

1

環境センサーを用いたロボットの無線遠隔制御のアーキテクチャの検討

大下 裕一† 安田 真也‡ 熊谷 太一‡
 吉田 裕志‡ 里田 浩三‡ 村田 正幸†

†大阪大学 大学院情報科学研究科
 ‡NEC システムプラットフォーム研究所

2


ロボットの無線遠隔制御

- ・ロボットを無線ネットワーク経由で制御

↓

- ・ロボットの稼働範囲が確保可能
 - ・様々な用途:
 - ・荷物の搬送
 - ・遠隔からの建物等の検査 等


無線ネットワーク経由でのロボットの制御への期待



3

環境センサーを用いたロボットの制御

- ・ロボットの動作環境内に設置されたセンサーでロボットの状態を把握
- ・把握したロボットの状態をもとに、制御



ロボットがセンサーを持つ必要がなくなり、ロボットをより安価に

4

研究目標

目標:
 環境センサーによりロボットの状況を把握する環境下で、ロボットを適切に制御するための機能配置を明らかにする

考慮が必要な点

- ・ロボットとコントローラ間の通信遅延
- ・ロボットの制御の精度

5

研究対象

ロボット: 対向二輪ロボット

- 2つの車輪の回転速度を決めることにより、ロボットを回転・前進・後退させる

$$x(t+1) = x(t) + \frac{v_l(t)+v_r(t)}{2} \cos(\theta(t) + \frac{v_r(t)-v_l(t)}{2W})$$

$$y(t+1) = y(t) + \frac{v_l(t)+v_r(t)}{2} \sin(\theta(t) + \frac{v_r(t)-v_l(t)}{2W})$$

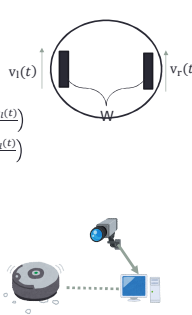
$$\theta(t+1) = \theta(t) + \frac{1}{W}(v_r(t) - v_l(t))$$

動作環境

- ・カメラによりロボットの位置・向きを観測
- ・カメラ・コントローラは有線LANで接続
- ・コントローラ・ロボット間は無線LANで接続

タスク

- ・ロボットを意図した軌跡



6

研究対象のロボットにおける誤差

各車輪の移動距離は、路面によってスリップするといった誤差が入る

実際の車輪の速度は、

- ・ $v_l(t) = v'_l(t) + a_l(v'_l(t))$
- ・ $v_r(t) = v'_r(t) + a_r(v'_r(t))$

$v'_l(t), v'_r(t)$: 制御時に入力された車輪の速度

7

ロボットの制御手順

1. ロボットの状態の把握:
 - ・ センサーから得られた情報をもとに、ロボットの座標・角度 (X, Y, θ) を把握する
2. ロボットの移動誤差モデルの更新:
 - ・ ロボットの車輪の移動距離に入る誤差のモデルを更新する
3. 予測:
 - ・ 今から投入する制御コマンドが実行される時刻におけるロボットの状態を予測する
4. ロボットの目標状態の計算:
 - ・ 制御コマンドを計算する際に考慮する、ロボットが次に目指す座標について計算する
5. ロボットに投入すべき車輪の速度の計算:
 - ・ 車輪の移動距離に入る誤差モデルを考慮した上で、与えられた目標状態に近づくために、ロボットに投入する各車輪の速度 $v'_1(t)$ 、 $v'_r(t)$ を計算する

8

ロボットの誤差モデルの更新: ロボットの誤差の把握

誤差 $a_1(v'_1(t))$ 、 $a_r(v'_r(t))$ の計算

- ・ 以前の制御時のロボットの位置・向きの状態情報を取得したのち、新たに取得されたロボットの位置・向きに至るまでに混入した誤差

連立方程式により取得

- ・ $r(t1, t2) = \frac{1}{W} (v'_1(t) + a_1(v'_1(t)) + v'_r(t) + a_r(v'_r(t))) (t2 - t1)$
- ・ $\theta(t1, t2) = \frac{1}{W} (v'_r(t) + a_r(v'_r(t)) - v'_1(t) - a_1(v'_1(t))) (t2 - t1)$
- ・ $r'(t1, t2) = \frac{1}{W} (v'_1(t) + v'_r(t)) (t2 - t1)$
- ・ $\theta'(t1, t2) = \frac{1}{W} (v'_r(t) - v'_1(t)) (t2 - t1)$

ただし、

- $r(t1, t2)$: 時刻 $t1$ から $t2$ までにロボットが移動した軌跡の長さ
- $\theta(t1, t2)$: 時刻 $t1$ から $t2$ までのロボットの角度の変化
- $r'(t1, t2)$: 時刻 $t1$ から $t2$ までに車輪の移動距離に誤差が入らなかった場合にロボットが移動する軌跡の長さ
- $\theta'(t1, t2)$: 時刻 $t1$ から $t2$ までに車輪の移動距離の誤差が入らなかった場合に生じるロボットの角度の変化

9

ロボットの誤差モデルの更新: ロボットの誤差モデルの更新

誤差モデル:
 v' を複数の bin に分け、それぞれについてモデル化

$$a(v') = \begin{cases} a_1(v') & (v' \in V_1) \\ \vdots & \vdots \\ a_n(v') & (v' \in V_n) \end{cases}$$

誤差モデルの更新:

- ・ 投入した車輪速度に対応する bin を見つける
- ・ 対応する bin の誤差の分布を更新

10

ロボットに投入すべき車輪の速度の計算: 必要な車輪速度の計算

目標点と現在のロボットの向きが $\theta_{threshold}$ 以上ずれた場合

- ・ その場回転

$$(v_1(t), v_r(t)) = (v, -v)$$

ただし、

$$v = \min \left(\frac{W}{2r} \text{Dif}(\theta^{target}, \theta(t)), \alpha \right) \times \text{sign}(\text{Dif}(\theta^{target}, \theta(t)))$$

$\text{Dif}(\theta^{target}, \theta(t))$: ロボットの向きと当該ロボットから目標地点への向きの差
 $\text{sign}(\text{Dif}(\theta^{target}, \theta(t)))$: $\text{Dif}(\theta^{target}, \theta(t))$ が正なら 1、負なら -1

それ以外の場合

- ・ 目標点に向かうように制御

$$(v_1(t), v_r(t)) = (K + AKW, K - AKW)$$

ただし、

$$A = \frac{\text{sign}(e_x)e_y}{e_x^2}$$

$$K = \frac{\text{sign}(e_x)\alpha}{1 + |A|}$$

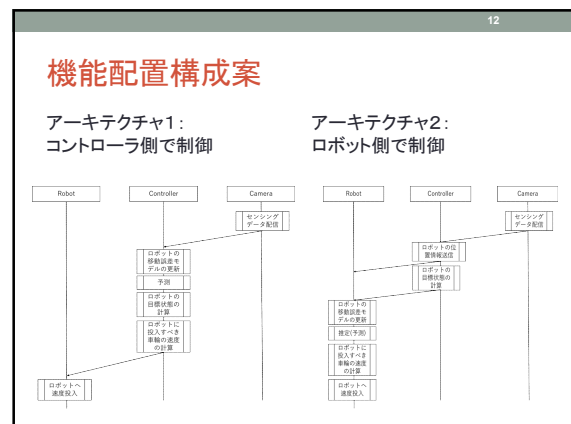
$$\begin{pmatrix} e_x \\ e_y \end{pmatrix} = \begin{pmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{pmatrix} \begin{pmatrix} x^{target} - x(t) \\ y^{target} - y(t) \end{pmatrix}$$

11

ロボットに投入すべき車輪の速度の計算 誤差の考慮

- ・ 以下の方程式の解を実際のロボットの車輪の速度とする

$$v_1(t) = v'_1(t) + a_1(v'_1(t))$$

$$v_r(t) = v'_r(t) + a_r(v'_r(t))$$


13

アーキテクチャ構成の違い

ロボットの状態の予測
 ・コントローラ側で制御
 ・コントローラ・ロボット間の遅延分先の状況を予測

アーキテクチャ1:
 コントローラ側で制御

Robot Controller Camera

センシングデータ取得

ロボットの移動速度モデルの更新
 予測
 ロボットの目標位置の計算
 ロボットに投入する目標位置の送信

ロボットの速度投入

14

アーキテクチャ構成の違い

ロボットの状態の予測
 ・コントローラ側で制御
 ・コントローラ・ロボット間の遅延分先の状況を予測
 (遅延がどの程度発生するのかわかっても予測する必要)

アーキテクチャ2:
 ロボット側で制御
 ・現在のロボットの状況を推定

Robot Controller Camera

センシングデータ取得

ロボットの目標位置の送信
 ロボットの目標位置の計算
 ロボットの速度投入

ロボットの移動速度モデルの更新
 推定予測
 ロボットに投入する目標位置の送信

ロボットの速度投入

15

シミュレーション評価 評価環境

・ロボット
 ・車輪間の距離: 14mm
 ・移動速度に関するパラメータ α : 271mm/s

・ロボットの移動誤差

投入された移動速度	予測値の誤差 (1秒間隔)		実際の移動速度の誤差 (1秒間隔)	
	平均	標準偏差	平均	標準偏差
400 mm/s 未満	-40.2mm/s	2.72 mm/s	-45.7 mm/s	5.71 mm/s
400 mm/s 以上	-123 mm/s	19.3 mm/s	-146 mm/s	33.6 mm/s

・コントローラ・ロボット間の通信周期: 10ms
 ・遅延
 ・ロボット・コントローラ間の遅延: 実測値
 ・カメラ・コントローラ間の遅延: 133ms

・目標軌跡

16

評価結果

Actual Path Target

コントローラ側で制御

ロボット側で制御

17

評価結果

・ロボット側で制御した方が目標軌跡からのズレは少ない
 ・コントローラ側では、将来の遅延についても予測した上で制御することが必要
 (予想よりも遅延が大きくなった場合に対応できない)

CDF

Controlled by robot
 Controlled by controller (25ms)
 Controlled by controller (50ms)
 Controlled by controller (100ms)

Error (mm)

18

ロボットからの情報送出手法の最適化

・ロボットからの情報送出手法の最適化
 ・ロボットからの情報送出手法の最適化
 →ネットワークが輻輳し、遅延が増大
 ・ロボットからの情報送出手法の最適化
 →ロボット側で古い情報を利用して制御

制御周期と通信遅延の和を最小化する制御コマンドの送信レート最適化の手法が提案^[1]

・レート最適化手法を適用した場合と、固定レートで情報を送出した場合についても比較

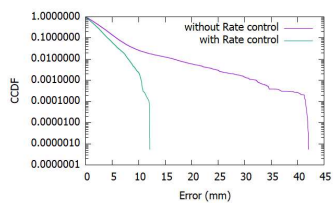
[1] 安田真也, 吉田裕志. "遠隔制御の応答性を改善するための制御周期の動的調整手法" 電子情報通信学会技術研究報告(IN2017-105) pp. 93-98, Mar. 2018.

19

レート制御の有無の比較結果

固定レートで制御した場合よりも、情報送出レートを調整した方が制御の精度がよい

- ・ 制御周期と通信遅延の和を最小化することにより、レート調整により、より新しい情報を制御に利用可能



20

まとめ

- ・ 環境センサーを用いて、ロボットを制御する際の機能配置の構成を検討
- ・ ロボット側で制御を行った方がより高い精度での制御を実現可能
- ・ ロボットからの情報送出レートの制御も効果的