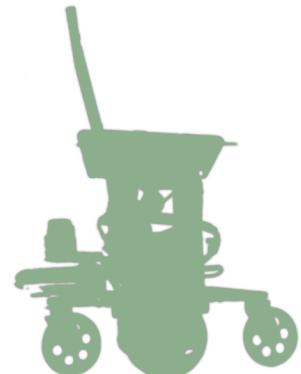


移動ロボットのモータ制御と走行制御

渡辺 敦志

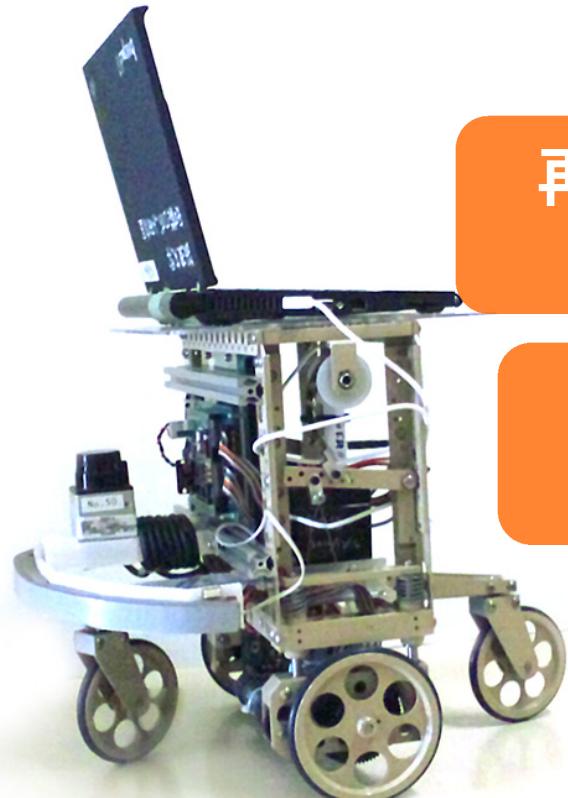


<http://openspur.org/~atsushi.w/>
atsushi.w@ieee.org

2 本セミナーの内容で実現できること

対象：車輪型移動ロボットなど

モータ + ギア + ホイール + ロータリエンコーダ

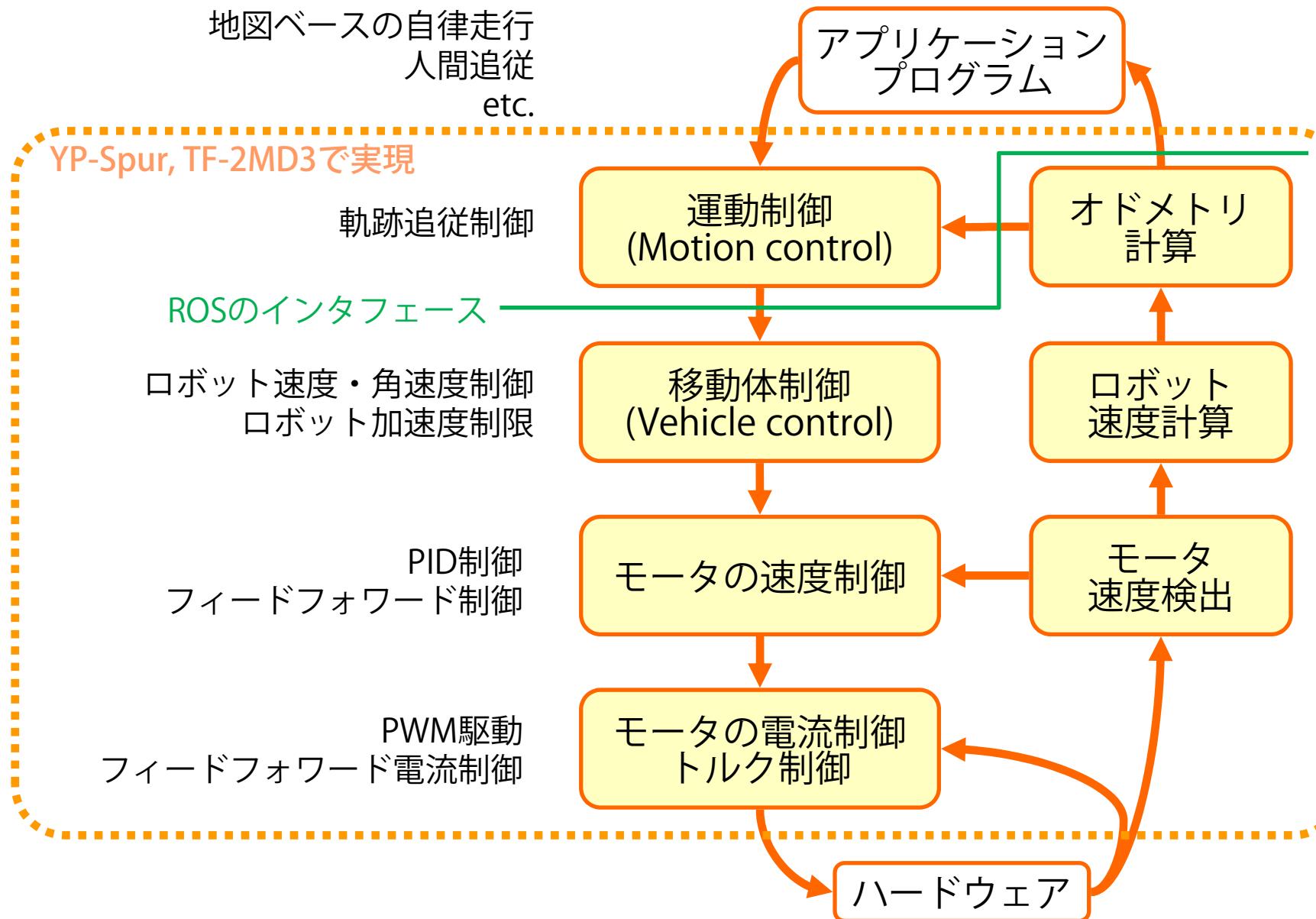


再現性が高く高精度な
位置決め・動作制御

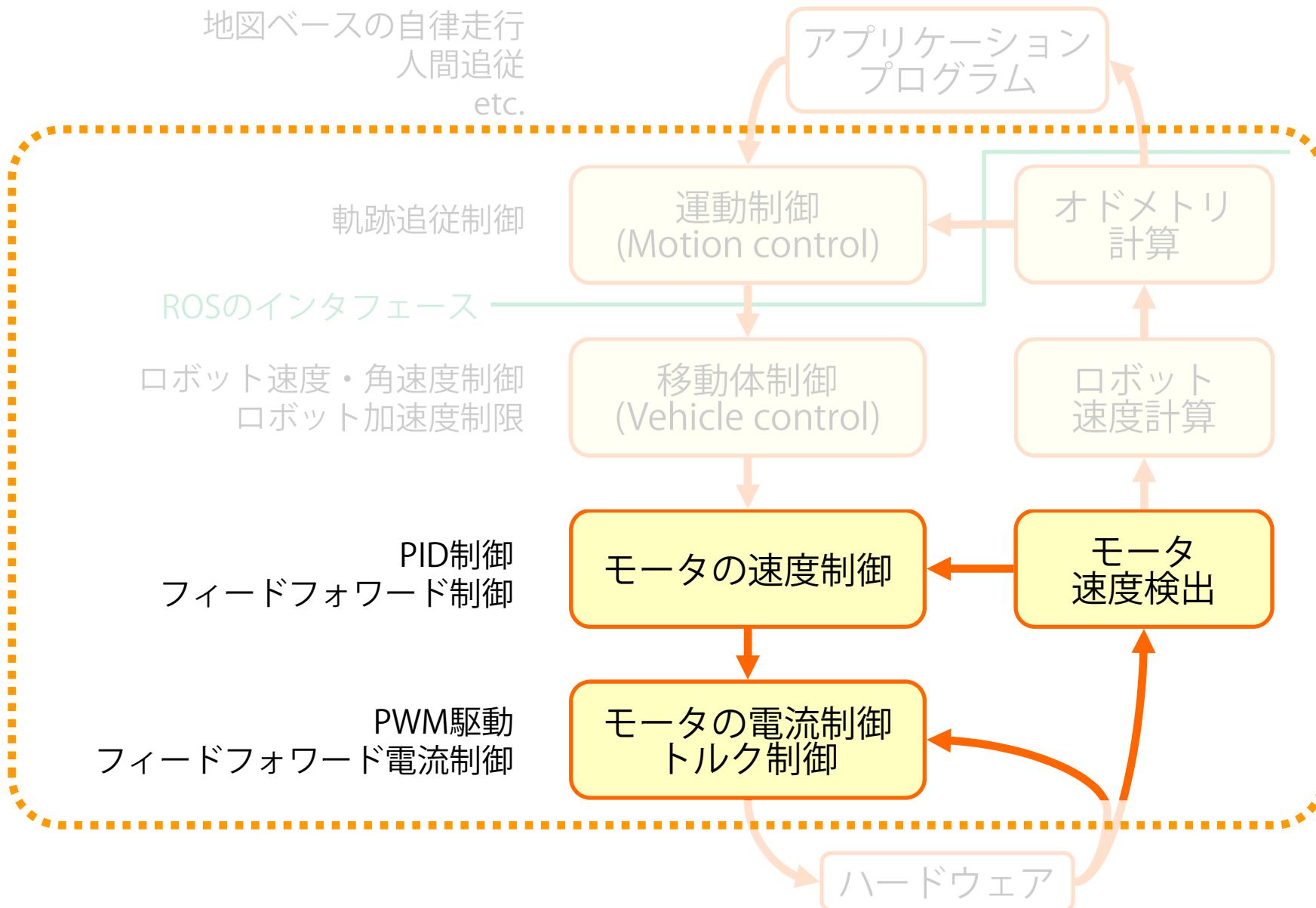
ダイナミクス等を考慮した
応答性能の良い制御

スリップ・横転を防ぐ
物理的制約を考慮した制御

移動ロボットを動かすには

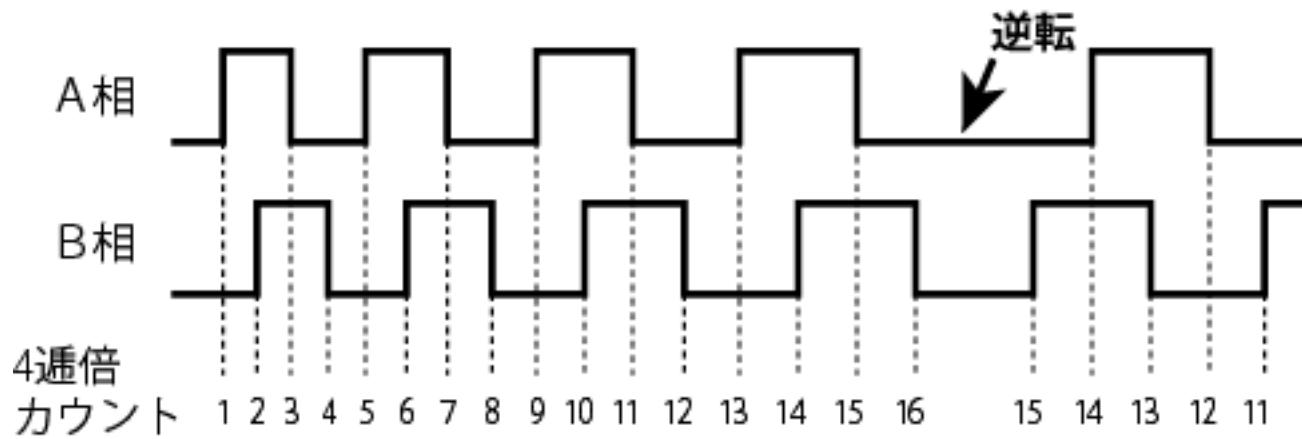


移動ロボットを動かすには (前半: モータ制御)

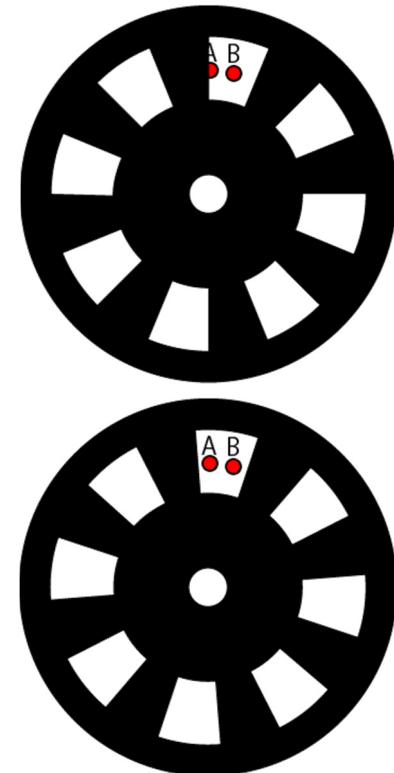


5 モータ回転数の検出

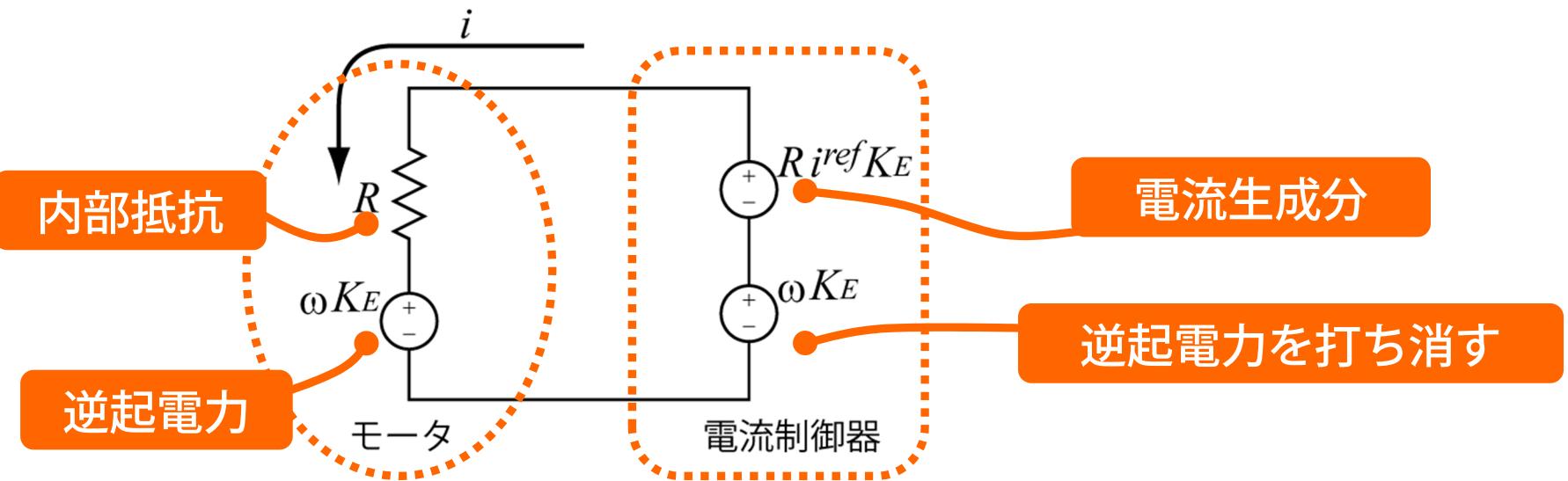
- モータに取り付けられた
2相インクリメンタルエンコーダを使用



- 1ミリ秒中の角度変化(パルス数)を
角速度として利用



6 モータの電流制御(トルク制御)



- 逆起電力 + a の電圧を与えると、 $i > 0$
 - 加速
- 逆起電力 - a の電圧を与えると、 $i < 0$
 - 減速

与えたいたトルクからPWM比を決定

- トルクに比例した電流を流す

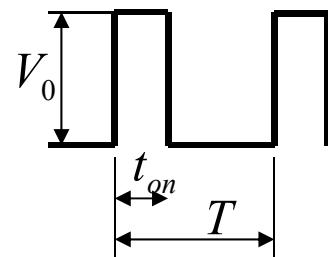
$$i^{ref} = \frac{\tau^{ref}}{k_\tau}$$

- モータの起電力に逆らって電圧を印加

$$V^{ref} = R i^{ref} + K_E \omega$$

- PWMで目標電圧と平均的に等価な電圧を与える

$$pwm = \frac{t_{on}}{T} = \frac{V^{ref}}{V_0}$$



τ^{ref} : 与えるトルク

i^{ref} : 与える電流

K_τ : モータのトルク定数

R : モータの端子間抵抗

K_E : モータ誘導起電力定数

V_0 : 電源電圧

8 ちなみに：トルク定数と逆起電力定数

- トルク定数[Nm/A]と逆起電力定数[V sec./rad]は単位は違うが物理的に等価

※逆起電力：電流を流す力 = 電圧 (powerではなくforce)

モータが角速度 ω で回転しているとき

コイルに発生する逆起電力は コイルに発生するトルクは

$$E = K_E \omega$$

$$\tau = K_\tau i$$

コイルに注入された電力は コイルが発した動力は

$$P = E i$$

$$P = \tau \omega$$

エネルギー保存則より $E i = \tau \omega$

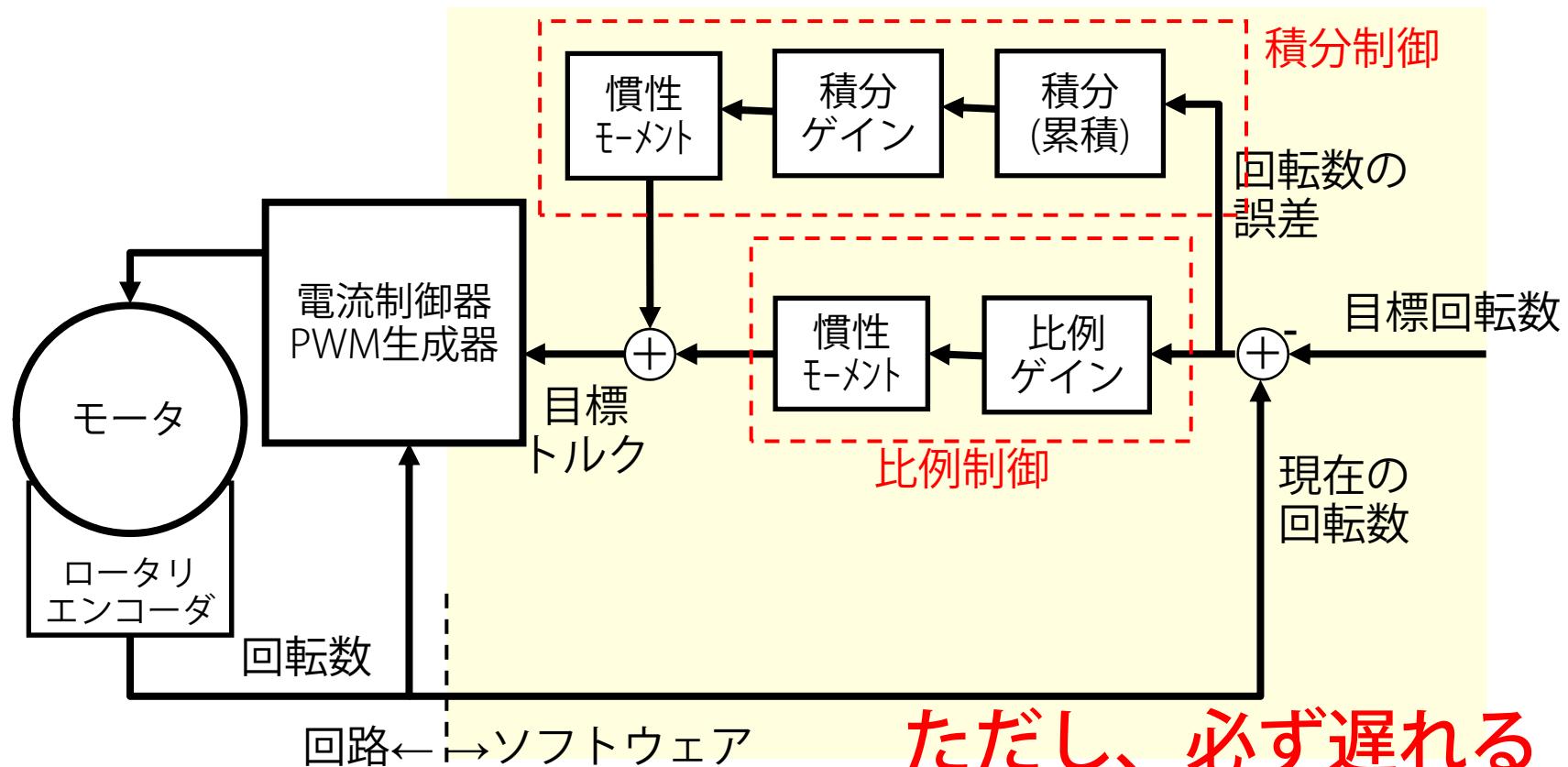
$$K_E \omega i = K_\tau i \omega$$

$$K_E = K_\tau$$

詳細不明なモータに出会ったら、回転させて電圧を測る

9 モータの速度 PI制御

- 回転数の誤差に比例したトルクを与える
(比例制御/P制御)
- 誤差が残り続けたときには更にトルクを与える
(積分制御/I制御)

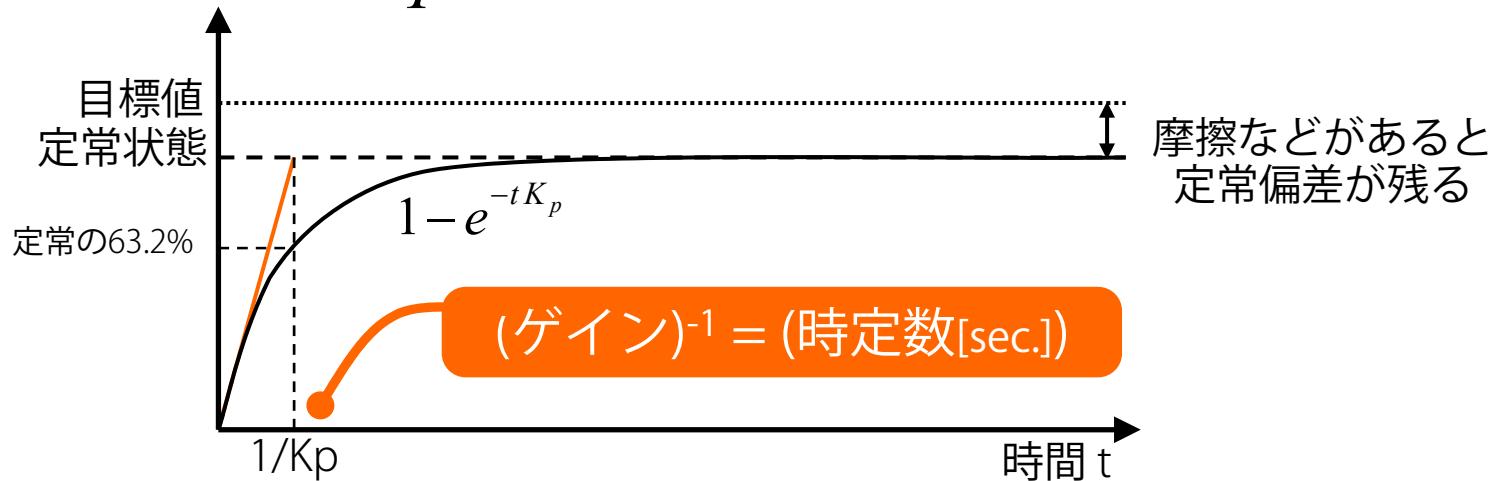


10 PI制御ゲイン

- 比例ゲインの意味

$\tau = -(\omega - \omega^{ref}) I K_p$ でフィードバックを与える場合

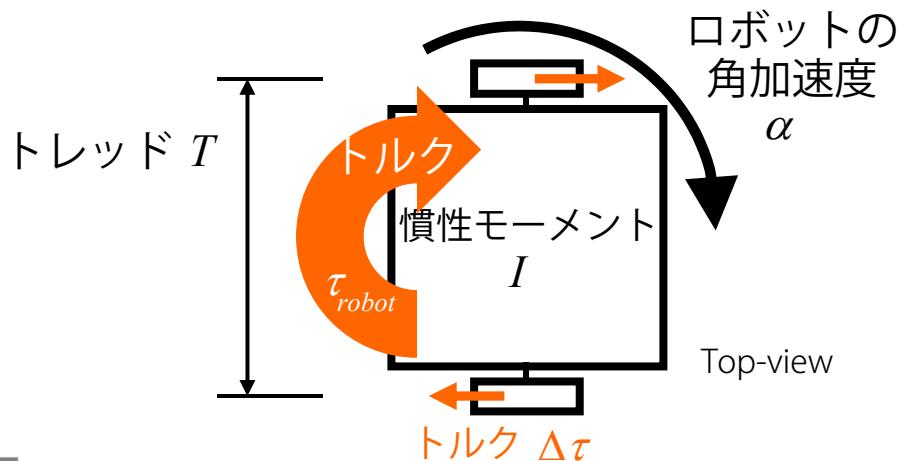
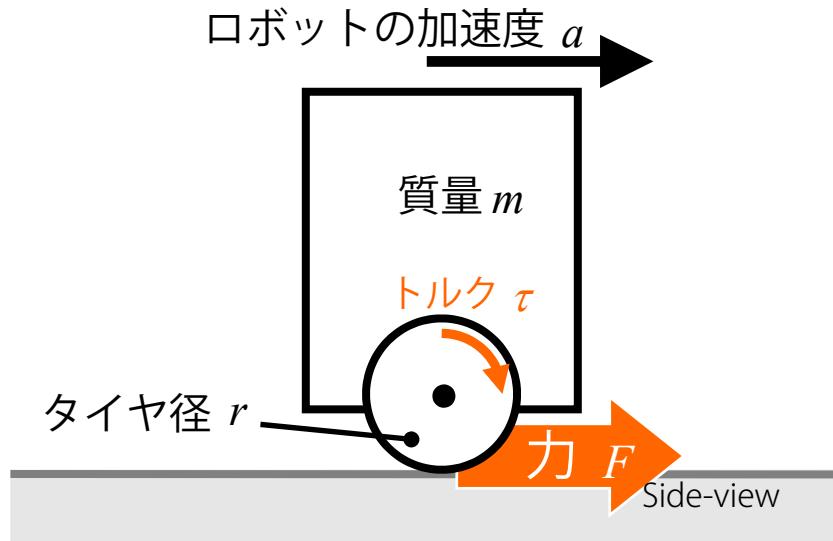
この微分方程式 $\frac{\tau}{I} = \alpha = -(\omega - \omega^{ref}) K_p$ を解くと…



ゲインを決める = 制御が何秒で応答してほしいか決める

モータの速度制御の力学

- 運動方程式 $F = m a$
- 必要トルク $\tau = F r$ (両輪で)

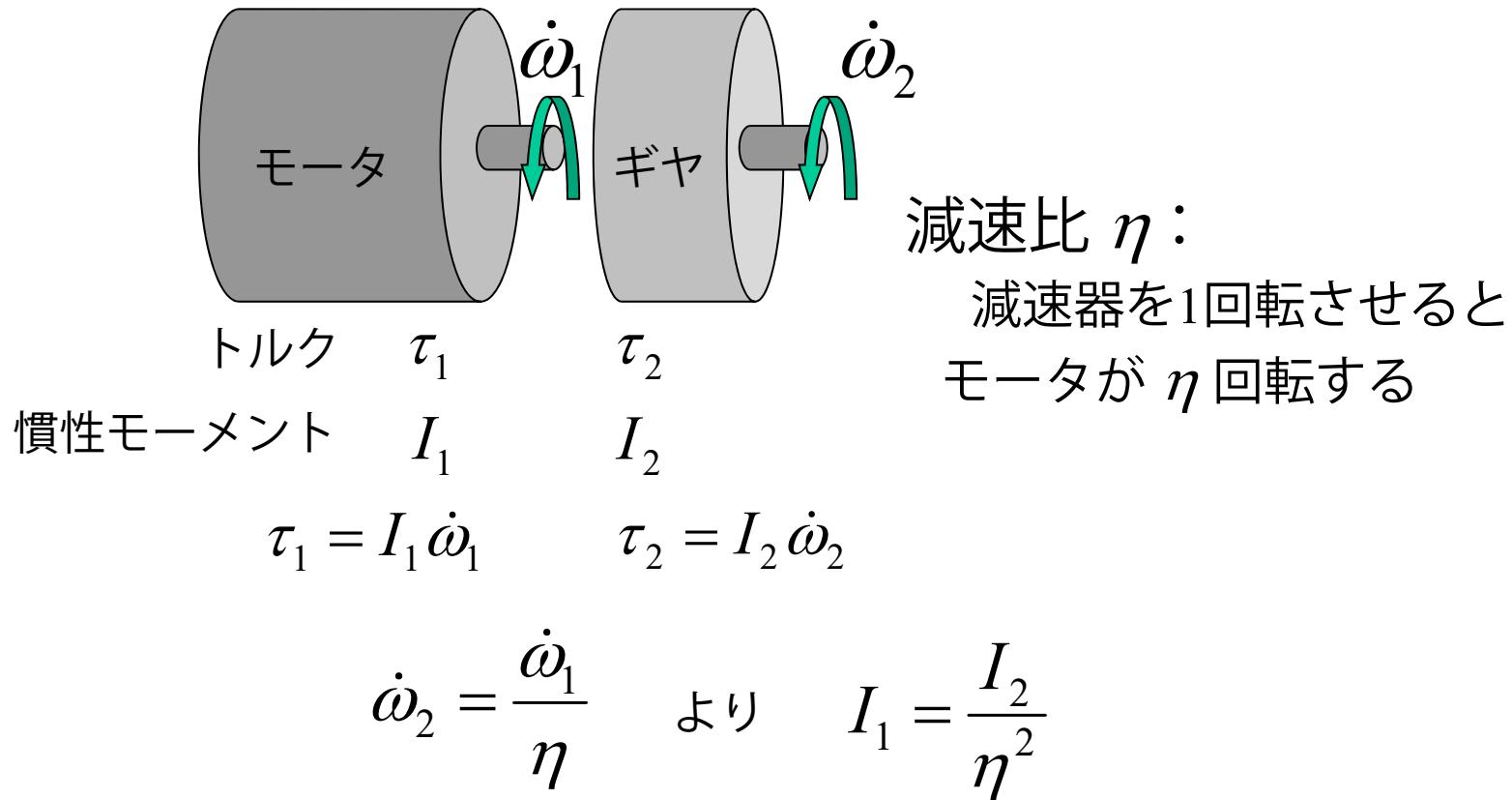


- 運動方程式 $\tau_R = I\alpha$
- 必要トルク $\Delta\tau = \tau_R r / T$

いま行おうとしている運動に必要なトルクを
予め計算して与える(遅れなし!)

ギヤがある時

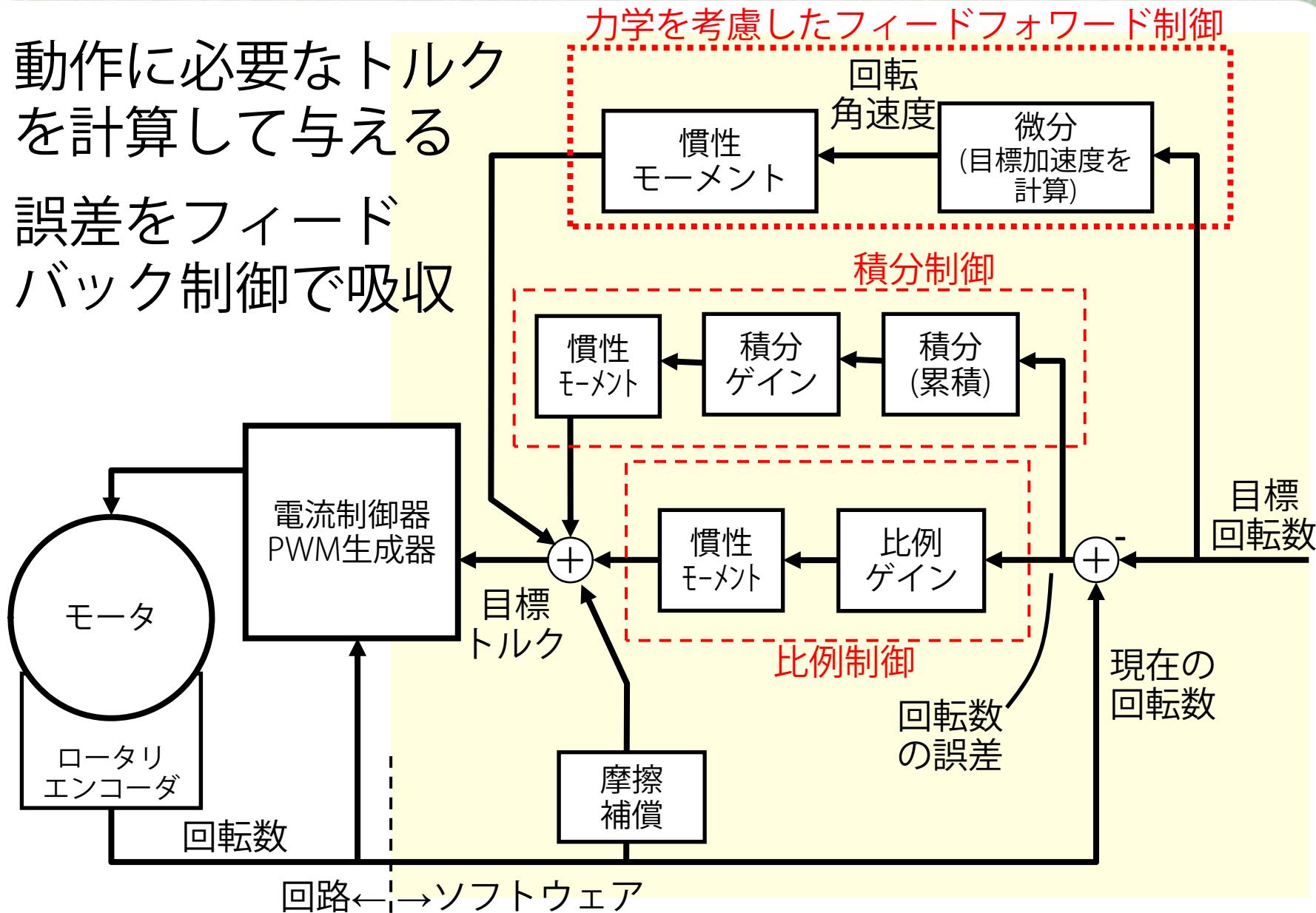
- 減速比 η のギヤが、モータに付いているとき



モータ軸から見た慣性モーメントは $1/\eta^2$ 倍に減少

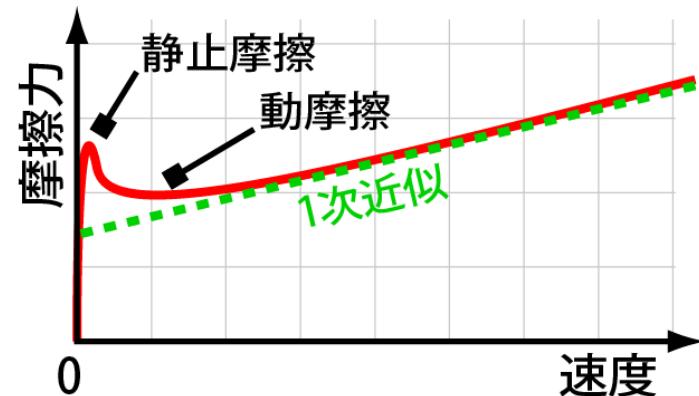
モータの速度フィードフォワード制御

- 動作に必要なトルクを計算して与える
- 誤差をフィードバック制御で吸収



摩擦補償

- ・ギア・ベアリングの摩擦
タイヤの受ける転がり抵抗
 - 小型移動ロボットでは、
ギアの摩擦が大きい
- ・速度によって摩擦が変化
 - この制御系では1次近似

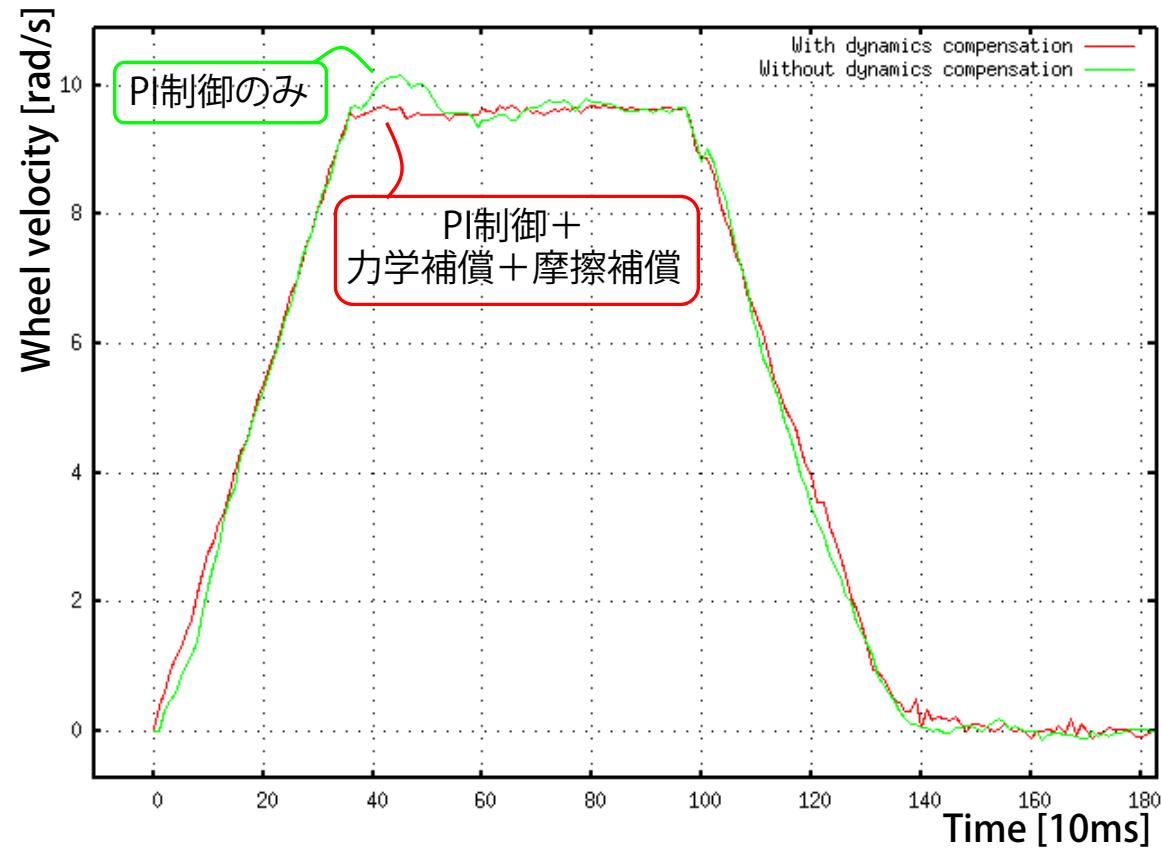


**速度に応じて摩擦力に相当する
トルクを予め与える**

※摩擦係数を大きく設定して摩擦補償すると暴走
→余分にトルクを発生 → 加速 → 更にトルクを発生 → 加速

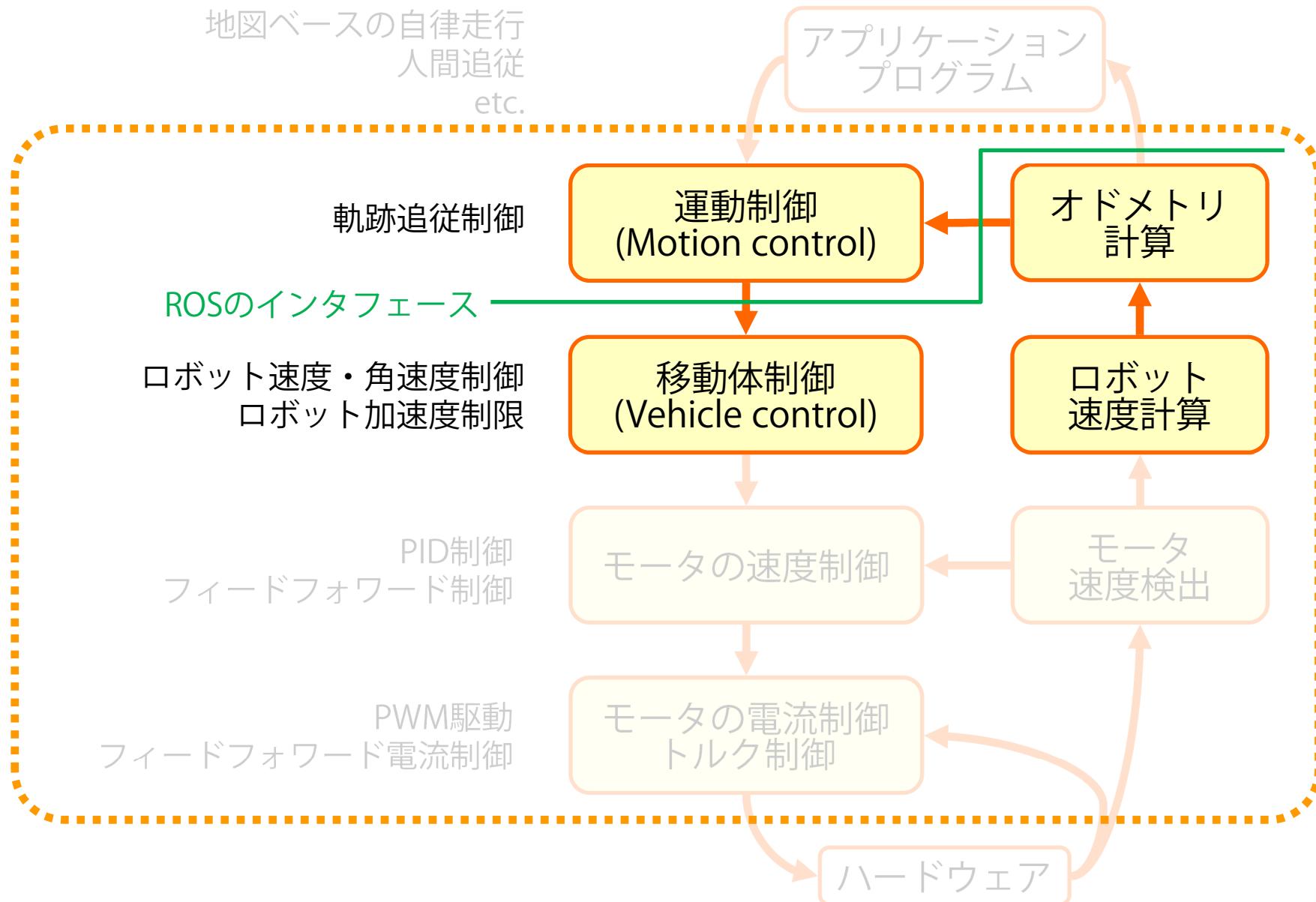
15 フィードフォワードを活用した制御

- モータ特性・力学をよく考慮することで、応答性能などを向上



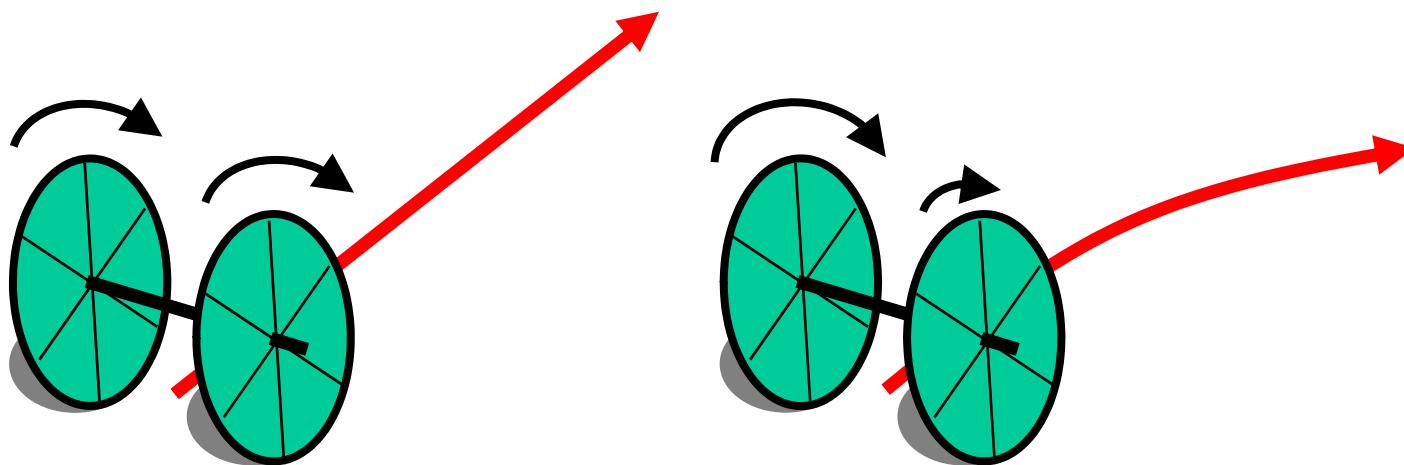
0.6 m/s, 1.0 m/s² で1.0 m 先で停止する位置制御

移動ロボットを動かすには (後半: ロボット全体の制御)



ロボットの速度・角速度(逆 kinematics)

- 左右動輪の平均回転数で、並進速度が発生
- 左右動輪の回転数の差で、角速度が発生



- v, ω から左右車輪の角速度 ω_L, ω_R を計算
 - この逆計算で、ロボット速度・角速度を計算

$$\begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{r_R}{2} & \frac{r_L}{2} \\ \frac{r_R}{T} & -\frac{r_L}{T} \end{bmatrix}^{-1} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{r_R} & \frac{T}{2r_R} \\ \frac{1}{r_L} & -\frac{T}{2r_L} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

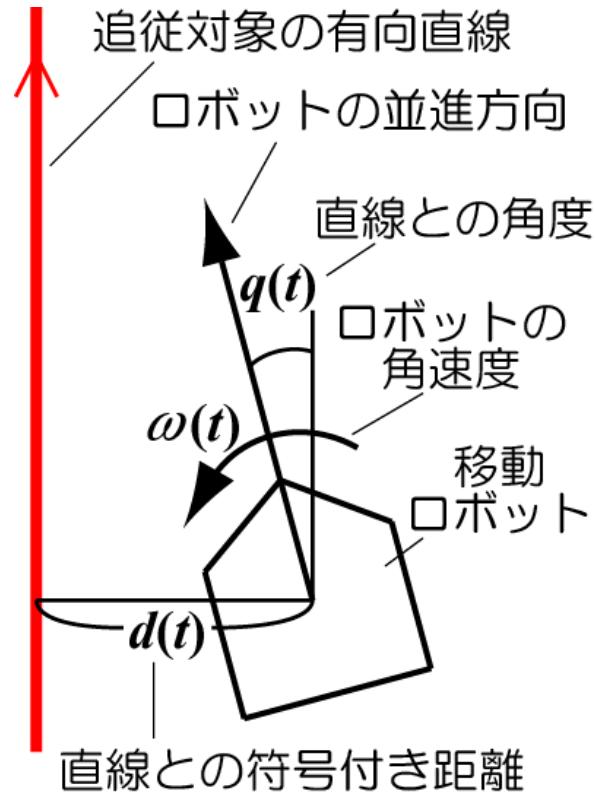
18 制御値のリミット

- 速度・加速度・角速度・角加速度の制限
 - ハードウェア制約
 - 転倒防止
 - スリップ抑制
- トルクの制限
 - モータの焼損防止
 - 接触時の安全性
 - スリップ抑制



動画：出力トルクを 0 にすれば、タイヤは滑らない

直線追従制御



位置・角度のずれと
角速度を、0にするように
フィードバック制御

[η_{max} でクリップした位置偏差]

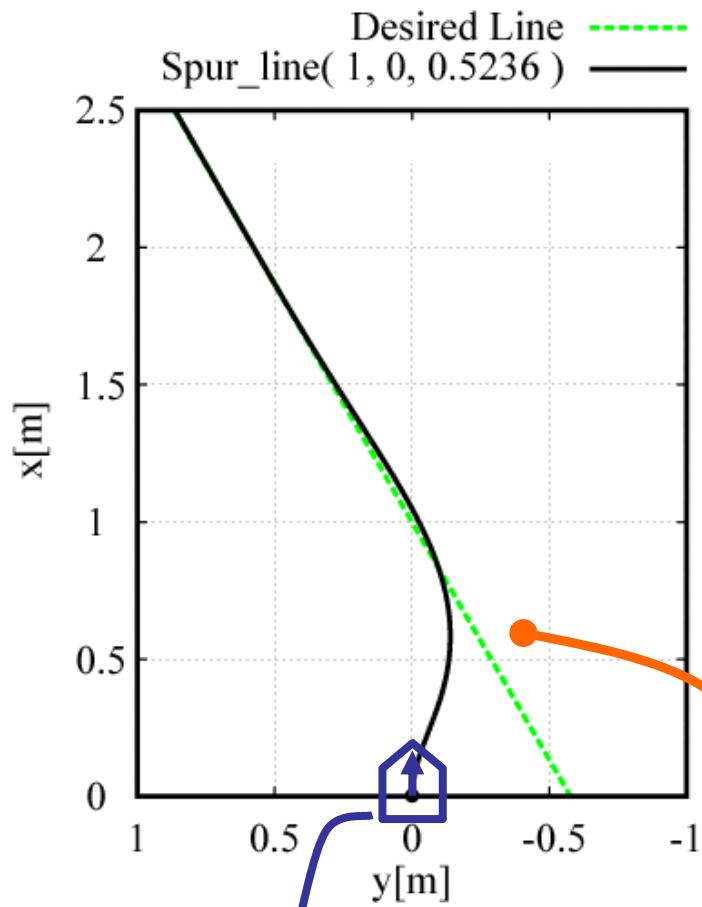
$$\begin{cases} \eta(t) &= \begin{cases} d(t) & |d(t)| < \eta_{max} \\ \eta_{max} & otherwise \end{cases} \\ \phi(t) &= q(t) \quad [角度偏差] \\ \omega_{diff}(t) &= \omega(t) \quad [角速度] \end{cases}$$

角速度目標値

制御周期

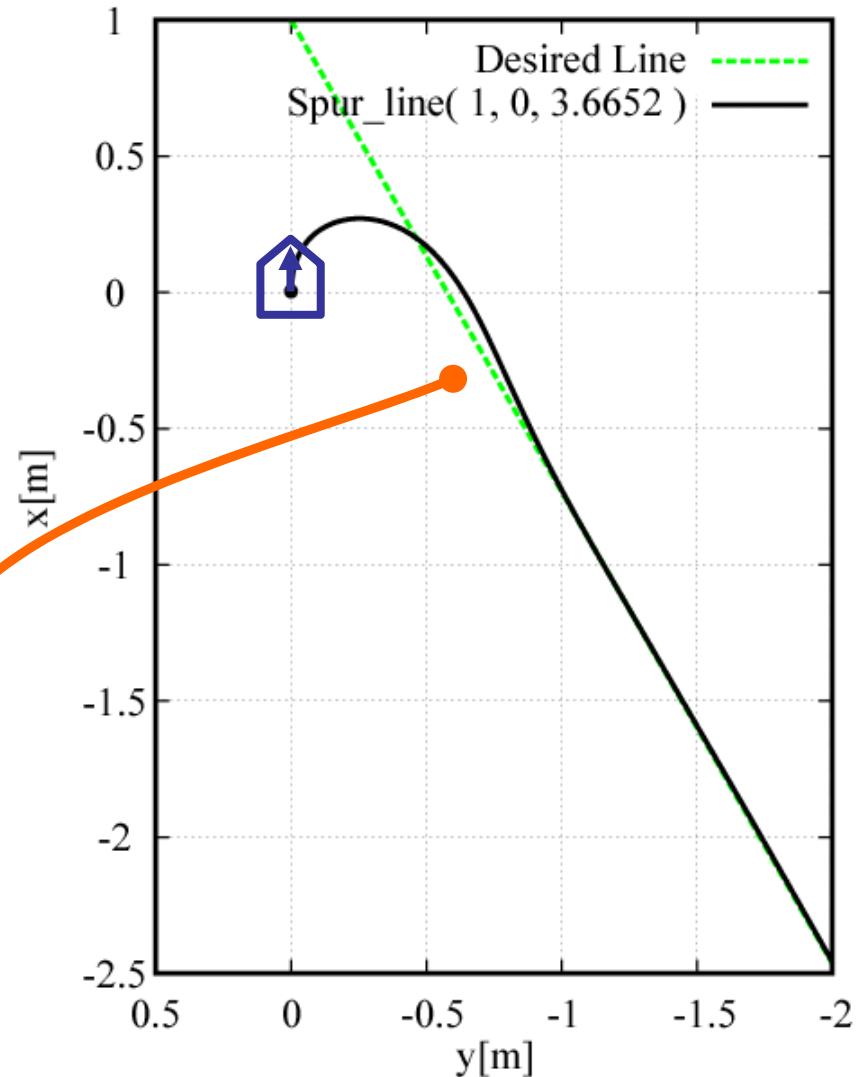
$$\omega^{ref}(t + \Delta t) = \omega(t) + \Delta t (-K_\eta \eta(t) - K_\phi \phi(t) - K_\omega \omega_{diff}(t))$$

直線追従制御の動作例

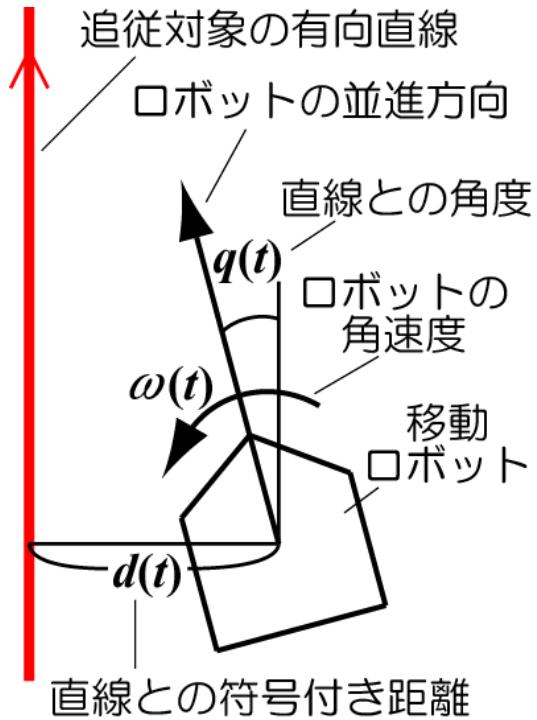


初期位置・姿勢

安定に追従



2.1 直線追従制御の特性



移動体のキネマティクスは0度付近で…

$$y(t) = \int_0^t v(\tau) \sin[\theta(\tau)] d\tau = v^{ref} \int_0^t \theta(\tau) d\tau$$

0度付近での各変数の関係は

$$\frac{d}{dt} \eta(t) = v^{ref} \phi(t)$$

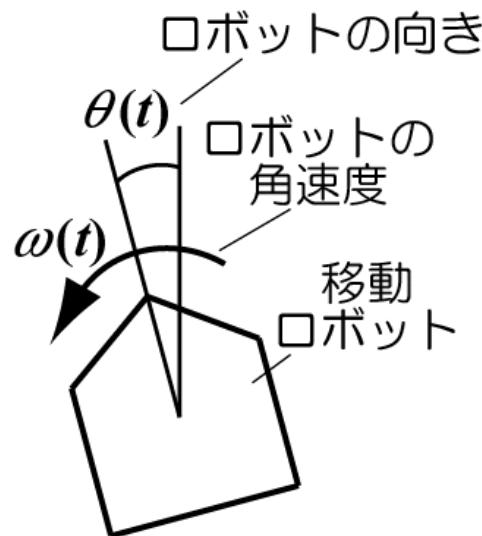
$$\frac{d}{dt} \phi(t) = \omega_{diff}(t) = \omega(t)$$

直線追従制御系の位置・姿勢の挙動

$$\frac{d^3}{dt^3} \phi(t) = -K_\eta v^{ref} \phi(t) - K_\phi \frac{d}{dt} \phi(t) - K_\omega \frac{d^2}{dt^2} \phi(t)$$

姿勢(角度)の制御

**最大角加速度で減速して
目標角度で停止するように制御
(最短時間制御)**



t_0 でちょうど停止するように

$$\left\{ \begin{array}{l} \omega(t + t_0) = 0 \\ \theta(t) + \int_t^{t+t_0} \omega(t) dt = \theta^{ref} \quad [\text{目標角度}] \\ \frac{d}{dt} \omega(t) = \alpha_{max} \quad [\text{最大角加速度}] \end{array} \right.$$

を満たせばよい

$$\omega^{ref}(t + \Delta t) = sign(\theta(t) - \theta^{ref}) \sqrt{2\alpha_{max} |\theta(t) - \theta^{ref}|}$$

を逐次与えれば、目標の角度でぴったり停止

- 位置姿勢の制御
 - 位置の最短時間制御
- 方位制御
 - 角度の最短時間制御+走行
- 緊急停止
 - 最大加速度で停止、ロック

詳しい使い方は
<http://openspur.org/>

移動ロボットの軌跡追従制御

- 世界座標系上で描いた軌跡に沿って走行
 - 地図座標系で軌跡を設計



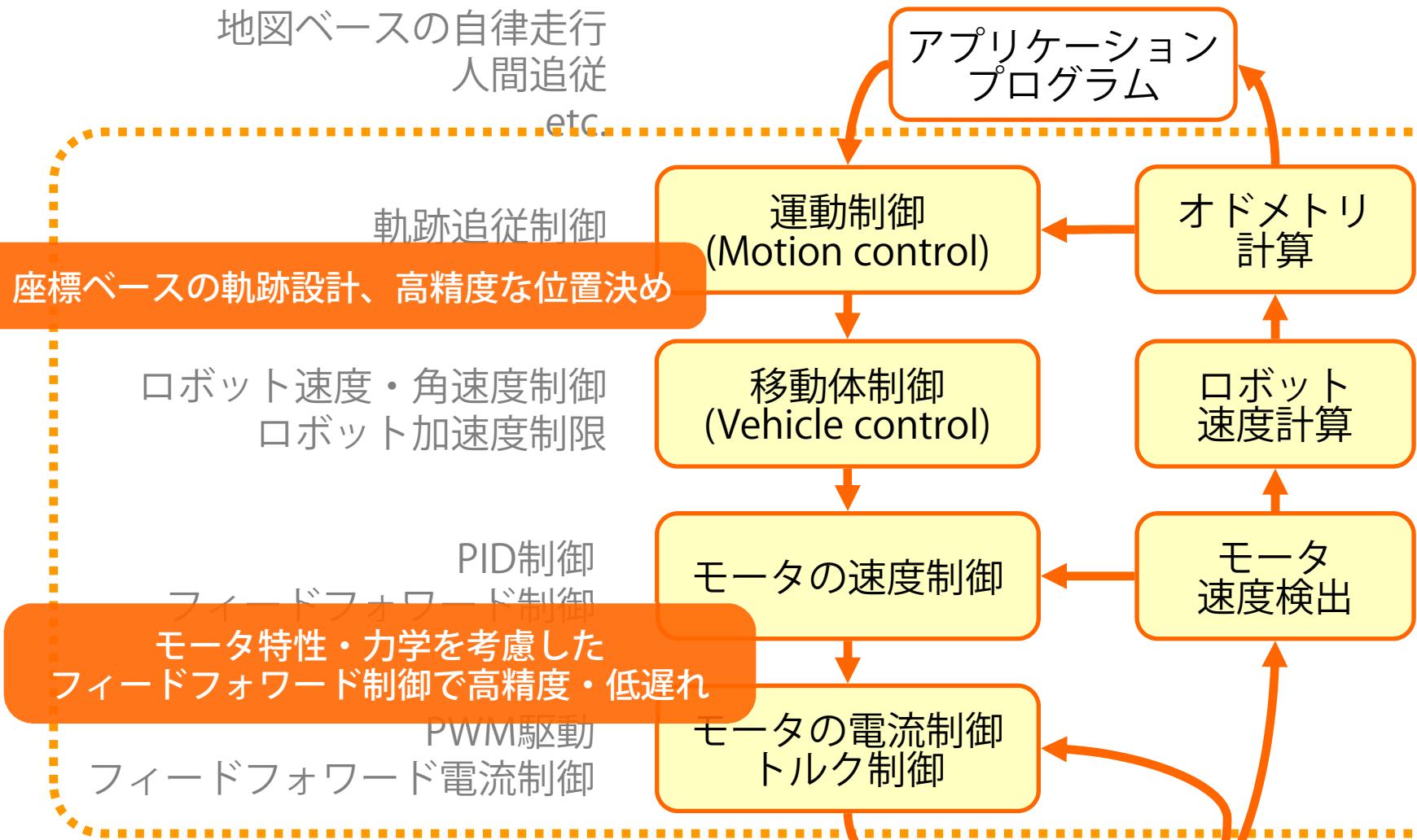
動画：ROSの自己位置推定と組み合わせて
高い再現性で移動制御



ロボ研 テモプログラムコンテスト2011 優勝：エクストリーム・センシング

動画：物を移動させる動作計画

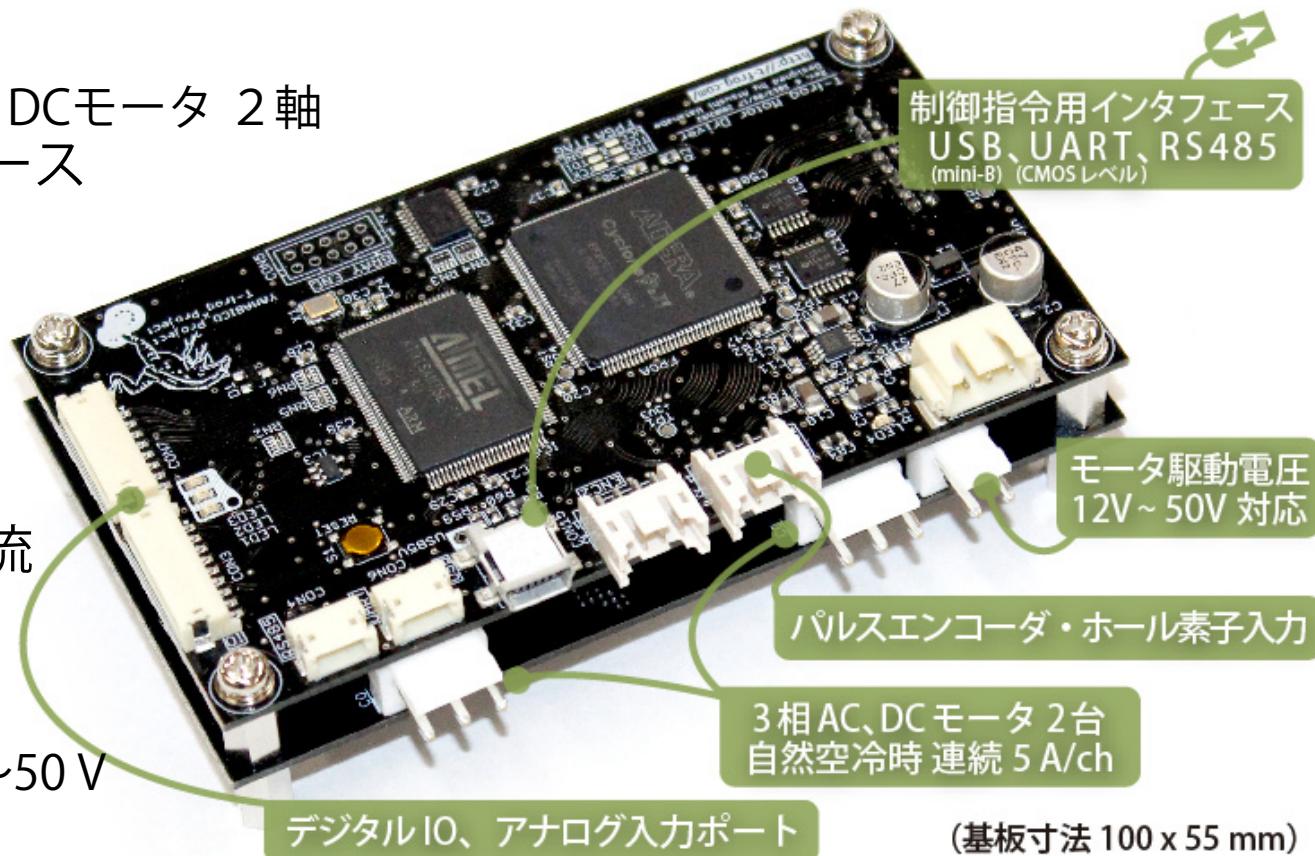
移動ロボットを動かすには



下位の制御をしつかり作ることで
上位のアプリケーションを作りやすく高性能に

本セミナーで使用している T-frogモータドライバ(TF-2MD3-R6)の仕様

- ・対応モータ
3相ブラシレス、DCモータ 2軸
- ・通信インターフェース
USB2.0、ほか



車輪型移動ロボットに適した構成

定価 ¥35,000- ツジ電子株式会社から販売中

27 そのほかの応用

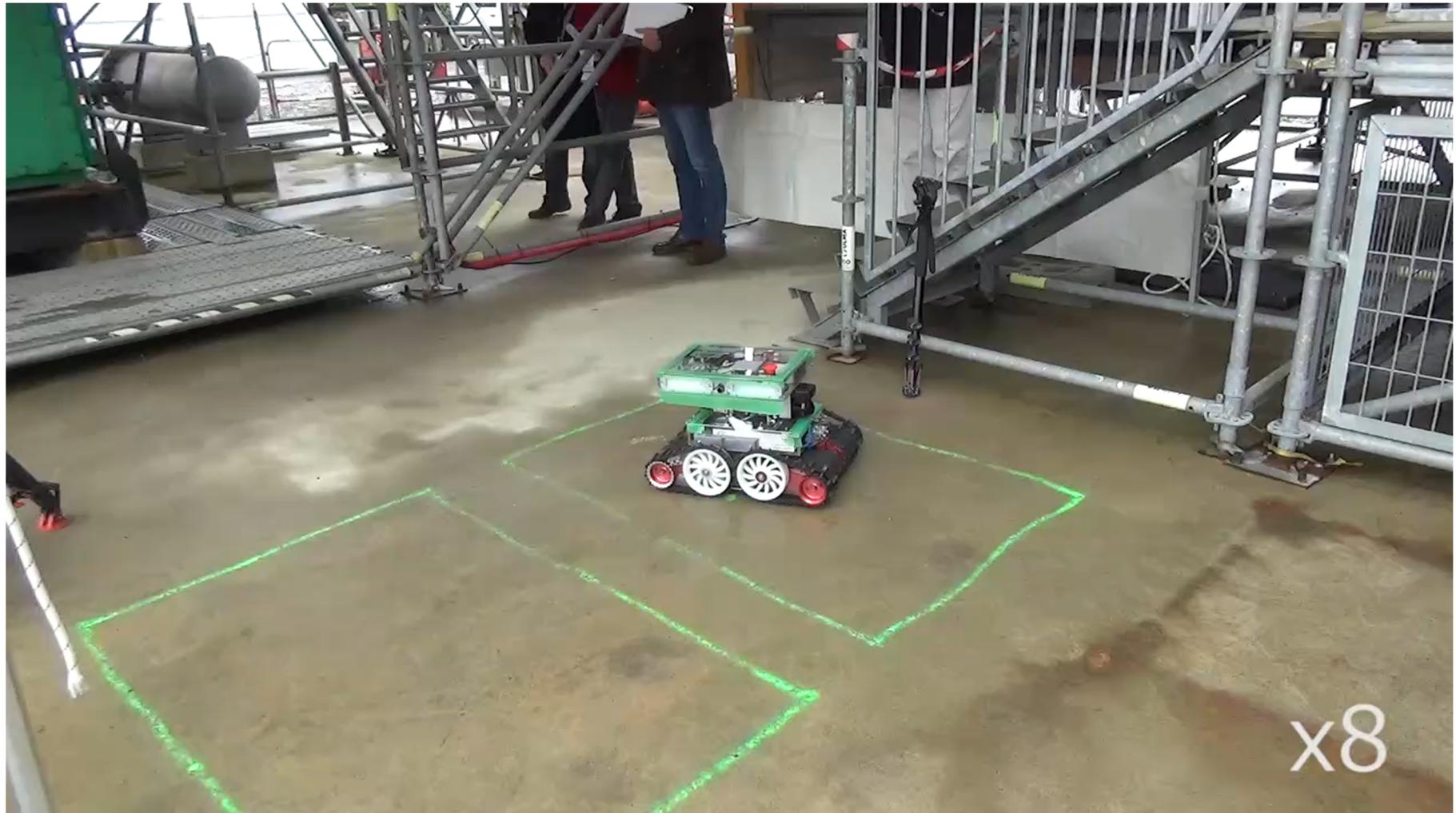
- 再利用性が高く、柔軟な構成・変更可能



工学院大学 羽田研究室でパラボラアンテナの雲台制御に利用した例

そのほかの応用

- 再利用性が高く、柔軟な構成・変更可能



ドライバ3台で6軸制御を行い、ROSと独自のナビゲーションパッケージで自律走行

YP-Spurパラメータの解説と簡易調整方法



<http://openspur.org/~atsushi.w/>
atsushi.w@ieee.org

簡易的な走行制御指令の与え方

- ypspur-coordinator を起動しておく
 - ypspur-interpreter を実行
 - FS> vel v w (速度v、角速度wの速度指令)
 - FS> line x y θ ((x, y)を通る θ 方向の直線追従)
 - FS> stop_line x y θ (lineコマンドの走行で(x, y)地点で停止)
- (FS は「コマンド発行時点のロボット座標系指定」モードを表す)
-
- FS> GL (GL 座標系に切り替え)
 - GL> set_pos x y θ (現在の座標を(x, y, θ)に設定)
- (GL は「グローバル座標系指定」モードを表す)

3.1 パラメータファイル

- モータパラメータ
- 機械的パラメータ
- 運動学(キネマティクス)パラメータ
- 力学パラメータ
- 制御パラメータ



適切に設定すると
ファインな制御が実現できます

モータパラメータ

- MOTOR_PHASE モータの種類
 - 0: DCモータ 3: 3相モータ(ブラシレス)
- MOTOR_R [Ω] 端子間抵抗
 - RCLメータ、テスター等で測定
- MOTOR_VC [rpm/V] 回転数係数
 - 最大無負荷回転数[rpm] / 定格電圧[V]
- MOTOR_TC [Nm/A] トルク係数
 - 回転数係数 K_{rpm} から $K_\tau = \frac{60}{2\pi K_{rpm}}$ で計算

PARAM[0], PARAM[1]
で右左モータ個別に設定

機械的パラメータ、運動学パラメータ

- GEAR
 - ギヤ仕様の値
- RADIUS [m]
 - 実測後に微調整
- TREAD [m]
 - 左右タイヤの接地点間距離を実測後に微調整

ギヤ比

PARAM[0], PARAM[1]
で右左モータ個別に設定

タイヤ半径

トレッド

力学パラメータ

- MASS [kg] ロボット質量
- MOMENT_INERTIA [kgm²] ロボット慣性モーメント
- TIRE_M_INERTIA [kgm²] タイヤ慣性モーメント
 - 計算、実測、推定、調整等で決定
- MOTOR_M_INERTIA [kgm²] モータロータ慣性モーメント
 - モータ仕様の値
- TORQUE_NEWTON [Nm] クーロン摩擦力
- TORQUE_VISCOS [Nm/(rad/s)] 粘性摩擦係数
 - 実測、推定、調整等で決定

35 モータ制御パラメータ 1/2

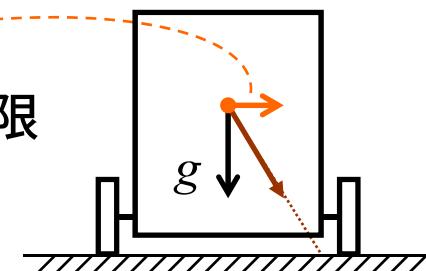
- VOLT [V] 電源電圧
- CYCLE [s] ドライバ速度制御周期
- TORQUE_FINENESS [Nm] トルク計算分解能
 - 0.000001 推奨（超大トルクのロボットの場合は大きく）
- COUNT_REV [cnt/rev] エンコーダ分解能
 - エンコーダパルス数 × 4 (4倍)
- TORQUE_MAX [Nm] モータトルク制限
- TORQUE_LIMIT [Nm] モータトルク出力制限
 - モータが焼けない範囲で目的に合わせて設定
 - MAXはPI制御ぶんの制限、LIMITは最終的な出力制限

36 モータ制御パラメータ 2/2

- GAIN_KP [1/s] 比例制御ゲイン
 - GAIN_KI = 0 の状態で、発振しない範囲で大きく調整
- INTEGRAL_MAX [rev] 積分最大値
 - 積分制御でどの程度の偏差まで吸収するか
 - タイヤの角度換算で与える 0.1程度
- GAIN_KI [1/s²] 積分制御ゲイン
 - 発振しない範囲で大きく調整

37 走行制御パラメータ 1/2

- CONTROL_CYCLE [s] 制御周期
 - 20 [ms] 推奨、振動抑制に10 [ms] 程度まで短くできる
- MAX_VEL [m/s] 最大速度
- MAX_W [rad/s] 最大角速度
- MAX_ACC_VEL [m/s^2] 最大加速度
- MAX_ACC_W [rad/s^2] 最大角加速度
 - 速度・加速度の制限 (コマンドでの設定を制限する)
- MAX_CENTRI_ACC [m/s^2] 遠心加速度制限
 - ZMPが車輪間に収まるように計算



38 走行制御パラメータ 2/2

- L_K1 $[(rad/s^2)/m]$ 軌跡追従制御 距離偏差に対するゲイン
- L_K2 $[(rad/s^2)/rad]$ 軌跡追従制御 角度偏差に対するゲイン
- L_K3 $[(rad/s^2)/(rad/s)]$ 軌跡追従制御 角速度偏差に対するゲイン
- L_C1 $[(m/s)/(rad/s^2)]$ 軌跡追従制御 減速係数
- L_DIST [m] 軌跡追従制御 距離偏差の制限
 - ちょうどよい軌跡を描くように調整
 - $L_C1 = 0, L_DIST = 0.5$ 程度

モータ・力学パラメータ調整の手順

1. 実測、カタログ仕様値、計算値を設定
2. 速度制御ゲイン調整
 1. 速度指令・負荷を与えながら
 $KI = 0$ で KP を発振しない範囲で大きくする
 2. KI を発振しない範囲で大きくする
3. モータパラメータ調整 (詳細は割愛) ↓ 精度が必要な場合のみ
4. ダイナミクスパラメータ調整・推定 (詳細は割愛)
運動時の出力トルクから推定
5. 速度制御ゲイン再調整
ダイナミクスパラメータが大きく変わった場合は再調整

1. 実測、カタログ仕様値、計算値を設定
2. 左右タイヤ径の比率調整
 1. 「line 0 0 0」で直進を指示
 2. 右に曲がる場合、RADIUS[0]を小さく
左に曲がる場合、RADIUS[1]を小さく
 3. 直進するまでくり返し
3. 平均タイヤ径の調整
 1. 「stop_line x^{ref} 0 0」で、 x^{ref} [m] 走行を指示
 2. 距離計などで実際の走行距離 x を測定
 3. $RADIUS[i] = (x/x^{ref}) RADIUS[i]$ に更新
 4. $x = x^{ref}$ に近づくまでくり返し

キネマティクスパラメータ調整の手順(2/2)

4. トレッドの調整

1. 「GL」でグローバル座標系に切り替え
2. 「set_pos 0 0 0」で座標系リセット
3. 「vel 0 1」等で旋回させ、 n 周させる
4. 「spin 0」で0度方向を向かせ、
角度誤差 $\Delta\theta$ [rad] を測定
5. $TREAD = (2\pi n / (2\pi n + \Delta\theta)) TREAD$ に更新
6. $\Delta\theta = 0$ になるまでくり返し

調整完了