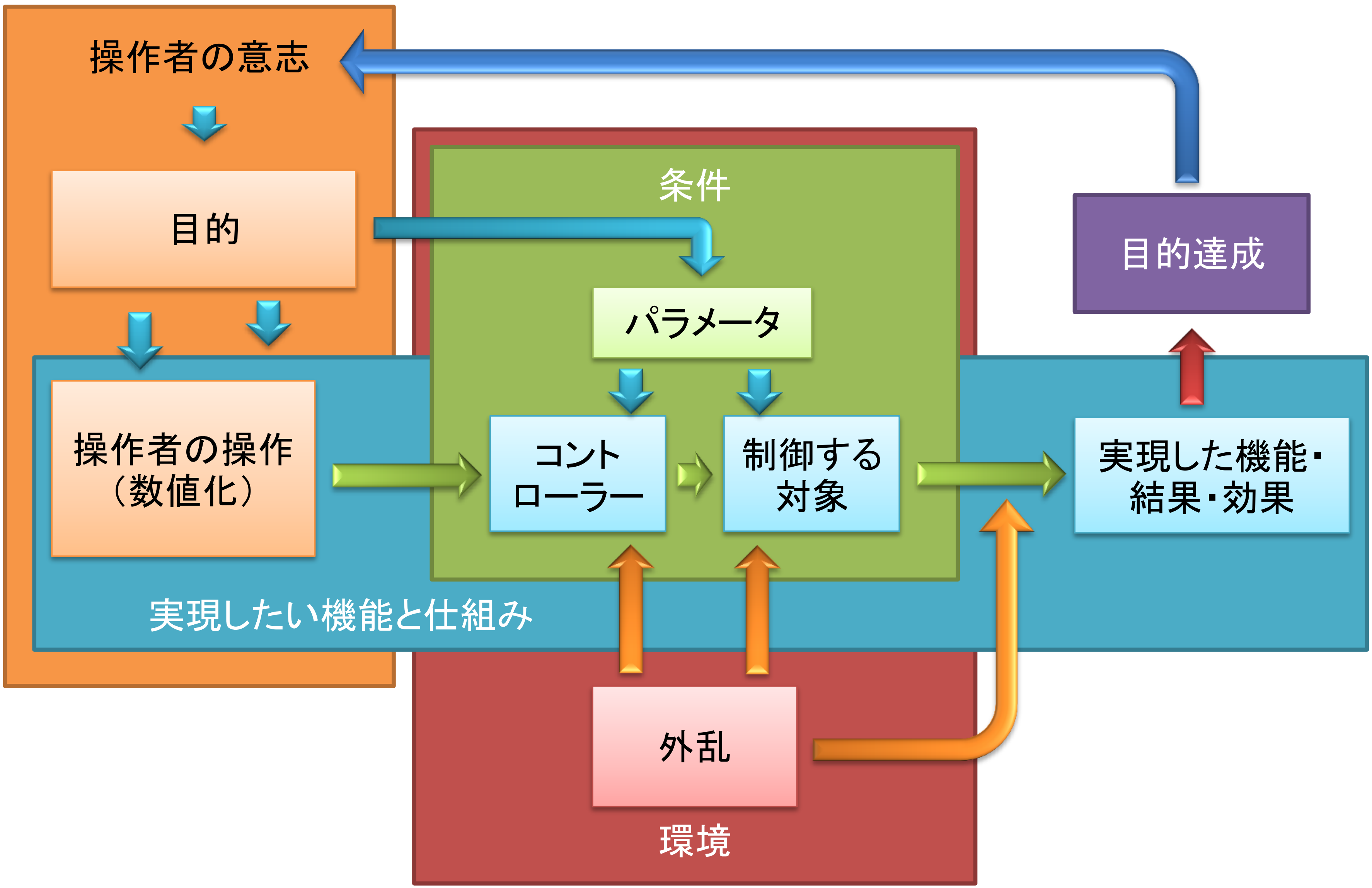
１．制御とは何か

人間や道具によるタスクの実行

- 体力（動力）
  - 人間の筋力を動力源として利用
- 技能
  - 人間の手の器用さ，道具の特性を把握・利用して作業
- 知能
  - 人間による認識，判別，計画，判断，実行，調整

人間の望む機能を実現できることが最終目標

制御の考え方

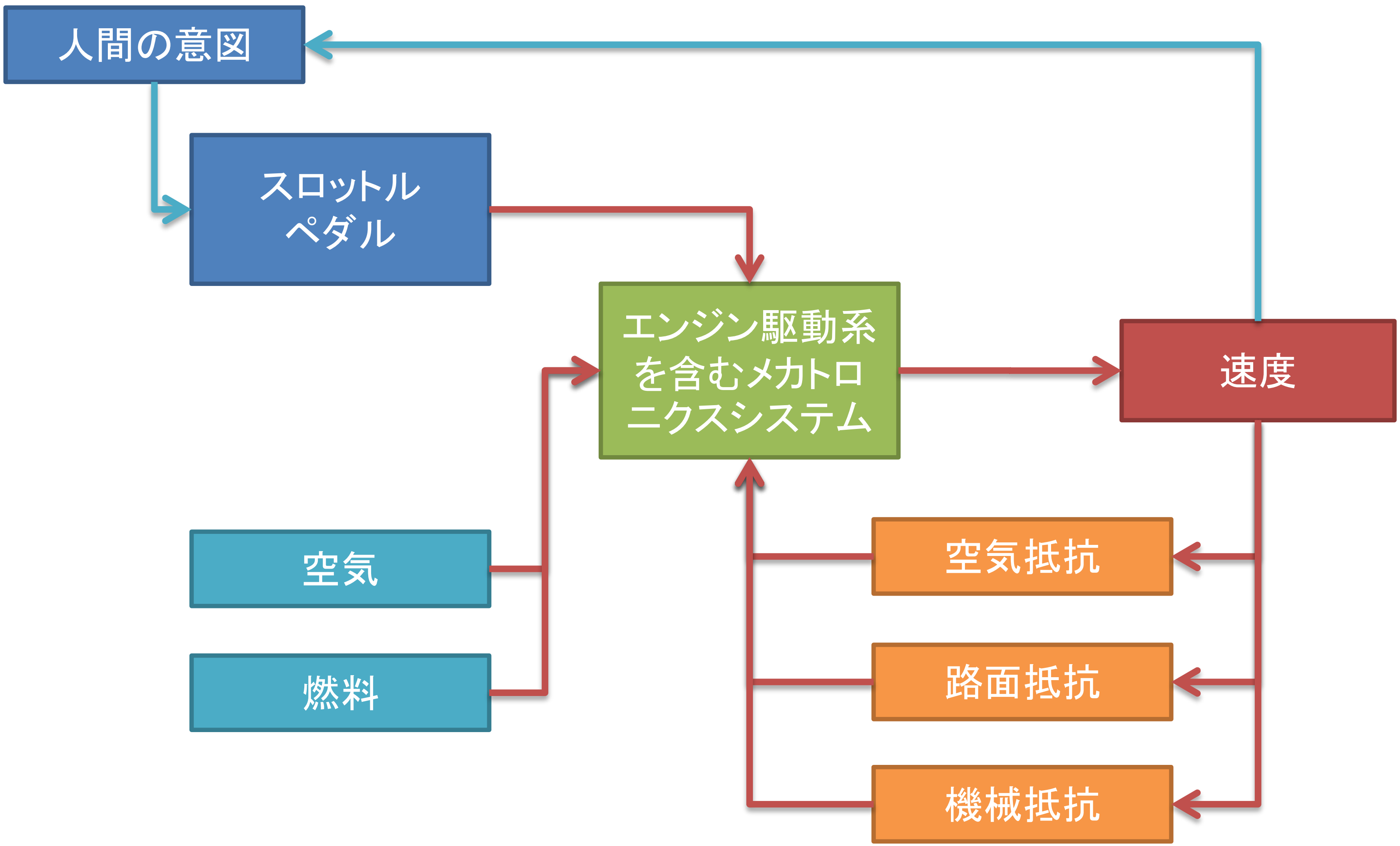

操作者の意志
目的
操作者の操作
（数値化）
条件
パラメータ
コントローラー
制御する対象
目的達成
実現した機能・結果・効果
実現したい機能と仕組み
外乱
環境

制御のいろいろ

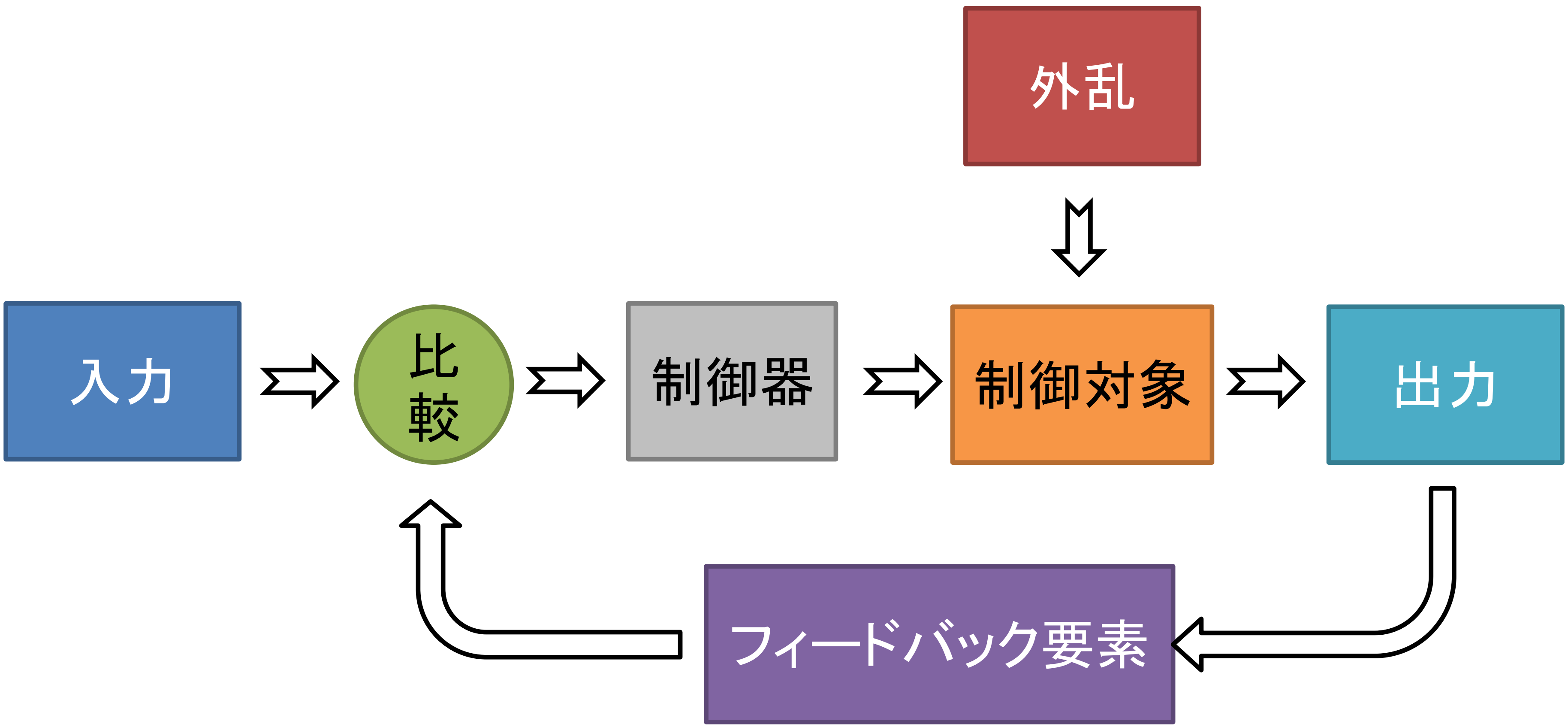
制御法	操作	自動制御	自律制御
役割	機能拡大	支援	代行
目的決定 （意志）	操作者	操作者	操作者
目標設定 （行動原理）	操作者	操作者	制御系
判断	操作者	制御系	制御系
動作	制御系	制御系	制御系
例（乗り物）			

制御の面から自動車を考える

操作を中心とした制御システム

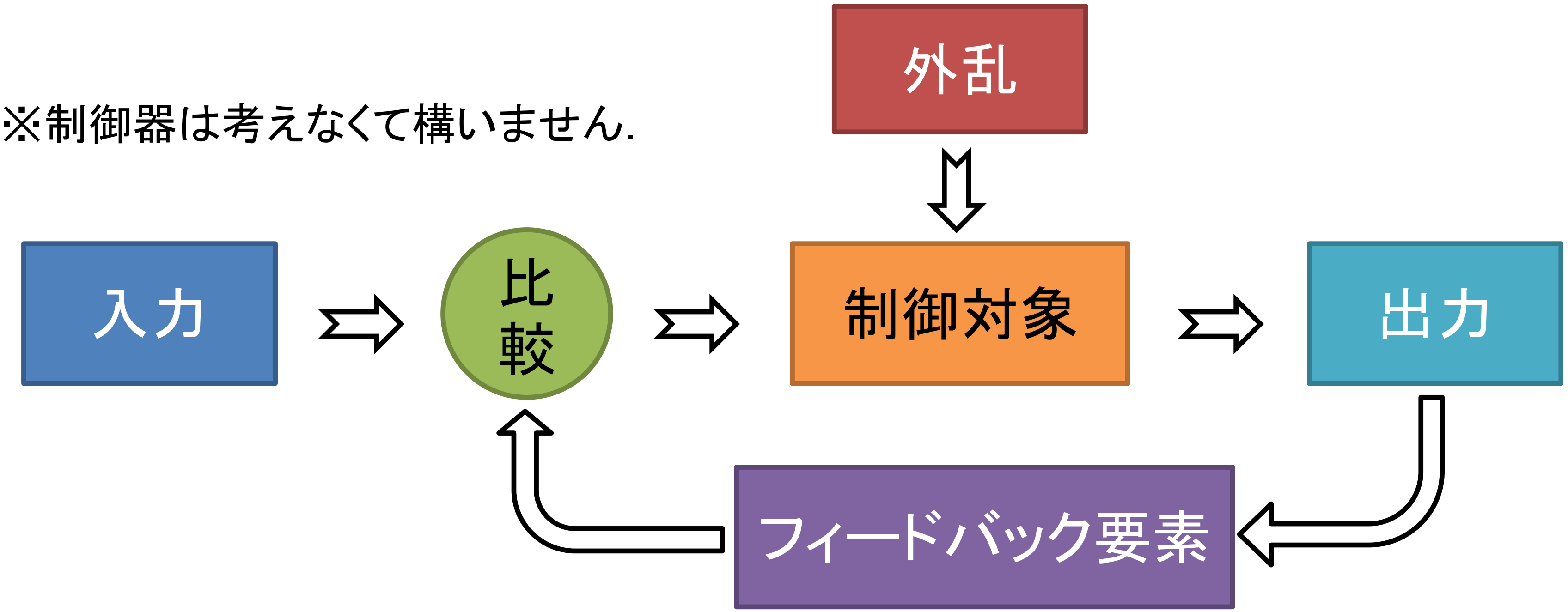


基本的な制御系の概念



ここで課題1

※制御器は考えなくて構いません.



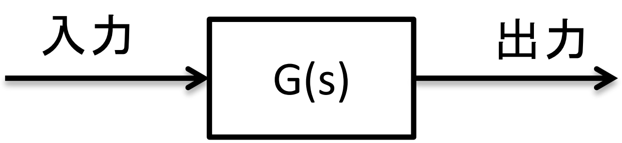
- 正しい答えを書く必要はありません. （周囲の人々と討論しながら考えてみましょう. ）
- 今の段階で想像する「制御」の仕組みを持ったものを考えましょう.
- レポート用紙半分～1枚くらい？
- 鉛筆書きで充分です.
- 授業中の分はペン入れの必要ありません.

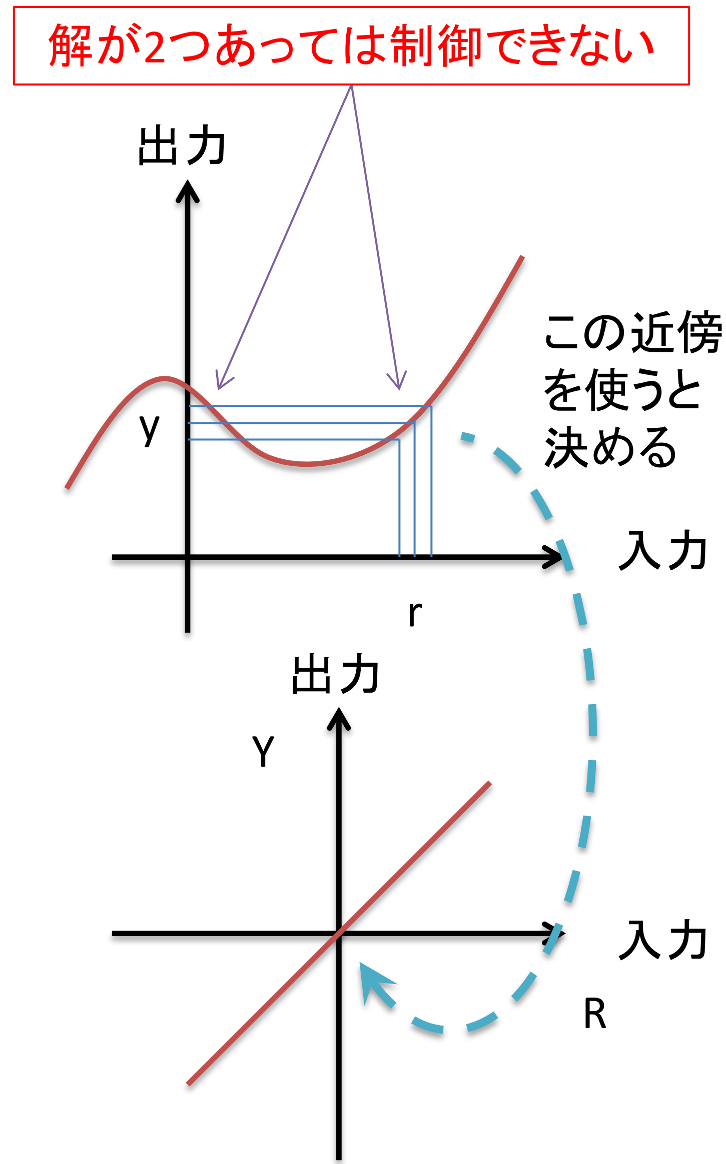
２．制御対象を知る

制御モデル

$$\boxed{m\frac{d^2}{dt^2}x + c\frac{d}{dt}x + kx + \alpha x^3 = f}$$

- 運動方程式のような時間変化の関係を表す**微分方程式**で示す
- 制御したい範囲（条件）※を決め，制御できる方程式に**近似**
- 指令値に対する応答の関係を示す**伝達関数**を作る





※主に扱う中心の状態（釣り合い）を平衡状態といい，制御則を決める基本

制御モデルの決め方

方法１

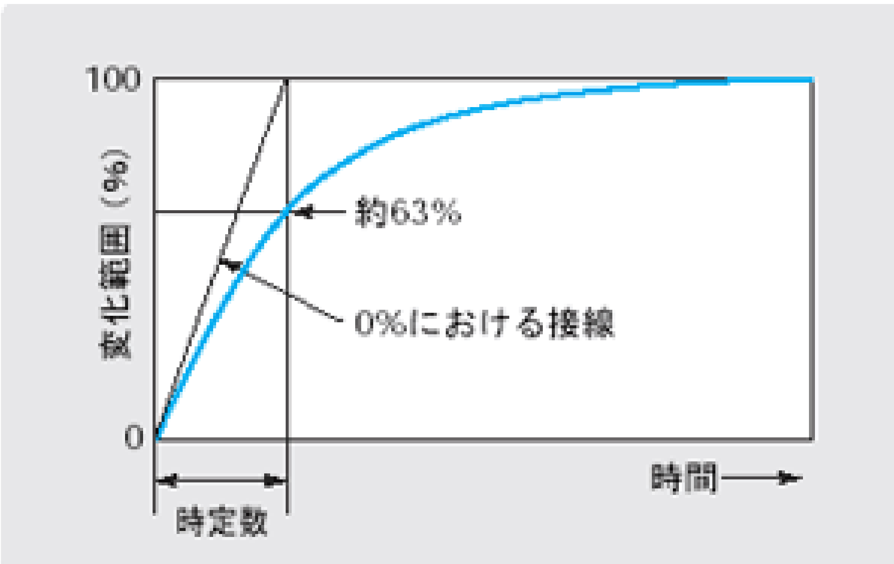
　実際に動作させ，計測してパラメータを得る

方法２

　選択した対象の既存の制御モデルに当てはめ，仕様書の値を用い，足りない分を実測してパラメータを得る

- 過去の知見を利用することは手抜きではありません．
- 利用できる知見を的確に選択する発想も工学的に重要です．

一次遅れの応答



$$y(t)=K\left(1-e^{-\frac{t}{T}}\right)$$

ラプラス変換した伝達関数

$$G(s)=\frac{K}{Ts+1}$$

ここが1次であり，遅れて応答するから1次遅れ

- 入力が加わった後
  - 初めは素早く応答
  - 時間が経つにつれて徐々になだらかに応答
  - 充分に時間が経つにつれ目標値に漸近

ラプラス変換

$$F(s)=L\{f(s)\}=\int_0^{\infty} e^{-st} f(t)dt$$

1. 重解，特殊解（右辺が0ではない）が存在するような複雑な微分方程式が楽に解ける
2. 初期値に注意しておけば，微分方程式を明快な代数方程式として解くことができるので簡単に解ける
3. 式の形から制御特性を見出しやすい

初期条件とラプラス変換表により簡単に解ける

ラプラス変換表

t 関数　$f(t)$	s 関数　$F(s)$	t 関数　$f(t)$	s 関数　$F(s)$
1，$u(t)$	$\frac{1}{s}$	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
a	$\frac{a}{s}$　(aは定数)	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
$\delta(t)$	1	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$
t	$\frac{1}{s^2}$	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$
t^2	$\frac{2}{s^3}$	$\sinh \omega t$	$\frac{\omega}{s^2-\omega^2}$
t^n	$\frac{n!}{s^{n+1}}$　$(n=1,2,\cdots)$	$\cosh \omega t$	$\frac{s}{s^2-\omega^2}$
t^p	$\frac{\Gamma(p+1)}{s^{p+1}}$　$(p>0)$	$t \sin \omega t$	$\frac{2\omega s}{(s^2+\omega^2)^2}$
e^{-at}	$\frac{1}{s+a}$	$t \cos \omega t$	$\frac{s^2-\omega^2}{(s^2+\omega^2)^2}$
te^{-at}	$\frac{1}{(s+a)^2}$	$\sin(\omega t+\theta)$	$\frac{\omega \cos\theta + s \sin\theta}{s^2+\omega^2}$
$t^n e^{-at}$	$\frac{n!}{(s+a)^n}$	$\cos(\omega t+\theta)$	$\frac{s \cos\theta - \omega \sin\theta}{s^2+\omega^2}$

<u>計算手順</u>

1. 時間に関する関数についてラプラス変換する
2. 必要な演算を行う
3. ラプラス逆変換により時間に関する関数に戻す

（DCブラシ付き）モータのコントロール

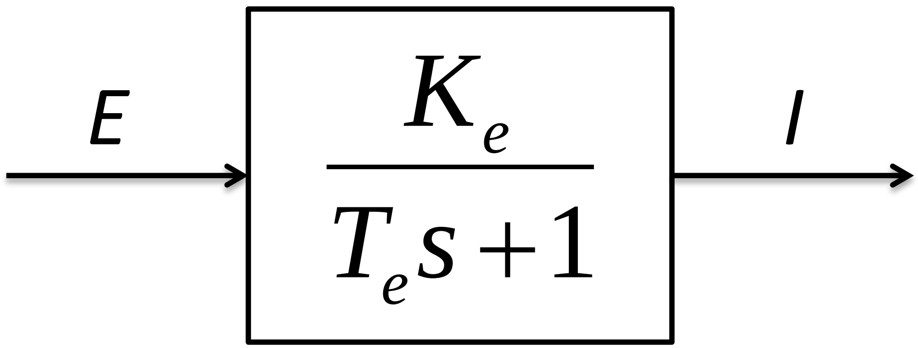
物理量	応用対象	センサ
回転角	ロボットアームの関節角	回転角センサ（ポテンショメータ，ロータリーエンコーダ，レゾルバ）
角速度	車両の速度，設備の運転状態	角速度センサ（タコジェネレータ，ロータリーエンコーダ，レゾルバ），電子ガバナ（回転速度を推測）
トルク	柔らかい制御	トルクセンサ
角加速度	飛行機，ヒューマノイドの姿勢制御	外付けのジャイロ
ジャーク	自律ヘリコプタの軌道制御，姿勢制御	演算によって算出

（参考）ヘリコプタの自律飛行制御

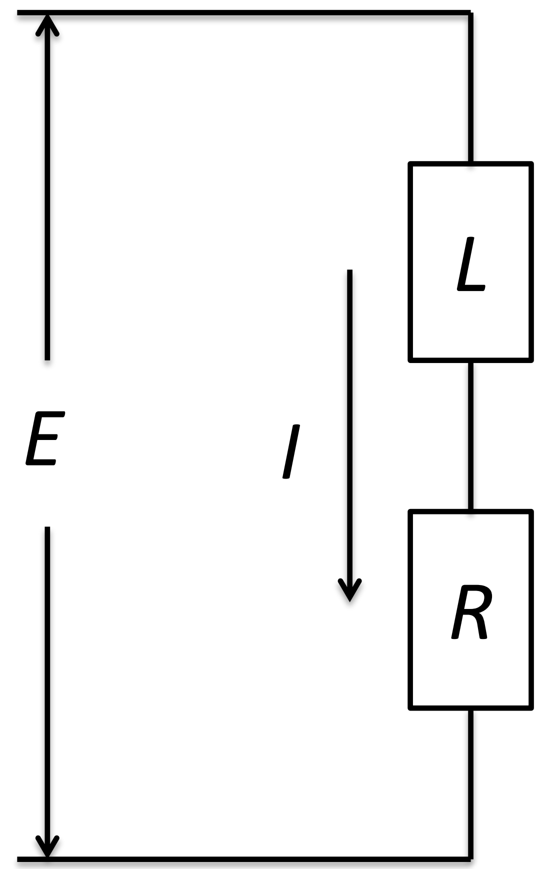
1. 1990年頃、スーパーコンピュータを用いてもヘリコプタの姿勢制御は不可能
2. 数年前、センサ性能の劇的な向上でヘリコプタの自立安定制御はマイコンレベルでも可能.
3. 画像処理による位置決めにより, スマートフォンでコマンドを送るだけで動かせるものすら出現.
4. 加速度, ジャーク（加加速度）までも扱う制御則の研究とマイコンのさらなる性能向上により, ヘリコプタの自律制御とアクロバット飛行も可能.

http://www.ted.com/talks/lang/ja/vijay_kumar_robots_that_fly_and_cooperate.html

電機子の電気特性

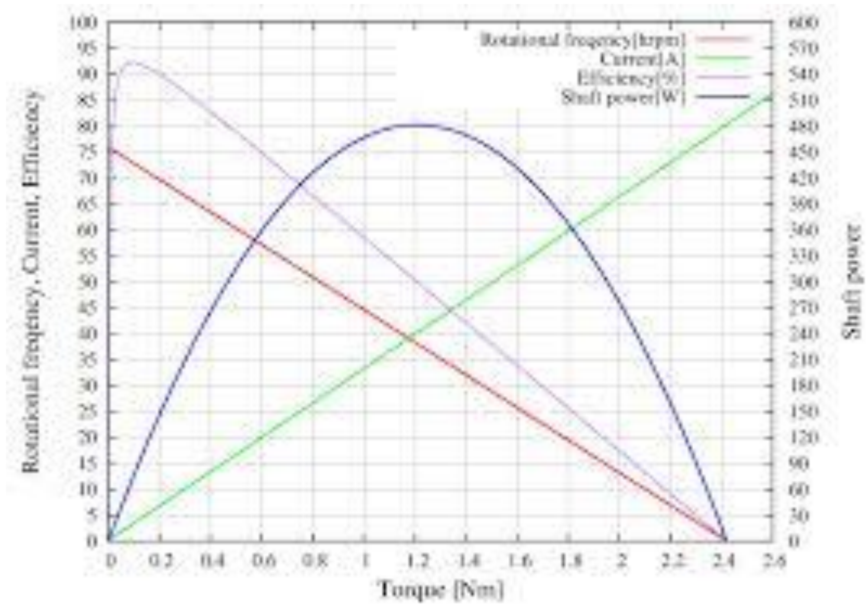


$$K_e = \frac{1}{R} \qquad T_e = \frac{L}{R}$$



磁気回路の特性

$$I \longrightarrow \boxed{K_I} \longrightarrow T$$



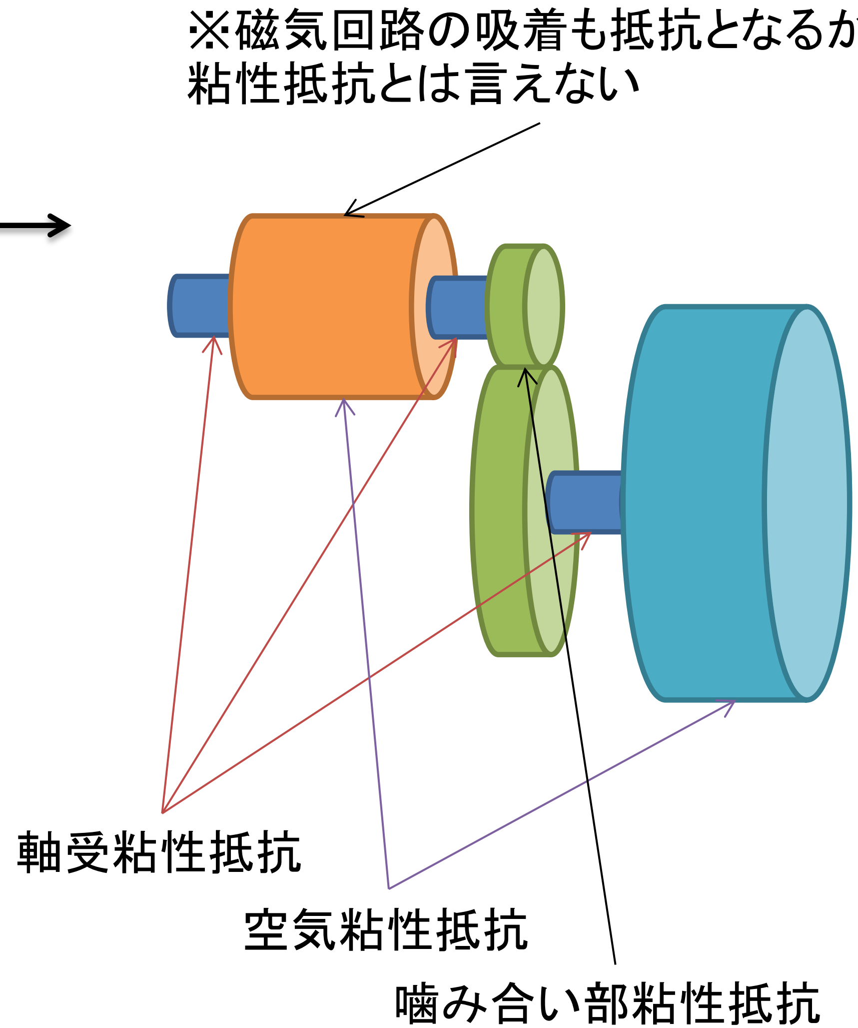
磁気回路の性能
- 飽和磁束密度
- 空隙
- 分布（角依存性）

が良いほど，
少ない電流でも大きなトルクが発生

モータの機械特性

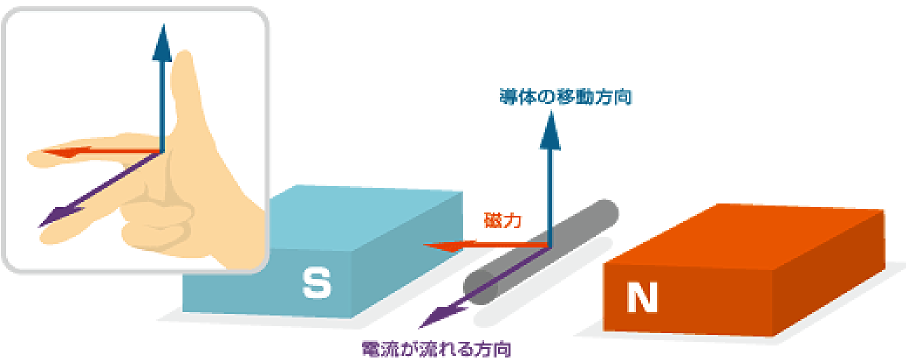
$$T \longrightarrow \boxed{\frac{K_m}{T_m s+1}} \longrightarrow \omega$$

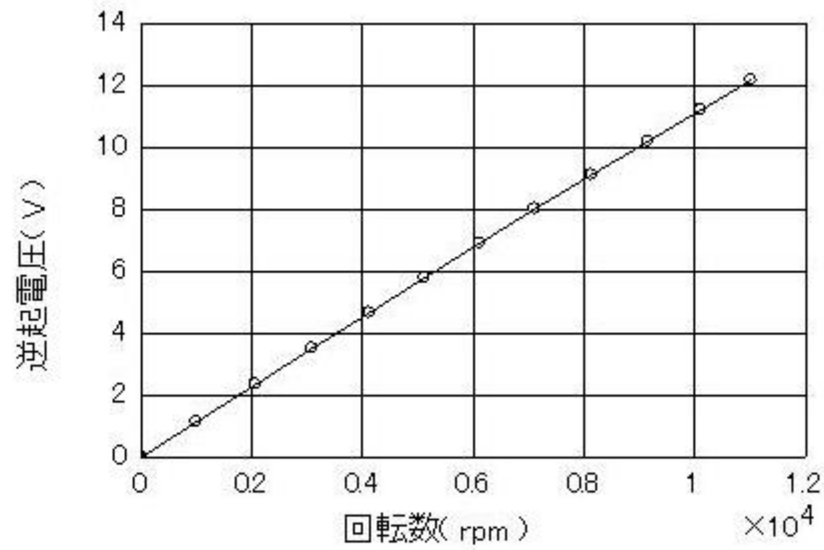
$$K_m = \frac{1}{K_v} \qquad T_m = \frac{J}{K_v}$$



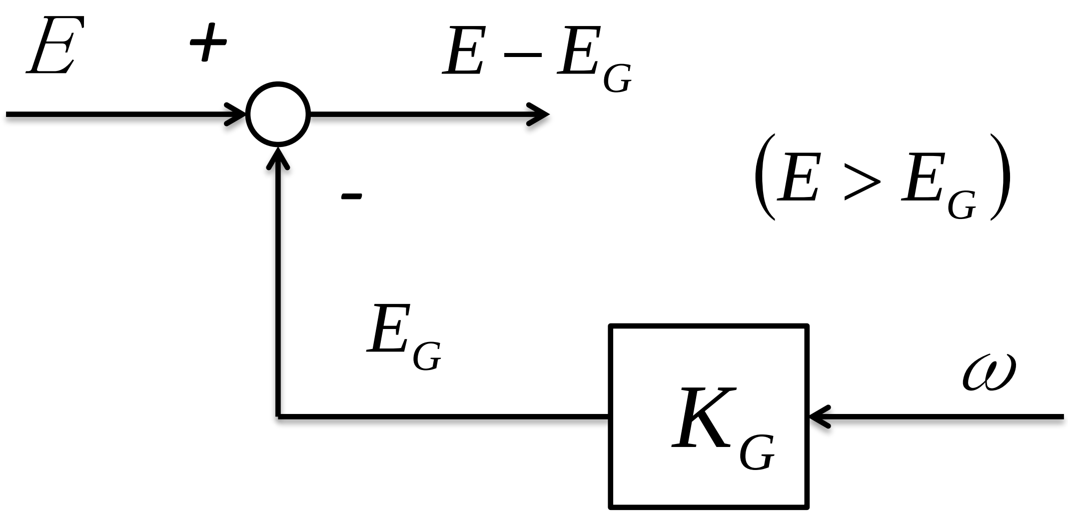
逆起電力の扱い

＜フレミングの右手の法則＞ 磁場で導体を動かすと電流が流れる



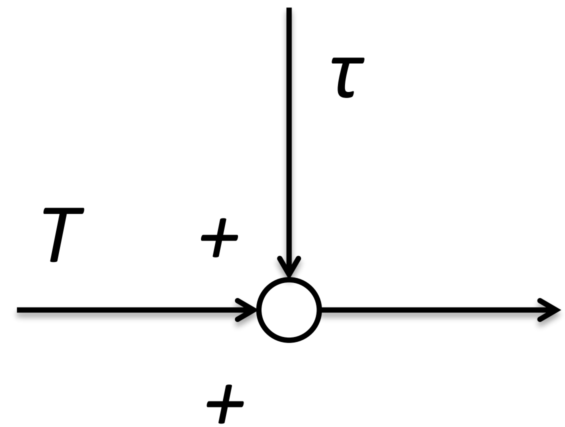


回転数と逆起電圧



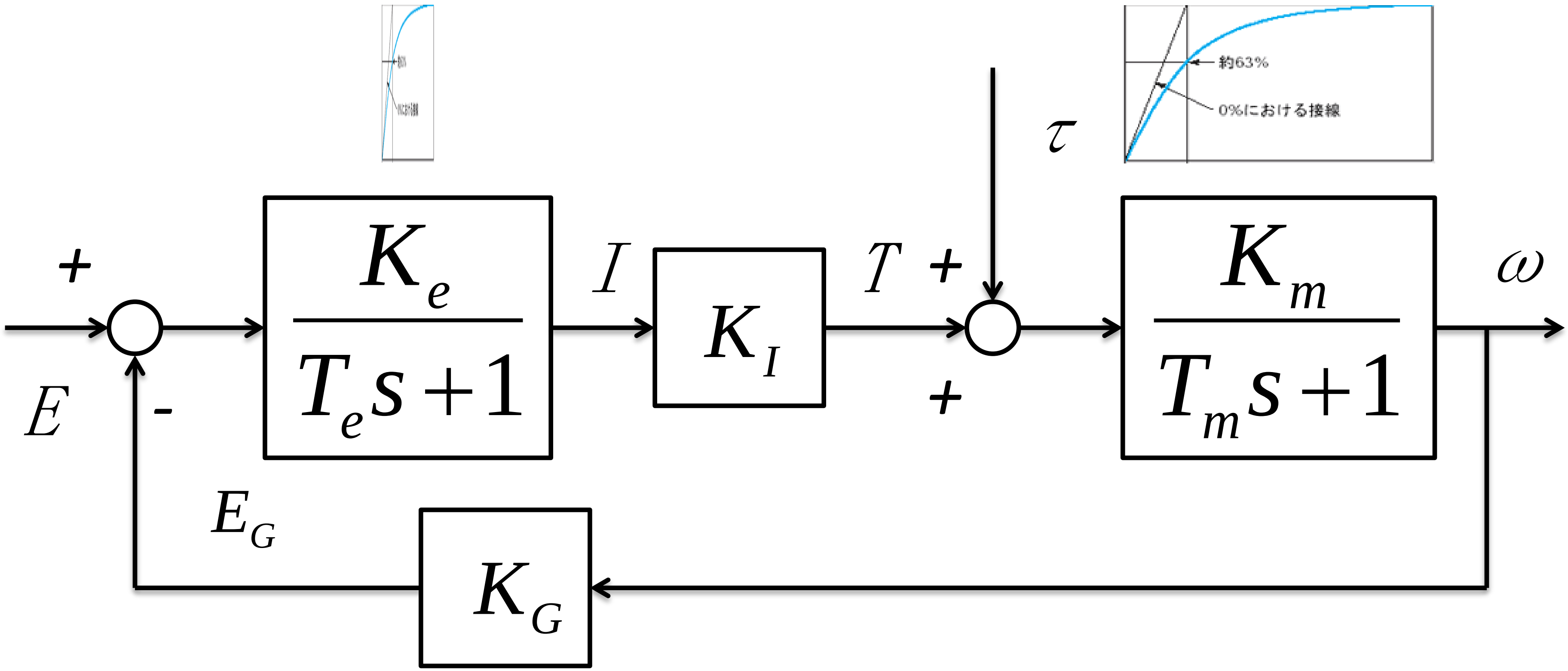
外乱

1. 想定外の要因により制御の平衡状態を乱す要素
2. 初期の状態から制御系の特性（定数）が変わってしまう



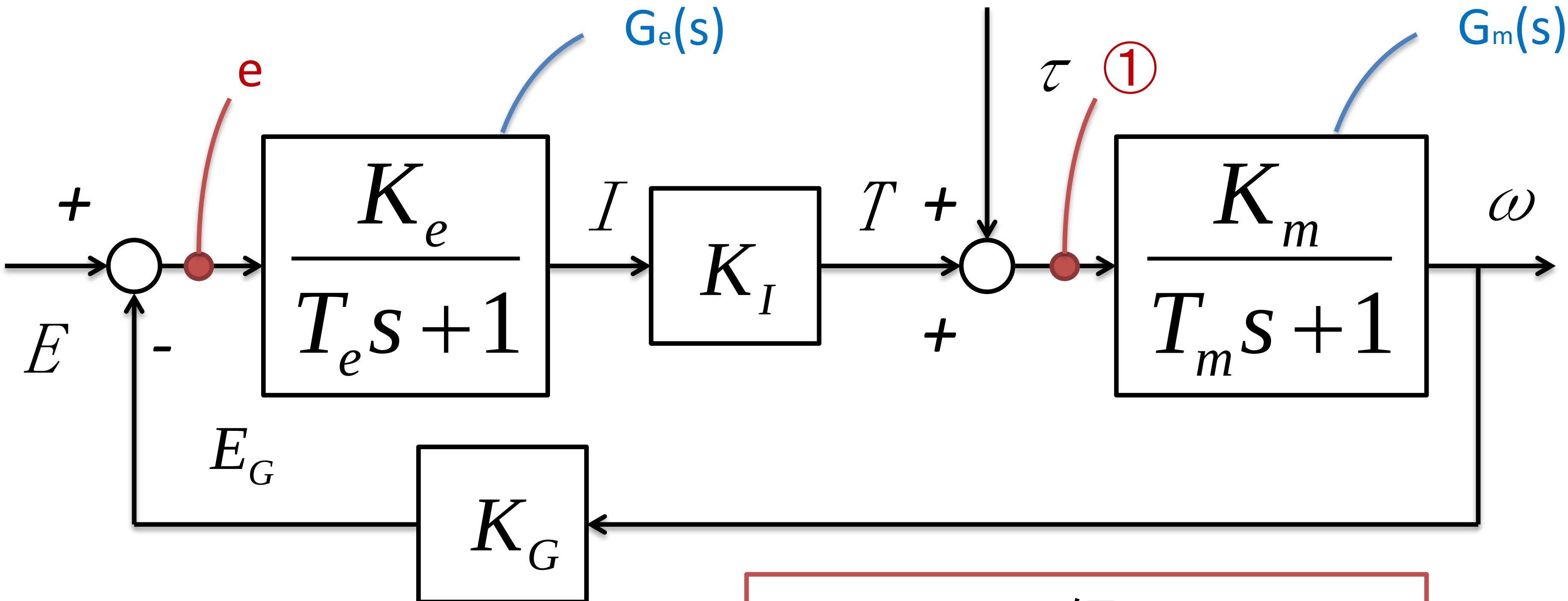
狙い通りの制御ができず，
欲しい機能が実現できない

モータ特性まとめ



各要素の特性をラプラス変換しておくと上のように
ブロック線図にまとめて表記し，代数演算ができる

ここで課題2



連立する方程式

$e = E - E_G$

$E_G = ???$

① $= T + \tau$

$T = ???$

$\omega = ???$

メモ

以下の変数を用いる

$K_G, E, G(s)_e, K_I, \tau, G(s)_m, \omega$

$G(s)_e =$ $\quad$ $G(s)_m =$

$\omega = ?E + ?\ \tau$

モータ伝達関数

$$\omega(s) = \frac{G_E(s)K_I G_M(s)}{1 + G_E(s)K_I G_M(s)K_G} E + \frac{G_M(s)}{1 + G_E(s)K_I G_M(s)K_G} \tau$$

$$G_E(s) = \frac{K_E}{T_E s + 1}$$

$$G_M(s) = \frac{K_M}{T_M s + 1}$$

入力と外乱の伝達関数の違いを見ておいてください．

$$\omega(s) = \frac{K_E K_I K_M}{T_E T_M s^2 + (T_E + T_M)s + 1 + K_E K_I K_M K_G} E$$

$$+ \frac{K_M}{T_E T_M s^2 + (T_E + T_M)s + 1 + K_E K_I K_M K_G} \tau$$

モータのモデリング精度

やたらと精密に動力学・制御モデルを作ってしまうと・・・

<u>古い例（10年前）</u>
電気特性，動力学特性，摩擦などの非線形要素を細かく設定した上で，歩行ロボット動かすシミュレータ
Pentium4 / 500MHzシングルコア（しかなかった）クラスで単純な歩行動作のシミュレーションに数時間かかる．

Core i7: 3.5GHz Quad core(8thred)

目的を良く考えて精度にこだわらなければ・・・

電気回路の特性は極めて早い応答，逆起電力も含んだモデルとして考えてもそれほど誤差はない

モータに加える電圧Eを入力とし，モータ回転数ωを出力として，単なる一次遅れのモデルだと考えても良い．

実時間で動力学シミュレーションを行いながら，その結果と指令値（軌道計画）と現在の観測値（応答）を比較しながら制御すると高度な制御を行うことができる．

目的を考えること，演算精度と演算速度のバランスを考える必要がある

課題2でせっかくモータの伝達関数を算出したのですが・・・.

要求**仕様**

- 誤差についての要求は厳しくない（数%あっても問題ない）
- 実時間で制御を行う必要がある

↓

- 制御モデルはシンプルでよく，シンプルである必要がある.
- 先程の制御モデルを用いず，もっと簡単な制御モデルを利用する.

一度，精密なモデルを作ってから，目的に立ち返り，何を重視するのか考え直すのは工学的に有効な考え方だと言って良い.

モータの伝達関数

$$\omega(s)=\frac{K_E K_I K_M}{T_E T_M s^2+(T_E+T_M)s+1+K_E K_I K_M K_G}E$$

$$+\frac{K_M}{T_E T_M s^2+(T_E+T_M)s+1+K_E K_I K_M K_G}\tau$$

電気回路の応答は充分速い（時定数が小さい）

$$T_E \to 0$$

$$T_E = 0.259[m\sec]$$

$$T_E = 4.67[m\sec]$$

$$\omega(s)=\frac{\dfrac{K_E K_I K_M}{1+K_E K_I K_M K_G}}{\dfrac{T_M}{1+K_E K_I K_M K_G}s+1}E+\frac{\dfrac{K_M}{1+K_E K_I K_M K_G}}{\dfrac{T_M}{1+K_E K_I K_M K_G}s+1}\tau$$

モータの伝達関数（実例）

$$\omega(s)=\frac{0.240*29.5*10^{-3}*40.8*10^{3}}{0.259*10^{-3}*4.67*10^{-3}s^{2}+\left(0.259*10^{-3}+4.67*10^{-3}\right)s+1+0.240*29.5*10^{-3}*40.8*10^{3}*3.46*10^{-3}}E$$

$$+\frac{40.8*10^{3}}{0.259*10^{-3}*4.67*10^{-3}s^{2}+\left(0.259*10^{-3}+4.67*10^{-3}\right)s+1+0.240*29.5*10^{-3}*40.8*10^{3}*3.46*10^{-3}}\tau$$

$$=\frac{289}{1.21*10^{-6}s^{2}+\left(4.93*10^{-3}\right)s+1+1}E+\frac{40.8*10^{3}}{1.21*10^{-6}s^{2}+\left(4.93*10^{-3}\right)s+1+1}\tau$$

$$G(s)=\frac{K_E K_I K_M}{T_E T_M s^2+2s+\omega_n^2}E$$

Ke	0.240	A/V
Ki	29.5	mNm/A
Km	40.8	rpm/mNm
Kg	0.00346	V/rpm

$$T_E=0.259[m\sec]$$
$$T_M=4.67[m\sec]$$

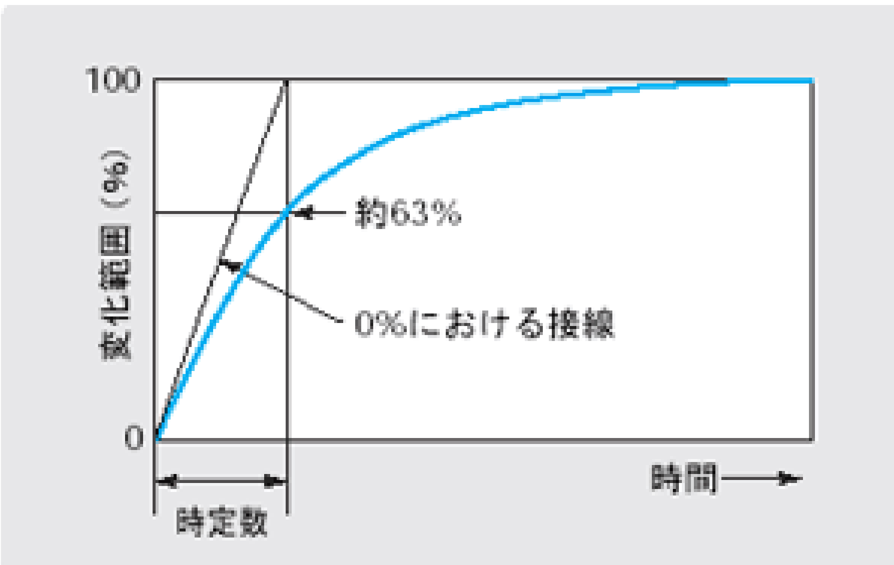
$$\omega(s)=\frac{\frac{289}{1+1}}{\frac{4.67*10^{-3}}{1+1}s+1}E+\frac{\frac{40.8*10^{3}}{1+1}}{\frac{4.67*10^{-3}}{1+1}s+1}\tau$$

$$=\frac{145[rpm/V]}{2.34*10^{-3}s+1}E[V]+\frac{20.4*10^{3}[rpm/Nm]}{2.34*10^{-3}s+1}\tau[Nm]$$

振動系でもζが大きいと1次遅れの応答と酷似する.

10V掛けると最終的に1450rpm位になる. もし, 最大トルクの10%の外乱(17.0mNm)があると, 280rpm位回転が落ちる（20%落ち）ことが分かる.

一次遅れの系のパラメータ

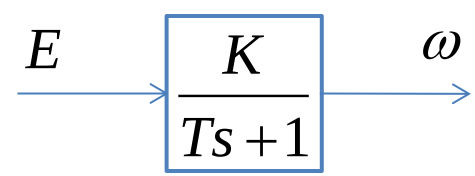


$$y(t) = K\left(1 - e^{-\frac{t}{T}}\right)$$

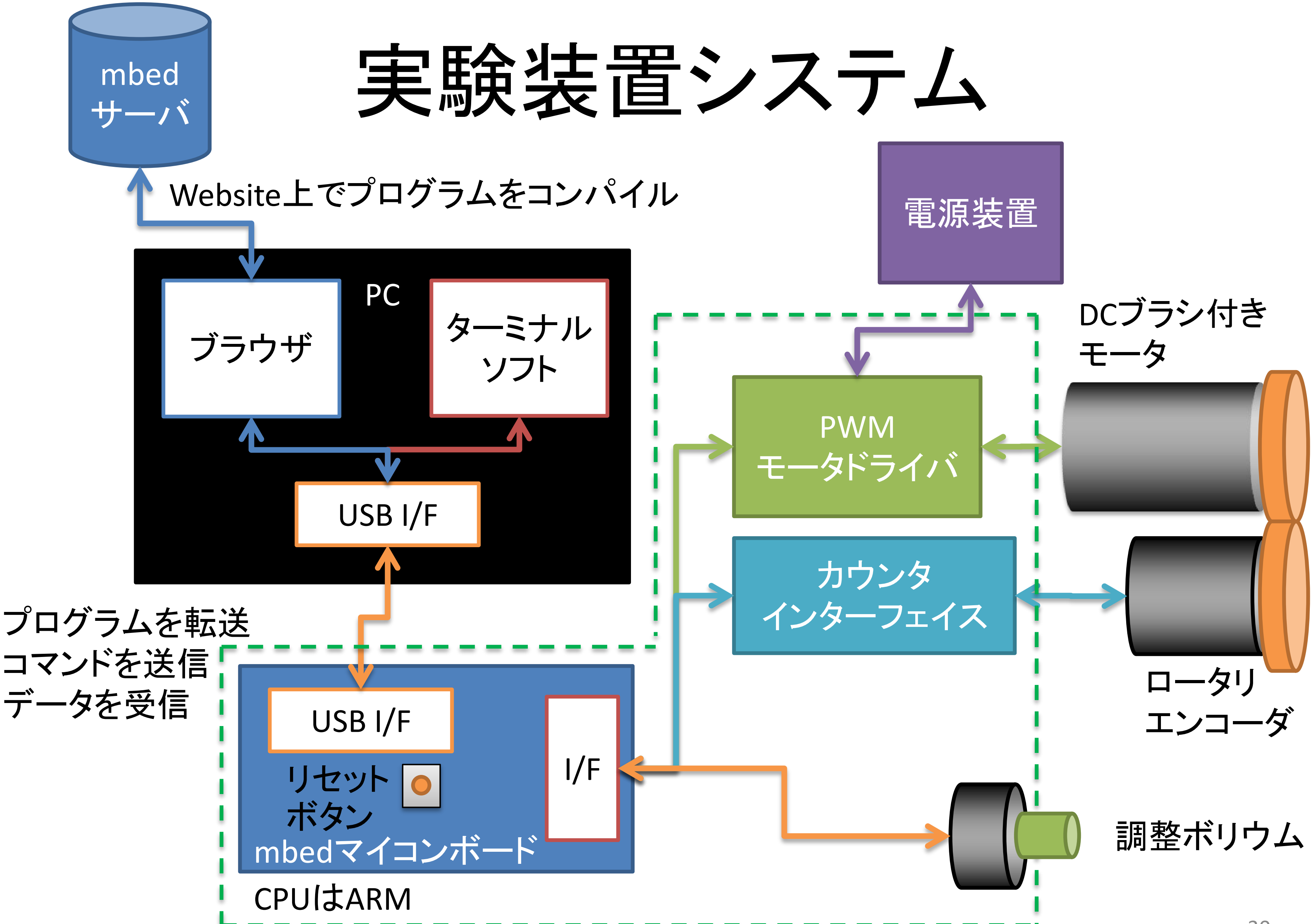
ラプラス変換した伝達関数

$$G(s) = \frac{K}{Ts+1}$$

- T:時定数（1-（1/e）≒63.2%になるまでの時間）
- K:ゲイン係数
- 4～5T:整定時間



実験装置システム



mbed
サーバ
Websiteでプログラムをコンパイル
電源装置
PC
ブラウザ
ターミナル
ソフト
USB I/F
DCブラシ付き
モータ
PWM
モータドライバ
カウンタ
インターフェイス
ロータリ
エンコーダ
プログラムを転送
コマンドを送信
データを受信
USB I/F
リセット
ボタン
I/F
mbedマイコンボード
CPUはARM
調整ボリウム

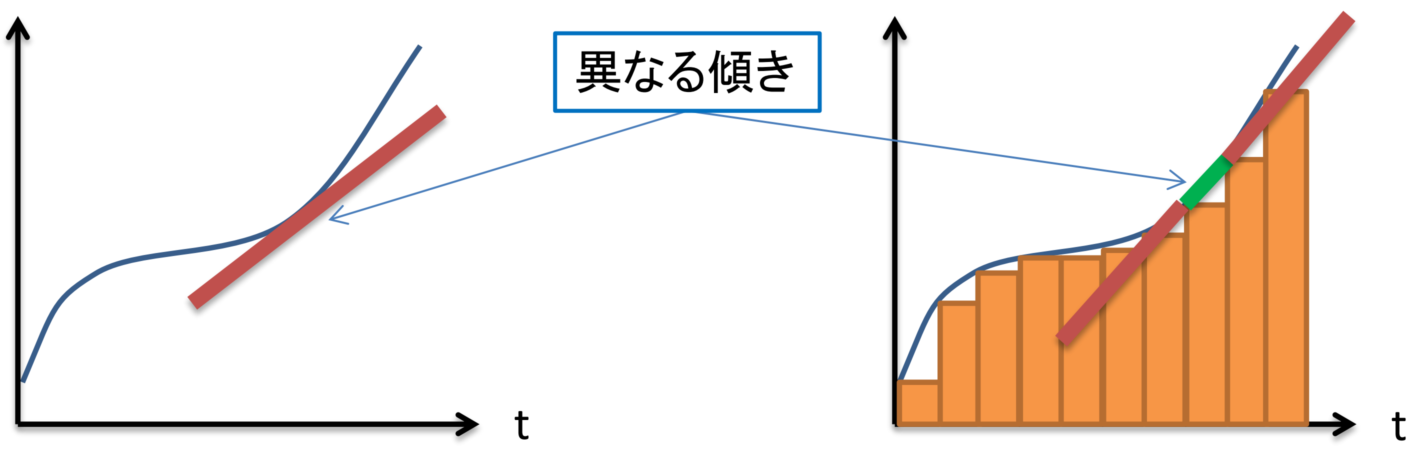
アナログ制御とデジタル制御

微分

- 文字通り微分を行いある点における接線の傾きを計算
- 回路としてはフィルタを通すことになり，ノイズ等の影響を減らす事が可能

差分

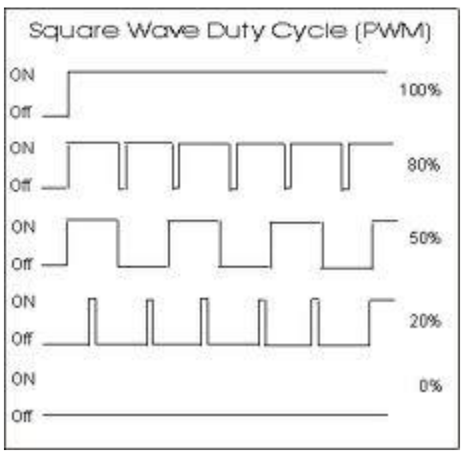
- 微分と似ているが，一周期前の値との差により傾きを計算
- ノイズなど突出した値の場合，極めて外れた値となるが，ソフトウェアでフィルタを構成することは可能



PWM制御 / ロータリーエンコーダ

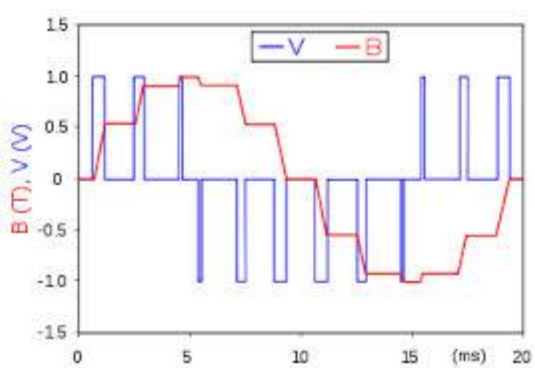
PWM制御

- 短時間のON-OFFの時間比で電圧を作り出す
- デジタル回路による制御に適す
- 回路がシンプルで効率の良いものが作れる



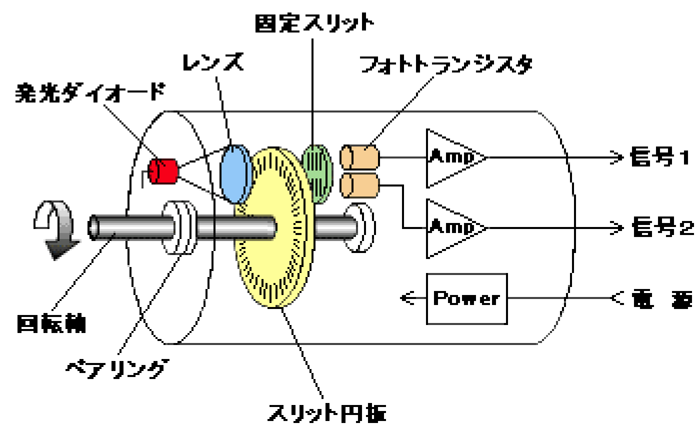
ON-OFF時間比
（デューティ比）
の変化と波形

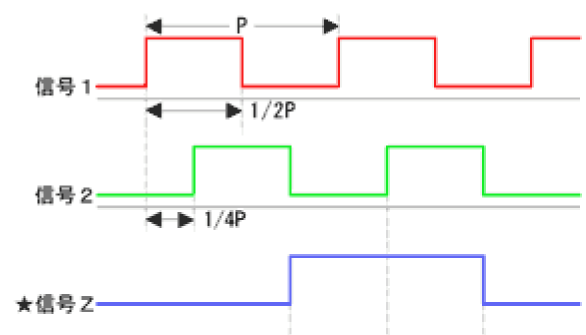
サイン波をPWM
で生成する原理



ロータリーエンコーダ

- 回転角や回転速度をパルス列として出力するセンサ
- 原理的にノイズは発生しない
- デジタル回路による制御に適す

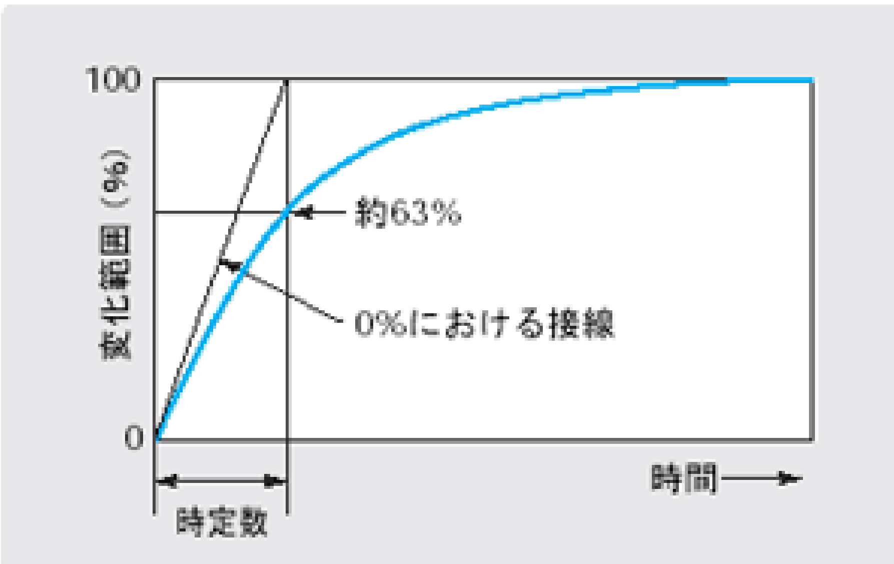




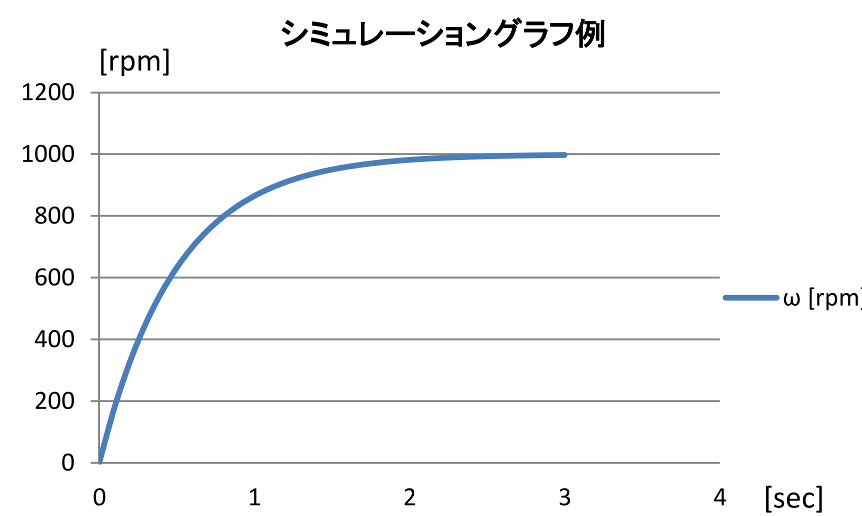
軸方向から見て右回転（CW）の時

ここで実験1
モータの「素」の特性を知る

課題3，課題4　独立に回答しても合わせた形で回答しても良い．



$$y(t) = K\left(1 - e^{-\frac{t}{T}}\right)$$



T：時定数（1-（1/e）≒63.2%になるまでの時間）

K：ゲイン係数

4～5T：整定時間

Kは出力（回転数）と入力（電圧）の比
K=N[rpm]/E[v]を求めてみよう．
K=ω[rad/s]/ E[v]でもよいが結果は変わらない

３．制御系を作る

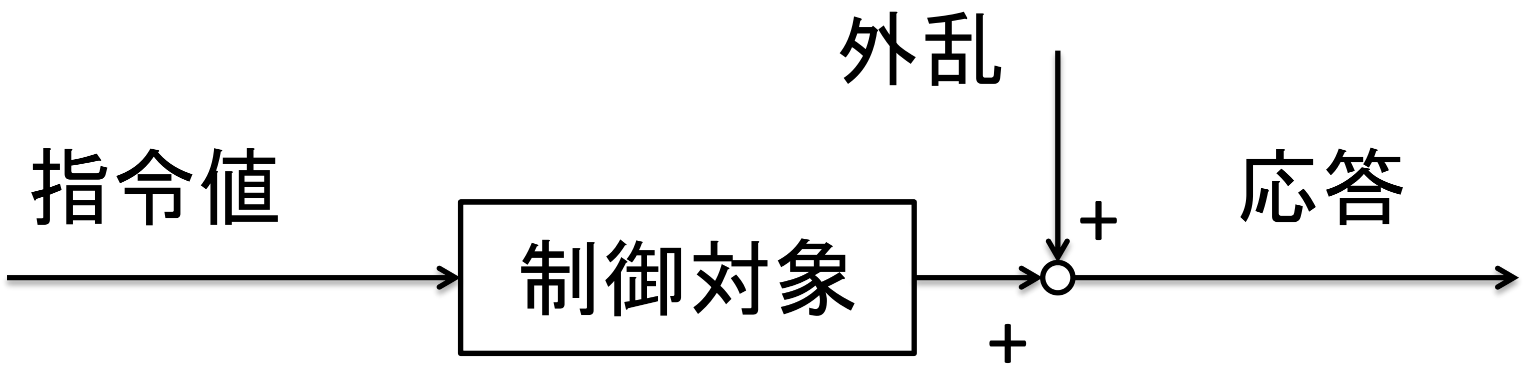
オープンループ制御系

利用できる条件

1. 入力（指令値）に対する出力（応答）の関係が一意に決まる
2. 制御対象の特性が変わらない
3. 外乱の影響が僅少または，外乱の影響が深刻でないと判断できる場合

応答に一意性・再現性がある

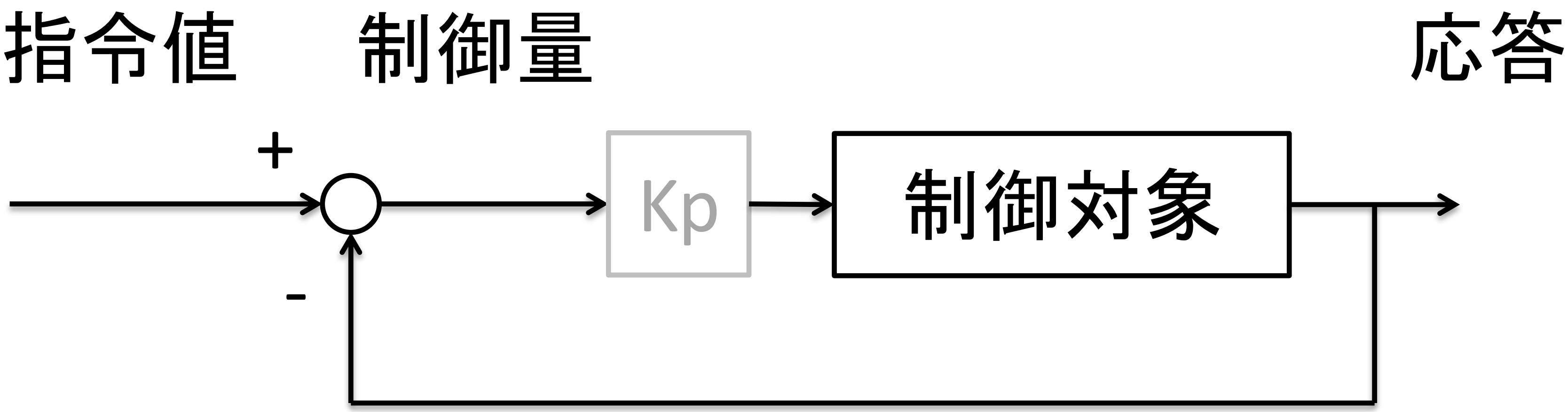
オープンループ制御系で起こる問題



- 外乱が加わってしまうと応答が変化
- 影響を取り除く手段がない

オープンループ制御系とは，<u>制御対象の素の特性を利用</u>する方法であるが，制御系であると考えた場合，必ずしも完全ではない．

単純フィードバック制御系（比例フィードバック制御系）



- 外乱の影響で応答が変わるような場合，応答（出力）と指令値（入力）を比較することで把握
- 入力ー出力による偏差を制御量として修正に利用

ジェームス・ワットの遠心調速機

1788年の発明：フィードバック制御の始祖

蒸気バルブ

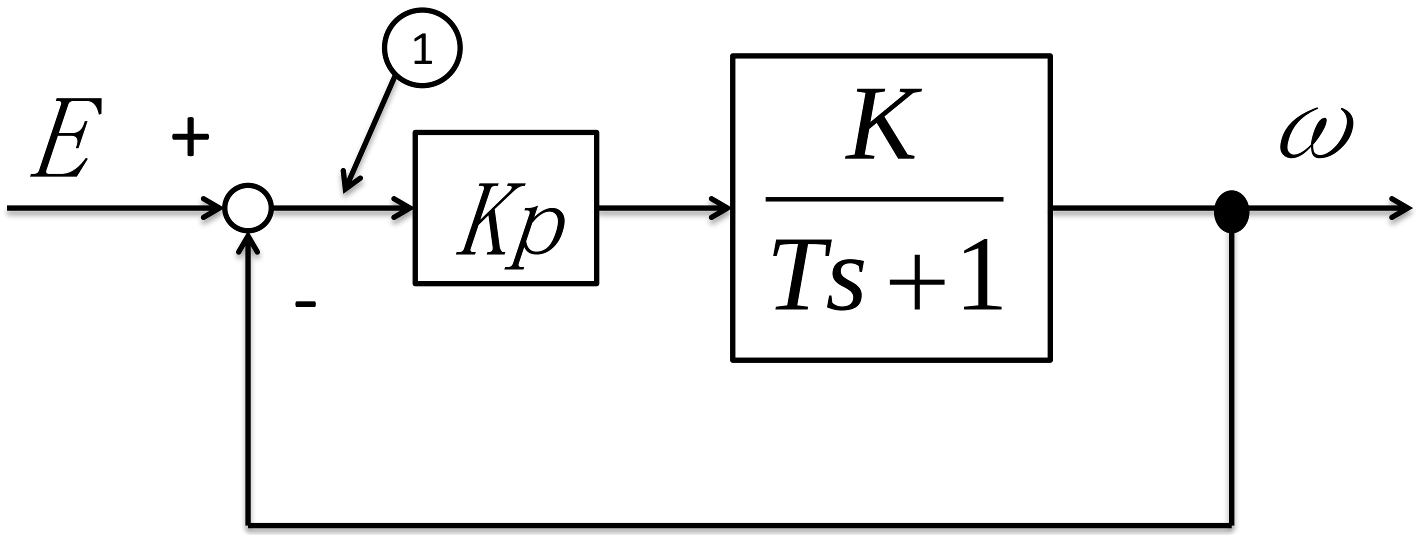
遠心調速機

蒸気エンジン

<u>その他の発明</u>

- 排気蒸気をシリンダ内で冷却することをやめ，シリンダ外で行うことで従来の4倍の効率を実現
- 蒸気機関の往復対向ピストンに蒸気を切り替えて送り込む機構により，従来の手動から自動化

ここで課題5



ヒント）以下の連立方程式を解けばよい.

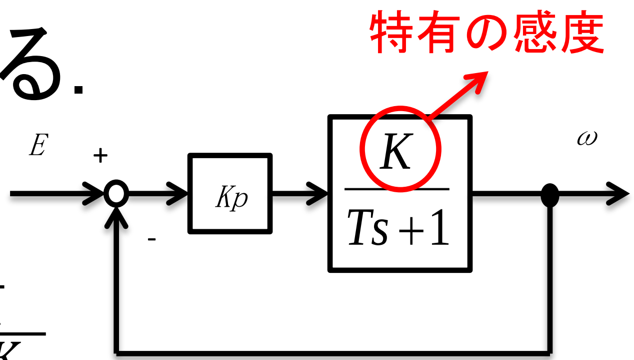
①＝？？？
ω＝？？？

レポートにはもとの伝達関数と比べてどのように変化したか示しておくこと

４．制御系の設計・改善

単純比例フィードバック制御の問題

1. 特有の感度が低すぎて応答が遅い.
2. 特有の感度が高すぎて発振する.
3. 定常偏差(ズレ)が生じる.



これがあるので制御器，制御対象の感度(ゲイン)が低いと誤差が増えることがある

$$G_c(s) = \frac{\frac{K_p K}{Ts+1}}{1 + \frac{K_p K}{Ts+1}}$$

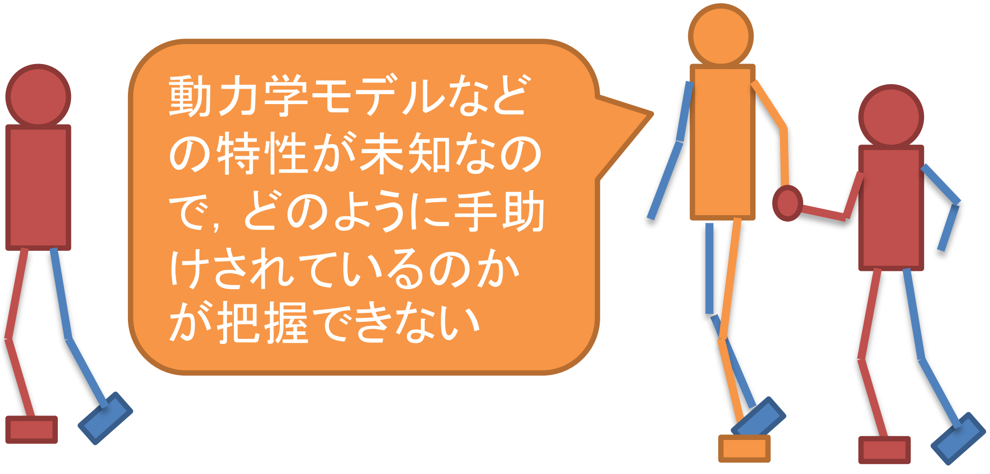
$$y(t) = K_p K\left(1 - e^{-\frac{t}{T}}\right) \Longrightarrow y(t) = \frac{K_p K}{1 + K_p K}\left(1 - e^{-\frac{t}{\frac{T}{1+K_p K}}}\right)$$

4. 外乱あるいはモデル化していない要素の影響が大きくて，欲しい応答にならない.

必ずしも欲しい応答特性であるとは限らない

（余談）人間はロボットを手助けしているか

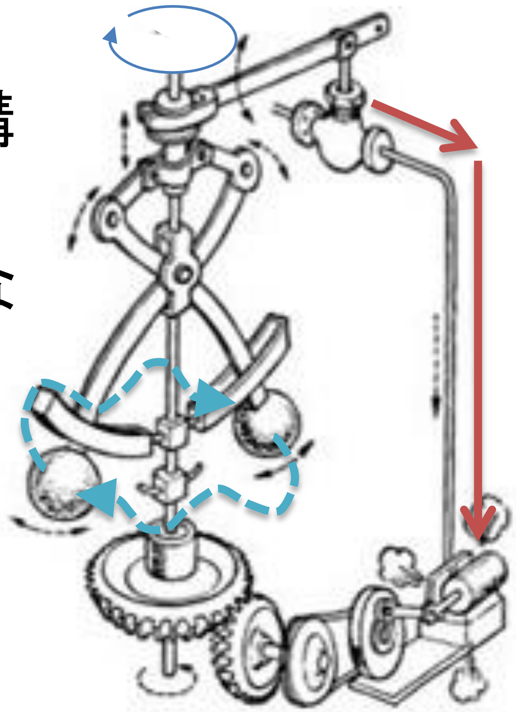
- 人間は手を繋ぐことで相手を支持する事が可能
- ロボットは，単体でバランスを取って，移動する仕組みだけの場合，手を繋ぐことで，姿勢を乱す予想外の外乱となる．



遠心調速機，実は制御的に問題がある

速度が安定しない

- 遠心力に対して機構の摩擦が少ない
- 錘が上下に振動しながら回り，バルブが振動的に開閉する
- ハンチングを起こし速度が振動的に変化を繰り返す



応答が遅れる

- バルブからエンジンまでの経路が長い
- バルブが調整されてもエンジンに辿り着くまで時間が掛かる
- 一定時間は応答できず，タイムラグを伴って応答する

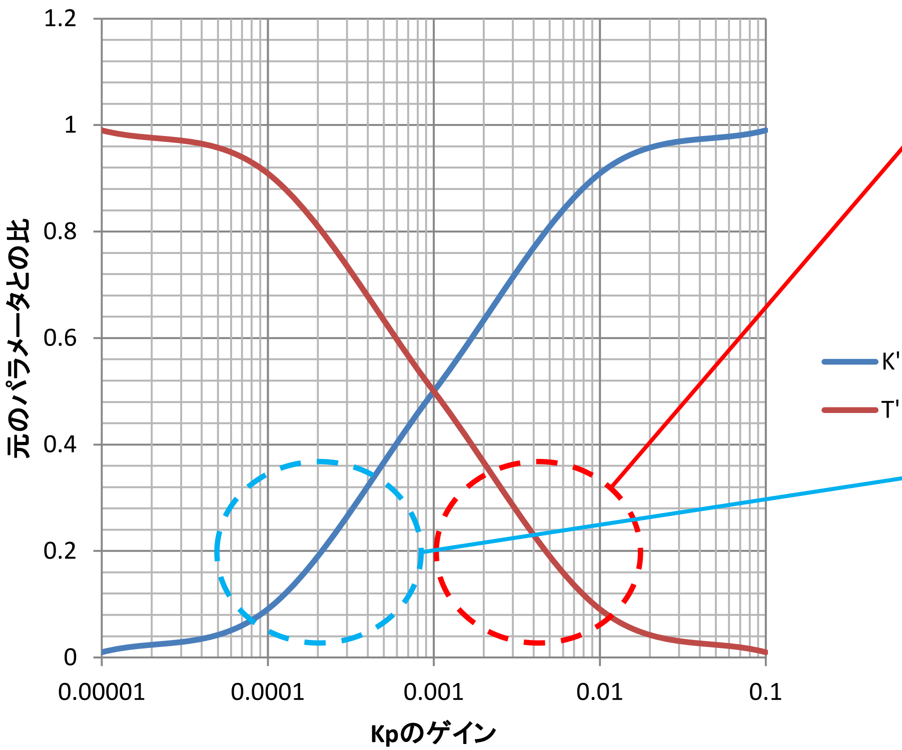
制御の安定性（感度）の問題

制御の無駄時間の問題

ここでもう一度，単純フィードバックを考えてみよう

$$G_c(s)=\frac{\dfrac{K_pK}{Ts+1}}{1+\dfrac{K_pK}{Ts+1}}=\frac{\dfrac{K_pK}{1+K_pK}}{\dfrac{T}{1+K_pK}s+1}=\frac{K'}{T's+1}$$

<u>例</u>
10Vで10000rpmのモータの場合
K=1000[rpm/V]となる



Kpを上げると応答が速くなる
時定数が1/5（元の5倍速く収束）の
実システムが実現できるだろうか？

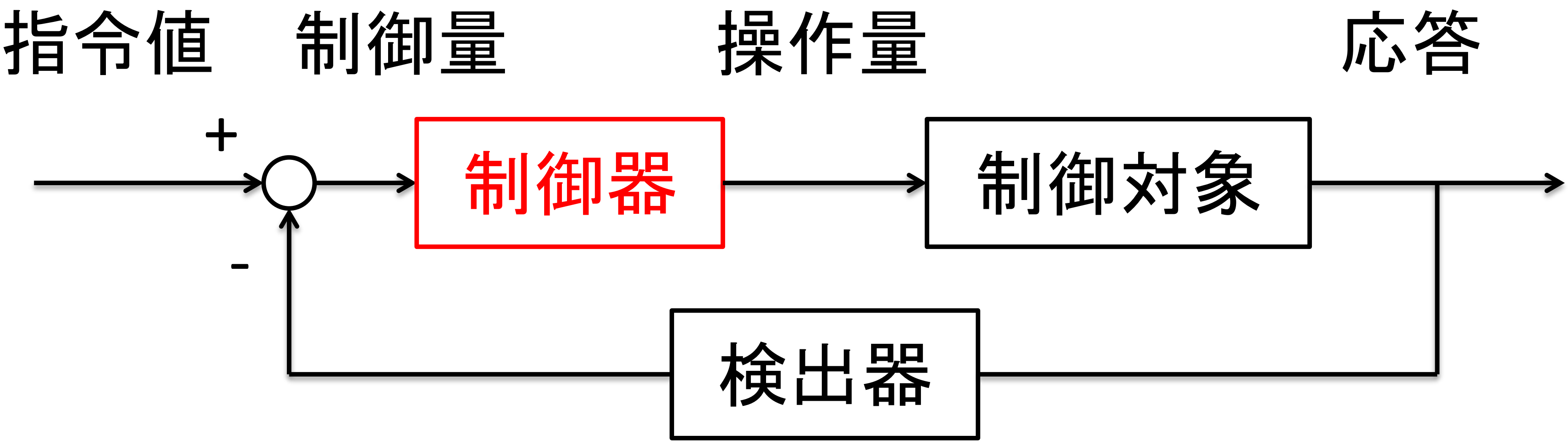
⇩

Kpを下げると現実的な時定数
（80%でちょっと速い）になる
応答の感度が1/5に下がった実
システムが役立つだろうか？

⇩

単純フィードバックだけではダメな気がしないだろうか？

フィードバック制御系の設計



<u>目指す性質</u>

- 制御対象の安定性向上
- 変化する目標値への追従性向上
- 外乱の影響の抑制
- 制御対象の特性変動による影響の抑制

PIDコントローラ

制御対象特有の特性に
合わせるゲイン係数

➡ 隠れていてわからない
ことがあるので注意

$$G_C(s) = K_n\left(K_P + K_I\frac{1}{s} + K_D s\right)$$

通常は()内を
考える

比例要素
制御系の基本となる
感度を調整する要素

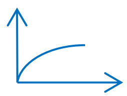
積分要素
応答の誤差を修正す
るように働く要素

微分要素
入力の変化に追従
するように働く要素

各種制御器と伝達関数

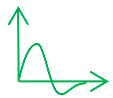
P制御器+1次遅れ

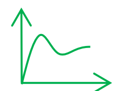
$$G(s)=\frac{\frac{K_pK}{1+K_pK}\,1}{\frac{T}{1+K_pK}\,s+1}$$



PI制御器+1次遅れ

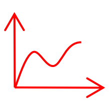
$$G(s)=\frac{s+\frac{K_PK}{T}\,1}{s^2+(T+K_PK)s+\frac{K_IK}{T}\,1}$$

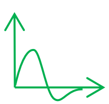


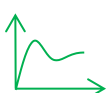


PID制御器+1次遅れ

$$G(s)=\frac{\frac{K_DK}{T+K_DK}s^2+\frac{K_PK}{T+K_DK}s+\frac{K_IK}{T+K_DK}\,1}{s^2+\frac{1+K_PK}{T+K_DK}s+\frac{T+K_DK}{K_IK}\,1}$$

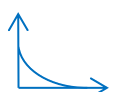


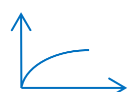




PD制御器+1次遅れ

$$G(s)=\frac{\frac{K_DK}{1+K_PK}s+\frac{K_PK}{1+K_PK}\,1}{\frac{T+K_DK}{1+K_PK}s+1}$$

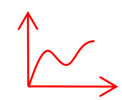




性質

1. 分母の形で大まかな応答が決まる．→1次遅れ，2次遅れ・・・
2. 分母の次数≧分子の次数でなければ安定に制御できない
3. 分子の次数は少ないほど制御しやすいことが多い

PIコントローラ

- PID制御器からD（微分）要素を省いたもの
- D要素は速応性を上げる反面，発散・不安定にすることもある扱いにくい要素  ←この成分
- D要素がなくても定常偏差を減らし，ゲイン（感度）を上げられるので，速応性をある程度上げられる

※但し，サーボ系など速応性（追従性）を重視する場合，PD制御を行う場合もある．

PID制御の調整法

1	限界感度法	発振を始める条件から設計
	発振するまでゲインK_cを上げ，その時の応答周期T_cを用いて制御器を調整	
	一周期で振幅が1/4に減衰するゲイン$K_{1/4}$を調整し，その時の応答周期$T_{1/4}$を用いて制御器を調整	
2	ステップ応答法	ステップ応答から設計
	ステップ入力時の無駄時間Lと時定数T（または反応速度R）を用いて調整	
3	試行錯誤法	実際に応答させて調整
	現場における調整法（決してゼロからではない）	

限界感度法

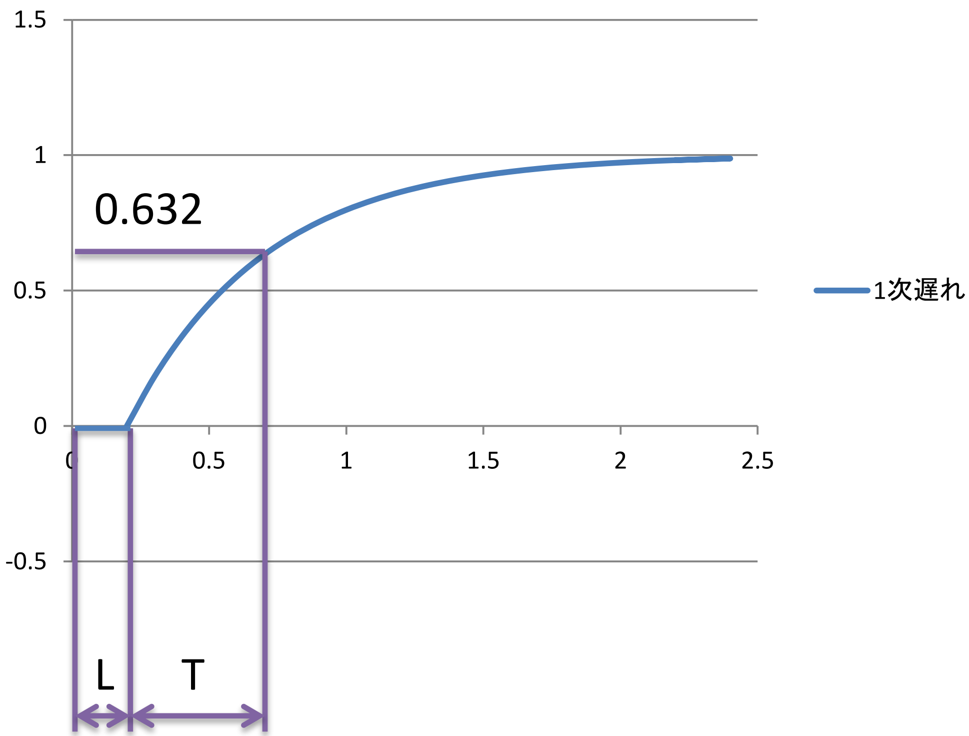
1. 通常の限界感度法
   - 振動を始めたときのゲインKc
   - 振動の周期Tc[sec]
2. ¼減衰法
   - 発振は危ないので2波目で1/4に減衰した値を利用

制御器	比例ゲインKp	積分時間TI	微分時間TD
P	0.5Kc	-	-
PI	0.45Kc	0.833Tc	-
PID	0.6Kc	0.5Tc	0.125Tc

制御器	比例ゲインKp	積分時間TI	微分時間TD
PID	$K_{(1/4)}$	$T_{(1/4)}/1.5$	$T_{(1/4)}/6$

TcやT$_{(1/4)}$は，時定数ではなく，周期であることに注意

ステップ応答法(1)
無駄時間　＋　1次遅れの系を前提



一般的な応答を無駄時間＋1次遅れの系と仮定

– 無駄時間：　初期に全く応答しない時間**L**を計測
– 1次遅れ：時定数**T**を計測

高次系などはこの応答から外れることもあるが，それらしい値を算出して設計する（再調整は必要であるのが基本）

ステップ応答法(2)

1. Ziegler and Nicholsの調整則

制御器	比例ゲインKp		積分時間TI	微分時間TD
P	1/(RL)	T/L	-	-
PI	0.9/(RL)	0.9T/L	3.33L	-
PID	1.2/(RL)	1.2T/L	2L	0.5L

2. Chien, Hrones and Reswickの調整則

制御器		比例ゲインKp		積分時間TI	微分時間TD
行き過ぎなし	P	0.3/(RL)	0.3T/L	-	-
	PI	0.35/(RL)	0.35T/L	1.2T	-
	PID	0.6/(RL)	0.6T/L	T	0.5L
行き過ぎ20%	P	0.7/(RL)	0.7T/L	-	-
	PI	0.6/(RL)	0.6T/L	T	-
	PID	0.95/(RL)	0.95T/L	1.35T	0.47L

3. Cohen and Coonの調整則

制御器	比例ゲインKp	積分時間TI	微分時間TD
P	1/(RL) + 1/(3TR)	-	-
PD	5/(4RL) + 1/(6TR)	-	(6L - 2(L^2/T))/(22 + 3(L/T))
PI	9/(10RL) + 1/(12TR)	(30L + 3(L^2/T))/(9 + 20(L/T))	-
PID	4/(3RL) + 1/(4TR)	(32L + 6(L^2/T))/(13 + 8(L/T))	4L/(11 + 2(L/T))

試行錯誤法

1. 現場での調整可能がメリット.
2. ゼロからの調整ではなく，設計値の調整に使う.
3. 無駄時間Lの正確な計測は困難なことがある.
4. 制御対象特有の隠れた係数（ゲインKn）が分からないことも多々あるため，設計しても実動作による**試行錯誤**が必要.

$$G_c(s) = K_nK_p\left(1 + \frac{1}{T_Is} + T_Ds\right) = K_n\left(K_p + \frac{K_p}{T_Is} + K_pT_Ds\right) = \overbrace{K_nK_p}^{K_p} + \overbrace{\frac{K_nK_I}{s}}^{K_I} + \overbrace{K_nK_D}^{K_D}s$$

制御器のK_p, K_I, K_Dを入力する際，上記のK_nを掛けた値を入力する必要がある.

K_p, K_I, K_Dとして調整する際，K_nの分だけ連動する.

制御器の調整は簡単ではない

1. PID（PI）制御は数学的にきれいな方法ではない．
   1. 経験則として算出されてきたが，前提条件に合わなければ要調整
   2. 積分ゲインを上げ過ぎると誤差に対してどこまでも応答を続けようとして応答が収束しない
   3. 微分ゲインを上げ過ぎるとわずかの誤差でも大きく応答して振動的になる
2. ハードウェアの特性によって大きく変わるシステム固有の不明な比例定数Knがある

無視したら意外と影響の大きかった要素，工作・組み立て精度に依存する非線形要素の影響が大きい，など．

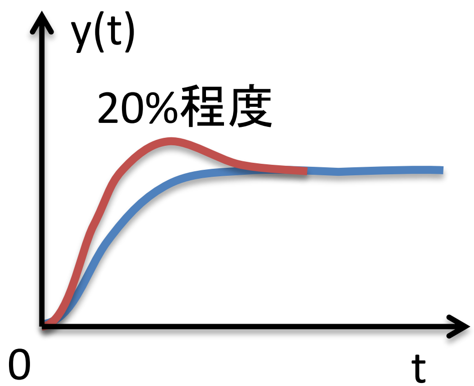
以上から必ずしも設計したパラメータ通りの調整をしても上手くいかないことがごく普通に起こることを体験して欲しい．

本日の実験では

- ステップ応答法を基に試行錯誤して調整する

制御器	比例ゲインKp		積分時間TI	微分時間TD
P	1/(RL)	T/L	-	-
PI	0.9/(RL)	0.9T/L	3.33L	-
PID	1.2/(RL)	1.2T/L	2L	0.5L

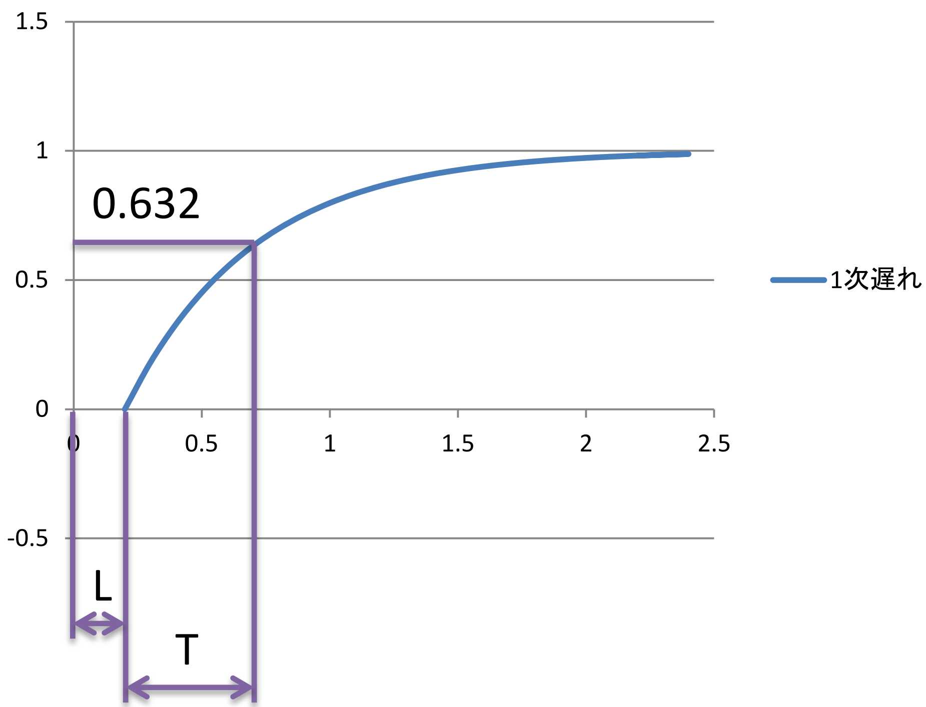
PI制御を掛けると2次遅れの応答となる

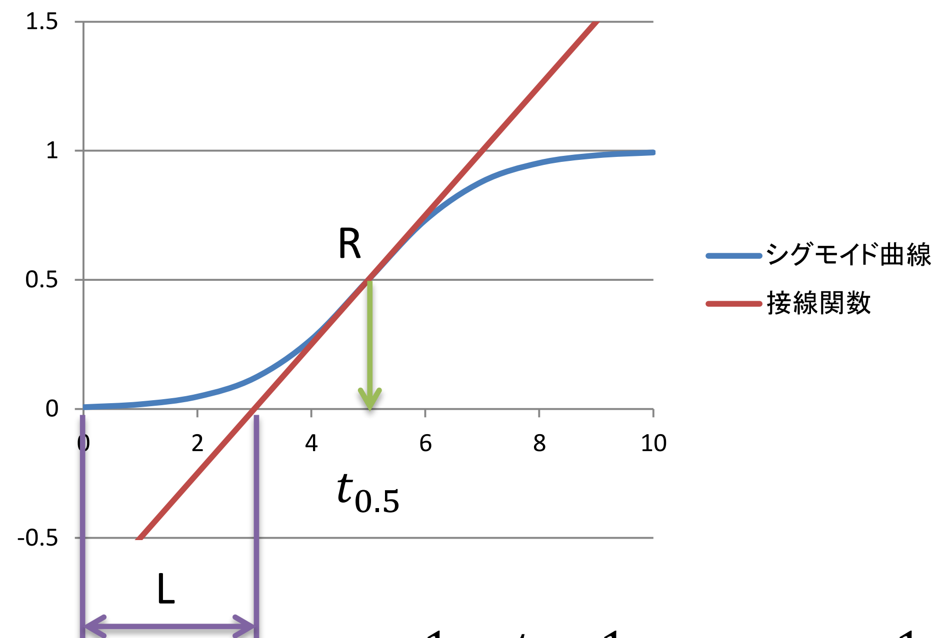


制御器		比例ゲインKp		積分時間TI	微分時間TD
行き過ぎなし	P	0.3/(RL)	0.3T/L	-	-
	PI	0.35/(RL)	0.35T/L	1.2T	-
	PID	0.6/(RL)	0.6T/L	T	0.5L
行き過ぎ20%	P	0.7/(RL)	0.7T/L	-	-
	PI	0.6/(RL)	0.6T/L	T	-
	PID	0.95/(RL)	0.95T/L	1.35T	0.47L

制御器	比例ゲインKp	積分時間TI	微分時間TD
P	1/(RL) + 1/(3TR)	-	-
PD	5/(4RL) + 1/(6TR)	-	(6L - 2(L^2/T))/(22 + 3(L/T))
PI	9/(10RL) + 1/(12TR)	(30L + 3(L^2/T))/(9 + 20(L/T))	-
PID	4/(3RL) + 1/(4TR)	(32L + 6(L^2/T))/(13 + 8(L/T))	4L/(11 + 2(L/T))

シグモイド曲線と無駄時間+1次遅れ





$$R \approx \frac{1}{T} e^{-\frac{t}{T}} \cong \frac{1}{T} e^{-0.69315} = \frac{1}{2T}$$

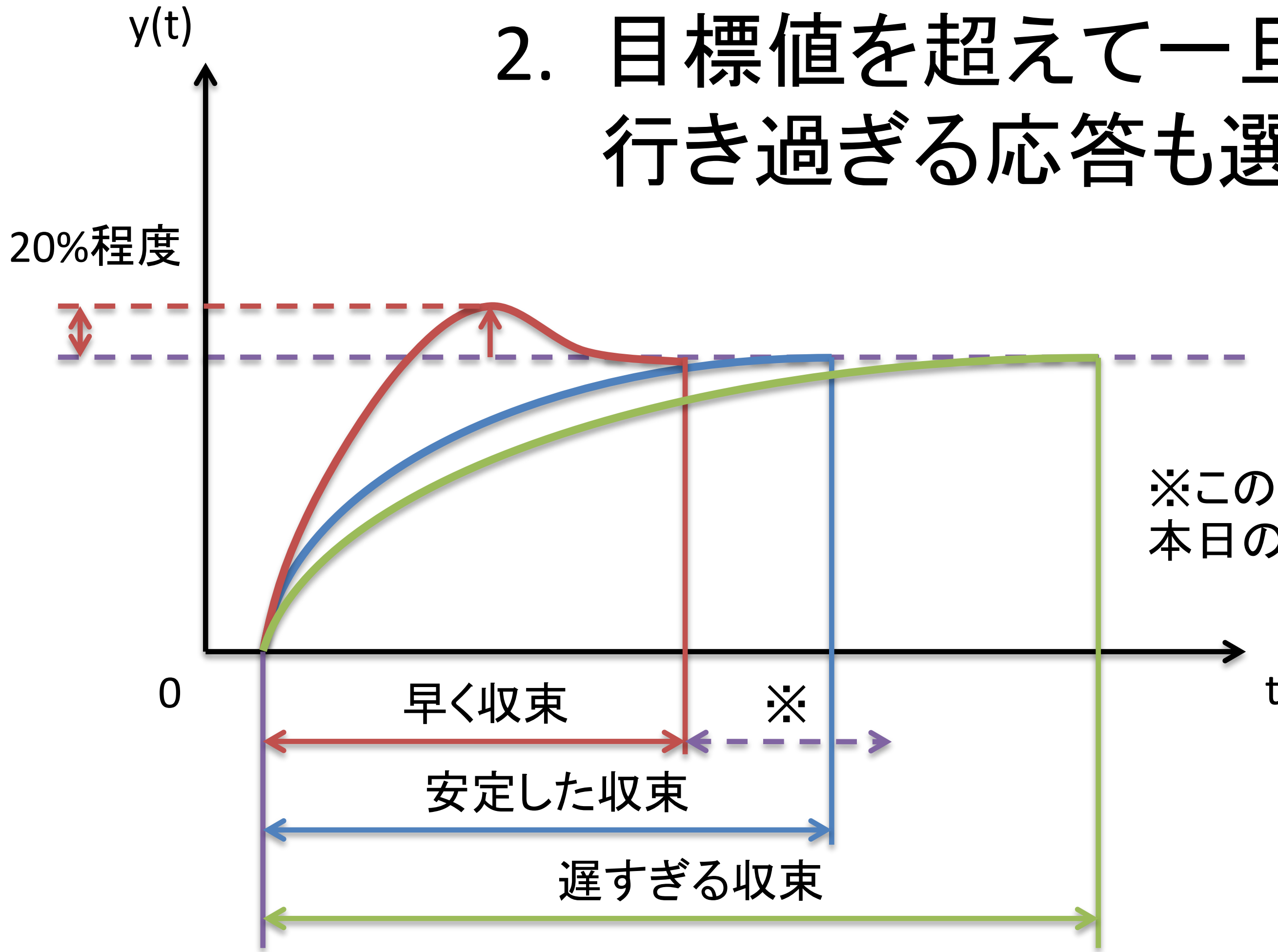
$$y_t = Rt + (0.5 - Rt_{0.5})$$

$$L \approx -\frac{0.5}{R} + t_{0.5}$$

- そもそもが異なる応答であるので，応答波形を見ながら近い方を選定すればよい．
- 右の式は目安程度だが，試してみて良好な結果が得られるか確かめてもよい．

行き過ぎ（オーバーシュート）

1. 基本は一次遅れの応答の範囲
2. 目標値を超えて一旦20%程度行き過ぎる応答も選択可能



※この範囲に収まるように本日の実験では調整

ここで実験2

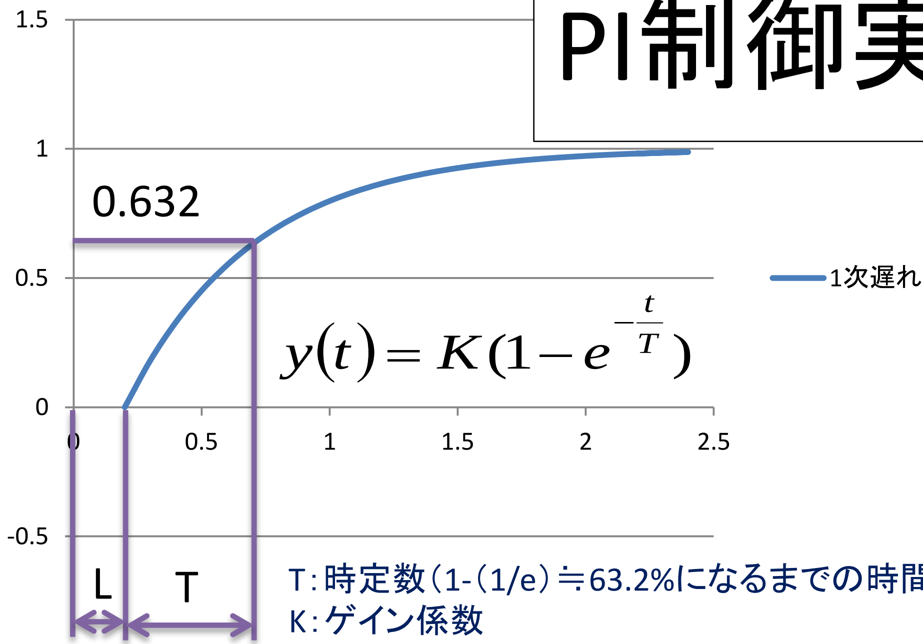
課題6

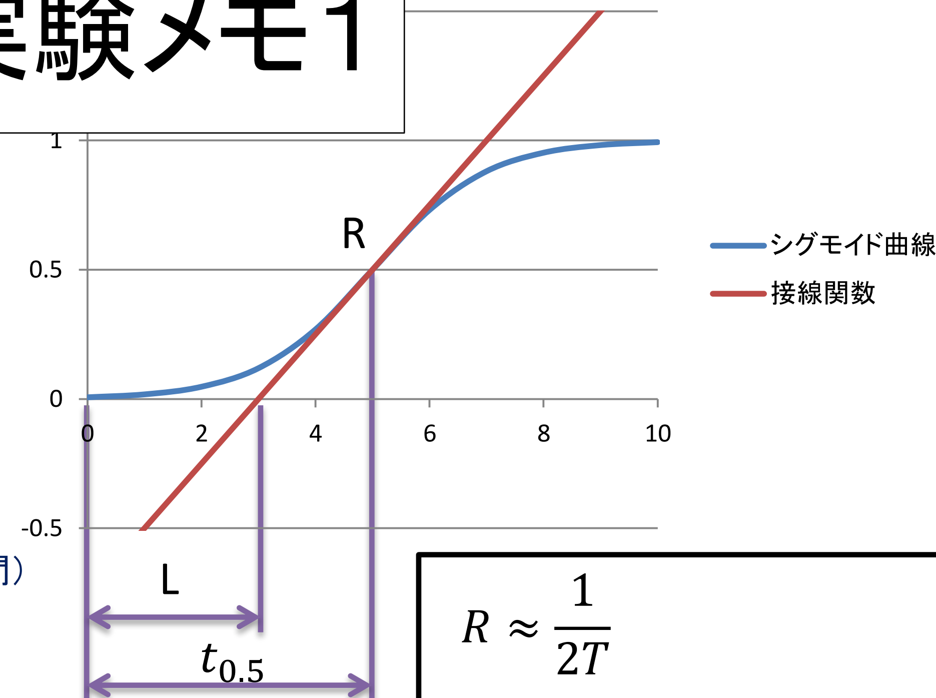
ここで実験３

課題7，課題８

レポートにする場合，独立に回答しても合わせた形で回答しても良い．

PI制御実験メモ1





T：時定数（1-（1/e）≒63.2%になるまでの時間）
K：ゲイン係数
4～5T：整定時間

$$R \approx \frac{1}{2T}$$

$$y_t = Rt + (0.5 - Rt_{0.5})$$

$$L \approx -\frac{0.5}{R} + t_{0.5} = -T + t_{0.5}$$

手法	比例ゲインKp		積分ゲインKI
Ziegler and Nichols	$\frac{0.9}{RL}$	$\frac{0.9T}{L}$	$\frac{K_p}{3.33L}$
Chien, Hrones and Reswick	$\frac{0.35}{RL}$	$\frac{0.35T}{L}$	$\frac{K_p}{1.2T}$
Cohen and Coon	$\frac{0.9}{RL} + \frac{1}{12TR}$		$\frac{K_p(9T + 20L)}{30TL + 3L^2}$

制御器のKp, KIを入力する際，Knを掛けて入力

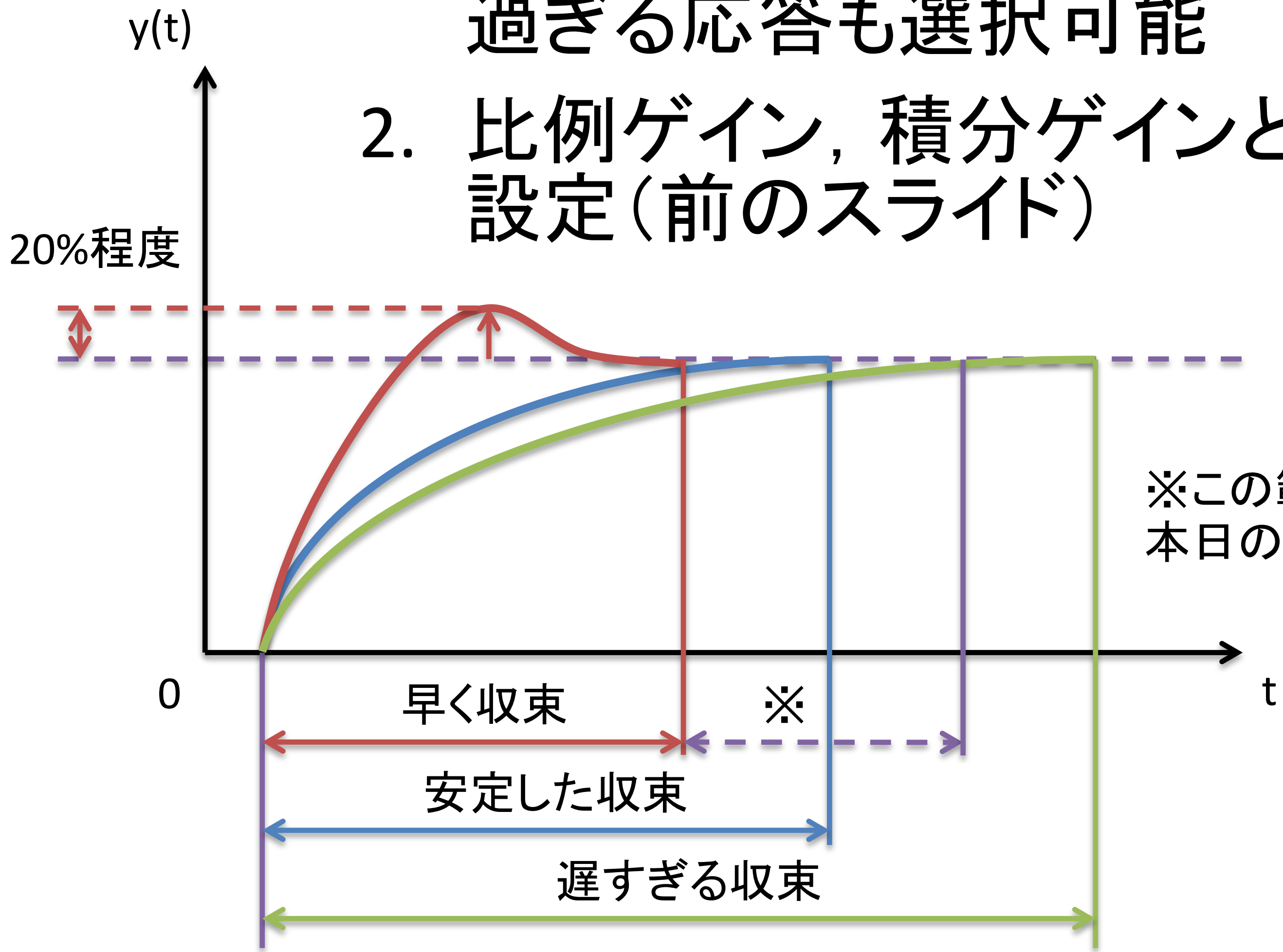
Kp, KIとして調整する際，Knの分だけ連動する．

$$G_c(s) = K_n K_p \left(1 + \frac{1}{T_I s}\right) = K_n \left(K_p + \frac{K_p}{T_I s}\right) = K_n K_p + \frac{K_n K_I}{s}$$

PI制御実験メモ2

PI制御により1次遅れの特性の制御対象の応答が2次遅れに変化している点にも注目

1. 目標値を超えて一旦20%程度行き過ぎる応答も選択可能
2. 比例ゲイン，積分ゲインとも高目に設定（前のスライド）



※この範囲に収まるように本日の実験では調整

５．実験後のまとめ・解説

モータの制御器の調整の意義

<u>目的</u>
- モータ特性と負荷特性のマッチング

<u>手段</u>
- 元々のモータの特性を変換して目的を実現するのが制御器の役割

<u>効果</u>
- 時定数を調整（追従性）
- 出力特性を調整

高性能モータに対して	汎用モータに対して
高すぎる出力を抑える	-
高すぎる速応性を抑える	-
-	弱い出力に対し摩擦等の外乱の影響を抑える

目的（応用，アプリケーション，使い道）に応じて，適切な特性のモータを選定した上で制御をかけるのが最も効果的なメカトロニクス系の運用法である.

制御とは抑制して御する（思い通りに動かす）という意味通り
- 素の対象に仕組みを用いて思い通りの特性にはできる
- それ以上のことができるわけではない

良いモータを使う意義

付加価値	決定要素
強力なトルク	大電流に耐える太い巻線（コイル）
	精度の良いコイル巻き付け（コアレス構造）
	精度の良い磁気回路（マグネット形状・材質）
高速応答	低イナーシャ（コアレス構造）
	低摩擦軸受
極低速時の滑らかな回転	低摩擦軸受
	角度依存性の低い磁気回路
	低摩擦のブラシ材質
高いエネルギ効率	精度の良い磁気回路
	低イナーシャ
	低摩擦軸受
耐久性・信頼性	各部材質，加工精度，組み立て精度