

国立大学法人

長岡技術科学大学

Nagaoka University of Technology

Ohishi Lab.

Nagaoka University of Technology



電気電子情報工学課程3年1学期

電気電子情報工学実験I

「サーボモータの制御」

長岡技術科学大学 電気電子情報工学専攻
助教 横倉 勇希



自己紹介

- 所 属 : 長岡技術科学大学 電気電子情報工学専攻 助教
- 氏 名 : 横倉 勇希
- 研究分野 :
モーションコントロール, モータドライブ, パワーエレクトロニクス,
メカトロニクス, ロボティクス, ハaptiクス, 制御工学
- 所属学会 :
 - 電気学会 産業応用部門(D部門)
 - Institute of Electrical and Electronics Engineers(IEEE),
Industrial Electronics Society(IES)
 - 日本ロボット学会(RSJ)
- Web :
<http://www.sidewarehouse.net/>

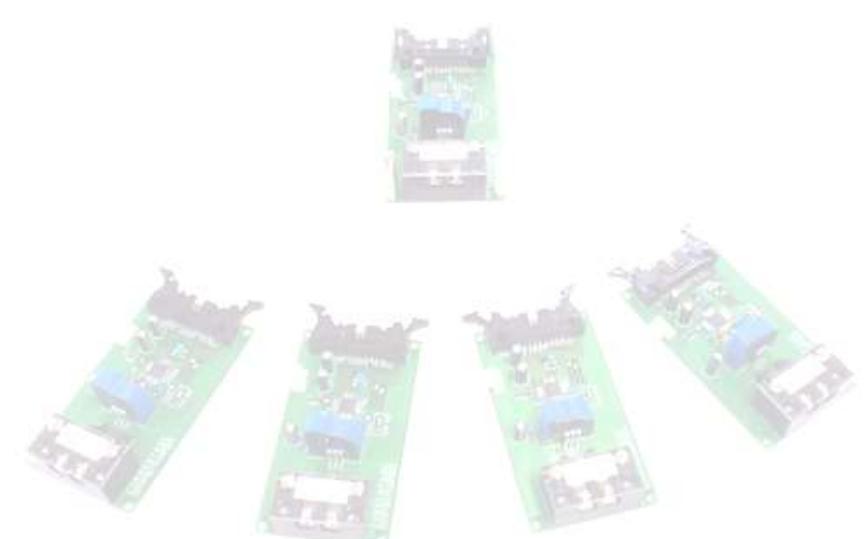
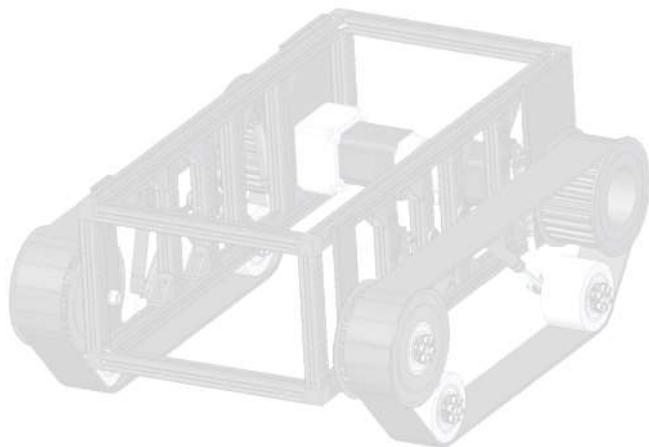
本日の講義内容

- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- ディジタル制御系での積分器の作り方
- 実験機ハードウェアの説明
- 実験の流れ（1日目）
- 実験の流れ（2日目）
- レポート課題

本日の講義内容

- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- ディジタル制御系での積分器の作り方
- 実験機ハードウェアの説明
- 実験の流れ（1日目）
- 実験の流れ（2日目）
- レポート課題

ACサーボモータとは？



ACサーボモータとは？

産業用ロボット

NCフライス盤

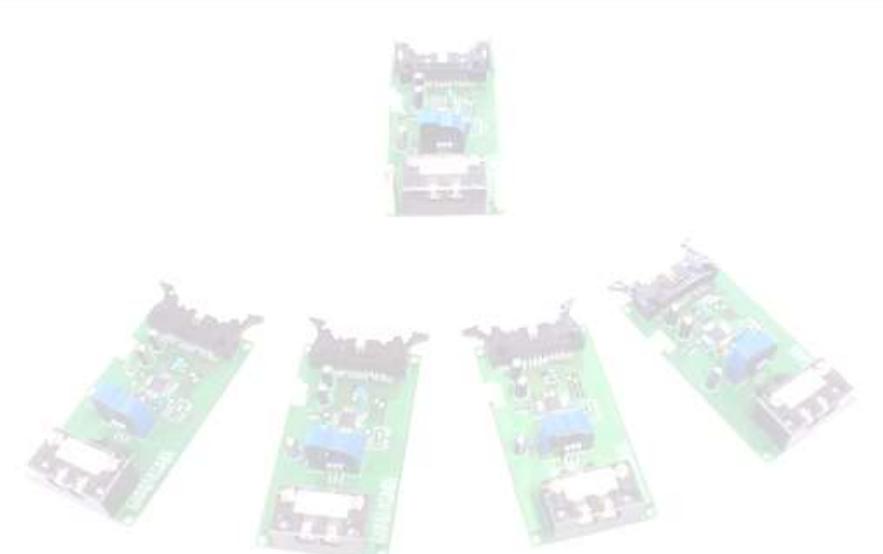
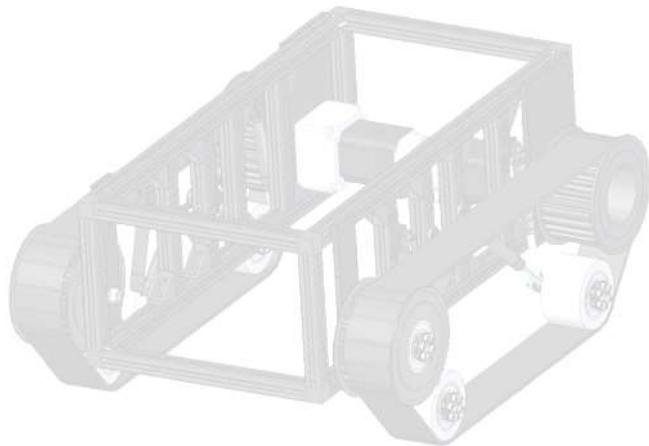
射出成形機

ACサーボモータ

チップマウンタ

半導体製造装置

ACサーボモータのモデル化



ACサーボモータのモデル化

モノを上手く制御するためのコツ

1. 制御したい対象を
理論的に可能な限り正確に表現する
2. やりたいことを可能にする、
その制御対象に適した制御器をくっつける
3. 制御器のゲインを理論的に計算する

職人技でゲインを人間が頑張って調整するなんて、言語道断！！

ACサーボモータのモデル化

モノを上手く制御するためのコツ

1. 制御したい対象を
理論的に可能な限り正確に表現する
2. やりたいことを可能にする、
その制御対象に適した制御器をくっつける
3. 制御器のゲインを理論的に計算する

職人技でゲインを人間が頑張って調整するなんて、言語道断！！

ACサーボモータのモデル化

■ ACサーボモータの構造

□ 永久磁石同期モータ

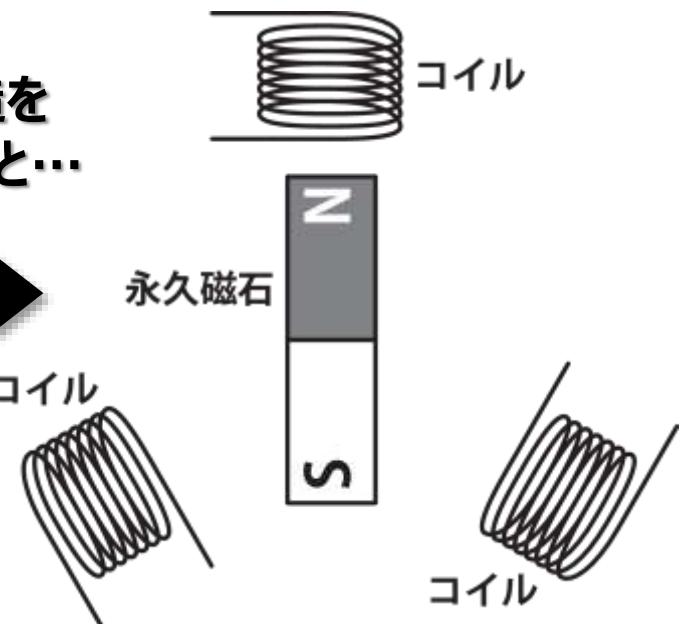
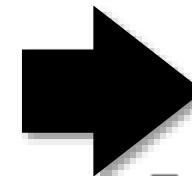
(PMSM Permanent Magnet Synchronous Motor)

□ ブラシレスDCモータ

(BLDC Motor Brushless DC Motor)



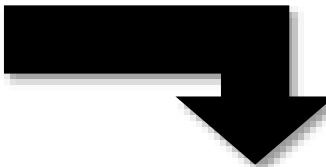
内部構造を
簡単に書くと…



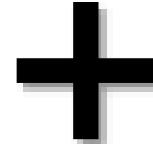
ACサーボモータのモデル化



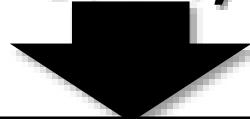
電気的モデル



機械的モデル



それぞれ個別に考えて、最後に合体！



ACサーボモータのモデル化

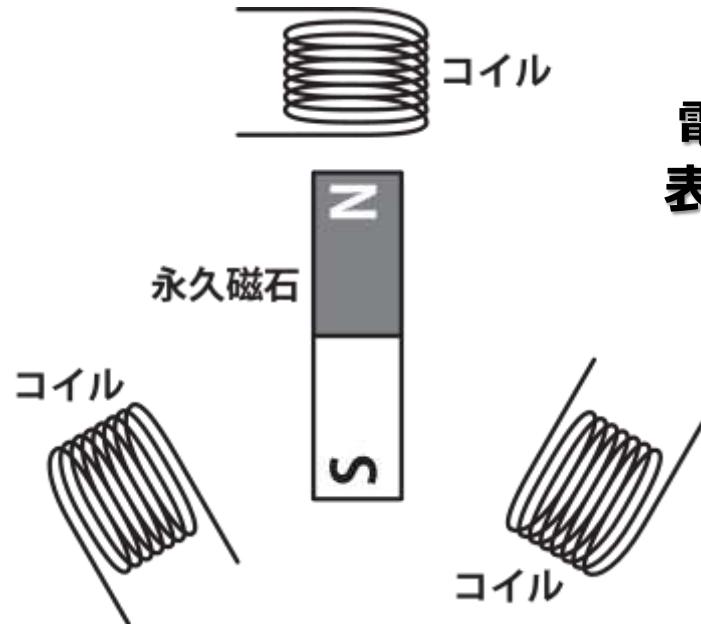


ACサーボモータのモデル化

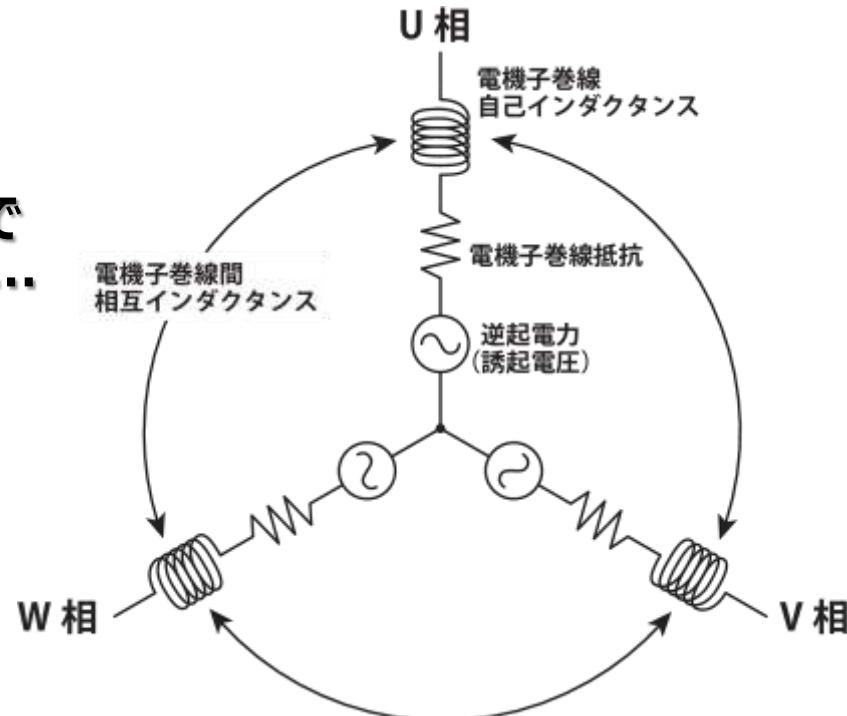
■ 電気的モデルの導出

- コイル状の導線で構成されている → インダクタンス成分
- 長い銅の導線で構成されている → 抵抗成分
- 永久磁石の磁束がコイルを横切る → 逆起電力成分

これらが
三相で結線



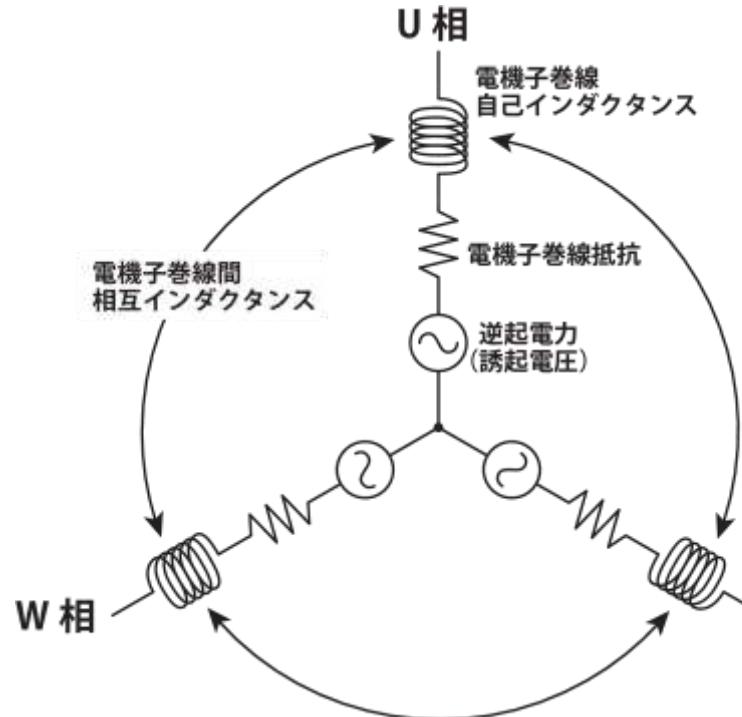
電気回路で
表現すると…



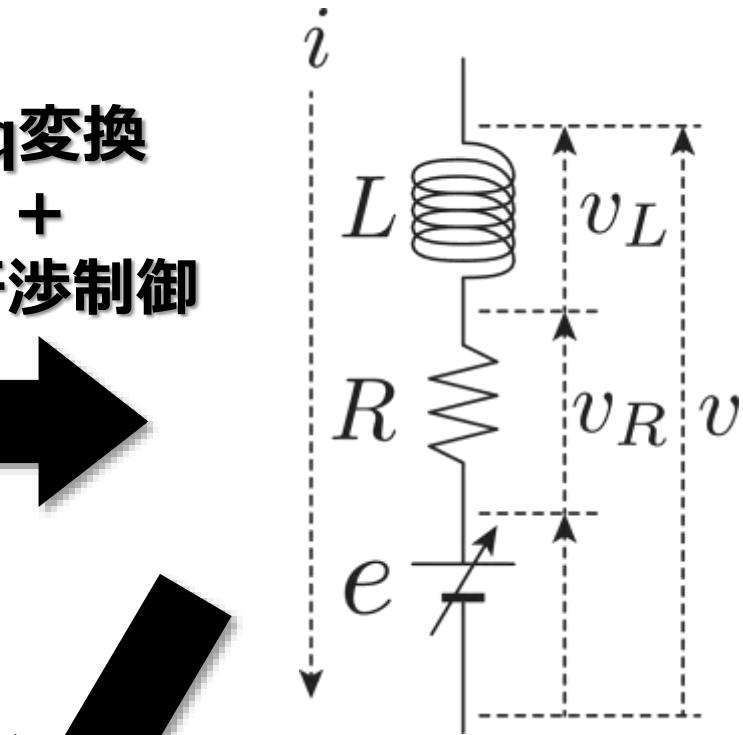
このままモデル化を進めていいても良いが、
このままだと複雑で面倒…

ACサーボモータのモデル化

■ 電気的モデルの導出



dq変換
+
非干渉制御

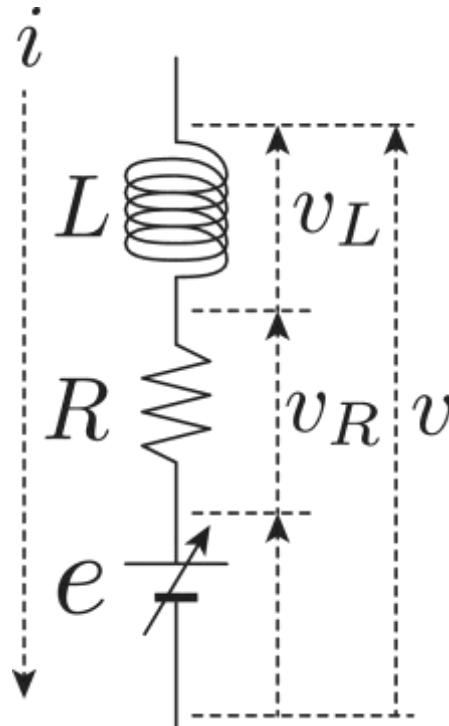


両端電圧 v は？

$$v = L \frac{di}{dt} + R_i + e$$

ACサーボモータのモデル化

■ 電気的モデルの導出



$$v(t) = L \frac{di(t)}{dt} + R_i(t) + e(t)$$

ラプラス演算子で表現すると…

$$\frac{d}{dt} \Leftrightarrow s$$

(ただし、すべての変数の初期値は零とする)

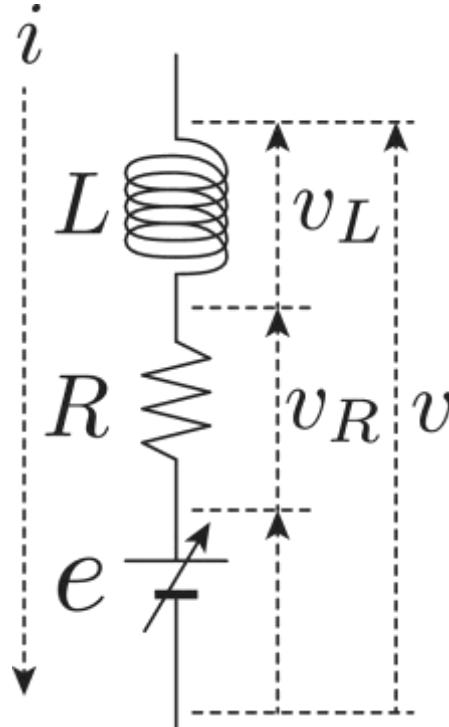
$$v(s) = Ls i(s) + R i(s) + e(s)$$

ちょっと変形

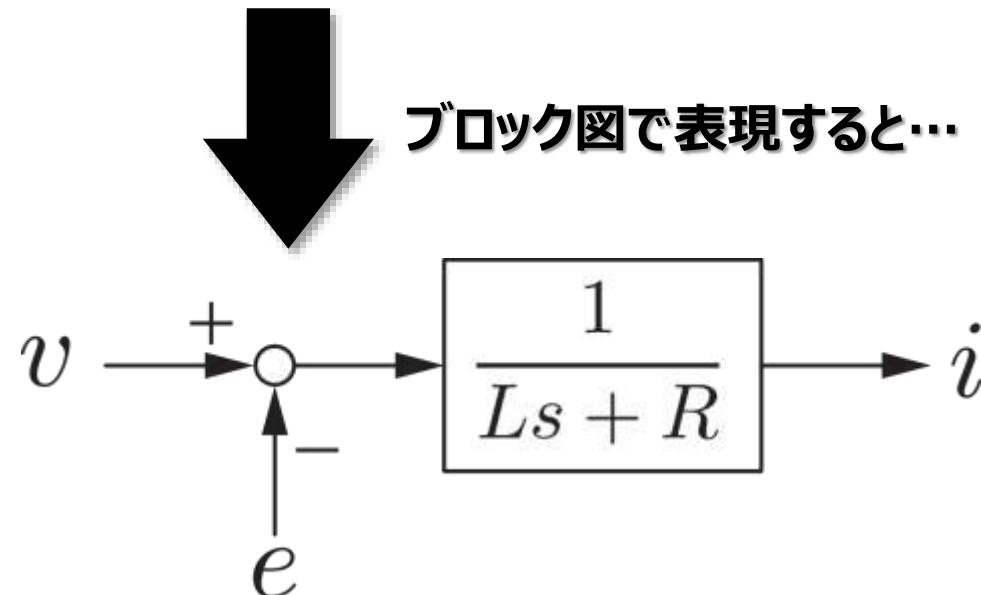
$$i(s) = \frac{1}{Ls + R} (v(s) - e(s))$$

ACサーボモータのモデル化

■ 電気的モデルの導出



$$\dot{i}(s) = \frac{1}{Ls + R} (v(s) - e(s))$$



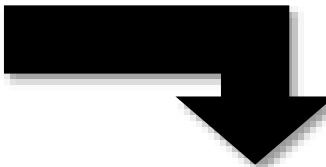
ブロック図で表現すると…

ACサーボモータの電気的モデルが完成！
(電圧を入力すると電流が出力されるモデル)

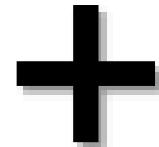
ACサーボモータのモデル化



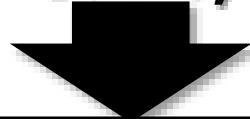
電気的モデル



機械的モデル



それぞれ個別に考えて、最後に合体！

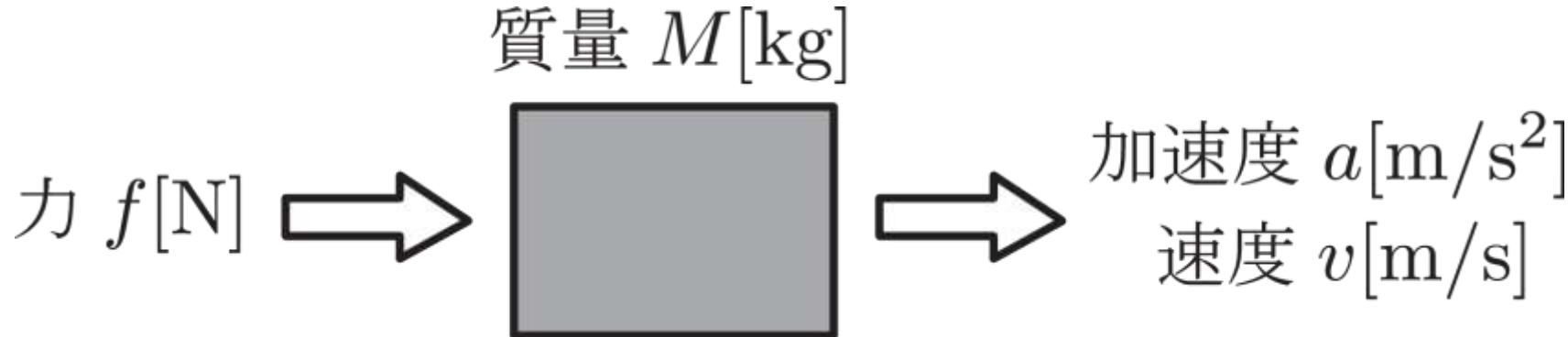


ACサーボモータ
全体のモデル

ACサーボモータのモデル化

■ 機械的モデルの導出

★ ただの運動方程式： 物理のおさらい I



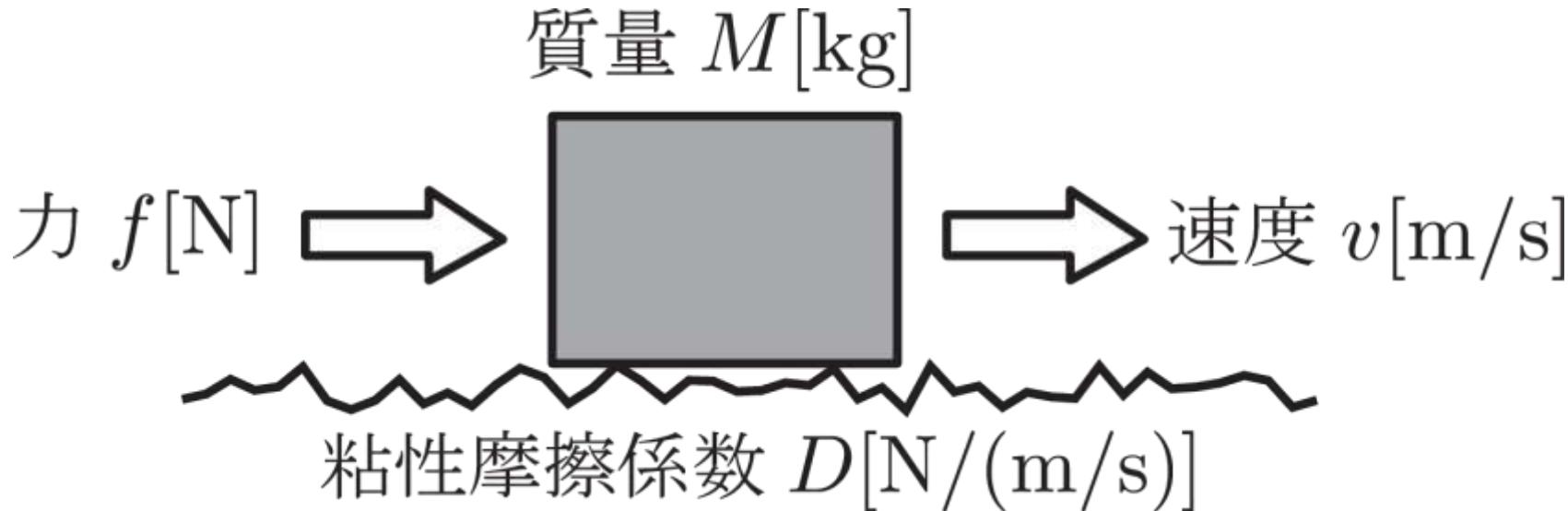
このときの力 f は？

$$\begin{aligned} f &= Ma \\ &= M \frac{dv}{dt} \end{aligned}$$

ACサーボモータのモデル化

■ 機械的モデルの導出

★ ただの運動方程式： 物理のおさらい II



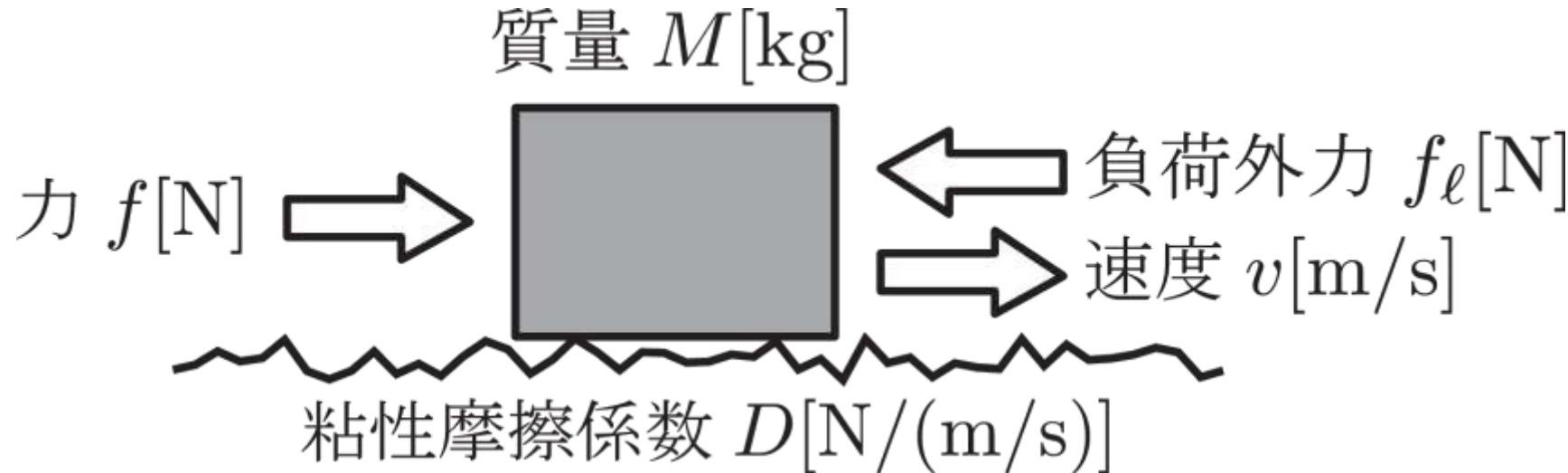
このときの力 f は？

$$f = M \frac{dv}{dt} + Dv$$

ACサーボモータのモデル化

■ 機械的モデルの導出

★ただの運動方程式：物理のおさらい III



このときの力 f は？

$$f = M \frac{dv}{dt} + Dv + f_e$$

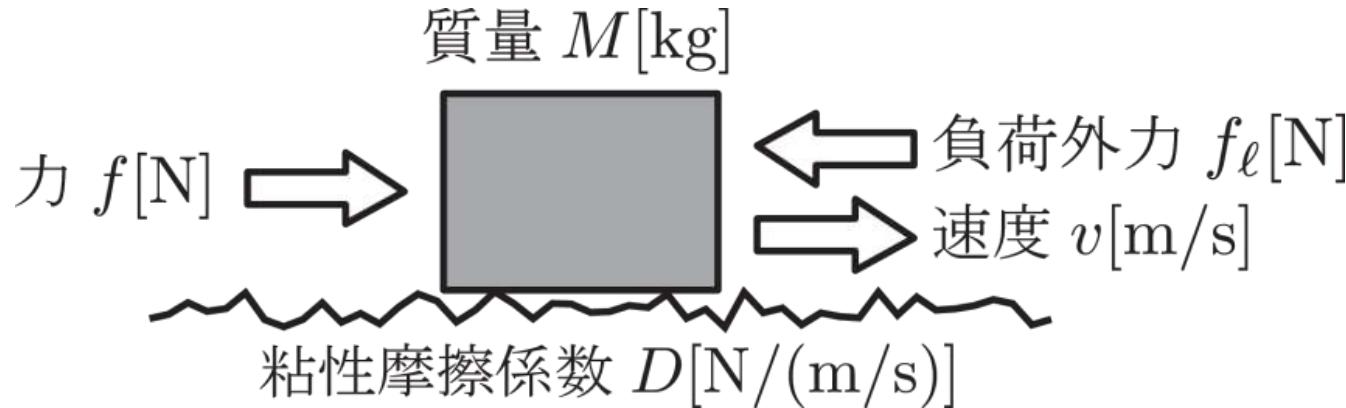
ただしこれは直動系の式。リニアモータならこれでよいが…

ACサーボモータのモデル化

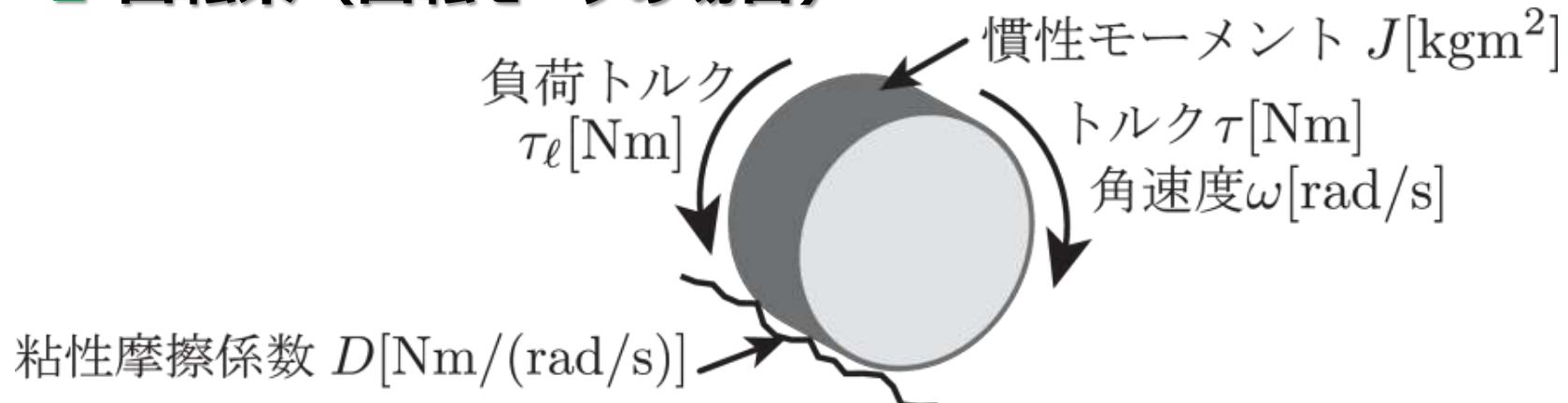
■ 機械的モデルの導出

★ ただの運動方程式：物理のおさらい IV

□ 直動系（リニアモータの場合）

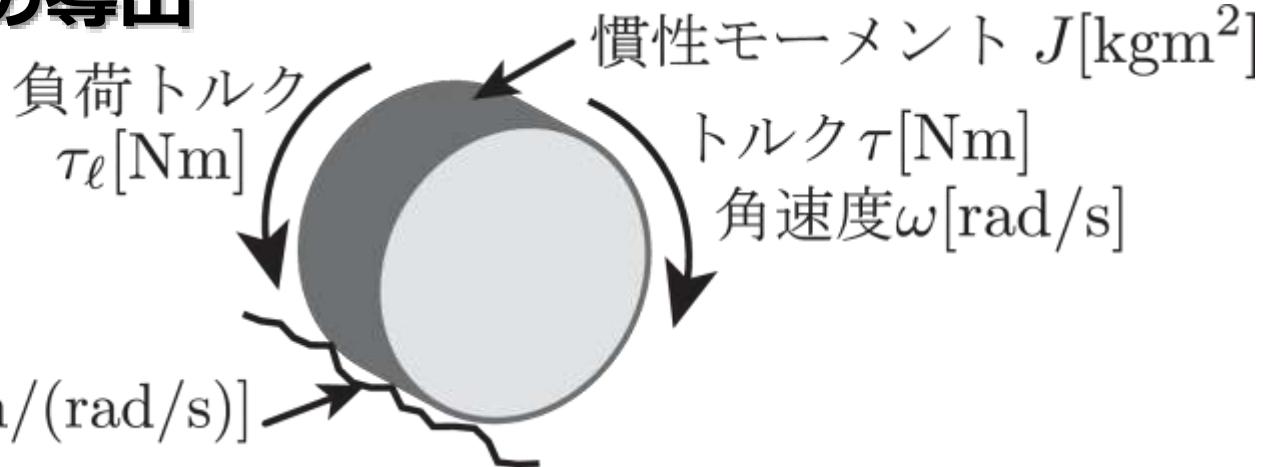


□ 回転系（回転モータの場合）



ACサーボモータのモデル化

■ 機械的モデルの導出



粘性摩擦係数 $D[\text{Nm}/(\text{rad/s})]$

□ 直動系と回転系の対応関係 □ 直動系 (リニアモータの場合)

力 $f \Leftrightarrow$ トルク τ

速度 $v \Leftrightarrow$ 角速度 ω

質量 $M \Leftrightarrow$ 慣性モーメント J

粘性摩擦係数 $D \Leftrightarrow$ 粘性摩擦係数 D

負荷外力 $f_\ell \Leftrightarrow$ 負荷トルク τ_ℓ

$$f(t) = M \frac{dv}{dt} + Dv + f_\ell$$

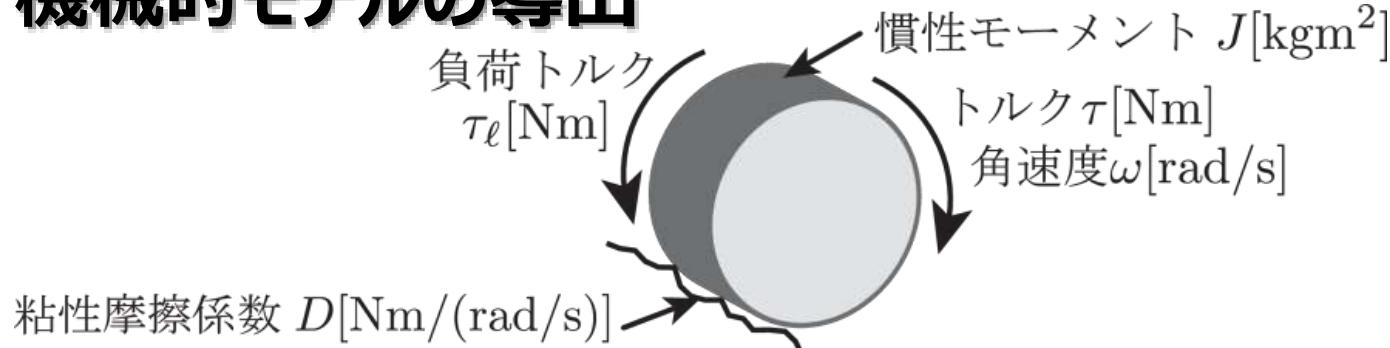
□ 回転系 (回転モータの場合)

$$\tau(t) = J \frac{d\omega}{dt} + D\omega + \tau_\ell$$

負荷外力
 慣性力
 粘性摩擦力

ACサーボモータのモデル化

■ 機械的モデルの導出



$$\tau(t) = J \frac{d\omega(t)}{dt} + D\omega(t) + \tau_l(t)$$

↓ ラプラス演算子で表現すると…
 (ただし、すべての変数の初期値は零とする) $\frac{d}{dt} \Leftrightarrow s$

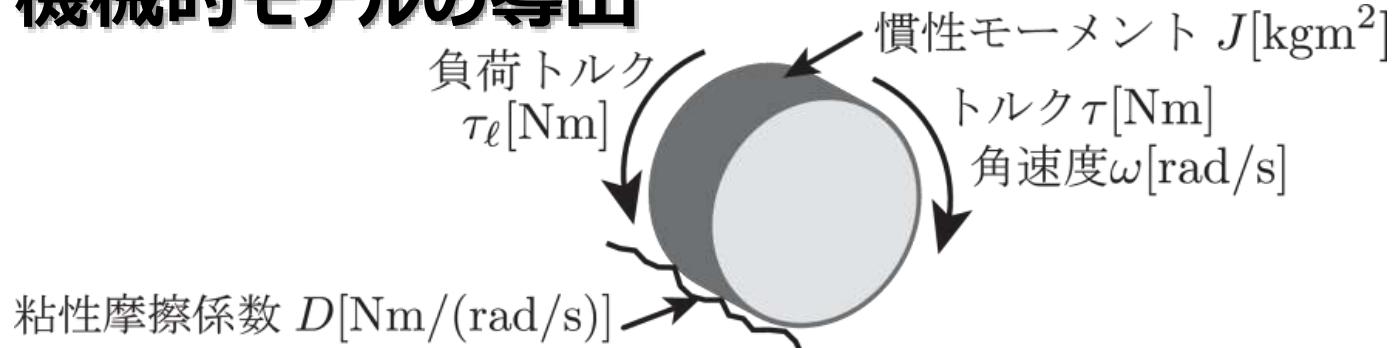
$$\tau(s) = Js\omega(s) + D\omega(s) + \tau_l(s)$$

↓ ちょっと変形

$$\omega(s) = \frac{1}{Js + D} (\tau(s) - \tau_l(s))$$

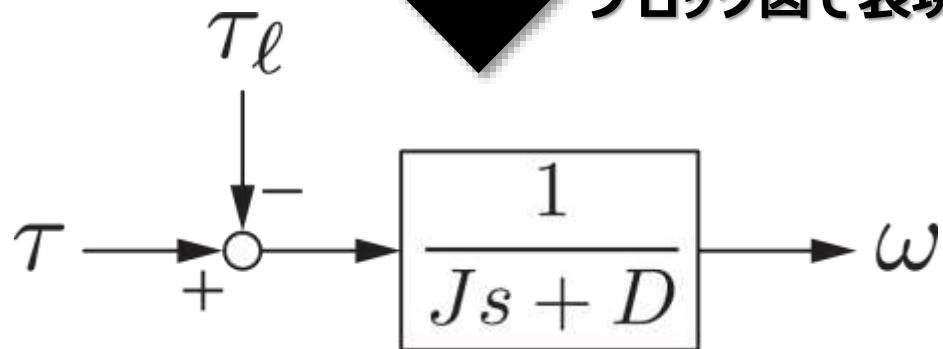
ACサーボモータのモデル化

■ 機械的モデルの導出



$$\omega(s) = \frac{1}{Js + D} (\tau(s) - \tau_\ell(s))$$

↓ ブロック図で表現すると…



ACサーボモータの機械的モデルが完成！
(トルクを入力すると速度が出力されるモデル)

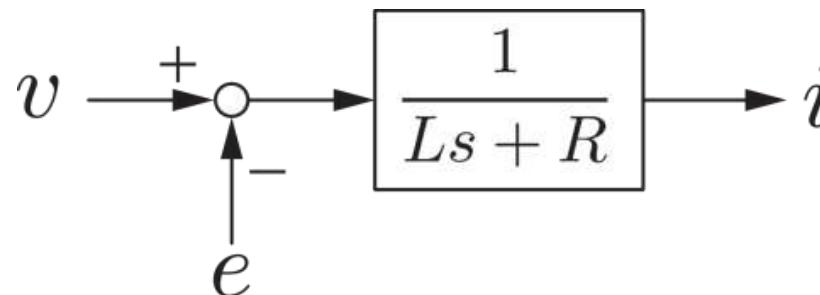
ACサーボモータのモデル化



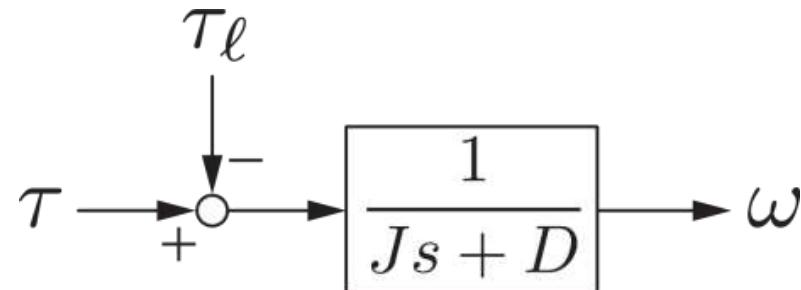
ACサーボモータのモデル化

■ 電気的モデルと機械的モデルの合体

電気的モデル



機械的モデル



これで、ACサーボモータの電気的モデルと機械的モデルがそれぞれ完成した。この2つのモデルを合体させればACサーボモータ全体のモデルとなる。

ではどのように合体させればよいか？

ACサーボモータのモデル化

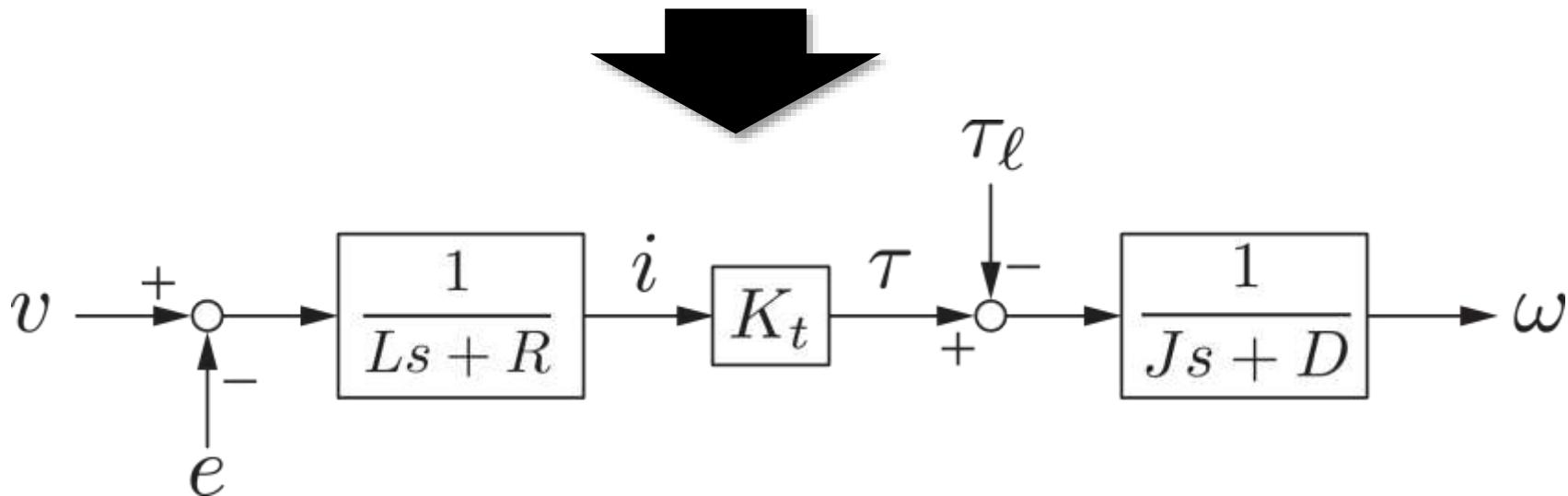
- 電気的モデルと機械的モデルの合体

- モータの特性 1

「モータに流れる電流」と「発生するトルク」には比例関係がある
すなわち,

$$\tau = K_t i$$

という関係式で書ける。この K_t [Nm/A] をトルク定数という。



ACサーボモータのモデル化

- 電気的モデルと機械的モデルの合体

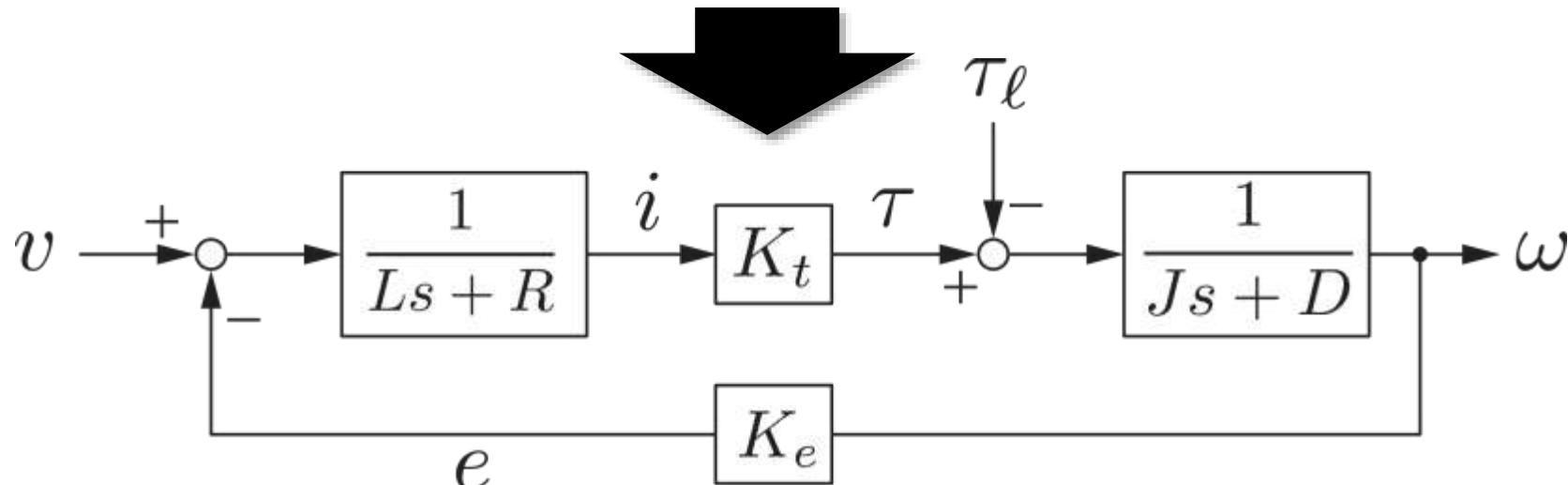
- モータの特性 2

「モータの速度」と「誘起電圧」には比例関係がある
すなわち,

$$e = K_e \omega$$

という関係式で書ける。

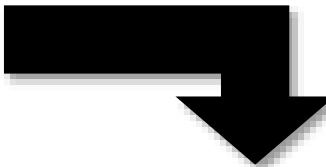
この K_e [V/(rad/s)] を誘起電圧定数という。



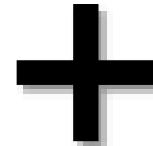
ACサーボモータのモデル化



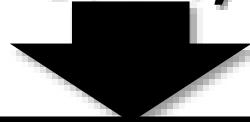
電気的モデル



機械的モデル



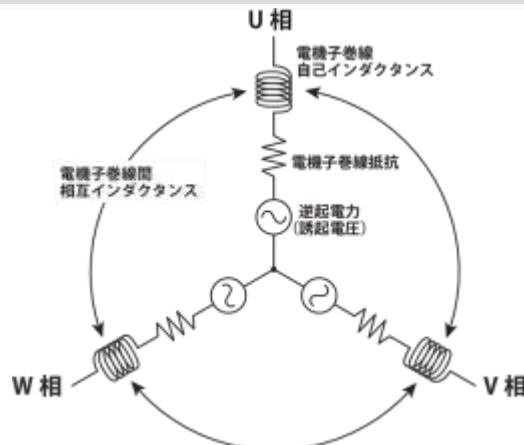
それぞれ個別に考えて、最後に合体！



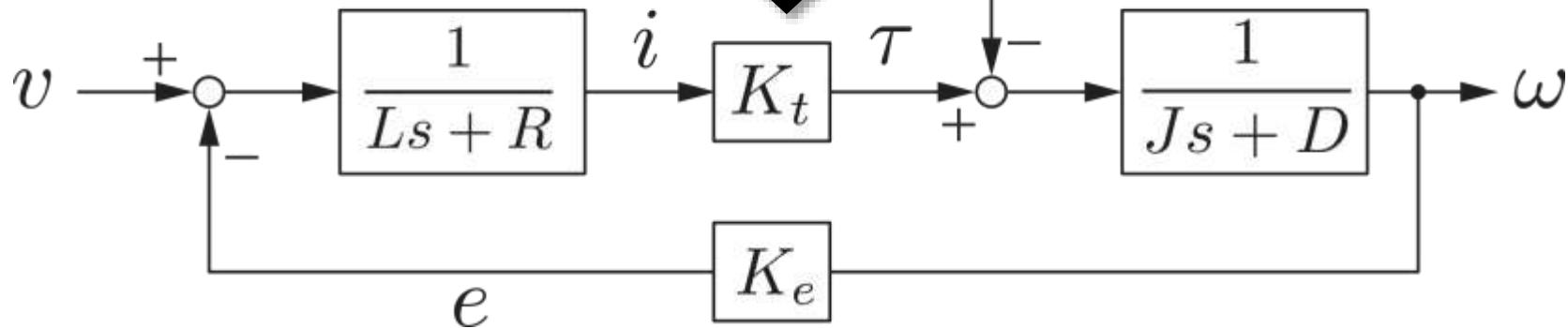
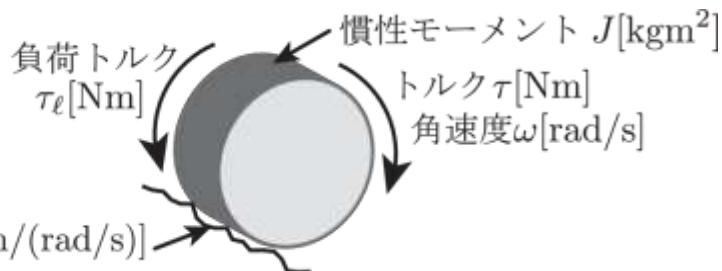
ACサーボモータ
全体のモデル

ACサーボモータのモデル化

電気的モデル



機械的モデル

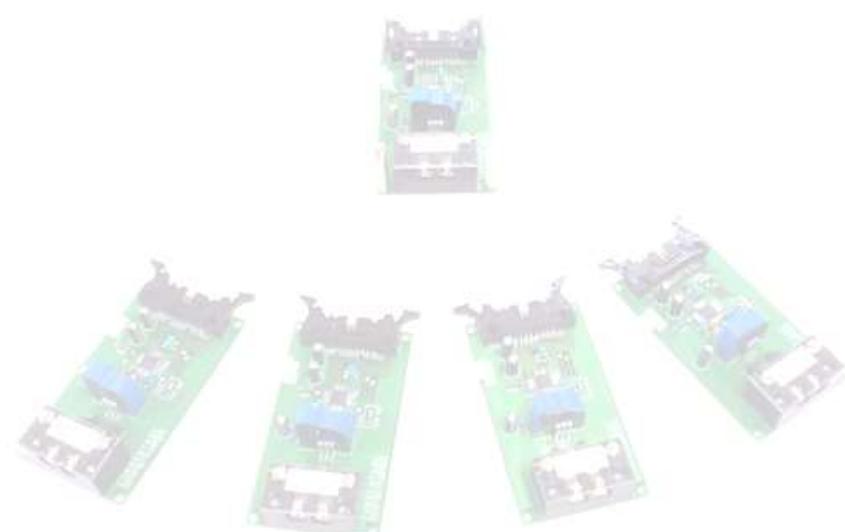
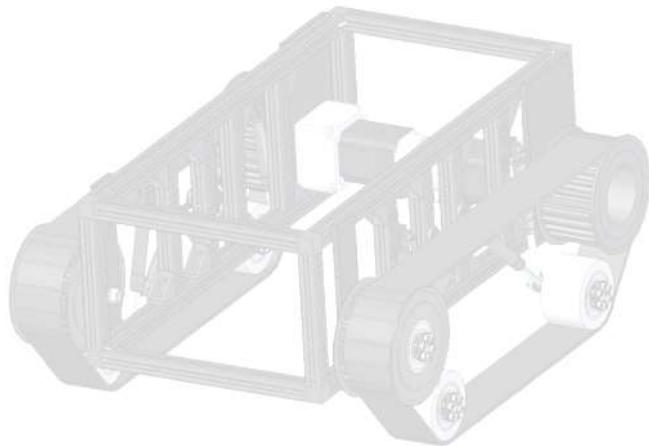


ACサーボモータの全体のモデルが完成！
 (電圧を入力すると速度が出力されるモデル)

本日の講義内容

- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- ディジタル制御系での積分器の作り方
- 実験機ハードウェアの説明
- 実験の流れ（1日目）
- 実験の流れ（2日目）
- レポート課題

ACサーボモータ制御の全体像



ACサーボモータ制御の全体像

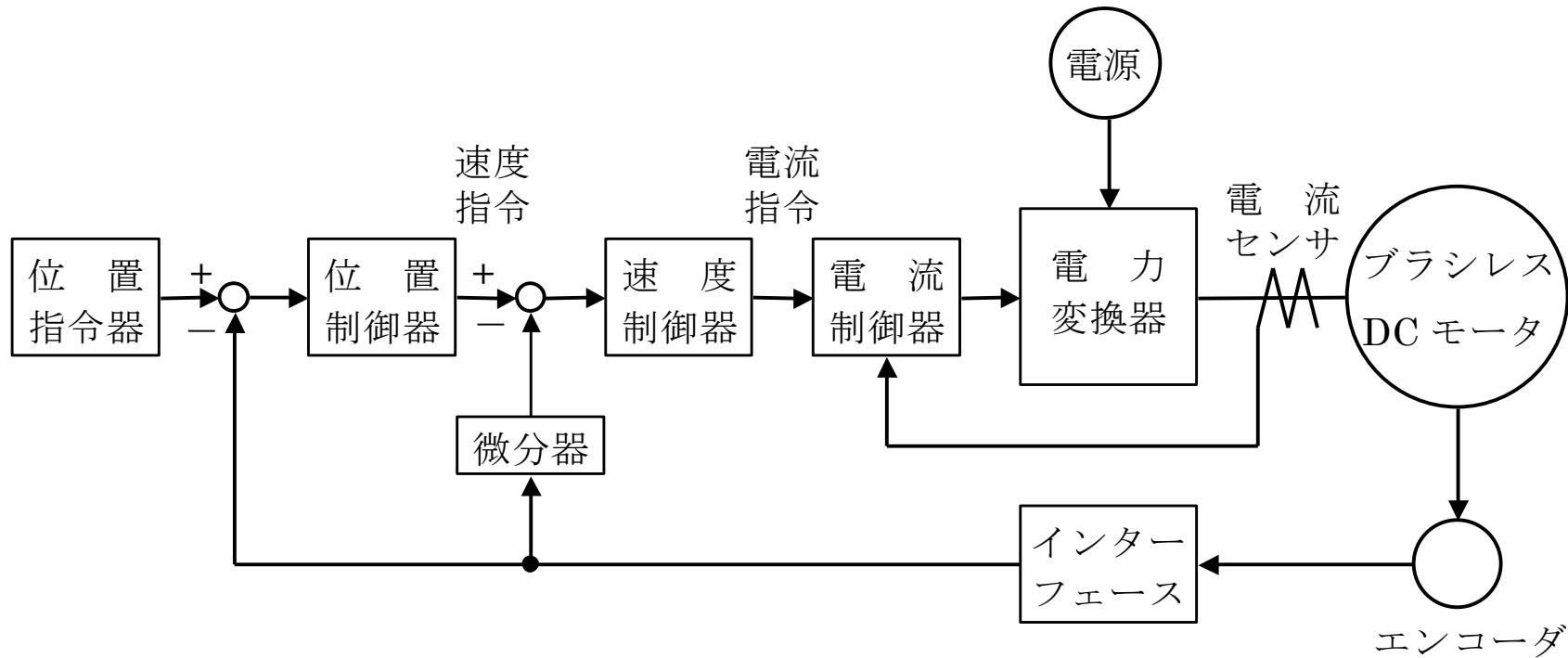
モノを上手く制御するためのコツ

1. 制御したい対象を
理論的に可能な限り正確に表現する
2. やりたいことを可能にする,
その制御対象に適した制御器をくっつける
3. 制御器のゲインを理論的に計算する

職人技でゲインを人間が頑張って調整するなんて、言語道断！！

ACサーボモータ制御の全体像

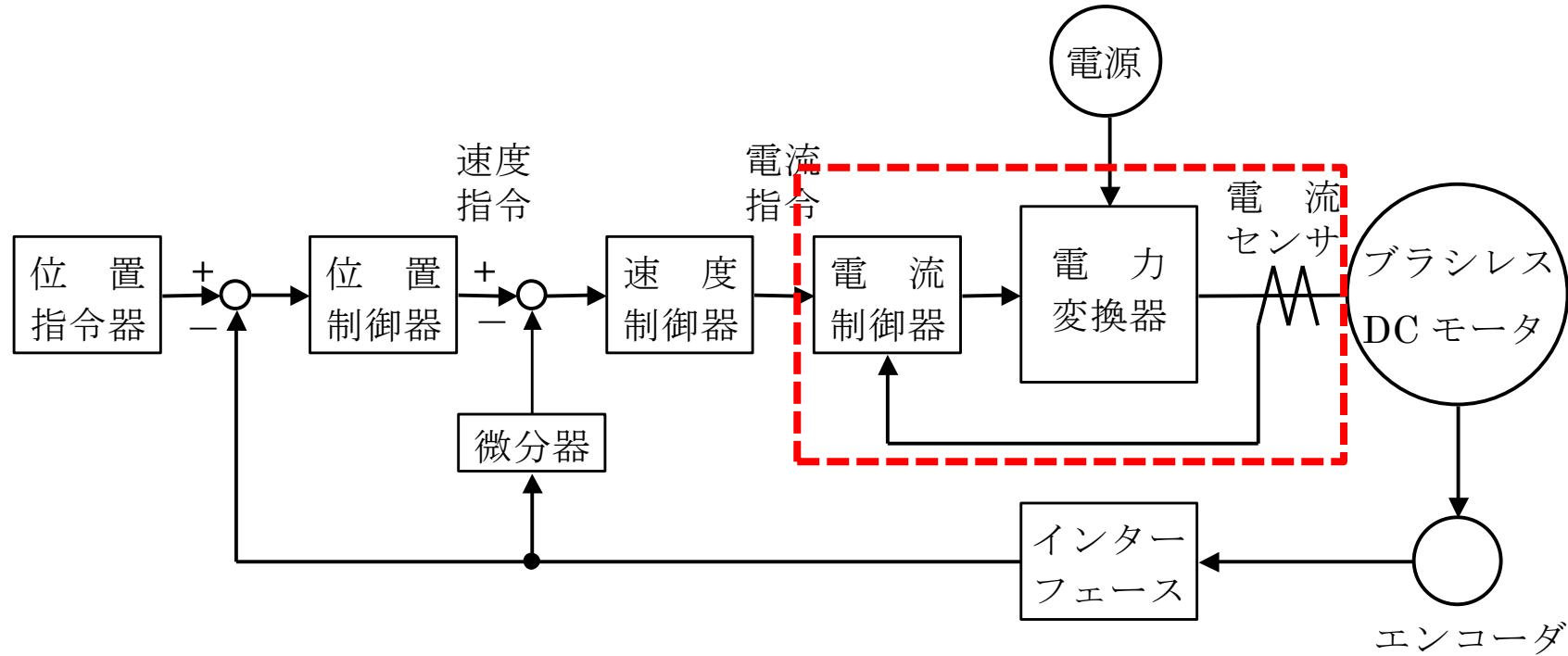
■ 永久磁石同期モータの位置制御系



ACサーボモータの位置制御系は、「電流制御系」、「速度制御系」、「位置制御系」の三重の制御ループで構成されるのが一般的である。

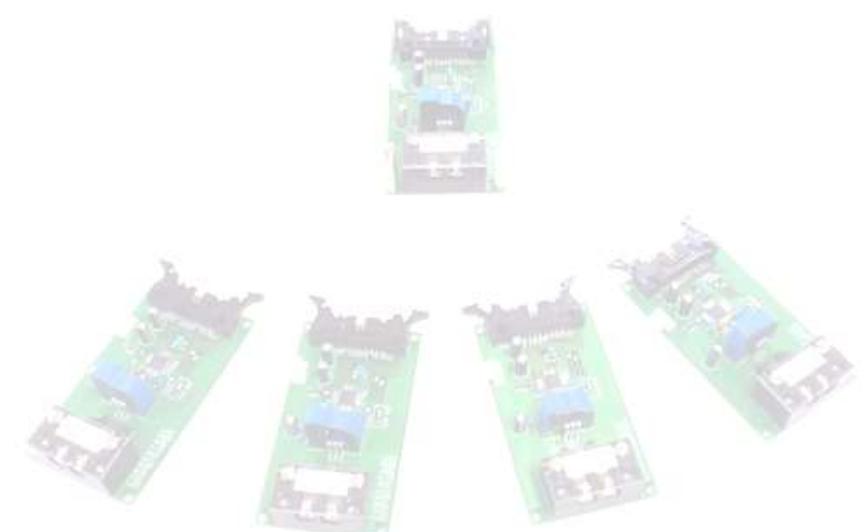
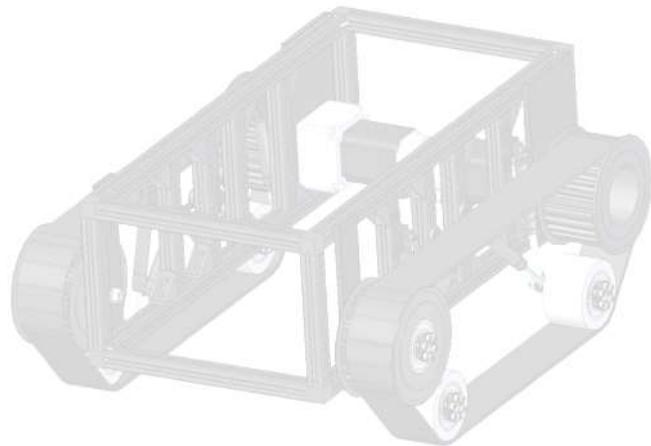
ACサーボモータ制御の全体像

■ 永久磁石同期モータの位置制御系



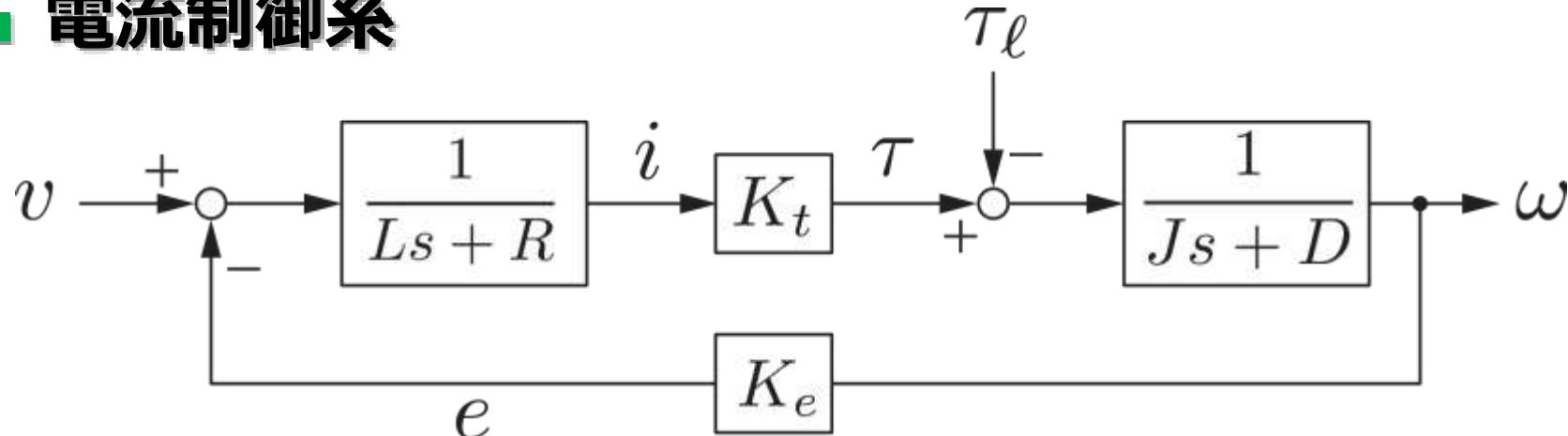
ACサーボモータの位置制御系は、
「電流制御系」、「速度制御系」、「位置制御系」
 の三重の制御ループで構成されるのが一般的である。

ACサーボモータの電流制御

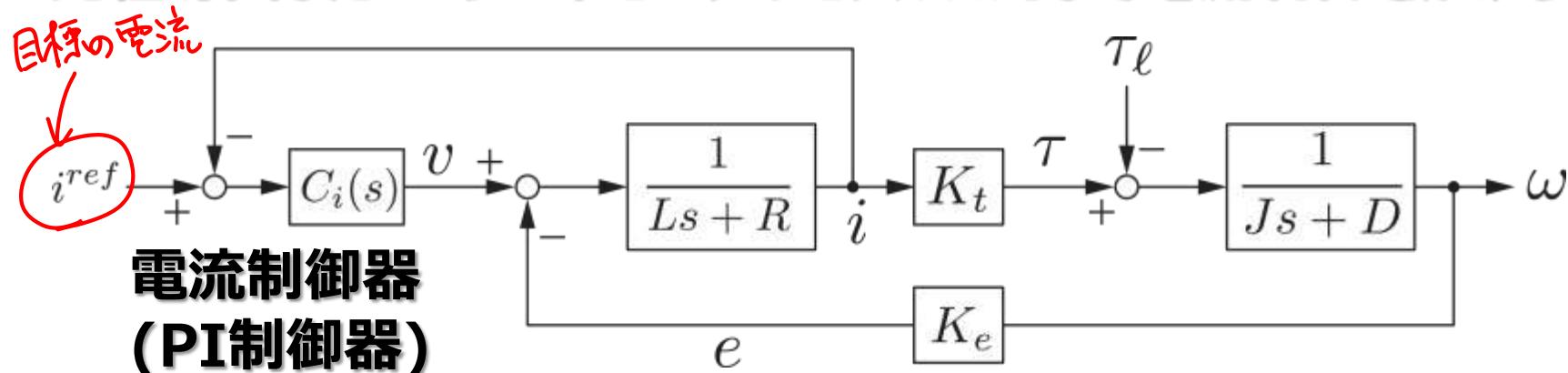


ACサーボモータの電流制御

■ 電流制御系



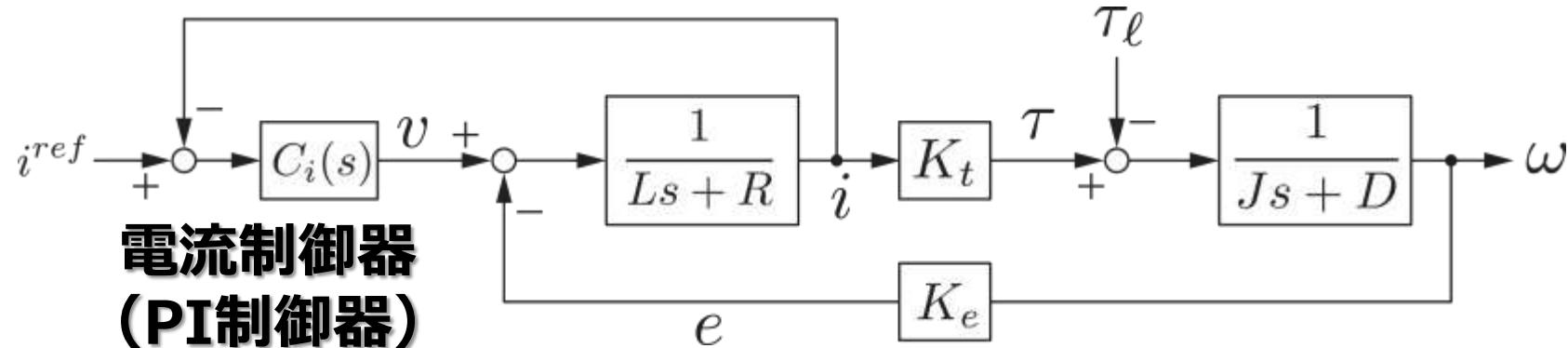
先程導出したACサーボモータのモデルに対して電流制御をかける



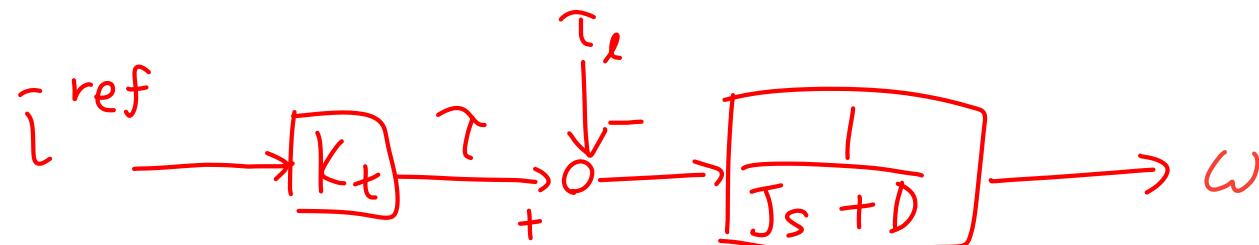
ここで問題：やりたいことはモータの速度制御や位置制御なのに、なぜ電流制御をするのか？ 目的は何？

ACサーボモータの電流制御

電流制御系



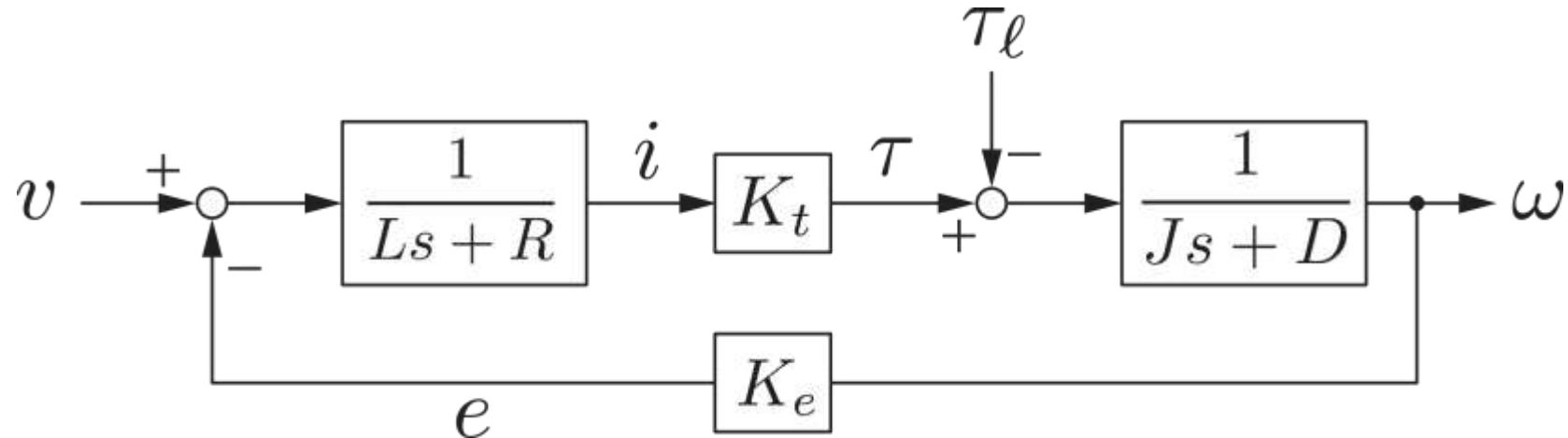
電流制御器が正常に動作していれば、
実電流は電流指令に必ず追従する。つまり…



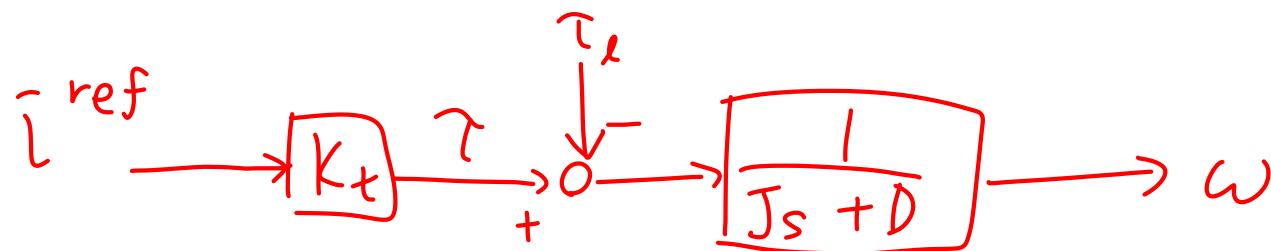
電流制御をかけられた永磁石同期モータのブロック図は
上記のよう書き換えられ、簡単なモデルで表せる。

ACサーボモータの電流制御

■ 電流制御をかけなかった場合 (素のモータ)

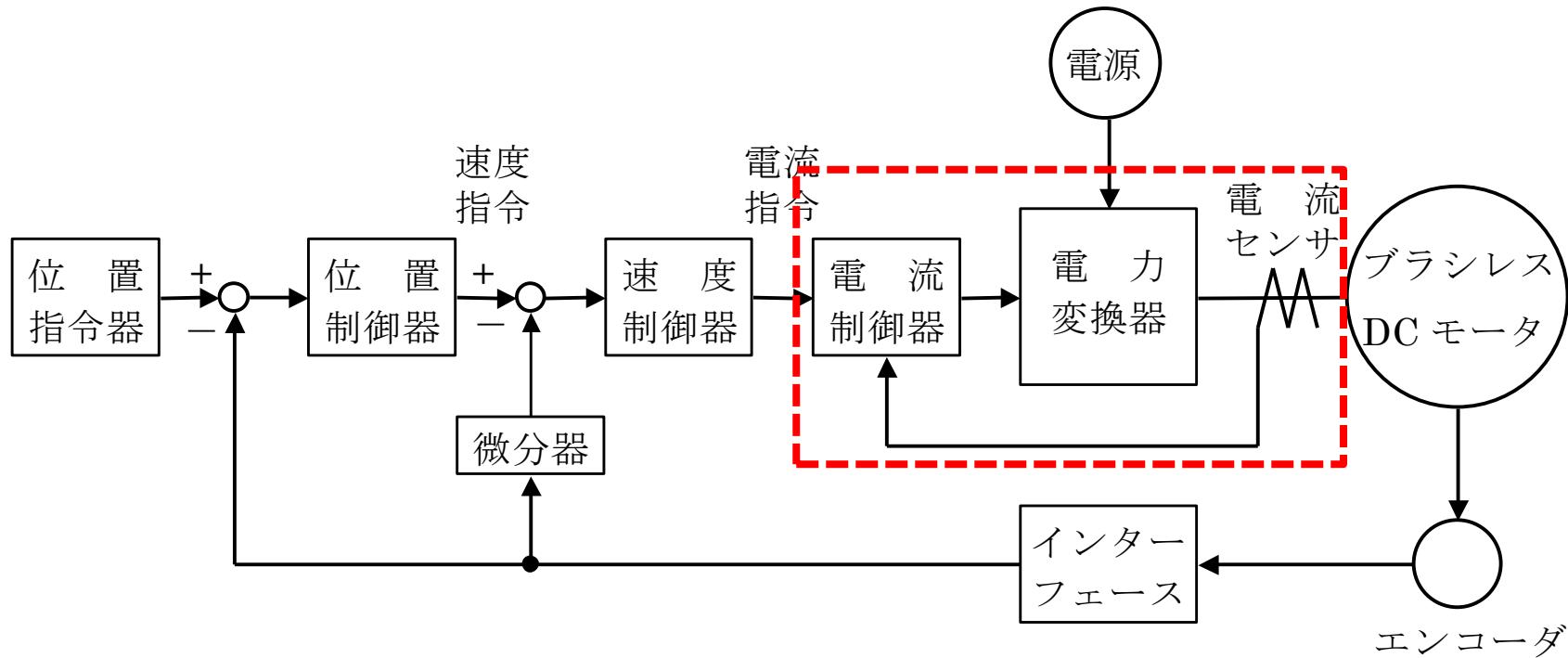


■ 電流制御をかけた場合



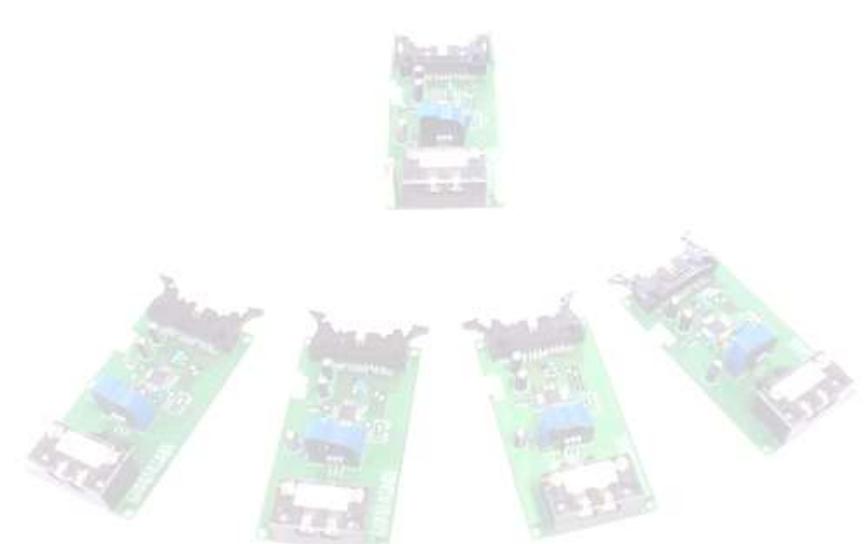
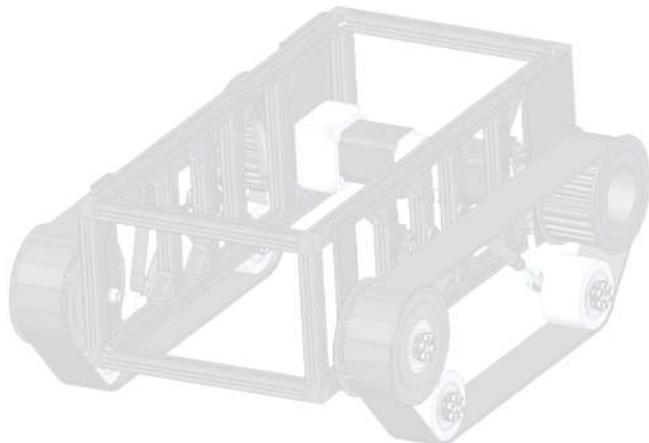
ACサーボモータの電流制御

■ 永久磁石同期モータの位置制御系



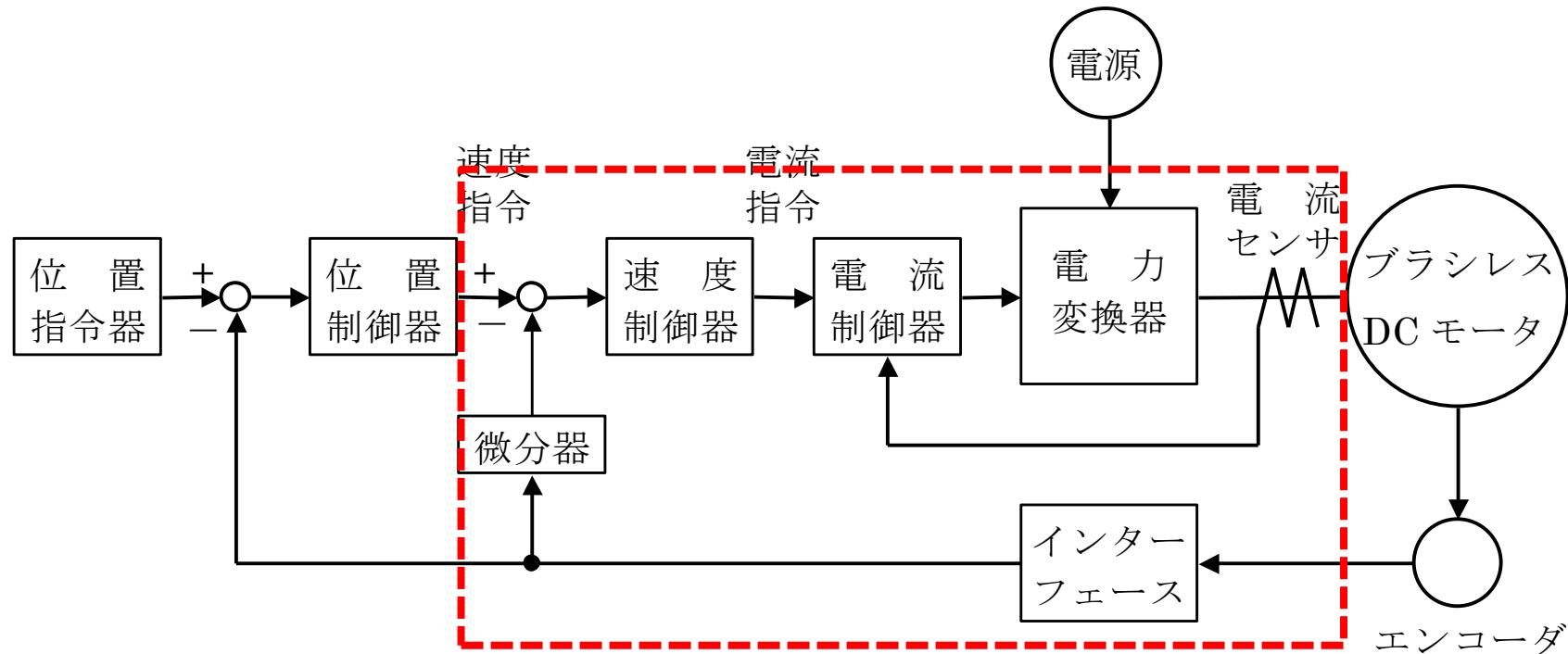
ACサーボモータの位置制御系は、
「電流制御系」、**「速度制御系」**、**「位置制御系」**
の三重の制御ループで構成されるのが一般的である。

ACサーボモータの速度制御



ACサーボモータの速度制御

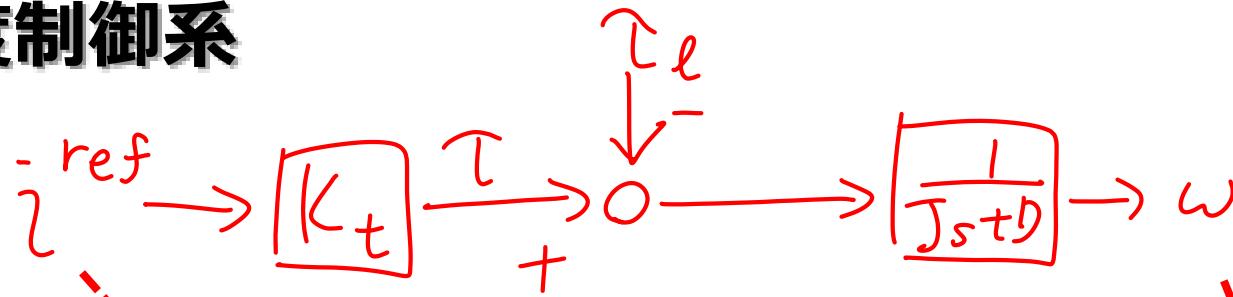
■ 永久磁石同期モータの位置制御系



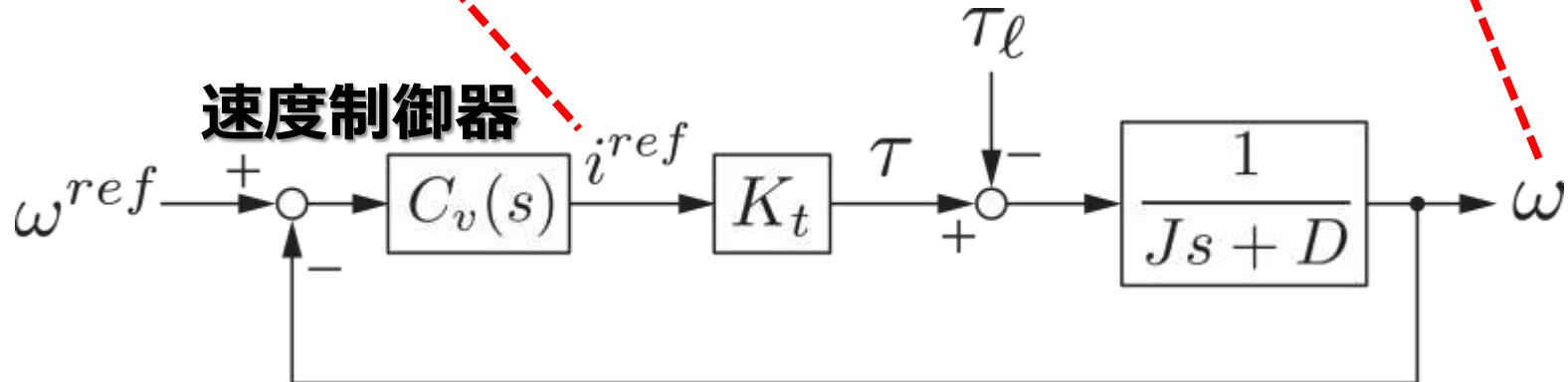
ACサーボモータの位置制御系は、
 「電流制御系」、「速度制御系」、「位置制御系」
 の三重の制御ループで構成されるのが一般的である。

ACサーボモータの速度制御

■ 速度制御系



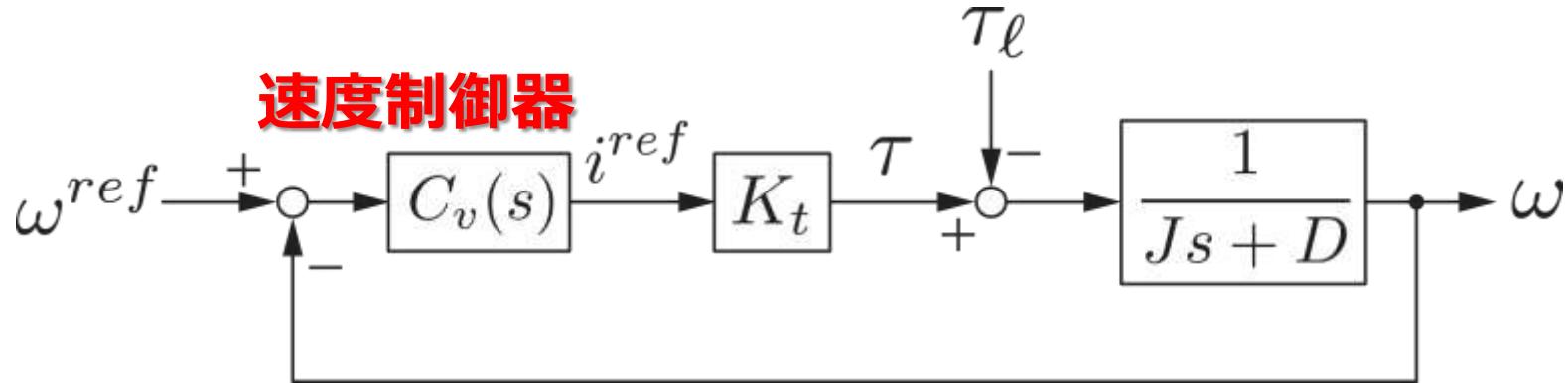
電流制御がかけられた
永久磁石同期モータのブロック図



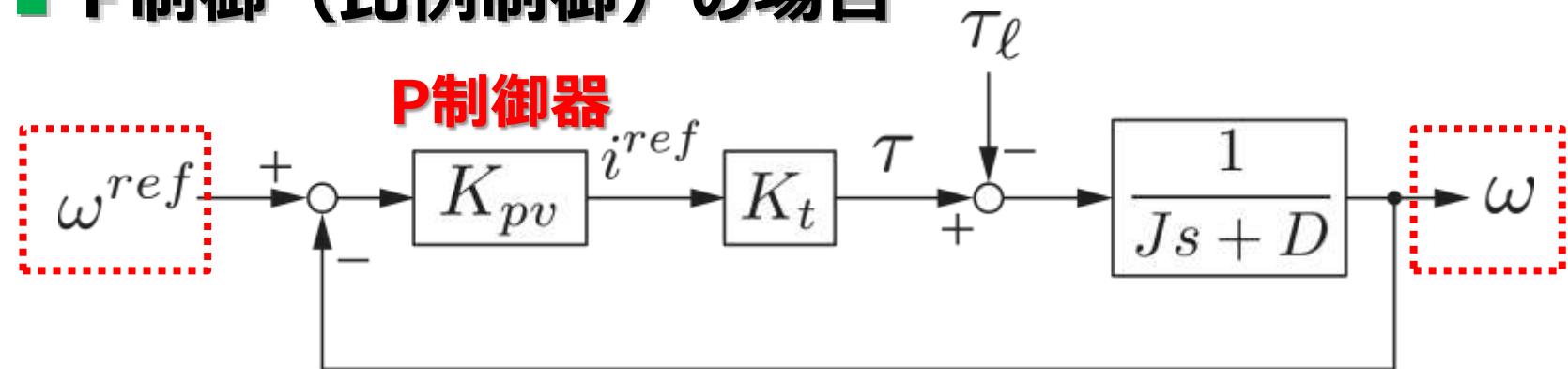
電流制御系の外側に速度制御系を構成
(電流マイナーループを持つ速度制御系)

ACサーボモータの速度制御

■ 速度制御器 $C_v(s)$ はどうする？



■ P制御（比例制御）の場合

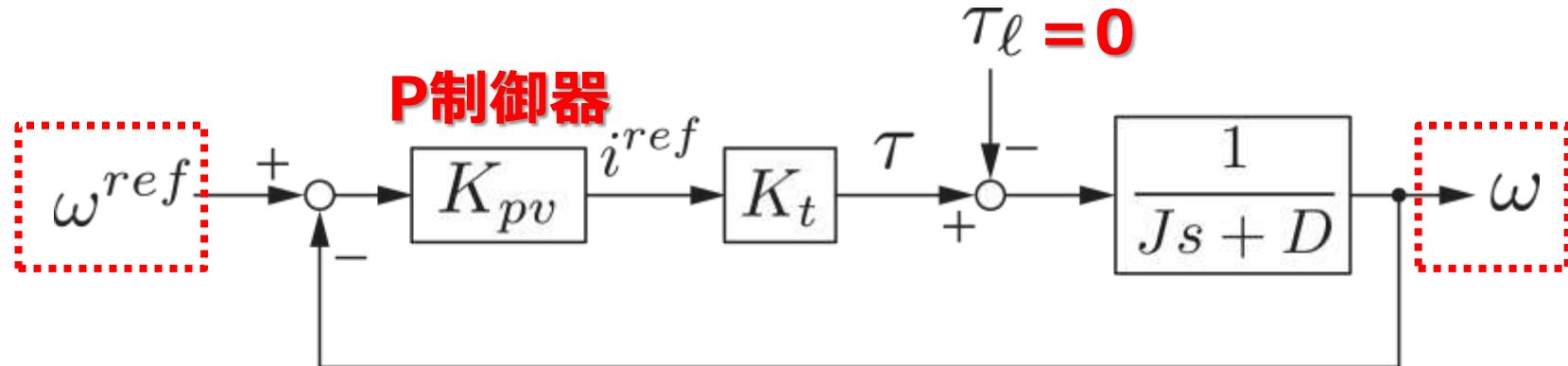


コレは一体どんな動作をするのだろうか？

そこで、速度指令 ω^{ref} から速度応答 ω の**伝達関数**を求めてみる。

ACサーボモータの速度制御

■ P制御器（比例制御器）の場合



速度P制御系の
目標値伝達関数

$$\frac{\omega(s)}{\omega^{ref}(s)} = \frac{K_{pv} K_t}{J s + D + K_{pv} K_t}$$

モータ速度が最終的にどんな値に落ち着くのか？

「最終値の定理」を使ってみる。 (ステップ応答における)

$$\lim_{s \rightarrow 0} s \frac{\omega(s)}{\omega^{ref}(s)} \cdot \frac{1}{s} = \frac{K_{pv} K_t}{J s + D + K_{pv} K_t} = \frac{K_{pv} K_t}{D + K_{pv} K_t} \neq 1$$

$$\lim_{s \rightarrow 0} s \frac{\omega(s)}{\omega^{ref}(s)} \cdot \frac{1}{s} = \frac{K_{pv} K_t}{D + K_{pv} K_t} < 0 \quad (0 < D)$$

摩擦係数は
絶対に正なので！

ACサーボモータの速度制御

■ P制御器（比例制御器）の場合

ステップ応答の
「最終値の定理」

$$\lim_{s \rightarrow 0} s \frac{\omega(s)}{\omega^{ref}(s)} \cdot \frac{1}{s} = \frac{K_{pv} K_t}{D + K_{pv} K_t} < 0 \quad (0 < D)$$

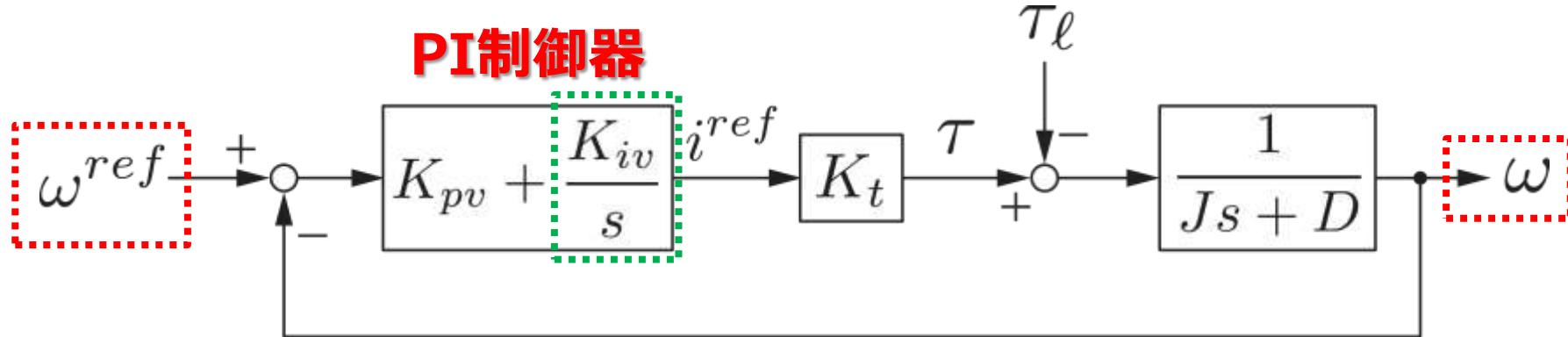
ココの物理的な意味分かります…？



速度P制御ダメじゃん。そこで…

ACサーボモータの速度制御

■ PI制御器（比例積分制御器）の場合



モータ速度が最終的にどんな値に落ち着くのか？
 「最終値の定理」を使ってみる。

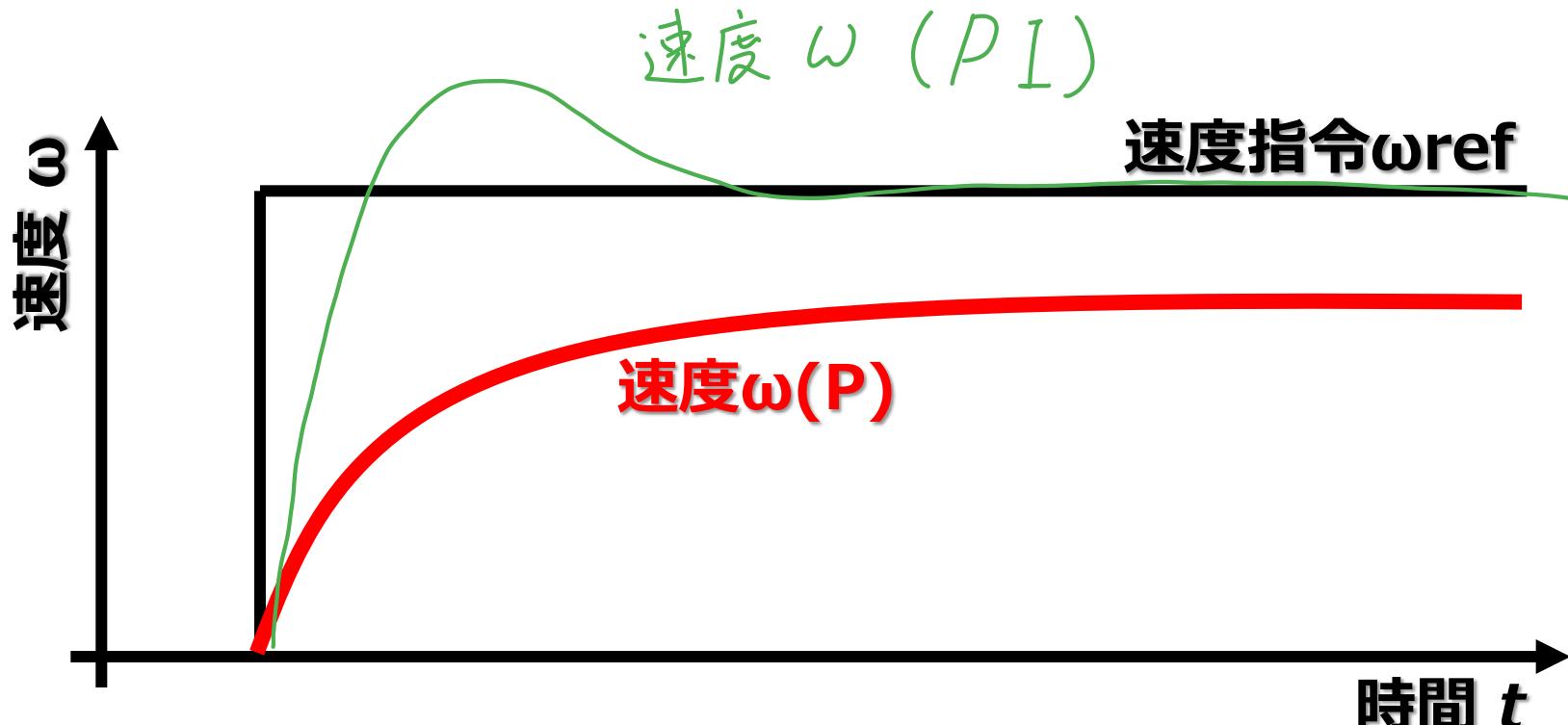
$$\lim_{s \rightarrow 0} s \frac{\omega(s)}{\omega^{\text{ref}}(s)} \cdot \frac{1}{s} = 1$$

足りない回数を τ に加える。

ACサーボモータの速度制御

■ PI制御器（比例積分制御器）の場合

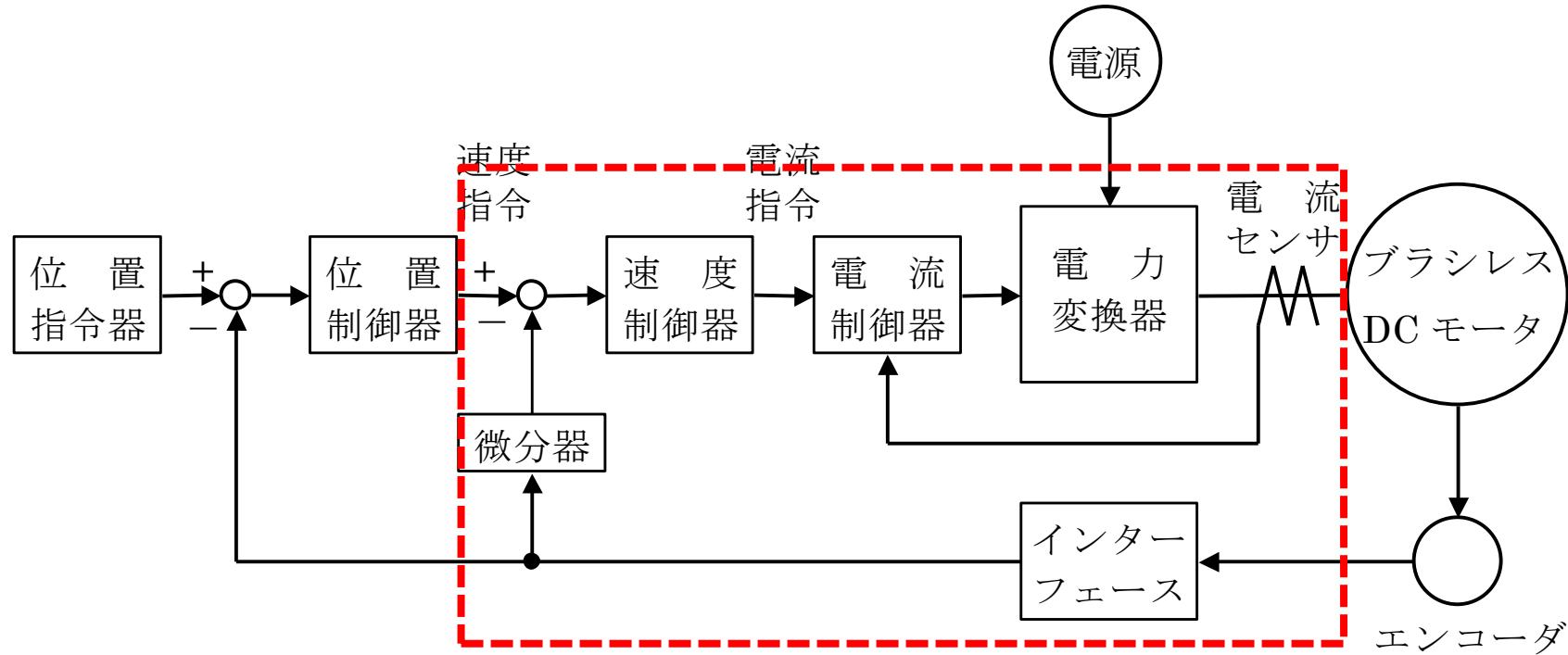
ステップ応答の
「最終値の定理」



速度PI制御すばらしい！

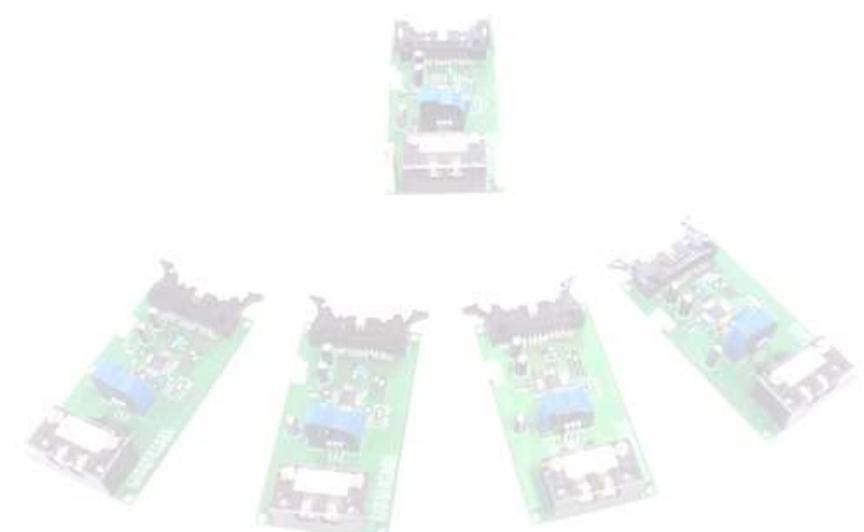
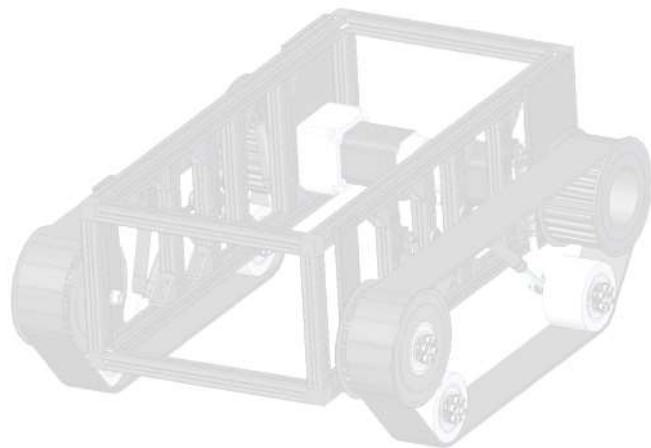
ACサーボモータの速度制御

■ 永久磁石同期モータの位置制御系



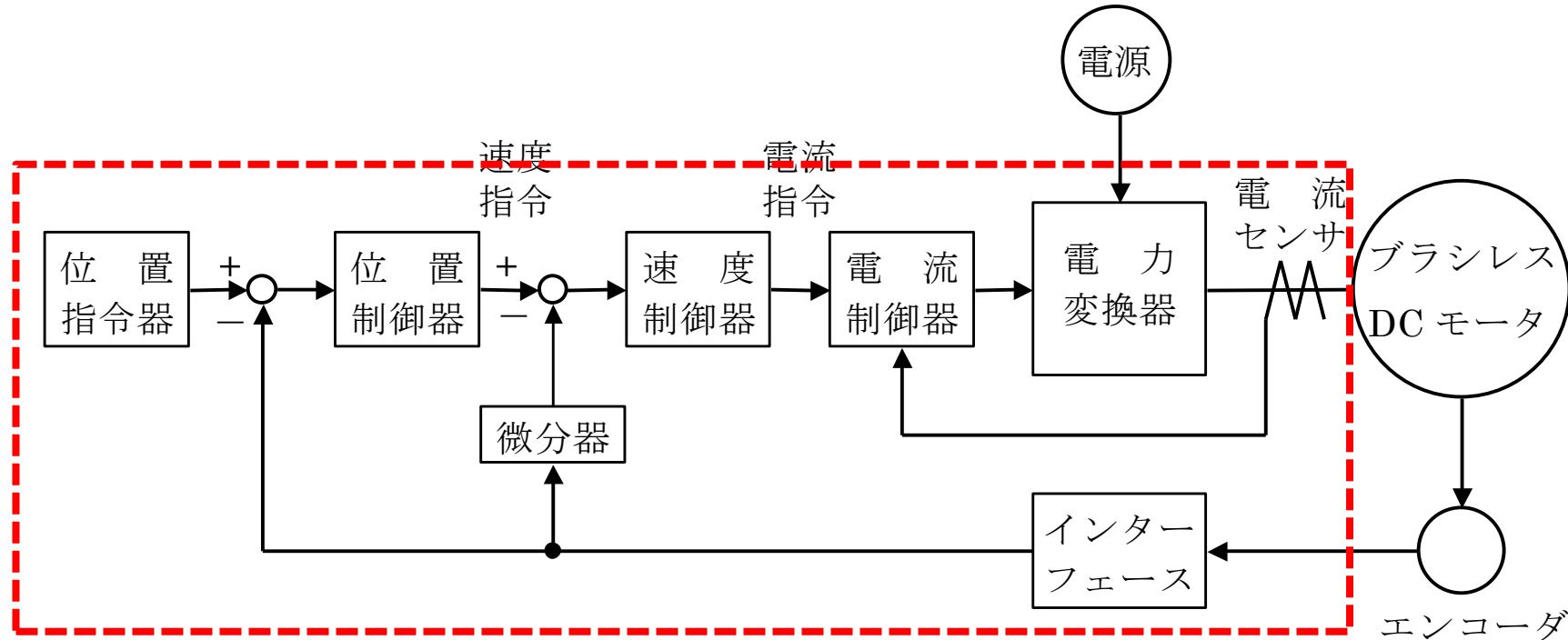
ACサーボモータの位置制御系は、
 「電流制御系」、「速度制御系」、「位置制御系」
 の三重の制御ループで構成されるのが一般的である。

ACサーボモータの位置制御



ACサーボモータの位置制御

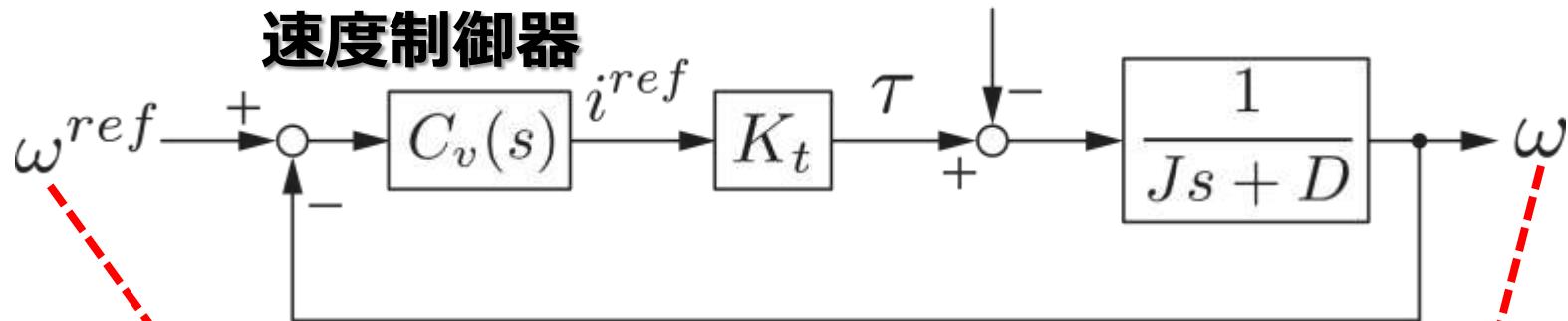
■ 永久磁石同期モータの位置制御系



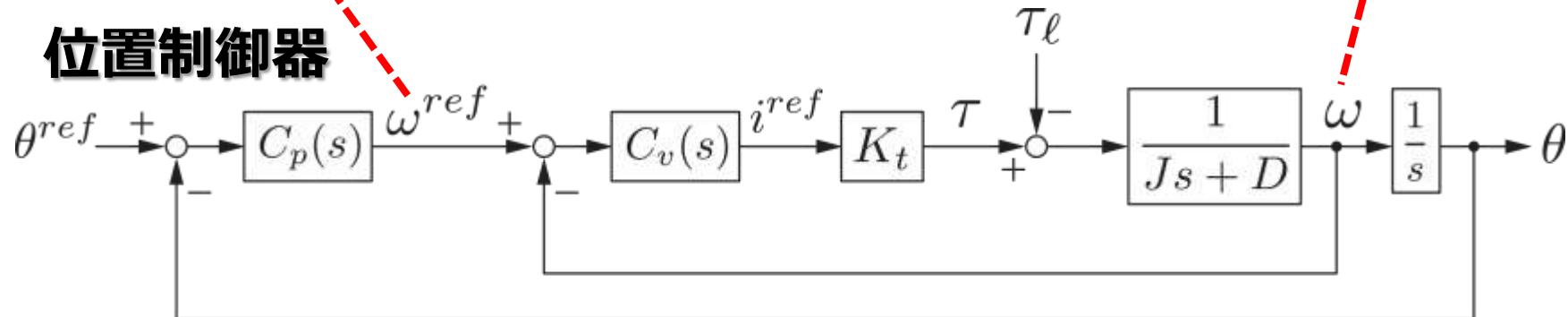
ACサーボモータの位置制御系は、
 「電流制御系」、「速度制御系」、「位置制御系」
 の三重の制御ループで構成されるのが一般的である。

ACサーボモータの位置制御

■ 位置制御系



電流制御と速度制御が施されたブロック図
(電流マイナーループを持つ速度制御系)

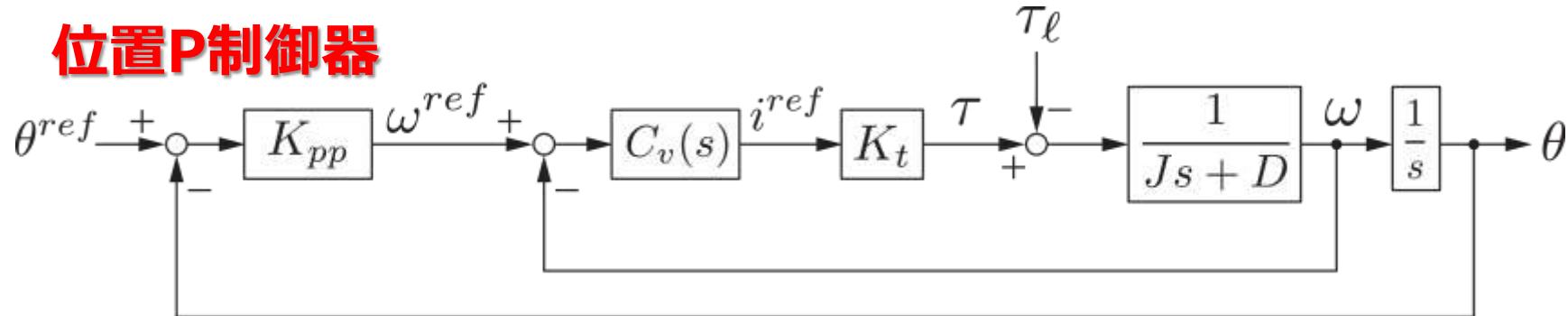


電流・速度・位置制御が施されたブロック図
(電流・速度マイナーループを持つ位置制御系)

ACサーボモータの位置制御

- 位置制御系はP制御器で十分！

位置P制御器



ステップ応答の
「最終値の定理」

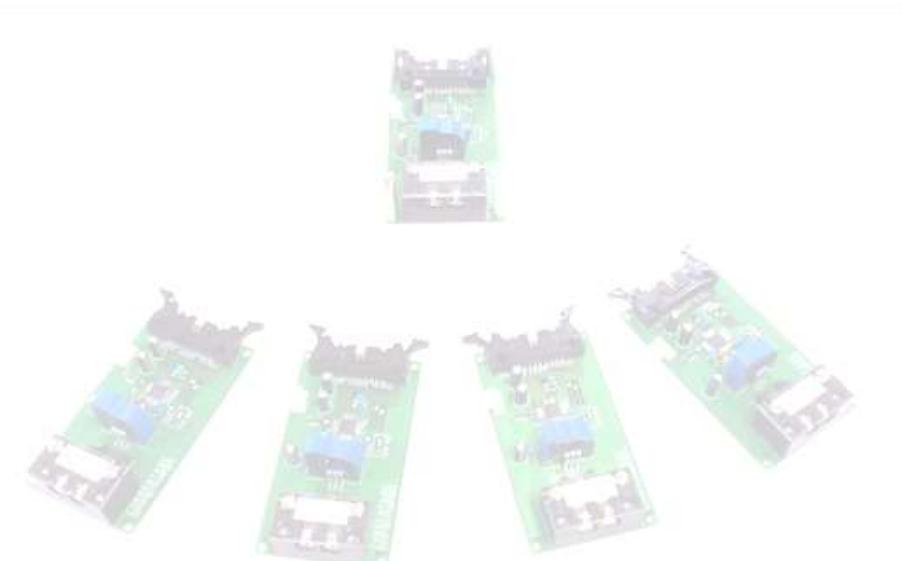
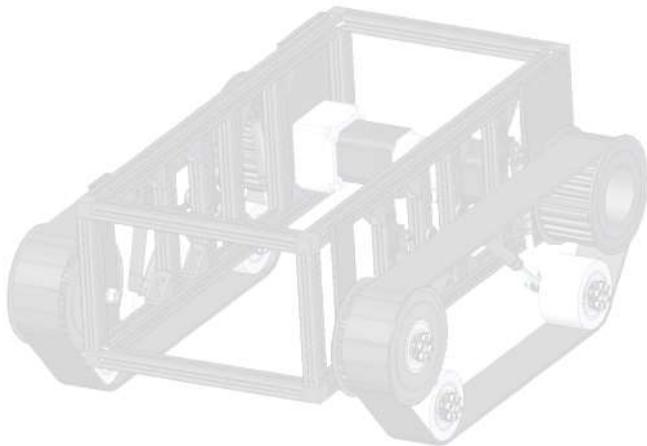
$$\lim_{s \rightarrow 0} s \frac{\theta(s)}{\theta^{\text{ref}}(s)} \cdot \frac{1}{s} = 1 \quad \checkmark \text{Pだけで十分収束する。}$$

安定性、高速応答性の点から 微分器を使わない
に越してはいけない。
使用する制御は伝達関数よく吟味してから選ぶべき。

本日の講義内容

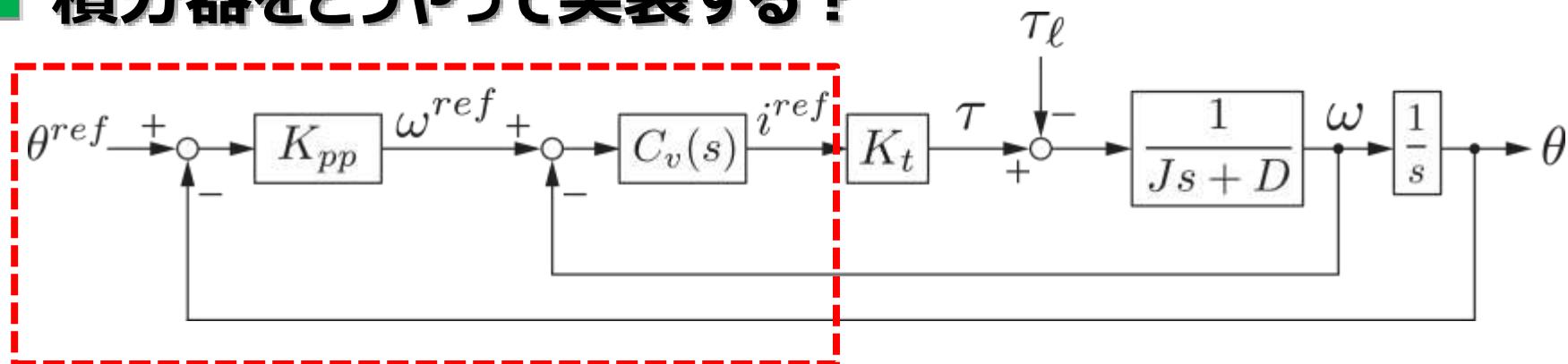
- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- **デジタル制御系での積分器の作り方**
- 実験機ハードウェアの説明
- 実験の流れ（1日目）
- 実験の流れ（2日目）
- レポート課題

デジタル制御系での積分器の作り方 (簡易版)



デジタル制御系での積分器の作り方（簡易版）

■ 積分器をどうやって実装する？



この部分をデジタル計算機で
実装しなければならない

例えば積分器の場合 : $Y = \frac{1}{s}U$ ←コレをどうやって実装する？
(実験ではC++を使う)

そこで双一次変換の出番！

$$s \rightarrow \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$y = \frac{T}{2} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} u$$

$$y - z^{-1}y = \frac{T}{2} (u + z^{-1}u)$$

(Tustin変換とも言う)

変換後の式 : $y = \frac{T}{2} \cdot (u + z^{-1}u) + z^{-1}y$

デジタル制御系での積分器の作り方（簡易版）

■ 積分器をどうやって実装する？

変換後の式： $y = \frac{T}{2} \cdot (u + z^{-1}u) + z^{-1}y$

z 演算子： z^{-1} ←どういう作用をするかご存知？

C++風に書き直すと： $y[k] = \frac{T}{2} \cdot (u[k] + u[k-1]) + y[k-1]$

\uparrow \uparrow \uparrow \uparrow
積分器
出力 今の
 入力 前回の
 入力 前回の
 出力

$$Y = \frac{1}{s}U \leftarrow \text{C++でどう実装していいか
分からぬけど…}$$

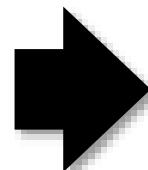
$$y[k] = \frac{T}{2} \cdot (u[k] + u[k-1]) + y[k-1] \leftarrow \text{これなら
20秒で実装できそう}$$

デジタル制御系での積分器の作り方（簡易版）

■ 積分器以外も作れる

$$Y = G(s)U \quad \leftarrow \text{欲しい任意の伝達関数でもOK}$$

$$s \rightarrow \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$



$$y[k] = \dots \dots \dots$$

コーディング向きの表現に変換！

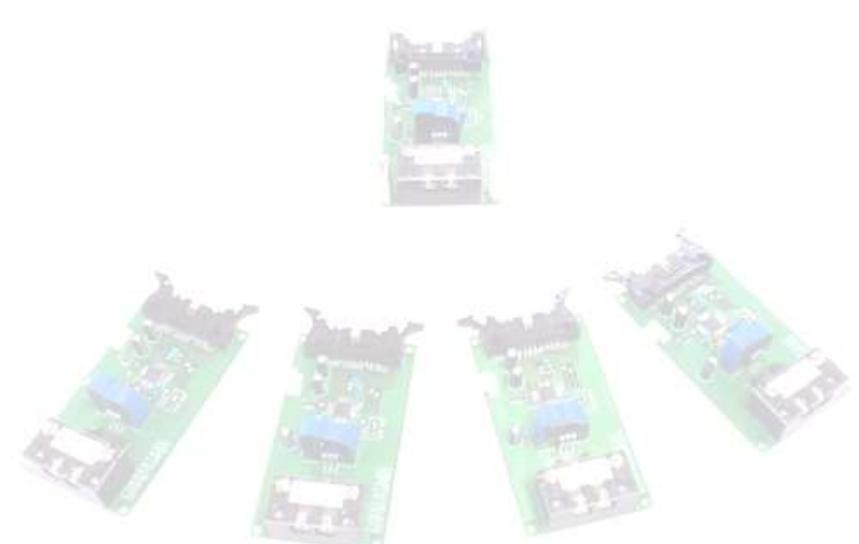
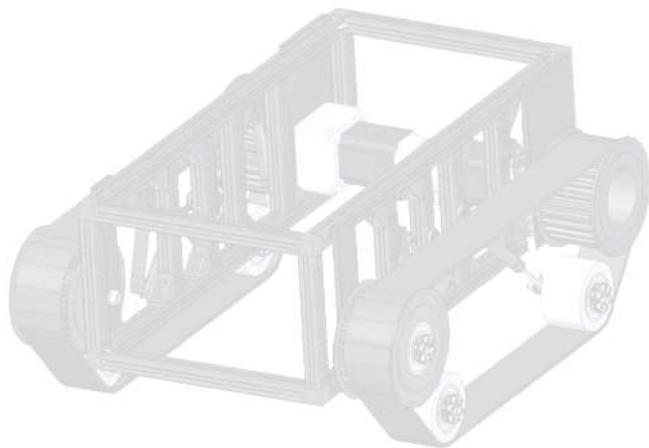
ただし…

伝達関数の次数が増えてくるとこの方法は使えない。
双一次変換はあくまで近似。真面目に正確にやるには、
ゼロ次ホールド・サンプラー・状態遷移行列などなど…

「s」と「z」の関係：

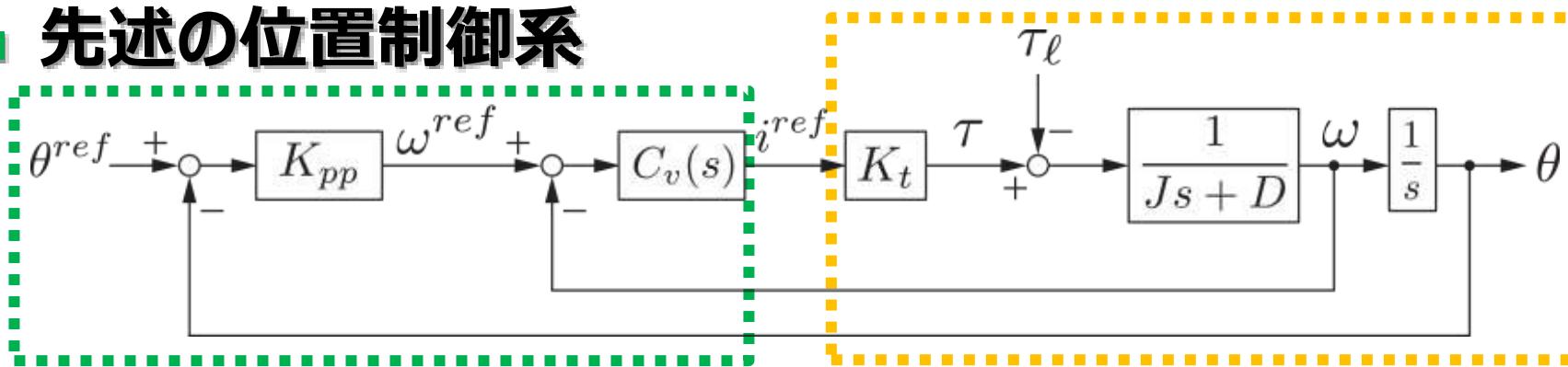
$$e^{-sT} = z^{-1} \rightarrow s = -\frac{1}{T} \log \frac{1}{z} \rightarrow \text{パデ近似を1次で打ち切り}$$

実験機ハードウェアの説明

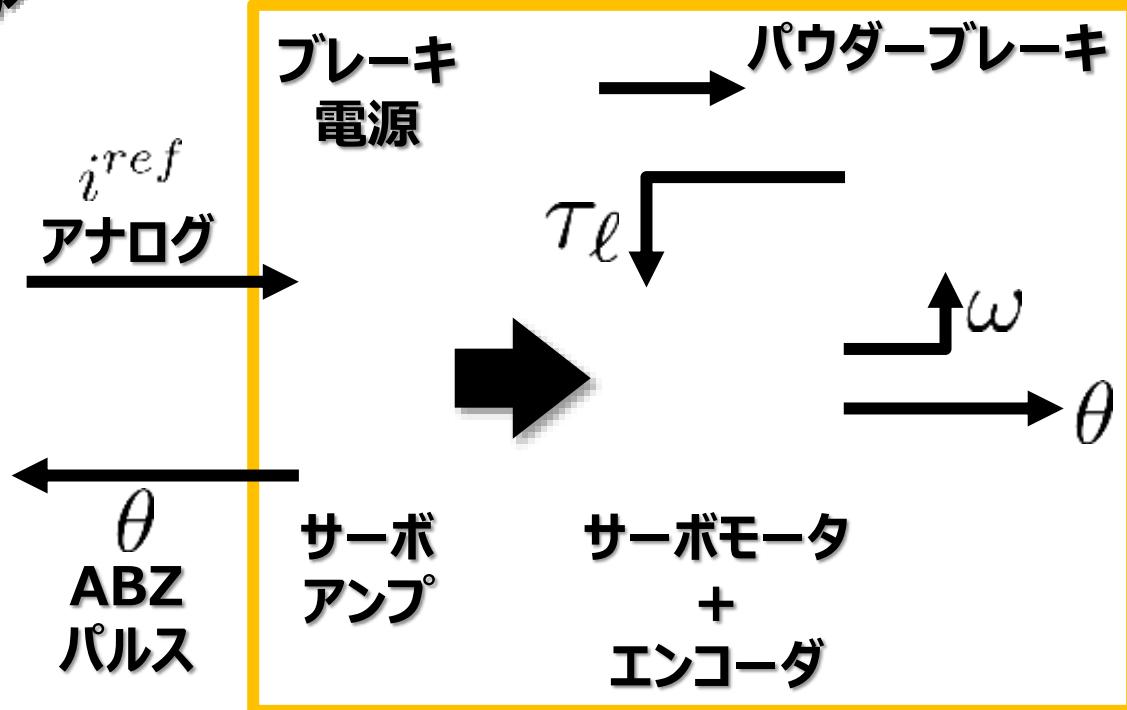


実験機ハードウェアの説明

先述の位置制御系

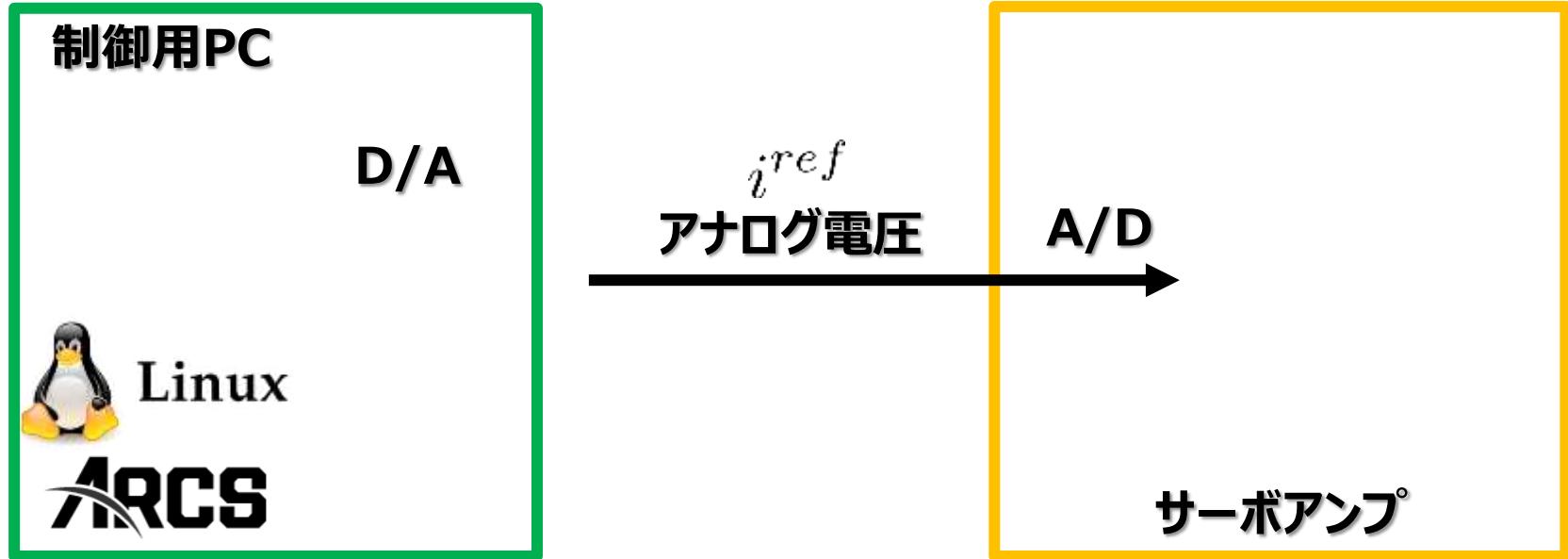


実際のハードウェア



実験機ハードウェアの説明

■ 電流指令の入力方法



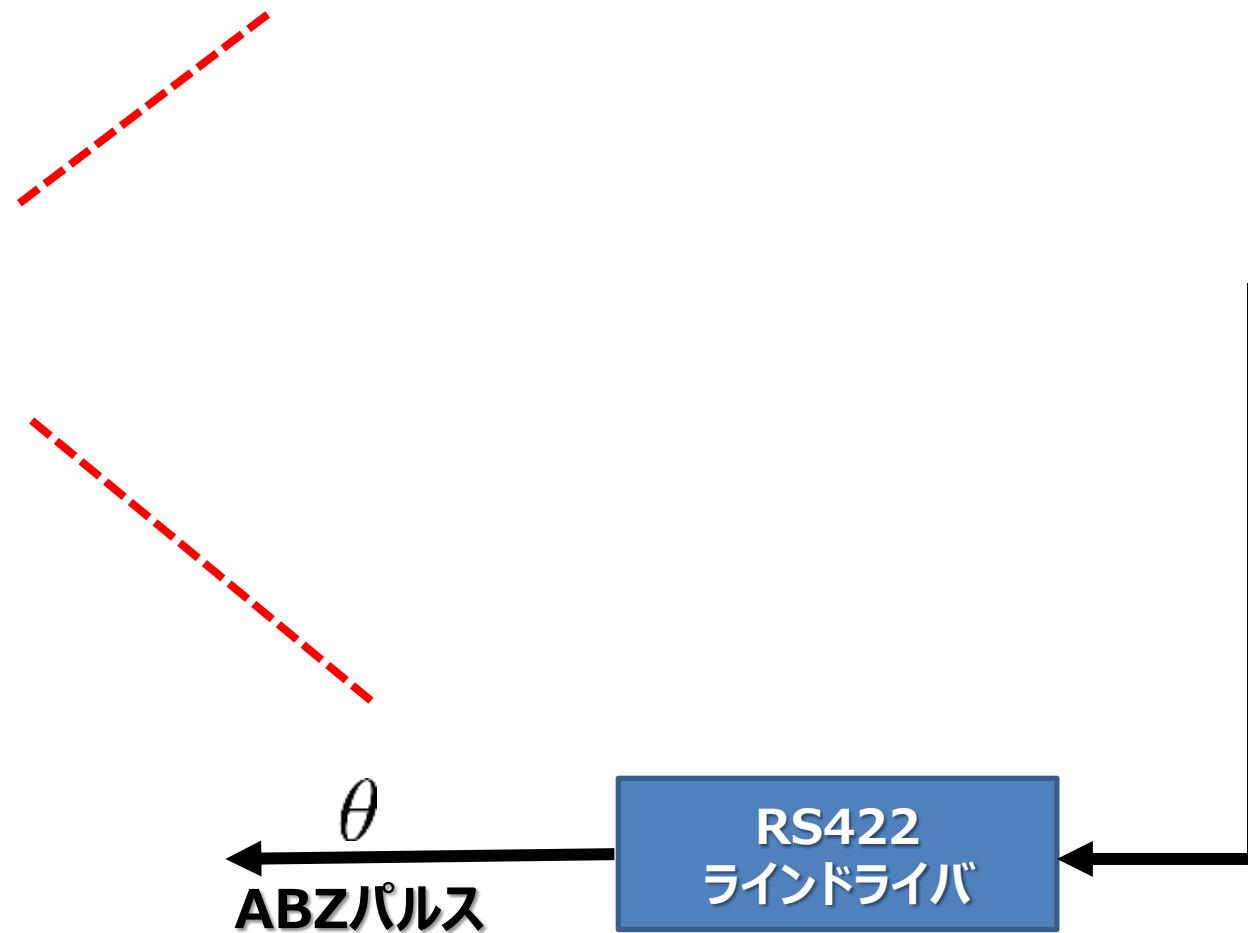
注意！ 電流指令をアナログ「電圧」で送る

**1V の電圧信号を入力すると
? A の電流がサーボモータに流れる**

実験機ハードウェアの説明

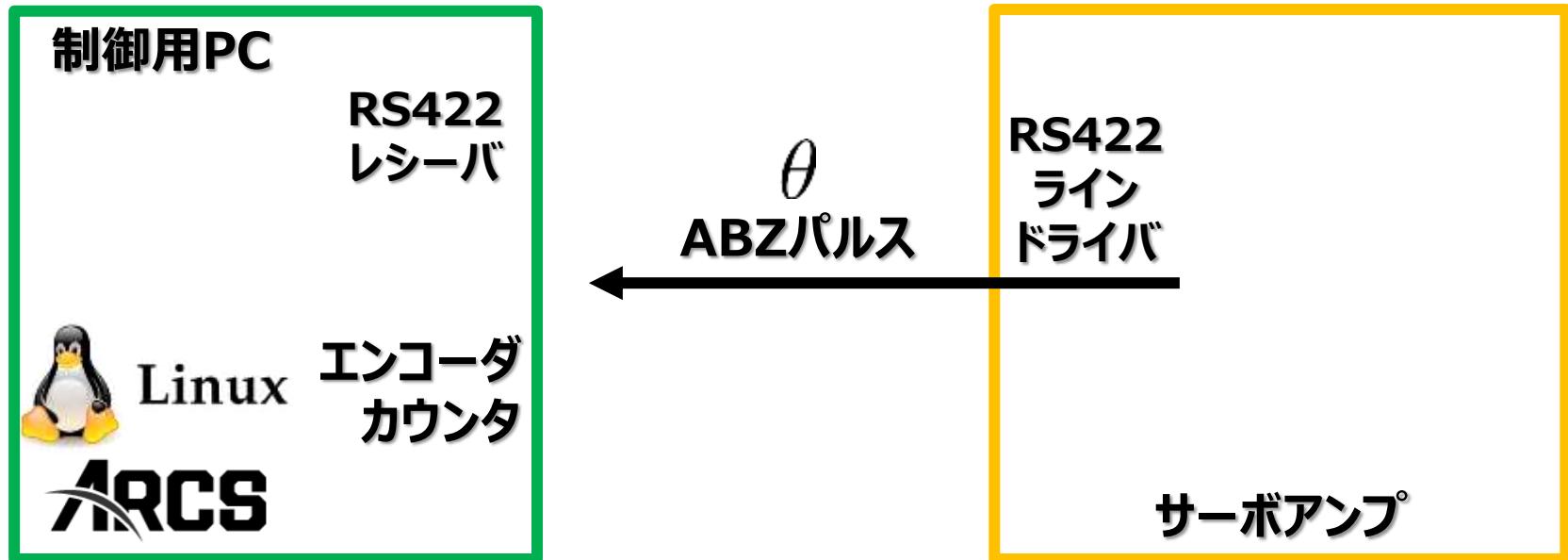
■ 位置応答の取得方法

サーボモータ
+
エンコーダ



実験機ハードウェアの説明

■ 位置応答の取得方法



注意！ 位置応答を「パルス信号」で貰う

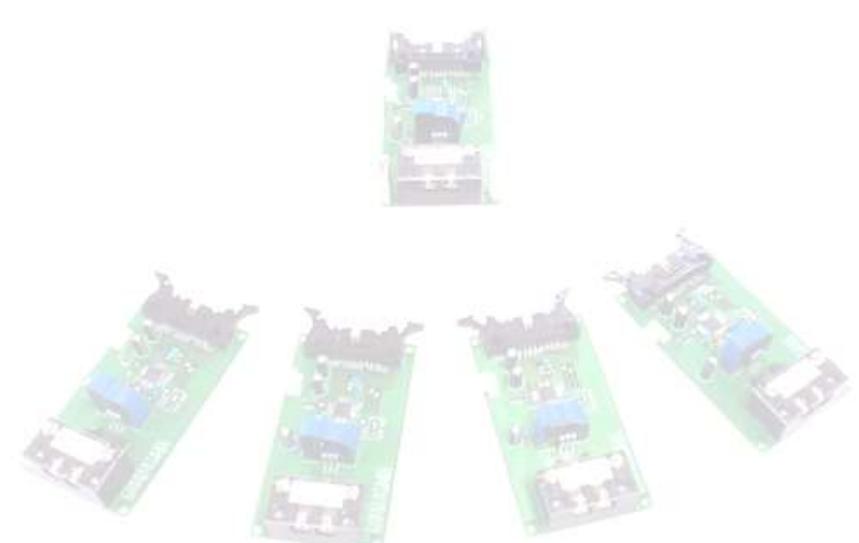
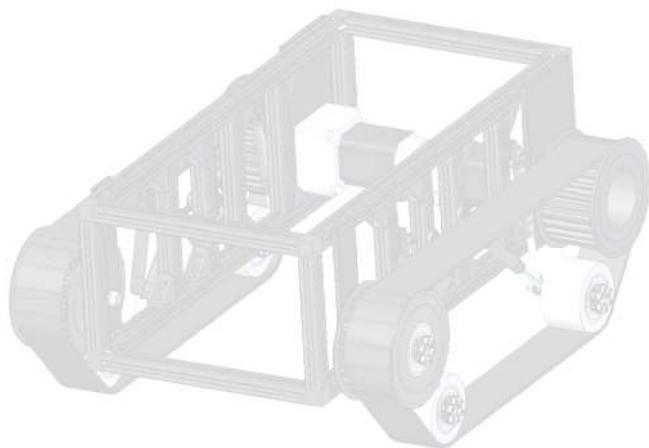
サーボモータが1回転すると？個のパルスが来る

4倍する場合では、1回転すると4×？個のパルスが来る

本日の講義内容

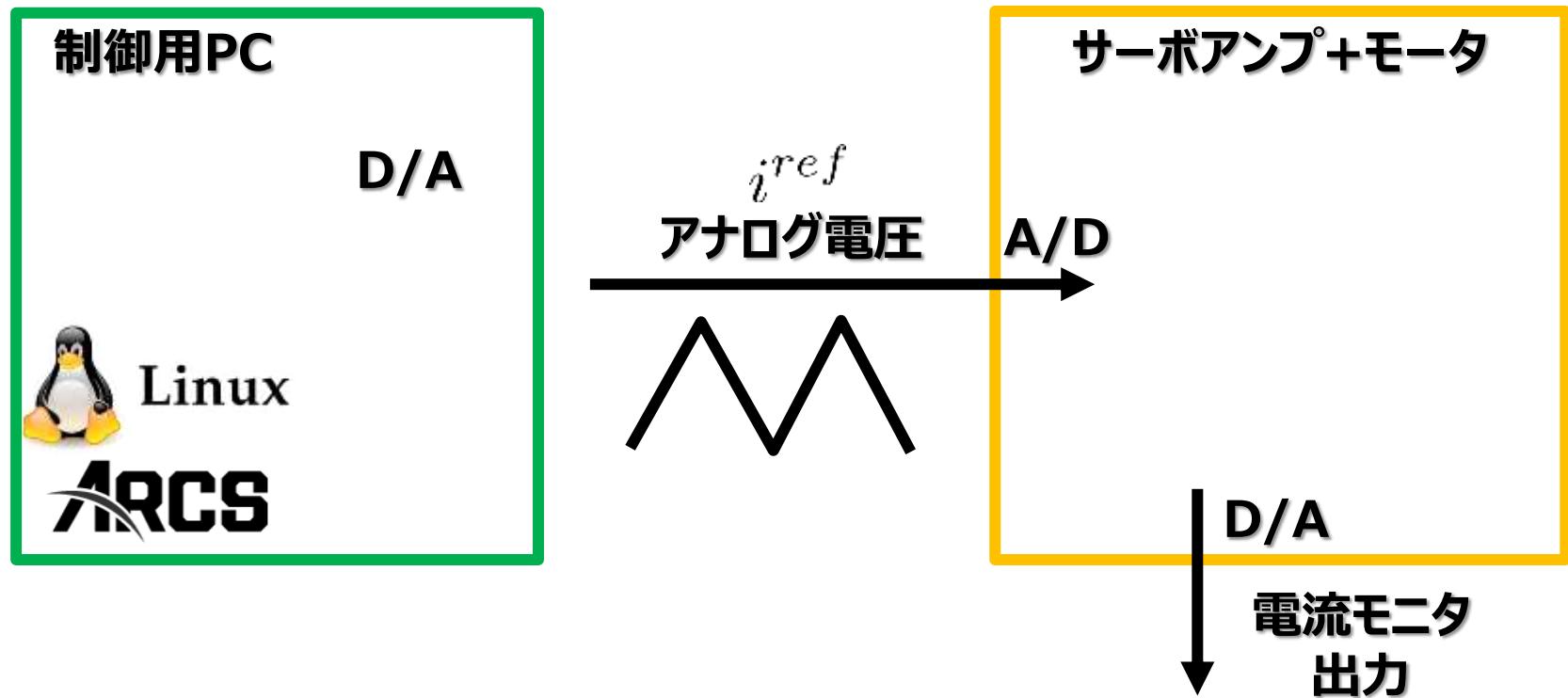
- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- ディジタル制御系での積分器の作り方
- 実験機ハードウェアの説明
- **実験の流れ（1日目）**
- **実験の流れ（2日目）**
- レポート課題

実験の流れ (1日目)



実験の流れ (1日目)

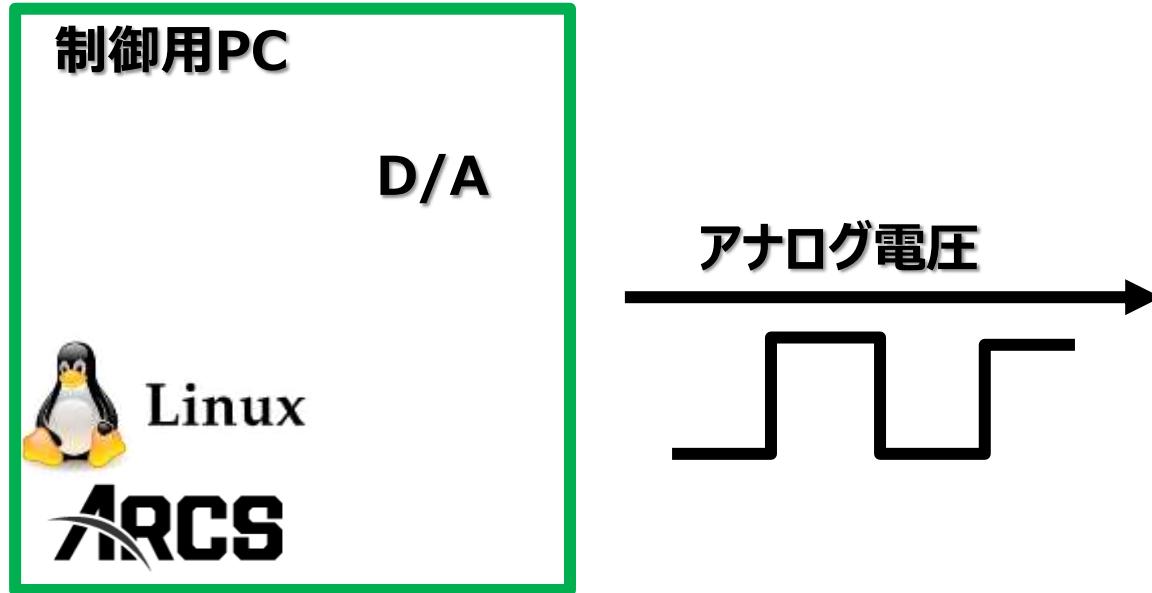
1.1. 電流制御系の応答特性のチェック



三角波の電流指令を出力して
その通りの電流が流れるかチェック

実験の流れ (1日目)

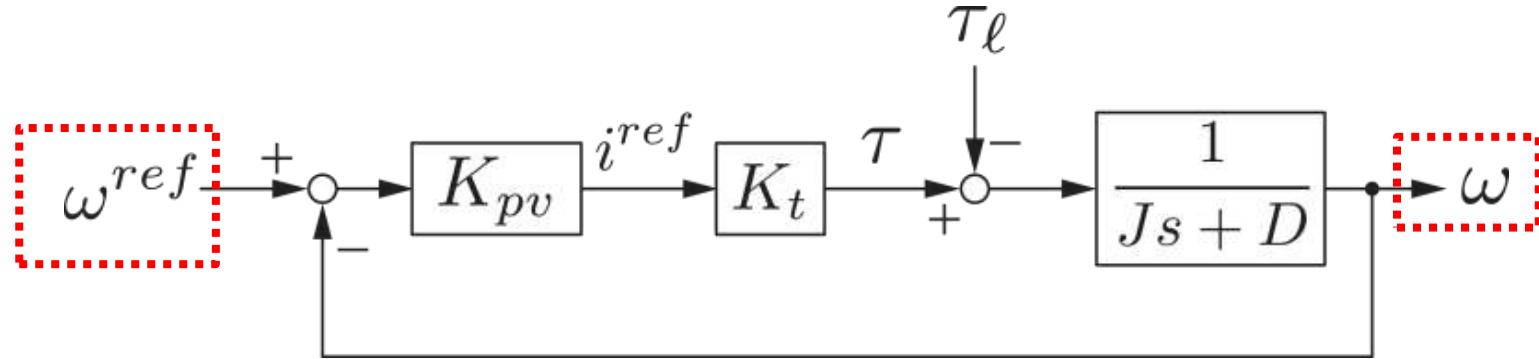
■ 1.2. 制御周期の確認



矩形波の電圧信号を出力して
制御周期が守られているかチェック

実験の流れ (1日目)

1.3. P速度制御系の仮設計



$$\frac{\omega(s)}{\omega^{ref}(s)} = \frac{K_{pv}K_t}{D + K_{pv}K_t} \cdot \frac{1}{\frac{J}{D+K_{pv}K_t}s + 1} = A \cdot \frac{1}{\tau s + 1}$$

Pゲイン **時定数**

係数比較によりPゲインが求まる

慣性モーメント (カタログ値) J 0.20×10^{-4} [kgm²]

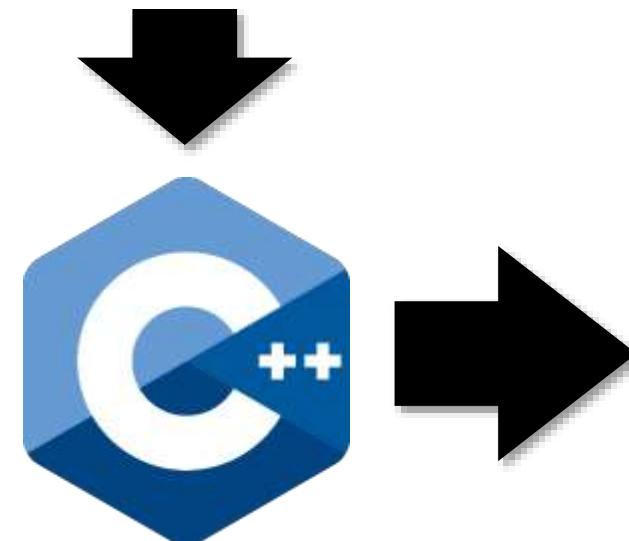
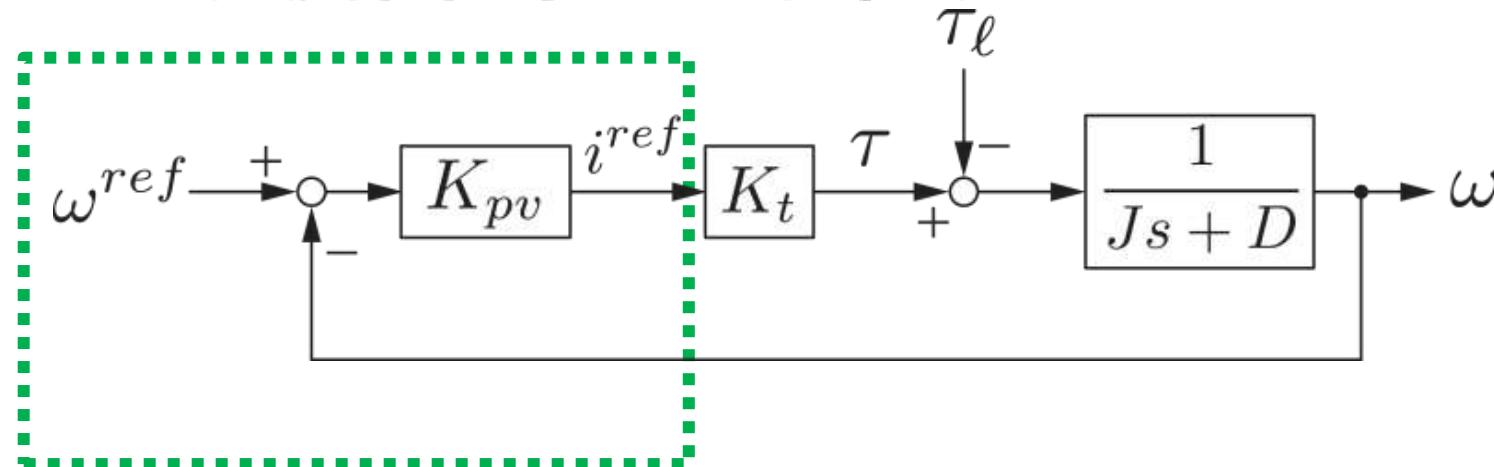
粘性摩擦 D 0.0 [Nm/(rad/s)]

トルク定数 K_t 0.20 [Nm/A]

所望の応答時間 τ 20 [ms]

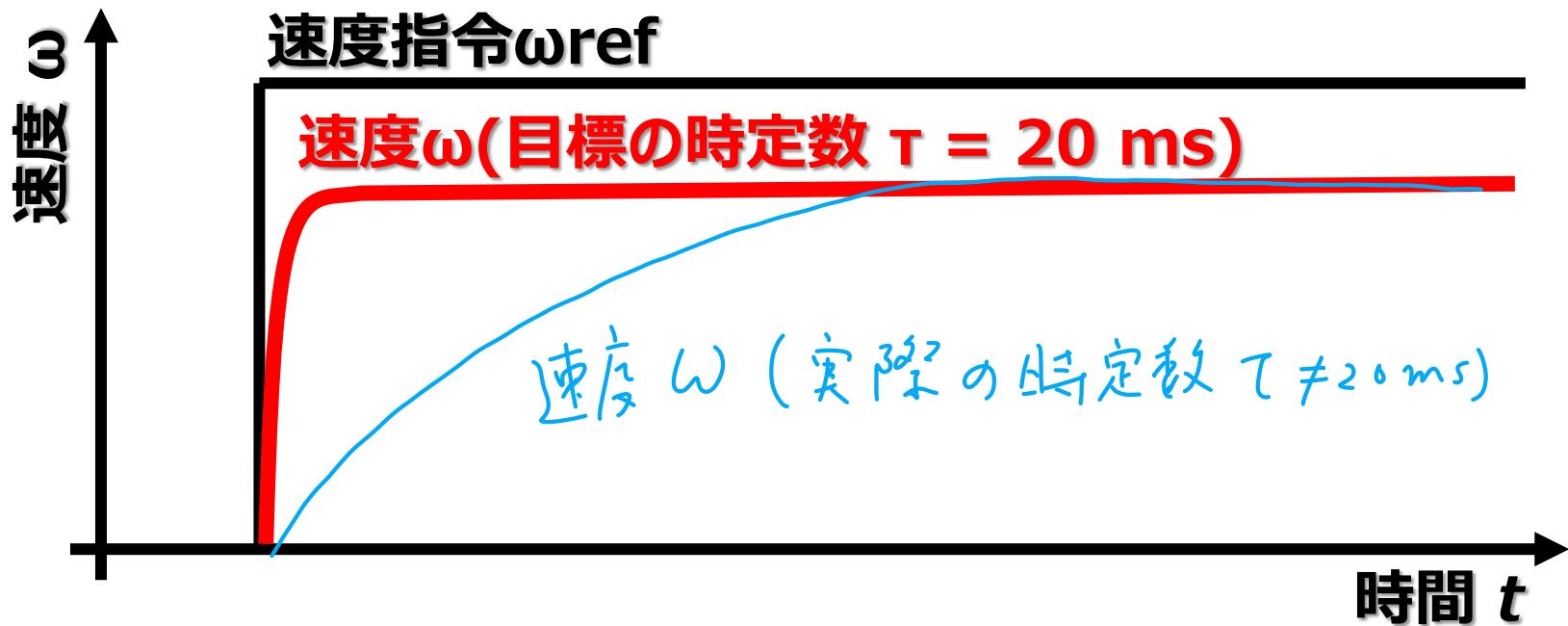
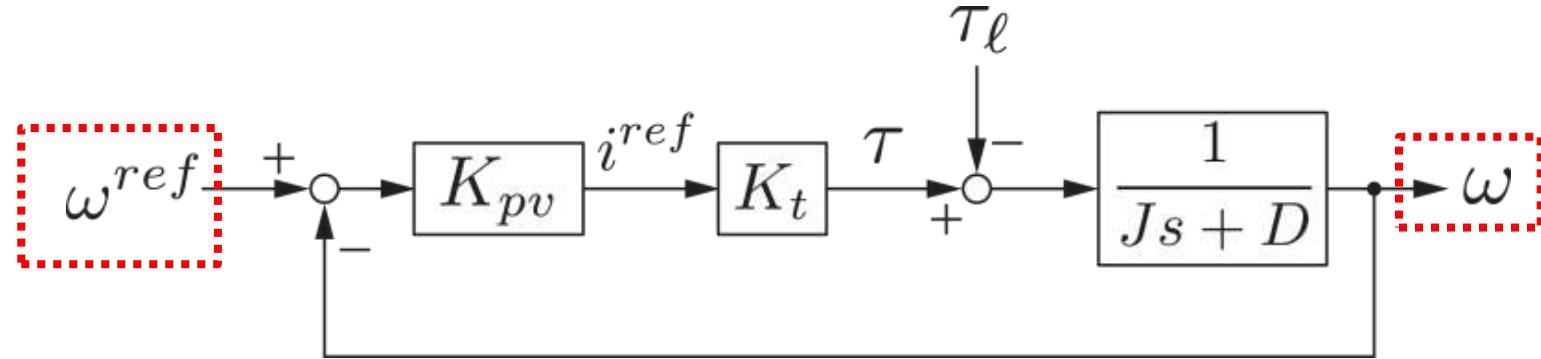
実験の流れ (1日目)

1.4. P速度制御系のコーディング



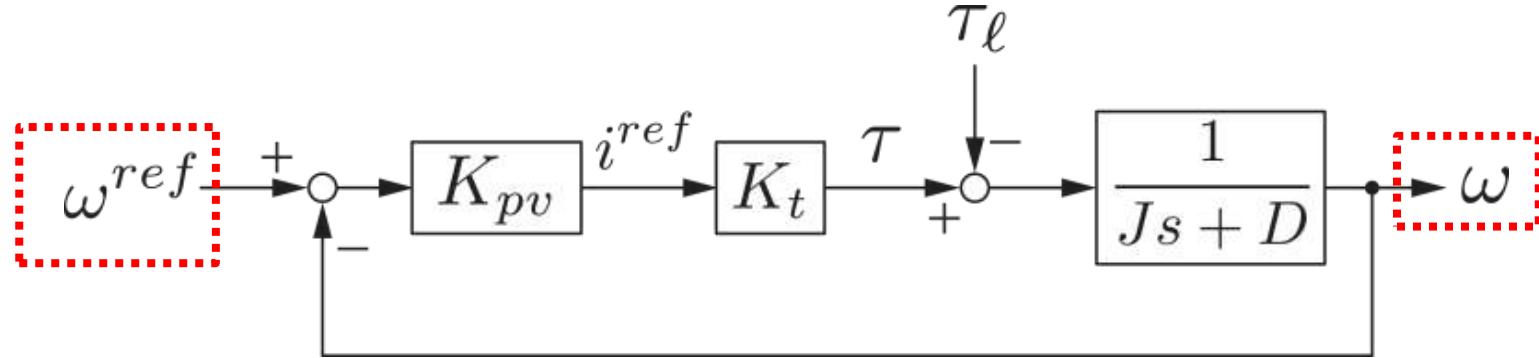
実験の流れ (1日目)

■ 1.5. 慣性モーメントJの同定



実験の流れ (1日目)

■ 6. P速度制御系の再設計



$$\frac{\omega(s)}{\omega^{ref}(s)} = \frac{K_{pv}K_t}{D + K_{pv}K_t} \cdot \frac{1}{\frac{J}{D+K_{pv}K_t}s + 1} = A \cdot \frac{1}{\tau s + 1}$$

Pゲイン **時定数**

係数比較によりPゲインが求まる

慣性モーメント (カタログ値) J ~~0.20×10⁻⁴~~ 同定値を使う! $[kgm^2]$

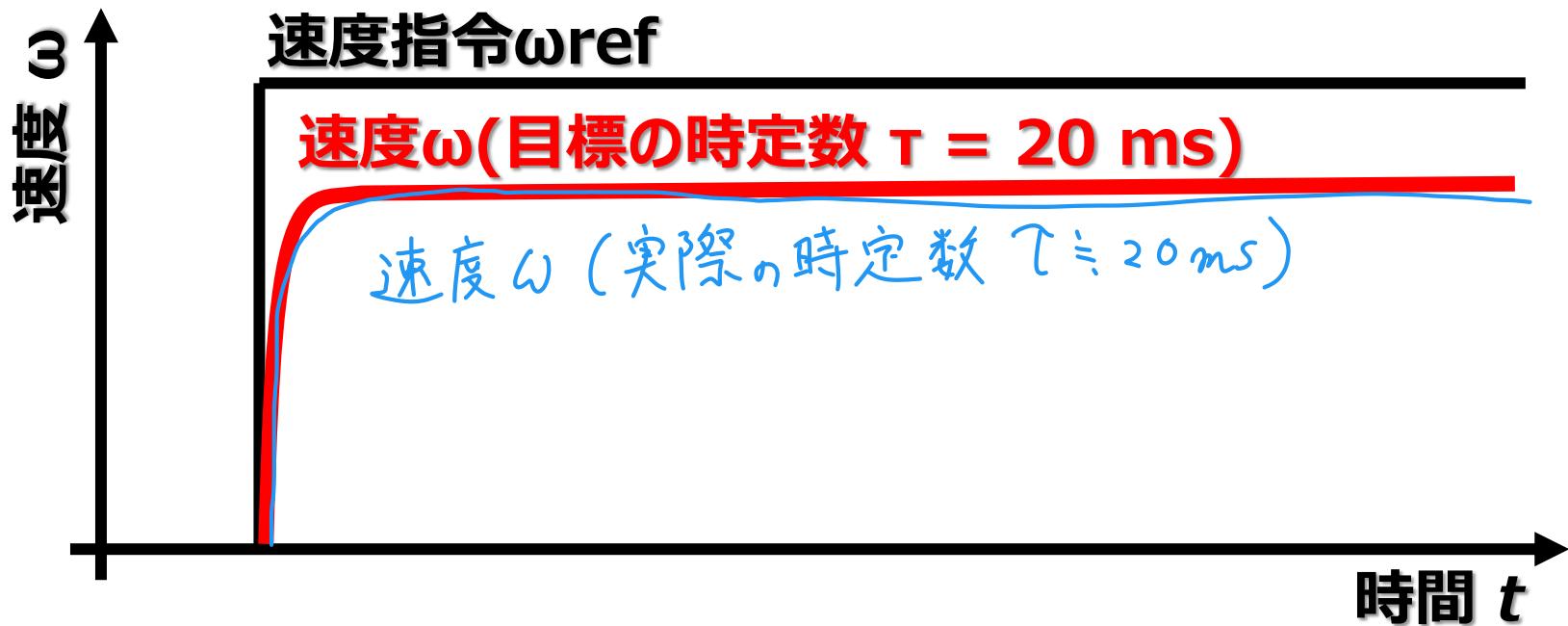
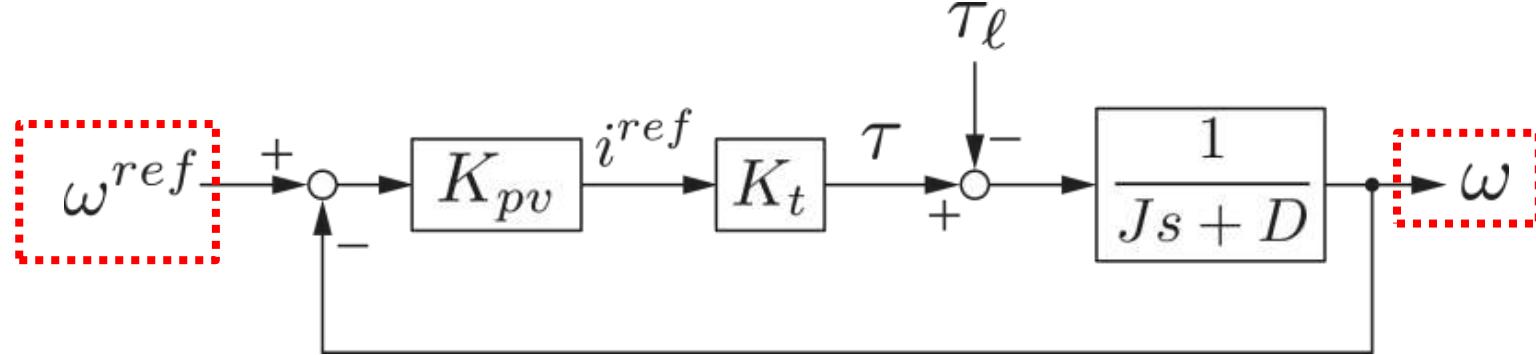
粘性摩擦 D $0.0 [Nm/(rad/s)] \rightarrow$ 実験後の考察課題

トルク定数 K_t $0.20 [Nm/A]$

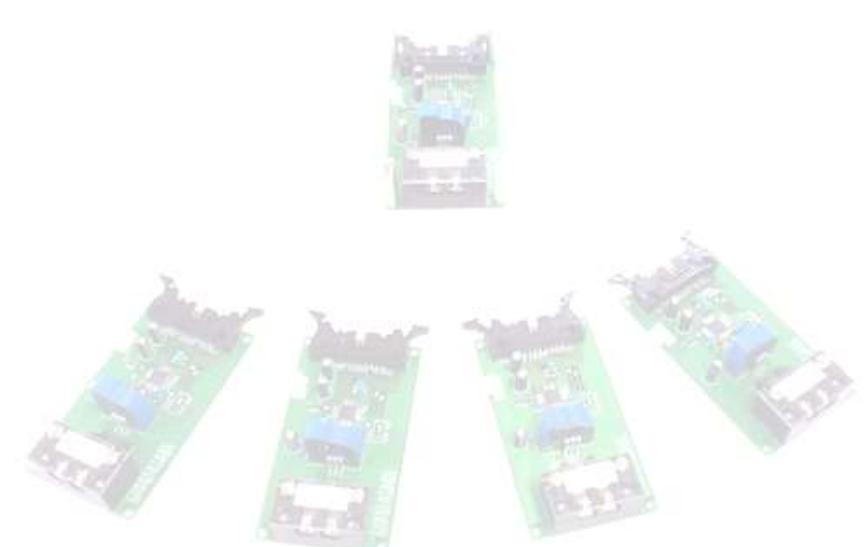
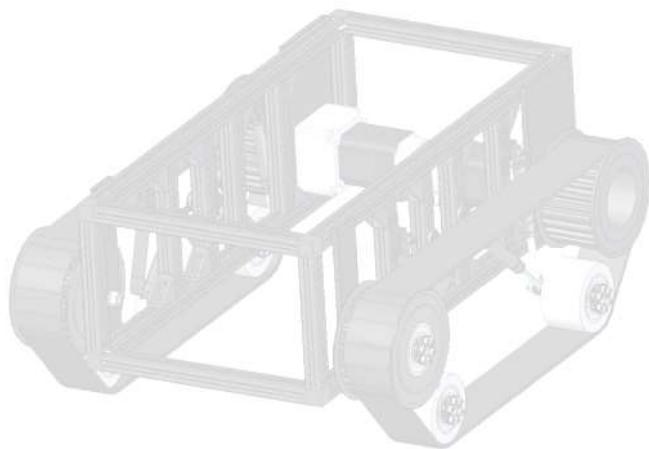
所望の応答時間 τ $20 [ms]$

実験の流れ (1日目)

■ 1.7. 再設計したP速度制御系の動作確認

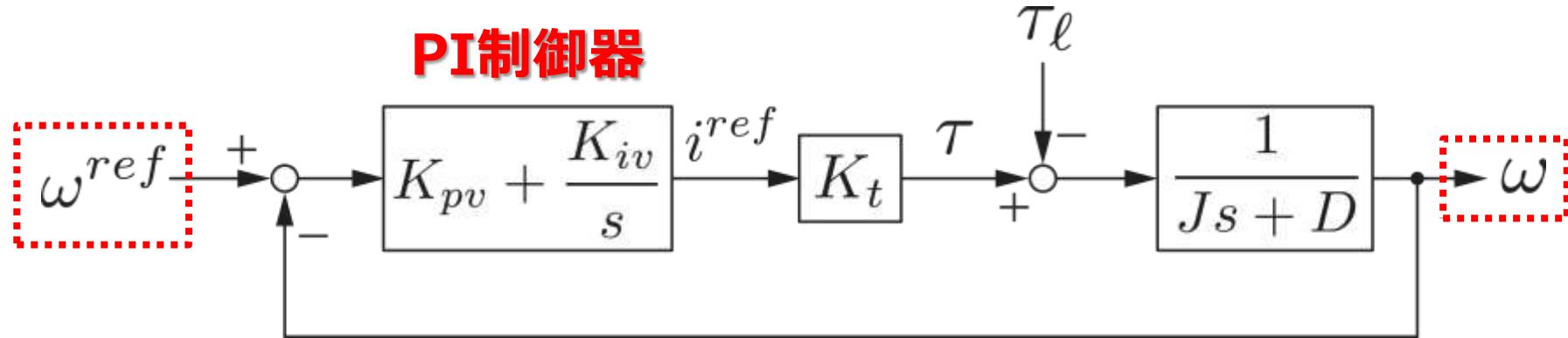


実験の流れ (2日目)



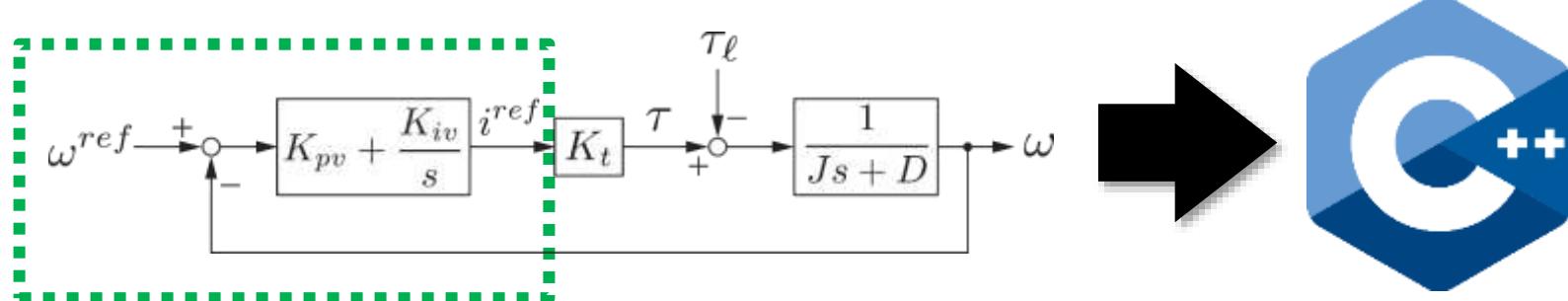
実験の流れ (2日目)

■ 2.1 PI速度制御系の設計



P制御のときと同じようにして、伝達関数を求めて、係数比較してPゲイン K_{pv} とIゲイン K_{iv} を算出

■ 2.2 PI速度制御系のコーディング

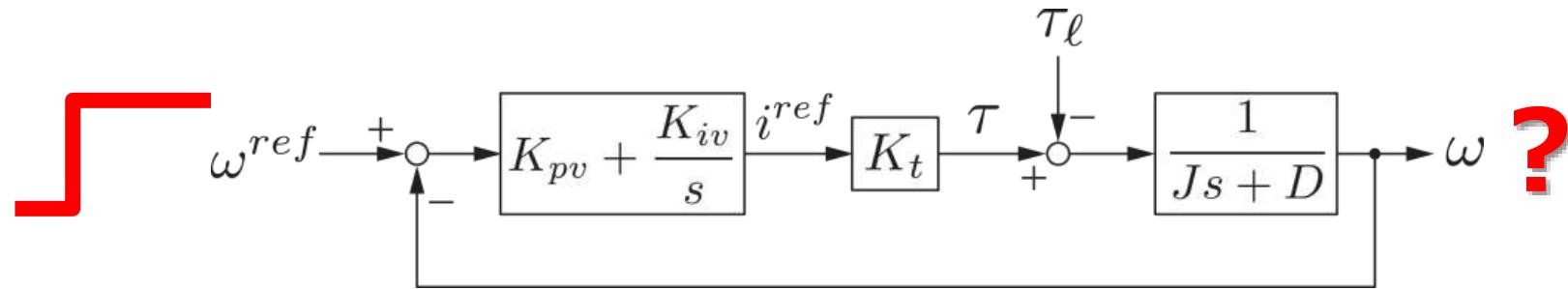


実験の流れ (2日目)

■ 2.3 PI速度制御系の動作確認

P制御のときと同じように動作確認する

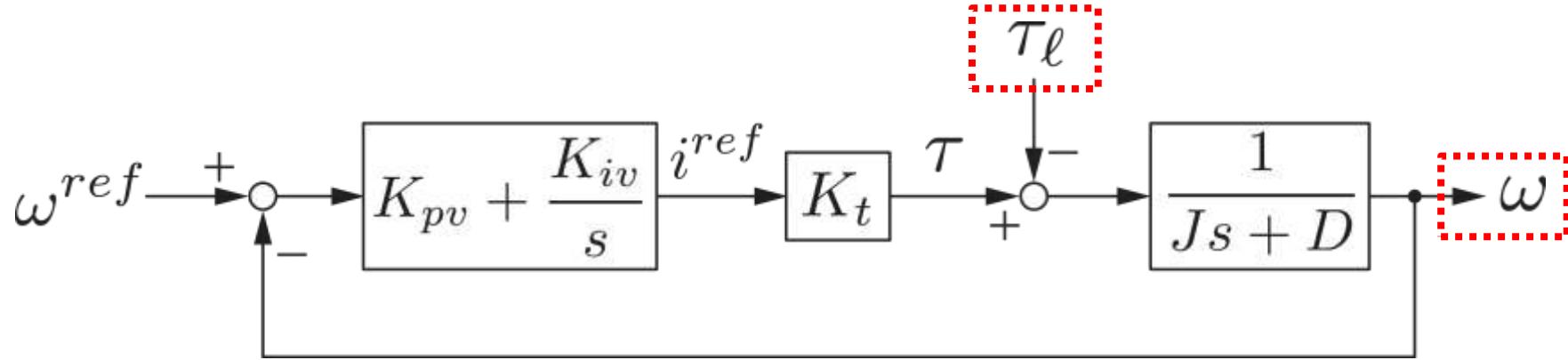
■ 2.4 PI速度制御系のステップ応答



速度指令にステップ応答を入力して速度応答を観測する
制動係数 ζ を3種類やってみて比較する
(目標値応答の観測)

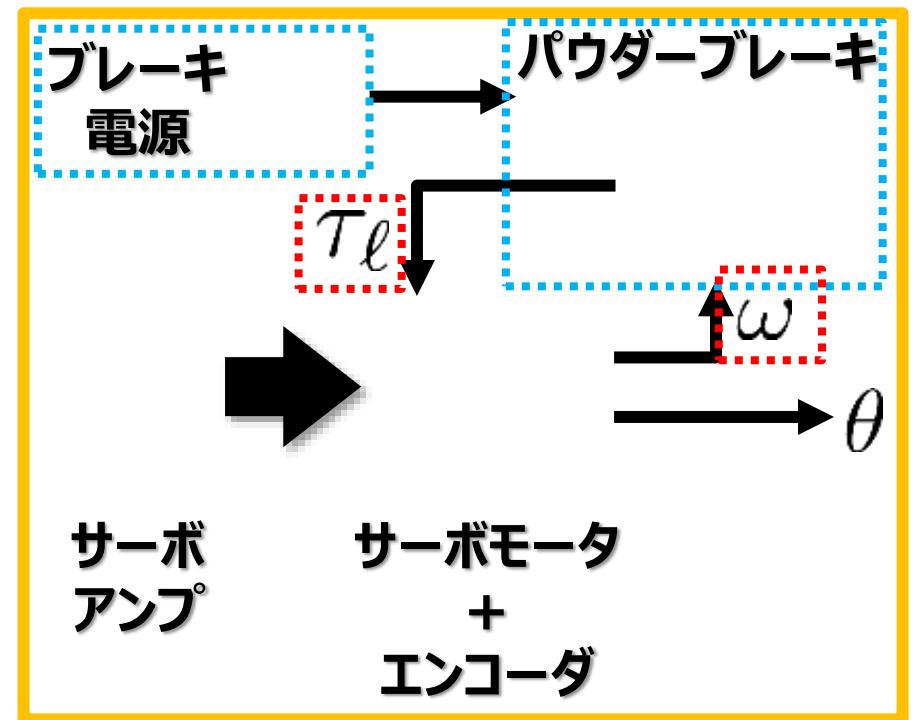
実験の流れ (2日目)

2.5 P速度制御系とPI速度制御系の外乱応答



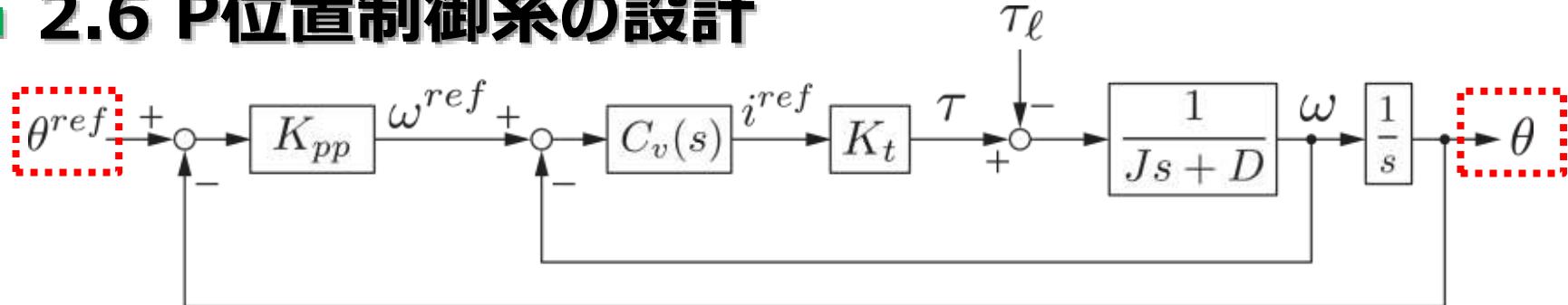
パウダーブレーキを動作させて
負荷トルクにステップ状の
波形を入力して
速度応答を観測する

(外乱応答の観測)



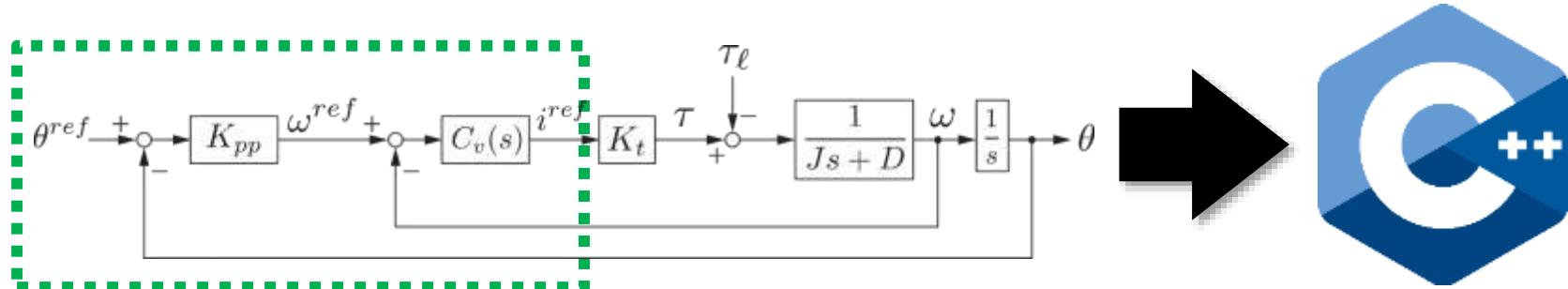
実験の流れ (2日目)

■ 2.6 P位置制御系の設計



PI速度制御のときと同じようにして、伝達関数を求めて、係数比較して位置Pゲイン K_{pp} を算出

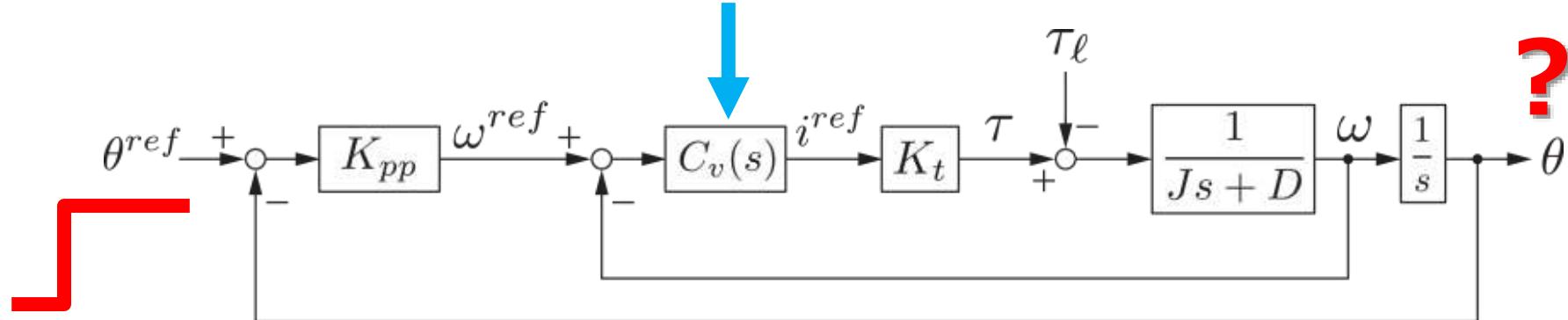
■ 2.7 P位置制御系のコーディング



実験の流れ (2日目)

■ 2.8 P位置制御系 + P/PI速度制御系のステップ応答

P制御器 or PI制御器



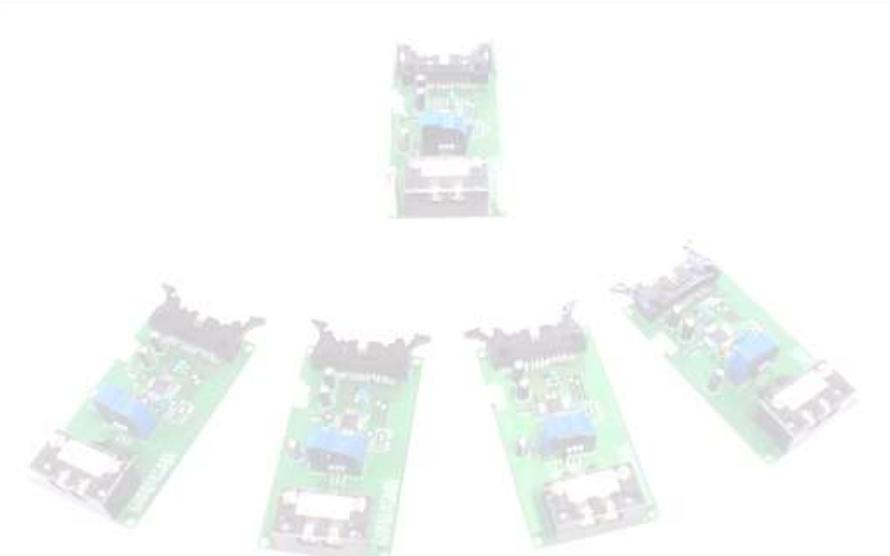
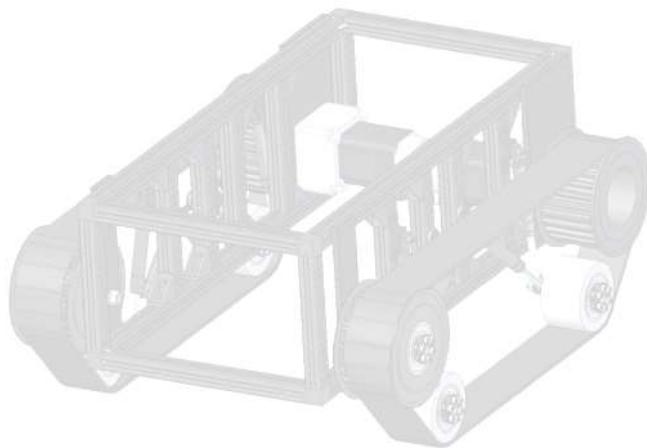
P位置制御系のマイナーループをPもしくはPI速度制御に変えたとき、どんな位置応答になるか？ 実験で確認

～ 実験終了～

本日の講義内容

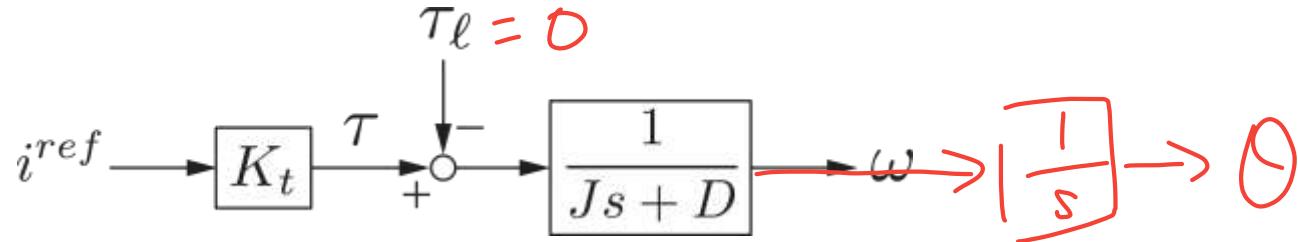
- ACサーボモータとは？
- ACサーボモータのモデル化
- ACサーボモータ制御の全体像
- ACサーボモータの電流制御
- ACサーボモータの速度制御
- ACサーボモータの位置制御
- ディジタル制御系での積分器の作り方
- 実験機ハードウェアの説明
- 実験の流れ（1日目）
- 実験の流れ（2日目）
- レポート課題

レポート課題



レポート課題

- 課題1：今日の講義内容を「自分の言葉で！」まとめよ。
- 課題2：下図の電流指令 i^{ref} から速度 ω までの伝達関数と、電流指令 i^{ref} から位置 θ までの伝達関数を求めよ。



- 課題3：P速度制御系とPI速度制御系の速度指令 ω^{ref} から速度 ω までと、負荷トルク τ_l （たうえる）から速度 ω までのそれぞれ2つの伝達関数を求めよ。
- 課題4：PI制御器を双一次変換せよ。

レポート課題

- **課題5 :**
電流指令からサーボアンプ（モータドライバ）までは以下のよう構成になっている。



<換算式Aについて>

本実験で使用するサーボアンプは、5Vのアナログ電圧信号を入力すると9.6Aの電流をモータに流すように作られている。電流指令[A]から電圧値[V]に変換する式を作れ。

ヒント：+9.6を入れると5を、0を入れると0を、
-9.6を入れると-5を出力する式を作ればよい。

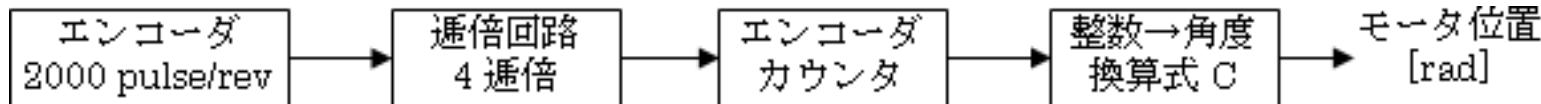
<換算式Bについて>

~~本実験で使用するD/A変換器の分解能は16ビットであり、表現方法は表1-1に示す通りである。アナログ信号の最大電圧振幅は±5Vである。所望の電圧値[V]からD/A変換器に入力すべき整数値（10進数）に変換する式を作れ。~~

ヒント：+5を入れると65536を、0を入れると32768を、
-5を入れると0を出力する式を作ればよい。
65535

レポート課題

- 課題6：位置センサとしてインクリメンタルロータリエンコーダを用い、エンコーダからのパルスをカウントしてモータ位置を取得する。



本実験で使用するエンコーダはモータが1回転すると2000パルスを発生し、その後に遙倍回路によってパルスが4遙倍されてからエンコーダカウンタに入力される。すなわち、モータ位置が0 [rad]から 2π [rad]まで回転すると8000パルスが出力され、エンコーダカウンタの整数値は0から8000に増加する。エンコーダカウンタの整数値からモータ位置 [rad]に変換する式を作れ。

ヒント：8000を入れると 2π を、
0を入れると0を出力する式を作ればよい。