

# 下肢障害者の自立生活支援ロボットの実用的制御法の検討

## Practical control methods for robots supporting

## independent living for people with lower limb disabilities

知能機械工学コース

知能ロボティクス研究室 久保 龍馬

### 1. 緒言

近年, 身体障害者の総数が年々増加している. 中でも大きな割合を占める肢体障害者の半数が下肢障害者である. 急速に増えてきた要介護者の介護は介護者に大きな負担がかかる. 本研究では, 下肢障害者の自立生活支援を目的として, 座位で移動することができる自立生活支援ロボットの開発している. 先行研究では, 最適制御法(LQR)を用いてシミュレーションによる検討を行っていた. 本論文では, より実用的制御の開発を目的として, 積分要素を入れる最適制御法(LQI)と比例積分制御法(PI)を用いて, 経路追従制御の結果について比較する. そのため, まず, 状態エラー方程式を構築して, LQRに基づいて最適制御則を導出する. 次に与えられる経路と軌道に対して, 両制御法による制御結果を比較して検討する.

### 2. 自立生活支援ロボットの概要

自立生活支援ロボット<sup>[1]</sup>は, 車輪に三輪のオムニホイールを搭載しており, ジョイスティックの操作などによって, 前後左右や回転といった全方向への移動が可能となっている. また, 一定の速度を与えるとその方向に対して移動を続ける補助モードでは, 立位の維持が困難な歩行機能障害者でも, 歩行訓練機として用いることができる.



Fig.1 Independent Living Support Robot

### 3. 三輪ロボットのモデル

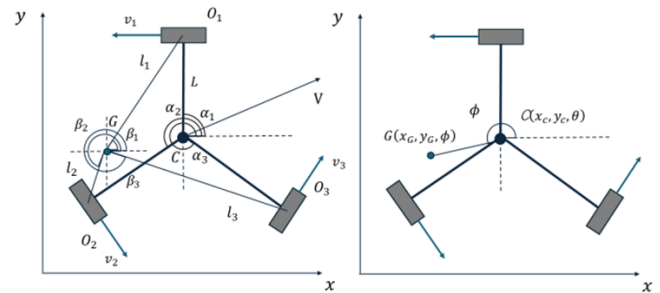


Fig.2 Model of three wheel robot

$C$ : 三輪ロボットの中心位置

$G$ : 三輪ロボットの重心位置

$O_i$ : 各オムニホイールの位置

$V$ : 三輪ロボットの移動速度

$\dot{\theta}$ : 三輪ロボットの角速度

$v_i$ : 各オムニホイールの移動速度

$D$ : 三輪ロボットの中心と重心の距離

$\phi$ :  $x$ 軸と線分 $GO$ との距離

$l_i$ : 線分 $Gom_i$ の長さ

$\alpha_i$ : 線分 $Gom_i$ と $x$ 軸との角度

$\beta_i$ : 線分 $Gom_i$ と $v_i$ 軸との角度

( $i = 1, 2, 3$ )

重心を考慮した状態方程式<sup>[2]</sup>を(1)に示す.

$$\dot{x} = Ax + Bu \quad (1)$$

$$x = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ y_0 \\ \dot{y}_0 \\ \theta \\ \dot{\theta} \end{bmatrix}, \quad u = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ d_1 & d_2 & d_3 \\ 0 & 0 & 0 \\ e_1 & e_2 & e_3 \\ 0 & 0 & 0 \\ f_1 & f_2 & f_3 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$d_i = -\frac{1}{M} \cos(\alpha_i - \beta_i) - \frac{D}{I} l_i \sin \beta_i \sin \phi$$

$$e_i = -\frac{1}{M} \sin(\alpha_i - \beta_i) - \frac{D}{I} l_i \sin \beta_i \cos \phi$$

$$f_i = \frac{1}{I} l_i \sin \beta_i$$

#### 4. LQI 制御法

LQI 制御法<sup>[3]</sup>では、式(1)と目標軌道 $\mathbf{x}_{ref}(t)$ に対し、エラー状態方程式(2)、出力(3)を導出する。

$$\begin{bmatrix} \dot{\tilde{\mathbf{x}}}_e(t) \\ \dot{\tilde{\boldsymbol{\omega}}}(t) \end{bmatrix} = \mathbf{A}_e \tilde{\mathbf{x}}_e(t) + \mathbf{B}_e \tilde{\mathbf{u}}(t) \quad (2)$$

$$\mathbf{e}(t) = \mathbf{C}_e \tilde{\mathbf{x}}_e(t) \quad (3)$$

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix}, \mathbf{B}_e = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \mathbf{C}_e = [-\mathbf{C} \quad \mathbf{0}]$$

$$\mathbf{x}_e(t) = \mathbf{x}_{ref}(t) - \mathbf{x}(t)$$

$$\boldsymbol{\omega}(t) = \int_0^t (\mathbf{C}\mathbf{x}_{ref}(t) - \mathbf{C}\mathbf{x}(t)) dt$$

$$\tilde{\mathbf{x}}_e(t) = [\mathbf{x}_e(t)^T \quad \boldsymbol{\omega}(t)^T]^T$$

$$\tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_\infty(t)$$

評価関数を式(4)で決められるものとして、制御入力は式(5)になる。ただし、係数 $\mathbf{K}_e$ はリカッチ方程式(7)を解くことにより得られる。 $\mathbf{Q}_{11} = \mathbf{Q}_{11}^T, \mathbf{Q}_{22} = \mathbf{Q}_{22}^T, \mathbf{Q} = \mathbf{Q}^T$ は半正定行列、 $\mathbf{R} = \mathbf{R}^T$ は正定行列とする。

$$\begin{aligned} J &= \int_0^\infty (\mathbf{x}_e(t)^T \mathbf{Q}_{11} \mathbf{x}_e(t) + \boldsymbol{\omega}(t)^T \mathbf{Q}_{22} \boldsymbol{\omega}(t) + \tilde{\mathbf{u}}(t)^T \mathbf{R} \tilde{\mathbf{u}}(t)) dt \\ &= \int_0^\infty (\tilde{\mathbf{x}}_e(t)^T \mathbf{Q} \tilde{\mathbf{x}}_e(t) + \tilde{\mathbf{u}}(t)^T \mathbf{R} \tilde{\mathbf{u}}(t)) dt \end{aligned} \quad (4)$$

$$\tilde{\mathbf{u}}(t) = \mathbf{K}_e \tilde{\mathbf{x}}_e(t) \quad (5)$$

$$\mathbf{K}_e = [\mathbf{K} \quad \mathbf{G}] = -\mathbf{R}^{-1} \mathbf{B}_e^T \mathbf{P} = -\mathbf{R}^{-1} \mathbf{B}_e^T \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21}^T & \mathbf{P}_{22} \end{bmatrix}$$

$$= [-\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}_{11} \quad -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}_{12}]$$

$$\mathbf{P} \mathbf{A}_e + \mathbf{A}_e^T \mathbf{P} - \mathbf{P} \mathbf{B}_e \mathbf{R}^{-1} + \mathbf{Q} = \mathbf{0}$$

$$\mathbf{Q} = \text{diag}(4, 0.4, 4, 0.4, 5, 0.8, 0.6, 0.6)$$

$$\mathbf{R} = \text{diag}(2.0, 2.0, 2.0)$$

#### 5. PI 制御法

PI 制御<sup>[4]</sup>とは比例制御と積分制御を組み合わせた制御法である。ある時刻  $t$  における目標値を $\mathbf{X}_{ref}(t)$ 、制御量を $\mathbf{X}(t)$ とすると偏差 $\mathbf{E}(t)$ は式(5)となる。

$$\mathbf{E}(t) = \mathbf{X}_{ref}(t) - \mathbf{X}(t) \quad (6)$$

また、操作量 $\mathbf{U}(t)$ は、比例定数 $K_P K_I$ を用いて

$$\mathbf{U}(t) = K_P \mathbf{E}(t) + K_I \int_0^t \mathbf{E}(t) dt \quad (7)$$

#### 6. 実機検証

PI 制御法と LQI 制御法を用いた場合について、実機での検証により評価をした。x 軸方向に 1[m] 走行させた時の x 軸誤差, y 軸誤差を図 3, 図 4 に示す。

$\mathbf{K}_P = \text{diag}(K_{Px} \quad K_{Py} \quad K_{P\theta})$ ,  $\mathbf{K}_I = \text{diag}(K_{Ix} \quad K_{Iy} \quad K_{I\theta})$  とする。

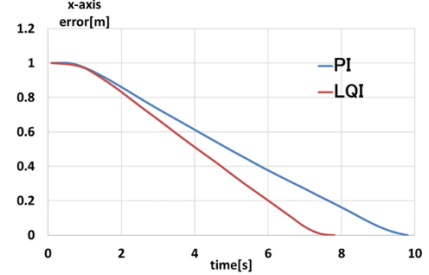


Fig.3 x-axis error

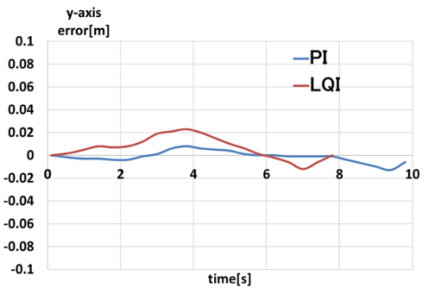


Fig.4 y-axis error

図 3 から、どちらも単調に誤差を減少させているが、時間は LQI 制御の方が短く、効率的に移動していることがわかる。

Fig4 からは、両制御法とも誤差は 0.02m 程度に抑えられており、大きな逸脱は見られなかったが、LQI 制御法の方がやや大きくなる傾向が確認された。

#### 7. 結言

本研究では、LQI 制御法と PI 制御法を自立生活支援ロボットに用いた場合の経路追従精度について検討した。x 軸方向の走行においては、LQI 制御の方が効率的な走行を可能にするが、前後誤差が大きくなることがわかった。

LQI 制御法は制御対象のモデル化が必須であり、計算過程が複雑であるが、PI 制御法はモデルに依存せず調整が容易である。よって実用性の観点では PI 制御が有利であると考えられる。

#### 参考文献

- [1] 王 碩玉, 石田 健司, 藤江 正克, “室内移動補助も可能な新型歩行訓練機”, 日本機械学会 2009 年度年次大会 講演論文集, pp283-284, 2009
- [2] 渡辺 悠人, 王 碩玉, 三浦 直樹, 姜 銀来, 石田 健司, 藤江 正克, “重心ずれを考慮した座位歩行訓練機の走行制御” 自動制御連合講演会論文集, 2010, p154
- [3] 久保 龍馬, “歩行困難者を対象とした自立生活支援ロボットの運動制御”, 日本機械学会 中四国学生会 第 54 回卒業研究発表講演会, 2024
- [4] 志水 清孝, PID 制御-現代制御論の視点から-, オーム社, 2020, p1-14, 73-96