

# 高校数学でマスターする 制御工学

—— 本質の理解から Mat@Scilab による実践まで ——

博士（工学）小坂 学 著

コロナ社

# まえがき

制御工学でなにができるのか？ そのためになにをすればよいのか？ その手順・方法は？ 制約は？ その答えは本書の【わかる編】にあります。

つまり，【わかる編】は制御工学の使い方がわかるマニュアルのようなものです。このマニュアルがあれば，とりあえず制御工学を使うことができるようになります。しかし，それをしっかり納得するためには，理論的裏付けが必要です。この理論的裏付けは，【ナットク編】でしっかり説明します。しかも，その説明は，高校の数学で十分理解できるものです。そして最後に，実際の制御工学の応用例が【役立つ編】にあります。そこでは現場の泥臭い制御設計をシミュレーションで実感できます。

以上のように，本書は3編に分かれて構成されています。

- (1) 【わかる (=方法手順) 編】：制御工学でできること・やりたいこと・解析設計手順と方法のマニュアル
- (2) 【ナットク (=理論証明) 編】：高校数学で理解できる【わかる編】の理論的裏付け
- (3) 【役立つ (=応用例) 編】：【わかる編】のマニュアルに沿った設計例 (MATLAB を利用)

これら3編を通して，制御工学をしっかりと自分のものにしてほしいと思います。

筆者は，企業の制御技術者として10年間，大学の制御工学の教員として10年以上の間，制御工学の研究と教育を続けています。この経験を生かして，わかりやすく，納得でき，そして企業の現場で役立つことを目指して本書を執筆しました。高校数学の知識で制御工学を理解できるように工夫し，懇切丁寧な説明を心がけました。また，式番号や図番号を参照するときは，その式や図が

載っているページ番号も並記しています。企業の現場で役立っている実例を示しながら、実際のモノのイメージが頭に浮かび、物理的な意味を把握できるよ  
うにしっかり制御工学を説明しています。

実際の制御系設計では多くの場合、MATLAB（マトラブと読む）という制  
御系 CAD ソフトが使われています。本書でも MATLAB の使い方を紹介しま  
す。MATLAB に似たフリーソフトとして、SCILAB（サイラブと読む）があ  
ります。これに Mat@Scilab（マト・アト・サイラブと読む）というフリーソ  
フトを組み合わせると、MATLAB の多くの関数を無料で実行できます。本書  
では、これも活用して実際にシミュレーションを行い、制御を実感できるよ  
うにしています。

なお、出版にあたって多大なご協力をいただいたコロナ社の関係者の方々に  
厚く御礼申し上げます。

2012 年 6 月

小坂 学

# 目 次

## —— Part I 【わかる編】 ——

### 1. 制御とはなにかを「わかる」

1.1 制 御 と は .....	1
1.2 制御技術者の仕事 .....	2
1.3 自動車の運転と制御 .....	3
1.4 制 御 の 目 的 .....	4
1.5 制御系設計とは .....	9
1.5.1 制御仕様の決定 .....	9
1.5.2 ブロック線図による制御対象と制御器のつながりの図的表現 .....	10
1.5.3 制御対象の把握とモデル化 .....	11
1.5.4 制 御 器 の 設 計 .....	12
1.5.5 制御仕様の実験による検証 .....	12

### 2. 制御システムの解析を「わかる」

2.1 システムとは .....	13
2.1.1 静的システム .....	14
2.1.2 動的システム .....	15
2.2 システムをわかりやすい図で表現するブロック線図 .....	16
2.2.1 ブロック線図とは .....	16

2.2.2	ブロック線図を書くときのルール	16
2.2.3	ブロック線図から伝達関数を求める手順	18
2.2.4	2自由度制御系のフィードバック制御とフィードフォワード制御と制御性能	20
2.3	ラプラス変換によるシステム解析	25
2.3.1	ラプラス変換とは	26
2.3.2	微分がかけ算に置き換わる	28
2.3.3	微分方程式から伝達関数を求める	29
2.3.4	制御工学でよく扱う関数のラプラス変換	31
2.3.5	微分方程式を解く	35
2.3.6	極による安定性・速応性・最終値の解析	40
2.3.7	制御対象に対する仮定	52
2.3.8	周波数特性によるシステム解析	56
2.3.9	安定解析	67

### 3. 制御対象の把握を「わかる」

3.1	モデル化とは	82
3.2	代表的なシステムの性質	83
3.2.1	比例要素	85
3.2.2	微分要素	85
3.2.3	積分要素	86
3.2.4	一次遅れ系	87
3.2.5	二次遅れ系	94
3.2.6	零点をもつ要素	103
3.2.7	むだ時間要素	105

## 4. 制御器の設計を「わかる」

4.1 制御系の性能を表す制御仕様とは .....	112
4.1.1 ステップ応答でわかる制御仕様 .....	112
4.1.2 ボード線図でわかる制御仕様 .....	113
4.2 制御仕様を満足させる制御器を設計するには .....	114
4.2.1 目標値応答特性を良くする 2 自由度制御 .....	114
4.2.2 定常特性を良くする内部モデル原理 .....	116
4.3 最も広く使われている PID 制御とは .....	117
4.3.1 P 制 御 .....	118
4.3.2 PI 制御と位相遅れ補償 .....	126
4.3.3 PD 制御と位相進み補償 .....	129
4.3.4 PID 制御と位相進み遅れ補償 .....	133
4.3.5 PID 制御の目標値応答特性の改善策 .....	134
4.4 PID 制御を設計するには .....	137
4.4.1 試行錯誤による調整 .....	137
4.4.2 限界感度法 .....	139
4.4.3 内部モデル制御 (ラムダチューニング) .....	141
4.4.4 極配置法 (部分的モデルマッチング) .....	145
4.4.5 ル ー プ 整 形 .....	149

## —— Part II 【ナットク編】 ——

### 5. 【わかる編】を理論的裏付けして「ナットク」する

5.1 高校の数学とその応用を「ナットク」する .....	154
-------------------------------	-----

5.1.1	アークタンジェント $\left(\theta = \tan^{-1} \frac{b}{a}\right)$ の $a$ と $b$ の符号と $\theta$ の範囲	154
5.1.2	$\log_{10} x^a = a \log_{10} x$ の証明	155
5.1.3	$\log_{10} xy = \log_{10} x + \log_{10} y$ の証明	155
5.1.4	$\lim_{i \rightarrow \infty} \frac{x^i}{i!} = 0$ の証明	156
5.1.5	テイラー展開	156
5.1.6	オイラーの公式 $e^{j\theta} = \cos(\theta) + j \sin(\theta)$ の証明	160
5.2	2章の制御システムの解析を「ナットク」する	161
5.2.1	ブロック線図のフィードバック接続の公式の証明	161
5.2.2	閉ループ系の伝達関数 $G_{yr}$ , $S$ , $T$ がすべて同じ特性方程式をもつことの証明	165
5.2.3	ラプラス変換の公式の証明	166
5.2.4	さまざまな関数のラプラス変換の証明	169
5.2.5	部分分数展開で便利な留数定理の証明	173
5.2.6	極による安定判別が $n$ 重解の極でも成り立つことの証明	174
5.2.7	複素数の極が複素共役の対をなすことの証明	175
5.2.8	定数×信号または定数×信号の時間微分の項だけをもつ微分方程式で表されるシステムが線形時不変系であることの証明	176
5.2.9	$G(s)$ からゲイン $K$ と位相 $\phi$ を求める公式の証明	177
5.2.10	$G(-j\omega) = \overline{G(j\omega)}$ の証明	179
5.2.11	ラウス・フルビッツの安定判別の証明	181
5.2.12	ナイキストの安定判別の証明	184
5.3	3章の制御対象の把握を「ナットク」する	189
5.3.1	微分要素 $s$ と積分要素 $\frac{1}{s}$ のボード線図の折れ線近似の作図	189
5.3.2	一次遅れ系	190
5.3.3	二次遅れ系	192
5.3.4	零点の性質の証明	196
5.3.5	むだ時間要素のボード線図の性質の証明	197

5.4	4章の制御器の設計を「ナットク」する	197
5.4.1	2自由度制御	197
5.4.2	内部モデル原理の証明	199
5.4.3	PID 制御	200
5.4.4	PID制御の設計	203

## —— Part III 【役立つ編】 ——

### 6. MATLAB を活用した制御系設計を行って「役立つ」

6.1	MATLABとは	211
6.2	制御対象を把握しよう	212
6.2.1	物理法則でモデル化しよう	212
6.2.2	ステップ応答で制御対象を把握しよう	214
6.2.3	ボード線図で制御対象を把握しよう	216
6.3	PID 制御を設計しよう	218
6.3.1	積分項と微分項の役割を確かめよう	218
6.3.2	極配置法による伝達関数を利用した PID 制御器を設計しよう	219
6.3.3	ループ整形によるボード線図を利用した PID 制御器を 設計しよう	222
	引用・参考文献	225
	索引	226



# — Part I 【わかる編】 —

## 1 | 制御とはなにかを「わかる」

この章では、自動車の運転を例として、制御とはなにかを理解しよう。

### 1.1 制 御 と は

制御とは、機械などを自動的にうまく操る技術である。例えば

- (1) 自動ドアの前に人が立つと 自動的に 開く。
- (2) エレベータの行き先ボタンを押すと 自動的に 行き先フロアに移動する。
- (3) 自動販売機のボタンを押すと 自動的に 商品を出す。
- (4) エアコンを室温 25°C に設定すると 自動的に 室温を 25°C に調節する。
- (5) 無人電車を時速 60 km に設定すると 自動的に 速度を時速 60 km に調節する。
- (6) 追尾ミサイルを敵機に発射すると 自動的に 追いかけてぶつかる。
- (7) ジェット機のオートパイロットは 自動的に ジェット機を操縦する。

このように、制御はあらゆる機械に使われている。これらのうち (1)～(3) は、スイッチを組み合わせて、それらをあらかじめ決めておいた順序で自動的にオン・オフさせるものであり、「シーケンス制御」という。本書では取り扱わないが、シーケンス制御については多くの参考書がある。(4)～(7) は、スイッチだけではうまく操ることができない。例えば、(4) のエアコンは、室温を 25°C に維持するために時々刻々と室温をチェックし、どれだけ冷やすか、あるいは暖めるかを調節し続けなければならない。(5) の無人電車は、急な坂道を上ると

きも、満員になって非常に重たくなっても、現在の速度を時速 60 km に保つように、時々刻々と自動的にアクセルやブレーキを調節し続けなければならない。このように時々刻々と温度や速度などを監視して時々刻々と調整し続ける制御をフィードバック制御（または閉ループ制御）と呼ぶ。本書では、このフィードバック制御を取り扱う。フィードバック制御はロボットなどの多くの機械に取り入れられており、例えば自動車の中だけでも、オートマ、パワステ、アクティブサス、カーエアコンなど多くのものに使われている。

## 1.2 制御技術者の仕事

自動車やロボットなどの機械を設計するためには、機械工学や電気電子工学といった多くの科目の知識が必要である。例えば子犬型愛玩ロボットを作って売ることを考えよう。まず、商品のおおまかな品質、価格、発売日（QCD: quality, cost, delivery）などを決める。これを商品のコンセプト立案という。子犬型愛玩ロボットの品質としては、子犬程度の大きさ、かわいい外見、飼い主の声に反応することなどがあるだろう。つぎに、もっと具体的に詳しく数値で表せる性能（仕様という）を決める。例えば、脚などの寸法、歩行の動作や声に反応する情報処理方法などだ。それらの仕様を設計するためには、機械工学や電気電子工学といった多くの科目の知識が必要である。仕様とそれらに関連する科目の例を表 1.1 に示す。ロボットを作るために多くの科目の知識が必要になることがわかるだろう。これらの科目のうち、本書では制御工学について学ぶ。

表 1.1 子犬型愛玩ロボットの仕様と関連する科目

仕 様	関連科目
脚などの寸法	製図, 設計, 材料力学
脚などの材質	材料工学, 材料力学
歩行の動作	制御工学, 電気電子回路
声に反応する情報処理	C 言語プログラミング, 論理回路

### 1.3 自動車の運転と制御

自動車を時速 60 km 一定で運転することを考えよう。ドライバは君だ。運転中に君は時速 60 km になっているかをチェックするために、スピードメータを見るだろう。このように制御結果を監視することをフィードバックという。そして、時速 60 km より遅ければ、アクセルを踏み込んで加速するだろう。逆に時速 60 km を超えていれば、アクセルを緩めて減速するだろう。この調節を時々刻々とし続けると、時速 60 km 一定で運転することができる。この調節を人の代わりにマイコンなどにやらせると自動運転が行える。じつは、このようにする調節こそがフィードバック制御の原理なのだ。人間の代わりにコンピュータがこの調節を行うことで、無人で自動運転を行えるのだ。このあと本書を読み進むと、途中で難しくなり、なにをやっているのかわからなくなってしまうかもしれない。しかし、この原理を忘れないでいてほしい。今後の難しい話は、この原理をうまく活用するための理論である。

工学にはさまざまな専門用語があり、それを表す記号として習慣的によく使われるものがある。例えば電圧として  $V$ 、力として  $f$  をよく使う。自動車の運転を例として、制御工学の専門用語とそれを表す記号を紹介しよう。図 1.1 に示すように、運転手は道路標識の制限速度と現在のスピードを見て、アクセルやブレーキを踏み、自動車のスピードを制御している<sup>†</sup>。

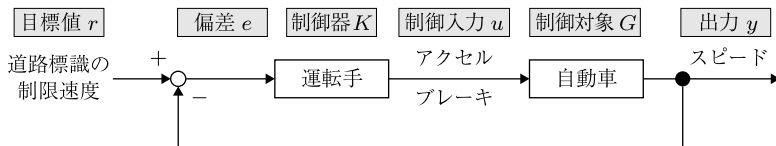


図 1.1 自動車の運転と制御工学の専門用語

ここでは自動車を操っているが、この操られるモノを制御対象（またはプラント）といい、その記号には  $G$  を使うのが通例である。 $G$  のほかに  $P$  も制御

<sup>†</sup> 実際には、交通ルールを守り、安全に運転しなければならない。

対象を表す文字としてよく使われる。自動車を運転する運転手、つまり制御対象を操るモノを制御器（またはコントローラ、調節器）といい、その記号として  $K$  を使うことが多い。 $K$  の代わりに  $C$  を使うこともある。操りたいのはスピードである。この「操りたい量」を制御量（または出力、出力応答） $y$  という。運転手は自動車を操るためにアクセルの開度を調節する。このように制御対象を操るために「調節する量」を制御入力（または操作量） $u$  という。運転手は道路標識を見て目標速度を知る。このように「目標とする量」を目標値（または目標入力） $r$  という。制御の目的は、スピードが目標速度に一致すること、つまり出力が目標値に一致することである。出力  $y$  と目標値  $r$  との差を偏差  $e$  ( $= r - y$ ) という。フィードバック制御器は、時々刻々と出力  $y$  を監視して、制御入力  $u$  を適切に調節している。

## 1.4 制御の目的

制御の目的は、理想的には出力  $y$  を目標値  $r$  に完全に一致させること、すなわち

$$y = r \quad (1.1)$$

とさせることだ。しかし現実には、これは不可能である。それはなぜか。もう一度、自動車の運転を考えてみよう。出力  $y$  はスピード、目標値  $r$  は道路標識に書かれた制限速度、例えば時速 60 km である。がんばって時速 60 km で運転できたとしよう。そのときは  $y = r$  となっているが、それはおよそ、あるいは一瞬の話である。つまり、60.00 km のように小数点以下 2 桁まで誤差のない正確な時速で長時間運転できる人はいないだろう。さらに、図 1.2 に示すように、向かい風が吹いたときや、上り坂になったときは、速度は低下してしまうだろう。これは減速させる力が加わるため、このように外部から制御を乱したり邪魔したりする力を、がいらん外乱という。運転の場合は外部から加わる力が、エアコンで室温を設定温度に保つ場合は外部から作用する熱が外乱である。また、上

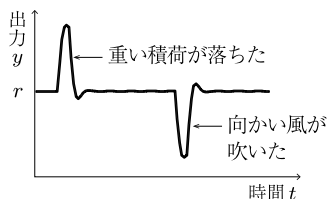


図 1.2 向かい風（外乱）が吹いたり積荷が落下（特性変動）したりしたときの出力  $y$

り坂を運転中に、かりに何百 kg もある重い積荷が車から落ちることを想像してほしい<sup>†</sup>。急激に車が軽くなったのに、アクセルを踏んだままにしていると、一気に加速してしまうだろう。このように重量などが変化することを、負荷変動という。また、自動車のエンジンは 10 年も経つとへたって、新車のときほど加速しなくなるだろう。このように、年月が経過してゆっくりと特性が変化することを、経年変化という。負荷変動や経年変化などのように、制御対象の特性が変動することを特性変動という。自然界には外乱と特性変動が必ず存在するので、正確に  $y = r$  とすることは不可能なのである。

もしも君が初めてレーシングカーを運転するとどうなるだろうか。恐る恐るアクセルを踏むと急加速し、驚いてブレーキを踏んで急減速してしまい、時速 60 km 一定で運転することは難しいだろう。ひどい場合は、図 1.3 に示すように加速と減速を繰り返す、加減速の度合いがどんどんひどくなってしまふかもしれない。この状態を不安定という。不安定になると、もちろん  $y = r$  とはならず、出力と目標値のズレはどんどんひどくなってしまふ。

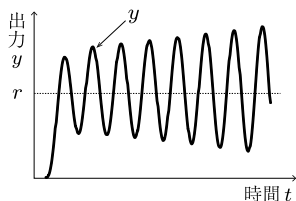


図 1.3 不安定になったときの出力  $y$

そこで制御工学では、制御の理想的な目的である、完全に  $y = r$  とさせることをあきらめる。その代わりに、制御の理想を邪魔する特性変動、目標値の変

<sup>†</sup> 積荷が落ちないように、運転前にしっかりチェックしておかなければならない。

化と外乱に対して、どれだけ  $y = r$  に近づけるかを表すつぎの三つの性能を、制御の目的として設定する。

- **安定性**：出力  $y$  がどんどん大きくなってしまふことを発散するといひ、この状態を不安定と呼ぶ。 $y$  は振動しながら大きくなることもある。不安定な制御対象の例として自転車がある。バランスをとらなければ自転車の傾き  $y$  はどんどん大きくなり、転倒してしまふ。安定性とは不安定になりにくい性質・能力である。上手な人が運転すると安定性が増す。普通の自転車なら運転できても、パンクして特性変動した自転車では、不安定になって転倒してしまふことがある。パンクしても運転できる人は安定性が優れているといえる。
- **外乱除去特性**：外乱の影響を受けにくい性質・能力である。自転車の運転中に突風が吹くと、ふらついてしまふことがある。突風は外乱である。ふらつかないで運転できる人は外乱除去特性が優れているといえる。
- **目標値応答特性**：目標値に出力が追従できる性質・能力である。運動場に曲がりくねったラインを引き、その上をずれないように自転車を運転できれば、目標値応答特性が優れているといえる。

これらは、制御性能とも呼ばれ、次章のシステム解析でそれぞれ解析することができる。

図 1.4 を見てほしい。横軸の  $t$  は時間で、単位は秒の英語である “second” の頭文字をとって [s] と表す。縦軸は自動車を時速 60 km 一定で運転しよう

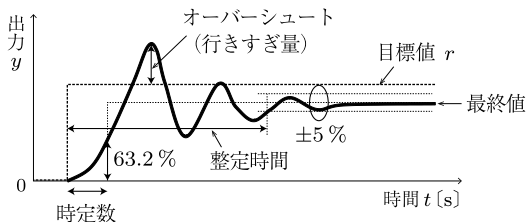


図 1.4 ステップ応答と制御目的の指標

# 索引

<b>【あ】</b>	過渡状態	7	厳密にプロパー
安定	過渡特性	7	186
安定限界	完全積分	128	<b>【こ】</b>
安定性	完全微分	131	コイル
安定余裕	観測ノイズ	20	89
	感度関数	22	公称値
			74
			古典制御理論
			31
			固有周波数
			94
			コンデンサ
			96
<b>【い】</b>	<b>【き】</b>		<b>【さ】</b>
位相	擬似積分	86	最終値
位相遅れ補償	擬似微分	86	7, 50
位相交差周波数	逆応答	104	—の定理
位相進み補償	逆ラプラス変換	169	50, 168
位相線図	共振	100	最小位相系
位相余裕	共振周波数	94, 100	105, 115
一次遅れ系	強制応答	40	再調整
一巡伝達関数	共役複素数	176	137
一般解	極	40	サスペンション
インパルス応答	—による安定判別	174	87, 95
インパルス関数	極零相殺	44, 208	サーボ制御
	極配置法	145, 208, 219	145
	虚部	45	三角関数
			34, 173
<b>【う】</b>	<b>【く】</b>		<b>【し】</b>
運動方程式	クロスオーバー周波数	73	シーケンス制御
			1
			次数
			31
			指数関数
			34, 171
			システム
			13
			システム同定
			12, 62, 82
			自然周波数
			94
			実部
			45
			時定数
			87
			時不変
			53
			時変
			53
			シミュレーション
			12
			遮断周波数
			92
			周期
			140
			周波数
			140
			周波数応答
			35, 57
			周波数応答法
			62
<b>【お】</b>	<b>【け】</b>		
オイラーの公式	経年変化	53	
オーバーシュート	ゲイン	56, 177	
オームの法則	ゲイン交差周波数	73	
折れ線近似	ゲイン線図	60	
	ゲインチューニング	137, 138	
	ゲイン余裕	73	
	限界感度法	139	
	減衰係数	94	
	減衰比	94	
<b>【か】</b>			
外乱			
外乱除去特性			
開ループ伝達関数			
角周波数			
カットオフ周波数			

周波数伝達関数 60  
 周波数特性 57  
 出力 4  
 純虚数 43  
 初期値 28  
 初期値応答 40  
 初期調整 137  
 ショックアブソーバ 88  
 自励振動 128  
 振 幅 56

【す】

数式モデル 11, 82  
 ステップ応答 7, 34, 214  
 ステップ関数 33, 171  
 スミス法 144  
 スモールゲイン定理 74

【せ】

制 御 1  
 制御器 4  
 — の設計 12  
 制御系設計 9  
 制御工学 2  
 制御仕様 9, 12, 21  
 制御性能 6  
 制御帯域 8, 73  
 制御対象 3  
 制御入力 4  
 制御目的 4  
 制御量 4  
 齊次方程式 40  
 整定時間 7  
 静的システム 14  
 積分ゲイン 127  
 積分項 127  
 積分要素 86, 189  
 折点周波数 61, 92, 94  
 零 点 40, 103, 196  
 線 形 52  
 線形時不変系 53, 176  
 線形性の公式 28, 167

【そ】

相補感度関数 22, 75  
 速応性 7

【た】

帯域幅 8  
 ダイナミックシステム 15  
 代表根 48  
 多項式 31  
 ダンパ 88

【ち】

チャタリング 130  
 チューニング 12  
 直結フィードバック 118

【て】

ディケイド 60  
 抵 抗 89  
 定常ゲイン 87, 94, 129  
 定常状態 7  
 定常値 7, 50  
 定常特性 7  
 定常偏差 7  
 テイラー展開 156  
 伝達関数 29  
 伝達関数表現 31

【と】

同 定 62  
 動的システム 15  
 特性変動 5, 53  
 特性方程式 40  
 特 解 40  
 トルク 213

【な】

ナイキスト軌跡 67  
 ナイキスト線図 67  
 ナイキストの安定判別 184  
 内部モデル原理 116, 199  
 内部モデル制御 141, 204

【に】

二次遅れ系 94, 192  
 2 自由度制御 20, 197  
 入出力間特性 11  
 ニュートンの運動方程式 15

【ね】

粘性摩擦係数 88  
 粘性摩擦力 88

【の】

ノミナルモデル 74

【は】

バックラッシ 55  
 発 散 42  
 ば ね 14  
 ばね・ダンパ系 87  
 ばね定数 14  
 ばね・マス・ダンパ系 95  
 パラメータ 11, 82  
 パラメータ調整 12  
 パラメータ同定 12, 82

【ひ】

非最小位相系 105, 115  
 ヒステリシス 55  
 非線形 54  
 微調整 138  
 微分ゲイン 129  
 微分項 129  
 微分公式 28, 166  
 微分方程式 11, 31, 35, 88  
 — を解く手順 36  
 微分要素 85, 189  
 比例ゲイン 118  
 比例項 118  
 比例要素 85

【ふ】

不安定 5, 42  
 — な  $G(s)$  の周波数特性 76



不安定極	44	ベクトル軌跡	67	<b>【ら】</b>	
不安定零点	44, 104	ベクトル線図	67	ラウス・フルビッツの	
フィードバック	3	偏差	4	安定判別	67, 181
フィードバック制御	2, 3	<b>【ほ】</b>		ラプラス変換	26, 169
フィードバック制御器	20	飽和	54	—の積分公式	167
フィードバックループ	17	ボード線図	59, 60, 216	—の線形性の公式	167
フィードフォワード制御器	20, 115	—の折れ線近似	62	—の微分公式	166
不完全積分	86, 128	<b>【ま】</b>		ラムダチューニング	141, 204
不完全微分	86, 130	マ ス	95	ランプ応答	34
不感帯	55	<b>【む】</b>		ランプ関数	34, 173
複素平面	45	無限大ノルム	75	<b>【り】</b>	
不確かさ	74	むだ時間要素	105, 197	留数定理	173
フックの法則	14	<b>【も】</b>		量子化誤差	55
部分的モデルマッチング	145, 208	モータ	89	<b>【る】</b>	
部分分数展開	37	目標値	4	ループ整形	149, 209, 222
プロセス制御	145	目標値応答特性	6	<b>【れ】</b>	
ブロック線図	10, 16	モデリング	82	レギュレータ	145, 151
プロパー	186	モデル化	11, 82, 212	<b>【ろ】</b>	
分母多項式	31	モデルマッチング	141	ロバスト安定	76
<b>【へ】</b>		<b>【よ】</b>			
閉ループ	17	1/4 減衰	140		
閉ループ制御	2				
閉ループ伝達関数	22				

<b>【D】</b>		LTI システム	53	<b>【R】</b>	
D ゲイン	130	<b>【M】</b>		RL 回路	89
decade	60	MATLAB	211	RLC 回路	96
<b>【H】</b>		Mat@Scilab	211	<b>【S】</b>	
$H^\infty$ 制御	76	<b>【P】</b>		s 平面	45
<b>【I】</b>		P ゲイン	118	SCILAB	211
I ゲイン	127	P 制御	118		
IMC	141	PD 制御	129		
I-PD 制御	135, 201	PI 制御	126		
<b>【L】</b>		PID ゲイン	133		
LTI	176	PID 制御	117, 133, 137, 200		
		PI-D 制御	136, 202		

		Im[ ]: 虚部	45	s: 複素数の変数	26
		j: 純虚数	44	t: 時間	6
C: 制御器	4	K: 制御器	4	u: 制御入力	4
e: 自然対数の底	26	P: 制御対象	3	y: 制御量, 出力	4
e: 偏差	4	r: 目標値	4		
G: 制御対象	3	Re[ ]: 実部	45		



		極を求める	67, 221	ボード線図を描く	109, 218
		ステップ応答を見る	147, 215		
安定余裕を求める	221	ナイキスト線図を描く	81		

— 著者略歴 —

1989年 大阪府立大学工学部電子工学科卒業  
1991年 大阪府立大学大学院工学研究科博士前期課程修了（電子工学専攻）  
1991～  
2001年 ダイキン工業株式会社 電子技術研究所  
1999年 大阪府立大学大学院工学研究科博士後期課程修了（電気情報系専攻）  
博士（工学）  
2001年 近畿大学講師  
2006年 近畿大学助教授  
2011年 近畿大学教授  
現在に至る

高校数学でマスターする 制御工学

本質の理解から Mat@Scilab による実践まで

Control Engineering Based on High School Math

— From the Essence to the Practice Using Mat@Scilab —

© Manabu Kosaka 2012

2012年 8月23日 初版第1刷発行

★

検印省略

著者 小<sup>こ</sup>坂<sup>さか</sup>学<sup>まなぶ</sup>  
発行者 株式会社 コロナ社  
代表者 牛来真也  
印刷所 三美印刷株式会社

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-03206-2 (柏原) (製本:愛千製本所) G

Printed in Japan



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられております。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めておりません。

落丁・乱丁本はお取替えいたします