについて

2. 自由課題

共通課題の実験内容

1. システム同定

- 与えられた方程式と入出力データからパラメータ推定
【ポイント】何を評価基準としてパラメータ推定するか

2. 制御系設計

- 与えられた数理モデルから、
閉ループ系を安定化させる制御器を設計
【ポイント】どのような制御性能を期待して制御するか

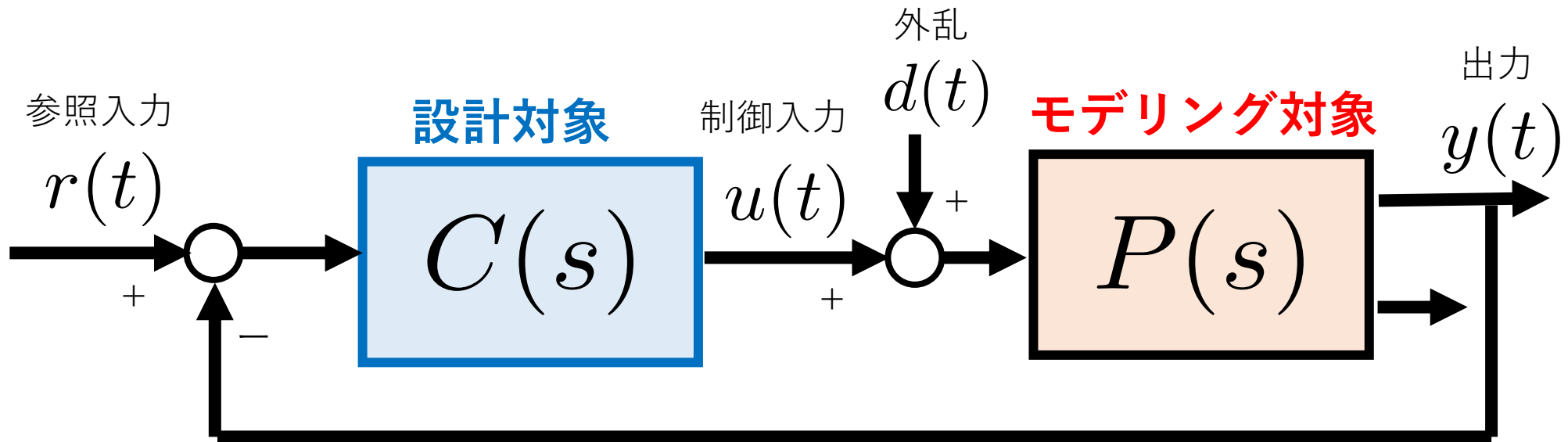


グループプレゼン

- ✓ 日時： 1/10（金） 13:30～
- ✓ テーマ： 未定（近づいたら決めます）
- ✓ 持ち時間： 7分（変更する可能性あり）
- ✓ 対象： 数理コース 2 回生終了程度の知識をもった人
- ✓ 質疑： 発表グループ以外の全員

共通課題における制御目標・課題

- 入力には外乱が生じている



課題

- $d(t) = \sin(4t + \phi)$, $r(t) = 0$ に対して $y(t) \rightarrow 0 (t \rightarrow \infty)$ となる安定化制御器を設計せよ。ただし ϕ は未知とする。
- 安定制御器の中で、何らかの意味で最適な制御器を作成せよ

目次

1. 共通課題

1. システム同定

2. 制御系設計

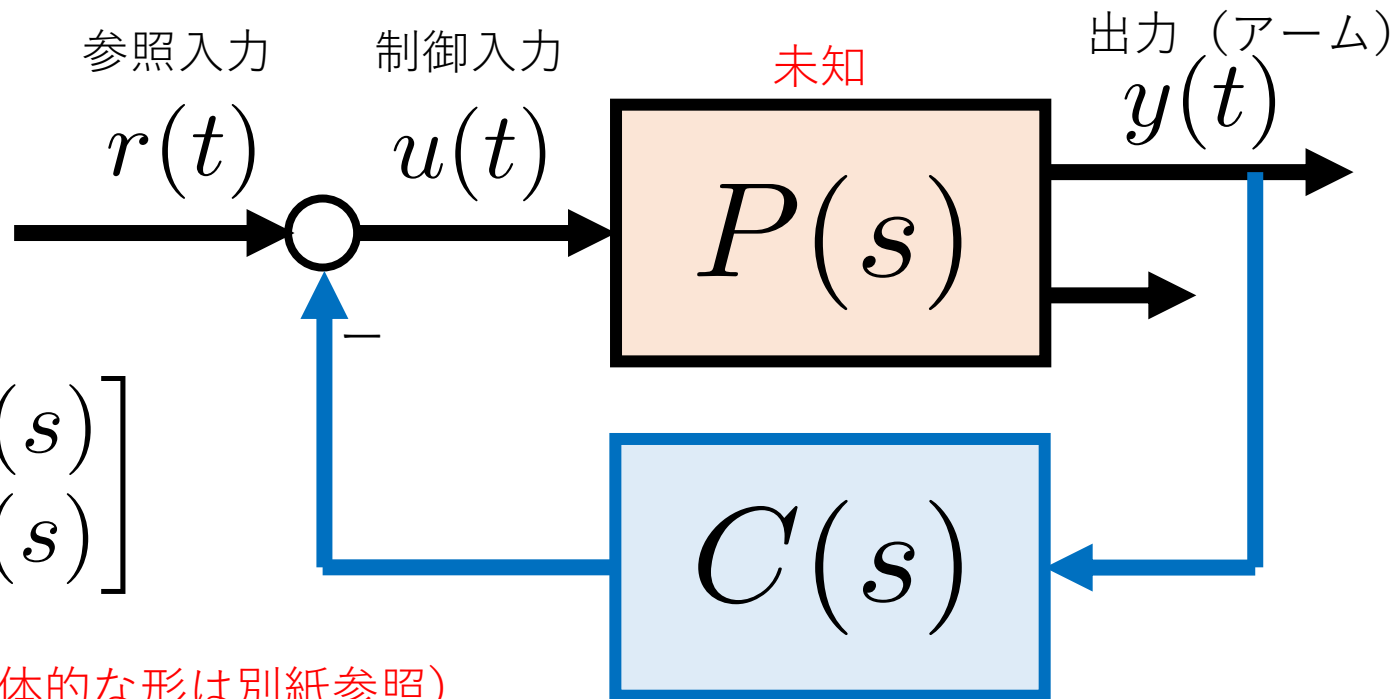
3. プレゼン資料について

2. 自由課題

システム同定

- 第一原理による数理モデリングとパラメータ推定を行う
- 必要なデータを取れるよう、実験計画
- パラメータ推定の**評価基準（コスト関数）**を定める
- 作成したモデルと実機とがどの程度一致するか確認
- 作成したモデルは制御班に渡す

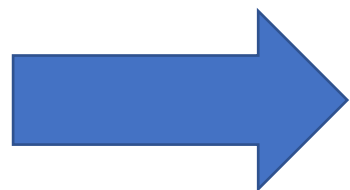
制御対象のモデリング



$$P(s) = \begin{bmatrix} P_{\text{arm}}(s) \\ P_{\text{link}}(s) \end{bmatrix}$$

求めたい伝達関数 (具体的な形は別紙参照)

(安定限界：データに非線形摩擦等の影響が乗る)



$$G(s) := \frac{P_{\text{arm}}(s)}{1 + P_{\text{arm}}(s)C(s)}$$

$P_{\text{link}}(s)$ は同定では使わない

フィードバック制御

$G(s)$ を求めてから、 $P_{\text{arm}}(s)$ を求める

伝達関数の形とパラメータ推定

- 求めたい伝達関数 :
$$P_{\text{arm}}(s) = \frac{b_1 s^2 + b_2}{s(s^3 + a_1 s^2 + a_2 s + a_3)}$$
- 初期の制御器 :
$$C(s) = \frac{K_p s + K_i}{s} \quad K_p = 1.65, K_I = 0$$

微分方程式の係数を求めればよい

- 複数の方法
1. 差分化して時系列データから求める
 2. 線型性を利用して周波数応答から求める
 3. その他

周波数応答

線形システムの特徴：解の重ね合わせ

$$r(t) = \sum_{i=1}^{N_d} g_i \sin(2\pi f_i t + \phi_i)$$

$$\Rightarrow y(t) = \sum_{i=1}^{N_d} |G(j\omega_i)| \sin(\omega_i t + \phi_i + \angle G(j\omega_i)), \quad \omega_i := 2\pi f_i$$

※ 正確には出力が右辺に指数的に漸近収束していく

確認事項

- 本当に線形システムとして扱ってよいか確認する
- フィードバック制御しないとどうなるか確認する

正弦波を入れたときの制御系の応答

ブロックパラメータ: Signal Generator

Signal Generator

各種波形を出力します:
 $Y(t) = \text{振幅} \cdot \text{波形(周波数, } t)$

パラメータ

波形: 正弦

時間 (t): シミュレーション時間を使用

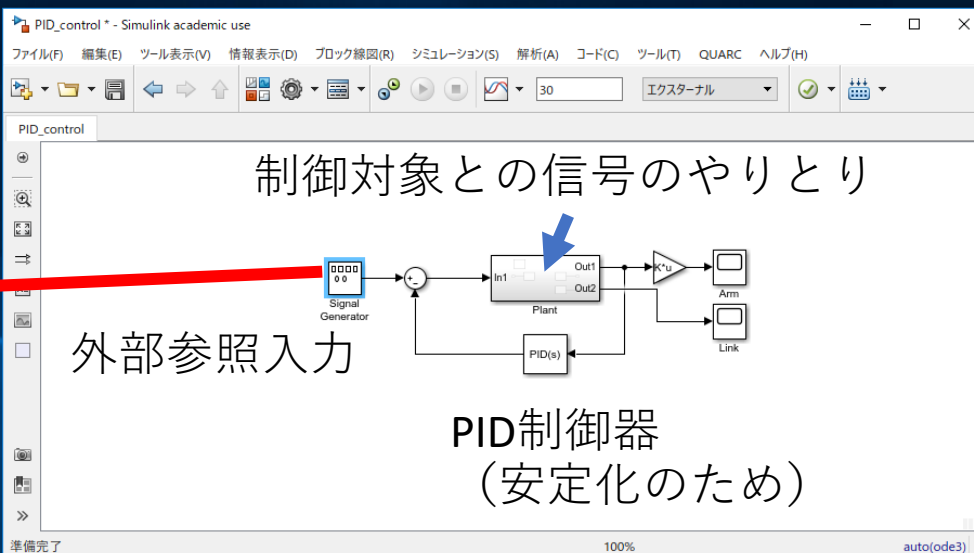
振幅: 10

周波数: 2

単位: Hertz

ベクトル パラメータを 1 次元として解釈

OK(O) キャンセル(C) ヘルプ(H) 適用(A)



Simulink/Sources

Commonly Used Blocks

Continuous

Dashboard

Discontinuities

Discrete

Logic and Bit Operations

Lookup Tables

Math Operations

Model Verification

Model-Wide Utilities

Ports & Subsystems

Signal Attributes

Signal Routing

Sinks

Sources

String

User-Defined Functions

> Additional Math & Discrete

> Quick Insert

> Audio System Toolbox

> Communications System Toolbox

> Communications System Toolbox HDL

> Computer Vision System Toolbox

> Control System Toolbox

Random Number

Repeating Sequence

Repeating Sequence Interpolated

Scenario

Signal Builder

Signal Editor

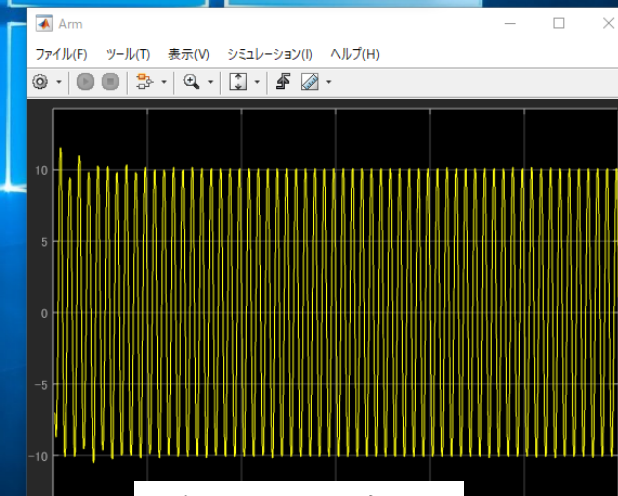
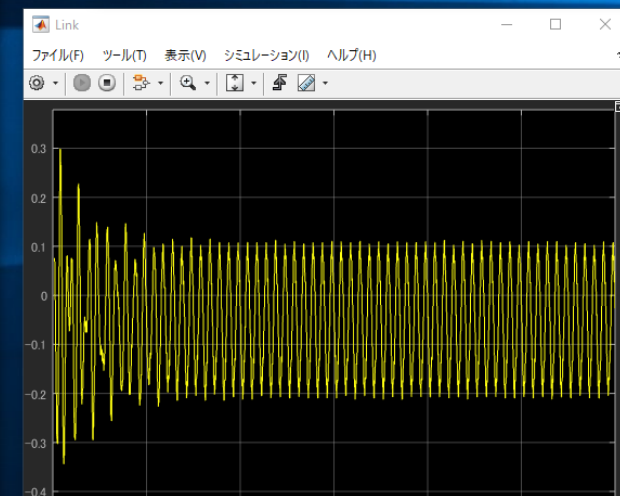
Signal Generator

Sine Wave

Step

Uniform Random Number

Waveform Generator



リンクの先端は、
弱い力で動かすと線形応答にならない

線形的な応答

“Plant”の中身:インプット・アウトプットの設定

The image displays a Simulink workspace with several windows and diagrams. On the left, the 'Simulink' library browser is open, showing various HIL blocks. The main workspace shows a 'HIL Initialization' block with a red circle highlighting the 'QuARC' logo. Below it, the 'Source Block Parameters: HIL Initialize' dialog is open, showing the 'Main' tab with the following settings:

- Board name: HIL-1
- Board type: q2_usb
- Board identifier: 0
- Board-specific options: d0=digital,d1=digital,led=auto,update_rate=normal,decimation=1
- Assume exclusive access to the board

At the bottom of the dialog, a photograph of the 'Quanser Q2-USB' hardware is shown. To the right, two block diagrams illustrate the output types:

- The top diagram shows an 'HIL Write Analog' block (HIL-1) with input 1 and output 0, connected to an 'HIL Read Encoder' block (HIL-1) with input 0 and output 1. This is labeled 'エンコーダ出力 (離散的)' (Encoder output (discrete)).
- The bottom diagram shows an 'HIL Write Analog' block (HIL-1) with input 1 and output 0, connected to an 'HIL Read Analog' block (HIL-1) with input 0 and output 2. This is labeled '歪みゲージ出力 (連続的)' (Strain gauge output (continuous)).

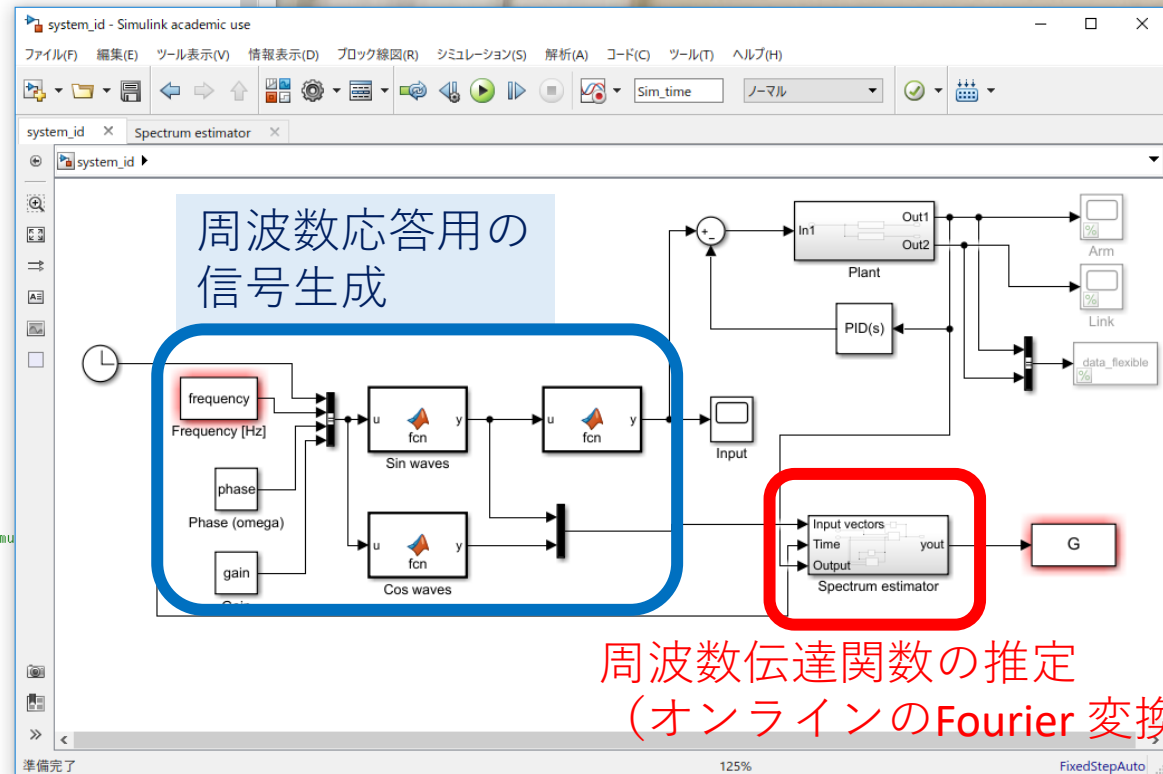
Board type は q2_usb を選ぶこと

同定用データ取得のための Simulink ファイル

```
1 clear all
2 close all
3
4 N_d = 50;% # of sampling f
5 N_for = 20;% 0(N)
6 Sim_time = 1800; % [sec] 0
7
8 %% Setting Simulink
9
10 f_low = -2;
11 f_up = 2.4;% log(250) = 2.3979..., 250Hz is the Nyquist frequency
12
13
14 open_system('system_id')
15 set_param('system_id','StopTime','Sim_time+10')% Set the terminal time
16 dt = 0.002;% sampling period [sec]
17 decim = round(Sim_time/dt);% To set the decimation for the "To Workspace" block
18 % decim = 1; % default
19 set_param('system_id/To Workspace','Decimation','decim')
20
21 % G = zeros(1,2*N_d,2);
22
23 % tentative parameters to build Simulink, updated in the for loop below
24 frequency = rand(N_d,1);
25 phase = rand(size(frequency));
26 gain = 1*rand(size(frequency));
27
28 qc_build_model;% Quarc command to build a model
29
30
31 % Data accumulation
32 k=1:N_for
33 k
34
35 freq_range = (f_up - f_low)*rand(N_d,1) + f_low;% from 10^f_low Hz to 10^f_up Hz.
36 % Note: Sim_time should be set more than 50/(10^f_low) in practice, morethan 10/(10^f_low) sec for simu
37 freq_range = sort(freq_range);% sorting
38 frequency = 10.^freq_range;
39 % frequency = 10.^[-1:0.2:1];% if specific
40 phase = rand(size(frequency));
41 gain = 1*rand(size(frequency));
42
43 qc_start_model;% Quarc
44 % set_param('system_id','SimulationCommand', 'start')% To start Simulink simulation
45 pause(Sim_time + 2);
46 set_param('system_id','SimulationCommand', 'stop')
47 % Obtain the frequency response
48 pause(1)
49
50
```

取得したい情報によって変更が必要なパラメータ
(N_d , N_{for} , sim_time , f_{low} , f_{up})

実時間処理には限界がある
⇒ やりたい実験を別々に行い
分割してデータ取得

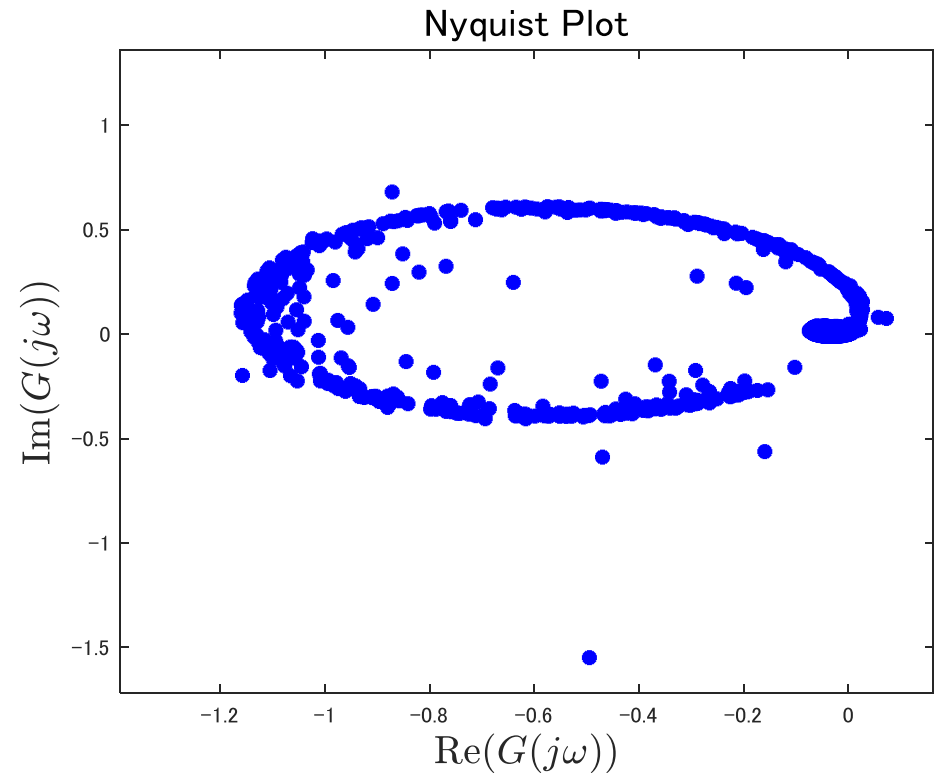
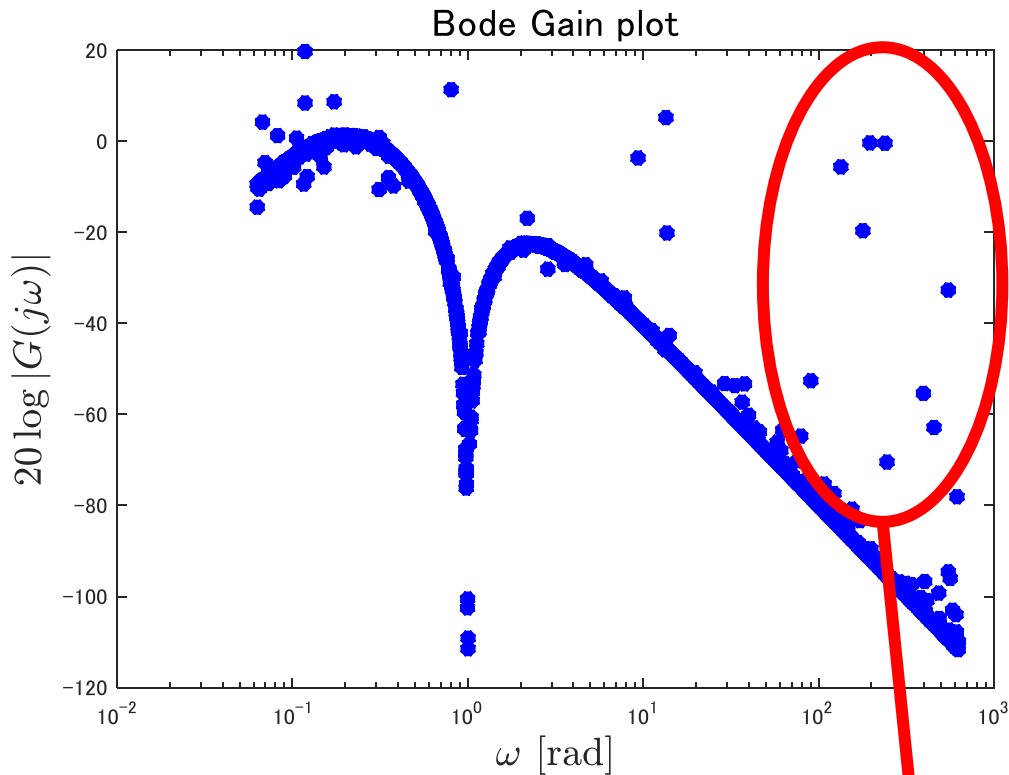


得られるデータ例

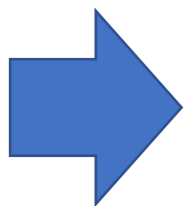
各50点の周波数データで30分間データ取得,
20回やった結果



角周波数, ゲイン, 位相の情報が保存

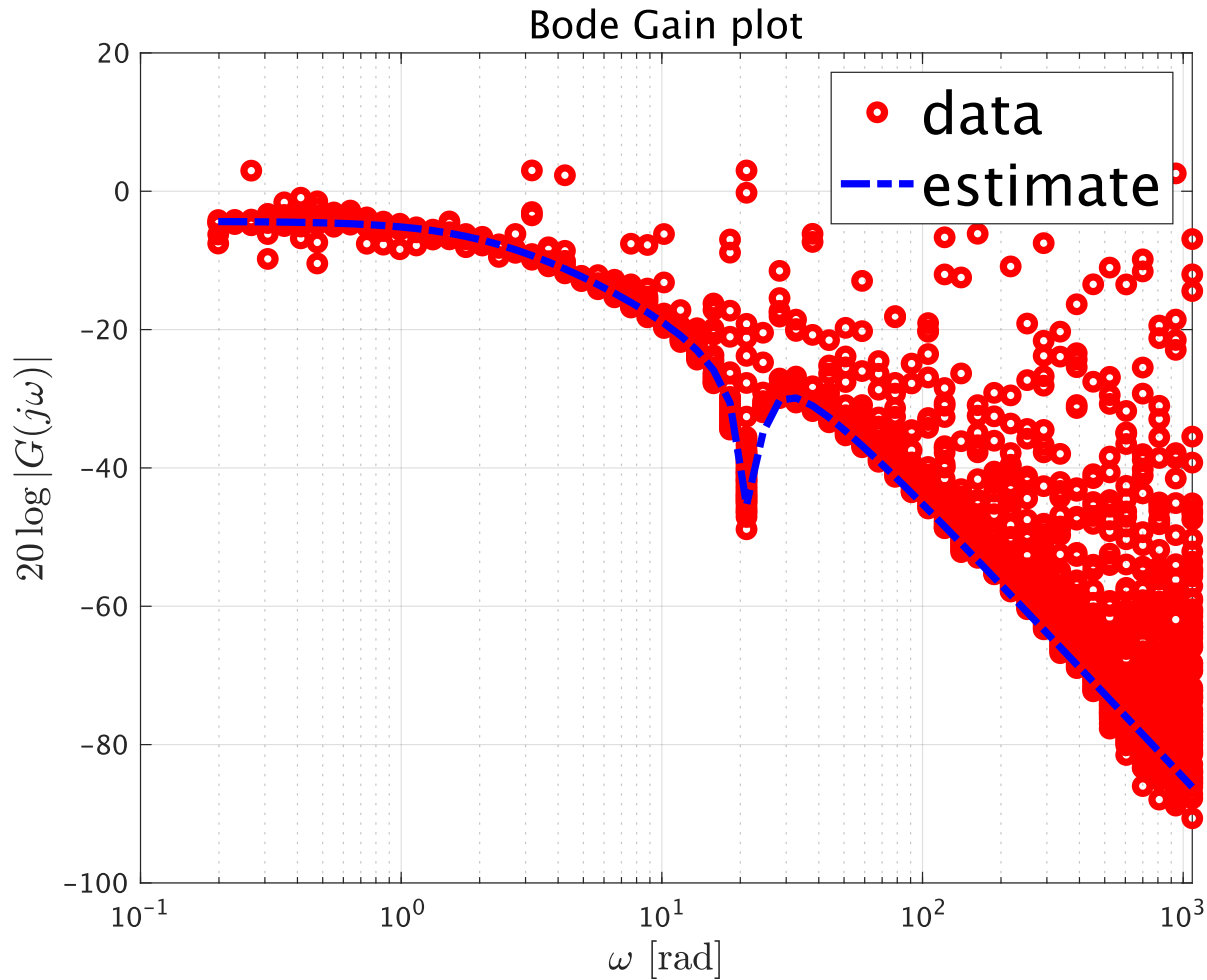


外れ値は工夫 (手動 or ロバスト統計) して除去する



ここから伝達関数 $P(s)$ を推定

推定結果の例



1. データを統計処理
(データの不要物を除去)
2. パラメータ推定

目次

1. 共通課題

1. システム同定

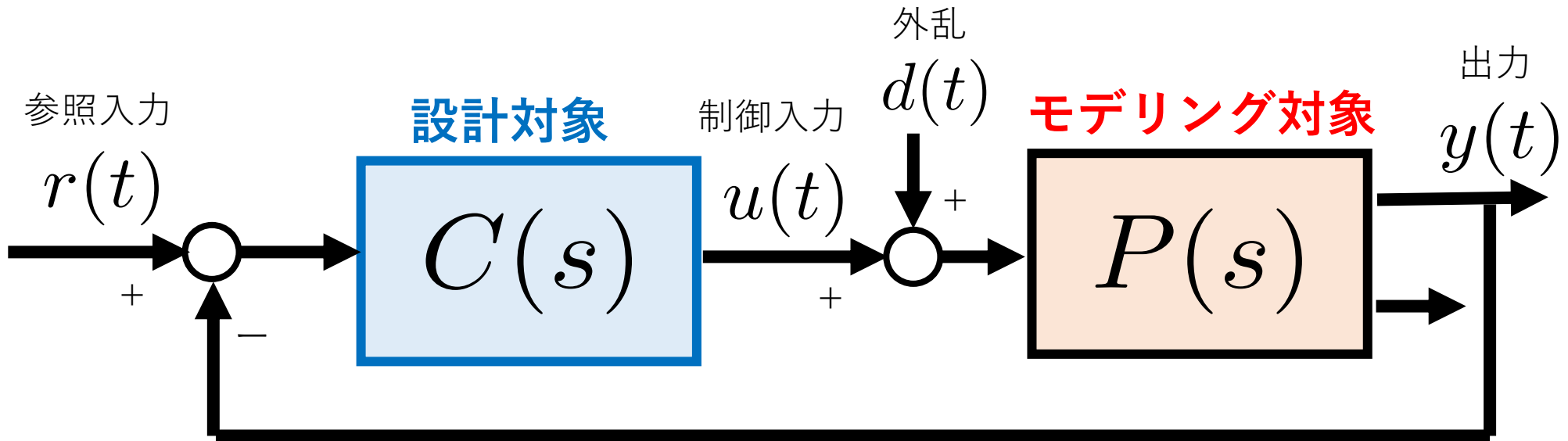
2. 制御系設計

3. プレゼン資料について

2. 自由課題

制御目標・課題（再掲）

- 入力には外乱が生じている



課題

- $d(t) = \sin(4t + \phi)$, $r(t) = 0$ に対して $y(t) \rightarrow 0 (t \rightarrow \infty)$ となる安定化制御器を設計せよ。ただし ϕ は未知とする。
- 安定化制御器の中で、何らかの意味で最適な制御器を作成せよ

制御系設計

- 制御器は数理モデルを引数として伝達関数に値を取る関数
 - 任意の**与えられた伝達関数**から、次を満たす制御器を作れるようにしておく
 - 閉ループ系を**安定化**する制御器
 - **制御性能**（スピード、モデル化誤差に強いなど）のよい制御器
 - **簡単な**制御器（スパーズ、PIDなど）
- シミュレーションは、Scilab を推奨する
 - Simulink と似た機能 Xcos がある
 - 制御系のライブラリが豊富（例えば <http://programmer-life.net/> を参照。他にもたくさんある。）
- 同定班からモデルが与えられたら、作成済の制御系設計法を用いて、制御器を作成する

課題の条件を満たすために

外乱から出力までの伝達関数: $G(s) := \frac{P(s)}{1 + C(s)P(s)}$

Laplace の最終値定理の条件を満たすように制御器を設計

$$\lim_{t \rightarrow \infty} y(t) \stackrel{\text{red}}{=} \lim_{s \rightarrow 0} sY(s) \stackrel{\text{blue}}{=} 0$$

それぞれの等号成立には何を満たさなければいけないか？
適切な制御器の条件は？

満たさない例

$$y(t) = e^t \iff Y(s) = \frac{1}{s-1}$$



$$\lim_{t \rightarrow \infty} y(t) = \infty, \quad \lim_{s \rightarrow 0} sY(s) = 0$$

Scilab での安定化制御器作成例

```
s=%s;  
P=1/(s-1);
```

```
Np=10;  
Ni=10;
```

```
K_antei = zeros(Np,Ni);  
C_antei = zeros(Np,Ni);
```

```
for ki=1:Ni
```

```
  for kp=1:Np
```

```
    C = (10^(0.1*ki))/s + (10^(0.1*kp));
```

```
    G=1/(1+P*C);
```

```
    kon = roots(G.den);
```

```
    if max(real(kon))<0
```

```
      K_antei(kp,ki)=1;
```

```
      C_antei(kp,ki)=C;
```

```
    end
```

```
  end
```

```
end
```

PI制御器の中から
安定化制御器を探索
Pを変更すれば

※ Scilab 6.0.1 は macOS Mojave に対応していないので注意

Python では後述の本の内容を駆使して作成
(多項式の係数の計算と根の求め方がわかれば書ける)



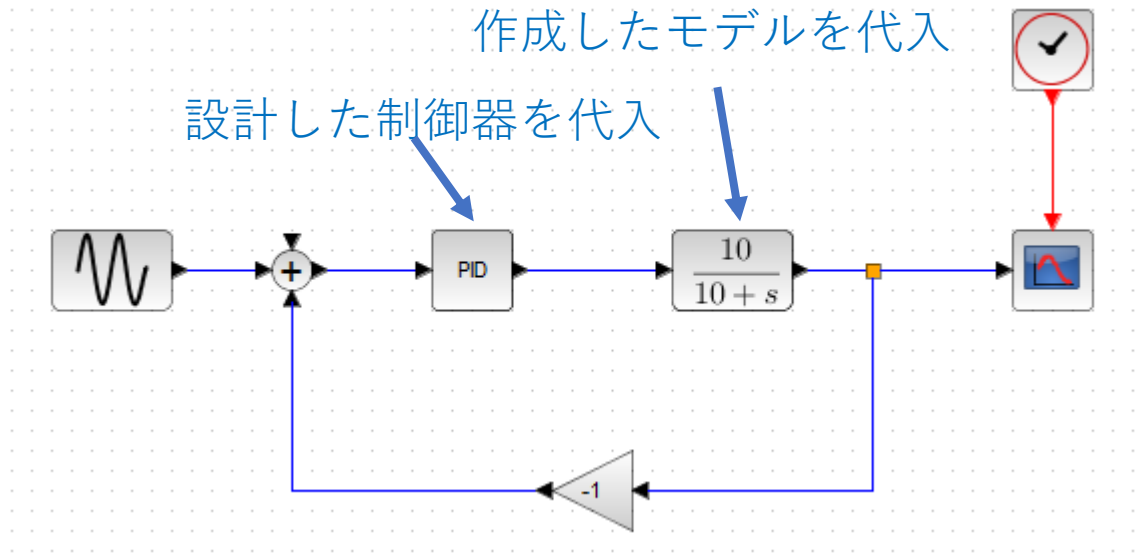
- ✓ 制御性能を決めるコスト関数を作成
- ✓ 安定化した制御器の中で、最適な制御器を求める

} 安定なゲインの組み合わせを記録

```
[index_kp,index_ki]=find(K_antei); 安定となる制御器のインデックスを記録
```

Scilab でのシミュレーション例

コマンド「Xcos」で Simulink と似た環境を作成し，制御器の性能を試す



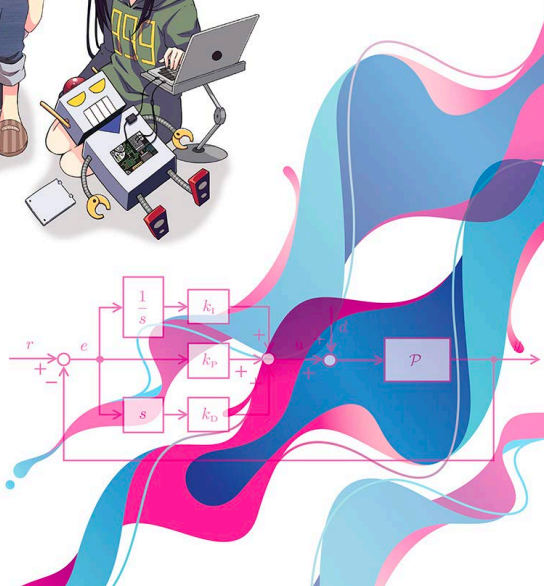
Python の場合

この本↓参照（環境の構築からコードまで載ってる）



環境構築は下記 リンクの「演習環境の構築」を参照

<https://y373.sakura.ne.jp/minami/pyctrl>



目次

1. 共通課題

1. システム同定

2. 制御系設計

3. プレゼン資料について

2. 自由課題

プレゼン資料について

• 意識すること

- プレゼンする対象は誰か？
- 「説明」ではなく「主張」をする
細かい数式や式変形は不要
- 1枚のスライドに情報を詰め込みすぎない
- 目的, 手法, 結果, 結論をストーリーにする

• スライドデザイン

- 4:3あるいはA4サイズで作成すること
(使用するプロジェクタの仕様では16:9は非推奨)
- 下記「見やすいプレゼン資料の作り方」を参照すること
<https://www.slideshare.net/yutamorishige50/ss-41321443>

目次

1. 共通課題

1. システム同定
2. 制御系設計
3. プレゼン資料について

2. 自由課題

自由課題

- フレキシブルリンクのデータを用いて、自分で設定した課題を行う
 - テーマは自由（例：時系列解析，スパースモデリング，ロバスト制御，）
 - 工作もできる範囲（後続の班が支障なく実験機を扱える）でやってよい（例：けん玉，）
 - グループでやっても個人でやってもよい
- 共通課題で求めたモデルは利用してよい
- 自由課題の内容を各自レポートで提出
 - 指定した tex ファイルで作成すること
 - グループでやる場合も，レポートは各自で書く
 - レポートについては別途指示