一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

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 〒565-0871 大阪府吹田市山田丘 1-5
‡NEC システムプラットフォーム研究所 〒211-8666 神奈川県川崎市中原区下沼部 1753
E-mail: †{y-ohsita, murata}@ist.osaka-u.ac.jp,
‡{s-yasuda@ce., t-kumagai@bq., h-yoshida@jh., k-satoda@cb.}jp.nec.com

あらまし 本稿では、ロボットが動作する環境におけるロボットの状態を監視するセンサーが配置されている状況下において、無線ネットワーク経由でロボットを制御するためのアーキテクチャの検討を行う。本稿で想定する状況では、環境センサーと有線接続されたコントローラでは、短い遅延でロボットの状況を観測できるものの、無線ネットワーク経由で行われるコントローラ・ロボット間の通信は、大きな遅延が生じ、また、遅延の変動も大きい。本稿では、このような環境下において、ロボットを制御する際に必要となる状況の推定や動作の決定といったタスクをコントローラ・ロボットのいずれで行うのかを検討する。そして、シミュレーションにより、環境センサーの値をもとにコントローラでロボットの状態を把握、ロボットに送出・ロボット側で、コントローラから受け取ったロボットの状態とその状態が観測された以後の自身の行動をもとに、現在のロボットの状態を推定し、行動を決定するという手法が、目標とする行動との誤差を抑えながら制御するのに有効であることを示す。

キーワード 無線ネットワーク，ロボット，制御

Architecture for wireless remote control of a robot using environment sensors

Yuichi OHSITA† Shinya YASUDA‡ Taichi KUMAGAI‡ Hiroshi YOSHIDA‡
Kozo SATODA‡ and Masayuki MURATA†

†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
‡System Platform Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara, Kawasaki, Kanagawa, 211-8666 Japan
E-mail: †{y-ohsita, murata}@ist.osaka-u.ac.jp,
‡{s-yasuda@ce., t-kumagai@bq., h-yoshida@jh., k-satoda@cb.}jp.nec.com

Abstract In this paper, we discuss architectures for wireless remote control of a robot in the case that environmental sensors are deployed. In this case, a controller that is connected to the sensors via a wired network can monitor the robot with a small delay. But, the delay between the controller and the robot is large and changes in time. In this paper, we discuss the roles of the controller and the robots. Then, we demonstrate that the architecture where the controller sends the monitored information to the robot and the robot estimates the current status and decides its behavior based on the estimated status is preferable.

Keywords Wireless network, Robot, Control

1. はじめに

近年、ネットワークに接続可能なロボットが様々な場面で用いられるようになってきた。それらの多くでは、ロボットの移動範囲を確保する等の目的のため、無線ネットワークを用いて制御される。そのため、無線ネットワーク経由でロボットを制御する研究が広く進められてきた[1]。

従来、多くのロボットでは、自身がセンサーを持ち、そのセンサーの情報をもとに、ロボットの制御を行ってきた。しかしながら、ロボットの状況を把握するセンサーは必ずしもロボットが持つ必要はない。例えば、ロボットの作業領域内を撮影するカメラ等の環境センサーを用いロボットの状態を監視し、制御に用いることも考えられる。これにより、各ロボットにセンサー

を搭載する必要はなくなり、特に多数のロボットの制御が必要な場合、各ロボットを安価に構成でき、低コストでのロボットの導入が可能となる。

本稿では、このような環境センサーが存在し、その環境センサーの値をもとに、無線ネットワーク経由でロボットを制御する環境下に合わせた、アーキテクチャの検討を行う。

2. 想定するロボット

2.1. 移動ロボットの概要

本稿では、対向二輪移動ロボットを想定する。対向二輪移動ロボットでは、2 つの車輪の回転速度を決めることにより、ロボットを回転させたり、ロボットを前進・後退させたりすることができる。

対向二輪移動ロボットにおいては、右車輪の速度$\mathrm{v_l}(t)$、左車輪の速度$\mathrm{v_r}(t)$をコントローラから設定が可能である。その設定された速度にもとづき、対向二輪ロボットは移動を行う。その結果、対向二輪ロボットは、ロボットの中心点の位置$(\mathrm{x(t)}, \mathrm{y(t)})$、角度 θ (t)を変化させる。$\mathrm{v_l(t)}$、$\mathrm{v_r}(t)$を 1 単位時間あたりに、左・右の車輪の回転により進む距離とすると、1 単位時間経過した後の位置は、以下のように変化する。

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

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

また、角度は以下のように変化する。

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

ただし、Wは車輪間の距離である。そのため、$\mathrm{v_l}(t)$、$\mathrm{v_r}(t)$を定めることにより、任意の位置にロボットを移動させることができる。

ただし、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。そのため、制御の際に与えた各車輪の回転により進む距離を$\mathrm{v'_l(t)}$、$\mathrm{v'_r}(t)$とすると、実際の車輪の回転により進む距離は、

$$\mathrm{v_l(t)} = \mathrm{v'_l(t)} + \mathrm{a_l(v'_l(t))}$$
$$\mathrm{v_r(t)} = \mathrm{v'_r(t)} + \mathrm{a_r(v'_r(t))}$$

となる。本モデルは、各車輪の移動の距離に入る誤差は、設定された車輪の速度に依存するとし、車輪の移動距離に入る誤差は、速度の関数$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r(v'_r(t))}$によって決まるものとする。

2.2. ロボット制御環境の概要

本稿では、制御環境下に 1 台のロボットと、当該ロボットを無線ネットワーク経由で制御するコントローラ、ロボットが稼働する環境をセンシングするセンサー（カメラ等）が配置されているものとする。センサーはコントローラと有線接続されているものとし、センサーにより、正確に現在のロボットの位置が把握できるものとする。そして、制御を行う際には、センサーから把握された情報をもとに、コントローラやロボットにおいて、ロボットの左右の車輪の回転速度を決定し、移動を行うことを繰り返すことにより、移動ロボットを意図した軌跡にあわせて動作させる。本環境では､移動ロボット自身にセンサーを配備しなくても、ロボットの状態を観測し、それをもとにロボットを制御することが可能となる。

以降、本稿では、センサー・コントローラ間の遅延の変動は十分に小さいのに対して、無線ネットワーク経由で通信を行う、コントローラ・ロボット間の遅延は時間や制御パケットの送出レートによって大きく変動するものとする。

3. 移動ロボットの制御とアーキテクチャ

3.1. 制御に必要な手順

ロボットの制御を行う際に必要な機能を以下に示す。

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、ロボットの状態を把握する
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$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r(v'_r(t))}$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状態について計算する
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$\mathrm{v'_l(t)}$、$\mathrm{v'_r}(t)$を計算する

本稿では、上記の 5 つの機能のうち、コントローラに配置すべき機能、ロボット側に配置すべき機能を議論する。以降、上記の機能をすべてコントローラ側に配備した場合（アーキテクチャ 1）と、可能な限りロボット側に配備した場合について、各機能の動作のタイミングと内容について述べる。

3.2. アーキテクチャ1： コントローラ側で制御

コントローラ側で制御を行う場合、コントローラはセンサーから得られた情報をもとに、ロボットの状態の把握から、ロボットに投入すべき車輪の速度の計算に至るまでのすべての機能を持ち、それらを実行した結果得られる車輪の速度を、無線ネットワークを通じてロボットに送出する。その際に、当該速度での車輪の回転を始める時刻を左右の車輪の速度に加えてロボットに送出する。これにより、送出する制御の実行時

刻を十分先の時刻として設定することができれば、無線ネットワークの遅延の変動があったとしても、変動の影響を受けず、意図した時間にロボットの制御を行うことができる。コントローラ側で制御を行う際の各手順の流れを図 1 に示す。

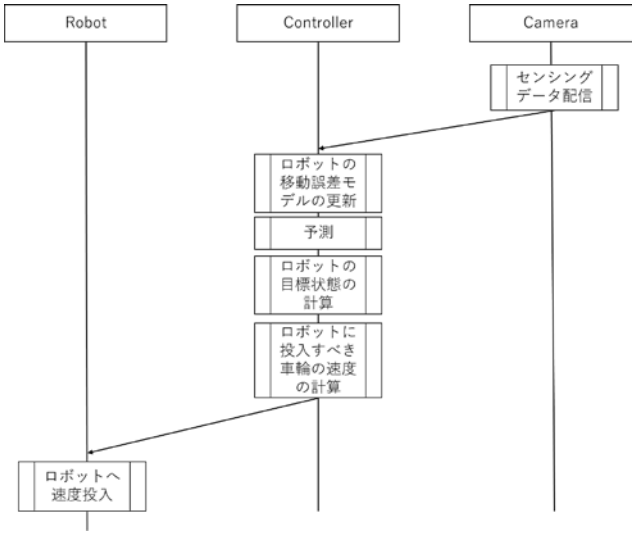

図 1　コントローラ側で制御を行う際の流れ

ロボットの状態の把握：
カメラ等のセンシング情報をもとに、当該センシングが行われた際のロボットの X 座標、Y 座標、角度 θ を得る。

ロボットの移動誤差モデルの更新：
移動誤差モデルを更新するには、まず、前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$\mathrm{a_l}(\mathrm{v'_l(t)})$、$\mathrm{a_r}(\mathrm{v'_r(t)})$を計算することが必要となる。これは、以下の連立方程式を解くことにより得ることができる。

$$\mathrm{r(t1,t2)} = \frac{1}{2}\left(\mathrm{v'_l(t)} + \mathrm{a_l}\left(\mathrm{v'_l(t)}\right) + \mathrm{v'_r(t)} + \mathrm{a_r}\left(\mathrm{v'_r(t)}\right)\right)(t2 - t1)$$

$$\mathrm{\theta(t1,t2)} = \frac{1}{\mathrm{W}}\left(\mathrm{v'_r(t)} + \mathrm{a_r}\left(\mathrm{v'_r(t)}\right) - \mathrm{v'_l(t)} - \mathrm{a_l}\left(\mathrm{v'_l(t)}\right)\right)(t2 - t1)$$

$$\mathrm{r'(t1,t2)} = \frac{1}{2}\left(\mathrm{v'_l(t)} + \mathrm{v'_r(t)}\right)(t2 - t1)$$

$$\mathrm{\theta'(t1,t2)} = \frac{1}{\mathrm{W}}\left(\mathrm{v'_r(t)} - \mathrm{v'_l(t)}\right)(t2 - t1)$$

ただし、$\mathrm{r(t1,t2)}$は時刻t1からt2までにロボットが移動した軌跡の長さ、$\mathrm{\theta(t1,t2)}$は時刻t1からt2までのロボットの角度の変化、$\mathrm{r'(t1,t2)}$は時刻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、$\mathrm{\theta'(t1,t2)}$は時刻t1からt2までに車輪の移動距離の誤差が入らない場合に生じるロボットの角度の変化を示す。

また、ロボットの軌跡の長さについては、ロボットの移動の始点・終点を結ぶ直線距離、すなわち、

$$\mathrm{r(t1,t2)} = \sqrt{\left(x(t1) - x(t2)\right)^2 + \left(y(t1) - y(t2)\right)^2}$$

と近似することができるものとする。$\mathrm{r'(t1,t2)}$についても、同様に近似できるものとする。この時、$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r}\left(\mathrm{v'_r(t)}\right)$は以下のように解ける。

$$\mathrm{a_l(v'_l(t))} = \frac{1}{2(\mathrm{t2} - \mathrm{t1})}\left(2(r'(t1,t2) - r(t1,t2)) + w\left(\theta'^{(t1,t2)} - \theta(t1,t2)\right)\right)$$

$$\mathrm{a_r(v'_r(t))} = \frac{1}{2(\mathrm{t2} - \mathrm{t1})}\left(2(r'(t1,t2) - r(t1,t2)) - w\left(\theta'^{(t1,t2)} - \theta(t1,t2)\right)\right)$$

$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r}\left(\mathrm{v'_r(t)}\right)$を得たのちに、$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r}\left(\mathrm{v'_r(t)}\right)$のモデルを更新する。本稿では、$\mathrm{a(v')}$は、複数の bin に分け、それぞれについてモデル化する。すなわち、

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

とモデル化するものとする。また、ここでは、各 bin に対応するモデルとして、正規分布を用いる。すなわち、$\mathrm{a_l(v'_l(t))}$、$\mathrm{a_r}\left(\mathrm{v'_r(t)}\right)$を得たのちに、$\mathrm{a_l}(v'_l)$の$v'_l$に対応する bin の平均、分散を更新、$\mathrm{a_r}(v'_r)$の$v'_r$に対応する bin の平均、分散を更新する。

予測：
コントローラが送出する車輪速度がロボットにおいて適用される時刻におけるロボットの位置、角度を、その間に投入されるロボットの各車輪の速度の情報と、ロボットの誤差モデルをもとに予測をする。

ロボットの目標状態の計算：
本稿では、到達すべきロボットの座標の系列を事前に与えられているものとし、現時点で未到達なロボットの座標のうち、もっとも順序が若い座標を次の目標地点とする。

ただし、到達、未到達の判断は、予測されたロボットの位置を基準に行うのではなく、センシングされたロボットの位置状態をもとに、目標地点と得られた位置情報との差が閾値以内になった場合に、到達したものとみなすこととする。

ロボットに投入する車輪速度の計算：
本制御においては、ロボットは目標地点に到達するように、左右の車輪の速度を決める。ただし、実際にロボットが左右の車輪の回転で移動する距離は、スリップ等の誤差が入るため、制御入力として与えた値とは異なる。そこで、本制御では、まず、左右の車輪で、本来移動すべき速度を求め、コントローラが把握している誤差モデルをもとに、それに補正を加えることにより、意図した移動ができるような制御入力を計算し、ロボットに送出する。

本制御においては、まず、制御コマンドが投入され

る時点でのロボットの向きと、当該ロボットの位置から現在の目的地へ向きを調べる。その向きが$\theta^{\mathrm{threshold}}$以上ことなっていれば、その場で回転を行う。回転を行う際には、左右の車輪を逆向きに等速度で回転させる。ただし、この回転は、次の制御コマンドが投入されるまで継続されるため、次の制御コマンド投入時に回転しすぎた結果、目標地点との向きのずれが生じることを防ぐ必要がある。そこで、制御コマンドを送出する周期Tを定め、以下のように速度vを計算した上で、その値をもとに車輪の回転速度を定める

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

ただし、αは速度の最大を表す定数であり、$Dif\left(\theta^{target}, \theta(t)\right)$は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、$\mathrm{sign}\left(Dif\left(\theta^{target}, \theta(t)\right)\right)$は$Dif\left(\theta^{target}, \theta(t)\right)$が正なら1、負なら-1となる値である。そして、各車輪の速度は

$$(v_{\mathrm{l}}(\mathrm{t}), \mathrm{v_r}(t)) = (v, -v)$$

と定める。

目標地点との向きの差が$\theta^{\mathrm{threshold}}$以内であれば、当該目標地点に向かうような左右の車輪の速度を計算する。本計算は、文献[1]と同様の計算を行う。具体的には、以下のように$v_{\mathrm{l}}(\mathrm{t}), \mathrm{v_r}(t)$を定める。

$$\left(v_{\mathrm{l}}(\mathrm{t}), \mathrm{v_r}(t)\right) = (K + AKW, K - AKW)$$

ただし、

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

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

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

である。本制御においては、パラメータαの値を変えることにより、ロボットの移動の速度を変えることが可能である。

本制御においては、上記のように、ロボットの各車輪で移動すべき速度が定まったのちに、実際にロボットに投入すべき左右の車輪の回転速度$v'_{\mathrm{l}}(\mathrm{t}), \mathrm{v'_r}(t)$を定める。実際の左右の車輪の回転による移動距離が$v_{\mathrm{l}}(\mathrm{t}), \mathrm{v_r}(t)$となるようにするためには、以下の方程式の解となる$v'_{\mathrm{l}}(\mathrm{t}), \mathrm{v'_r}(t)$をロボットに投入する制御コマンドとして与えればよい。

$$\mathrm{v_l}(\mathrm{t}) = \mathrm{v'_l}(\mathrm{t}) + \mathrm{a_l}(\mathrm{v'_l}(\mathrm{t}))$$
$$\mathrm{v_r}(\mathrm{t}) = \mathrm{v'_r}(\mathrm{t}) + \mathrm{a_r}(\mathrm{v'_r}(\mathrm{t}))$$

ただし、$\mathrm{a_l}(\mathrm{v'_l}(\mathrm{t}))$は速度により異なる誤差で、コントローラ側で計算をした誤差モデルに基づく関数である。本稿では、$\mathrm{a_l}(\mathrm{v'_l}(\mathrm{t}))$の誤差モデルは、速度ごとに bin にわけ、各 bin の中で正規分布によりモデル化をしている。そのような誤差モデルでは、上記の式の解を解析的に求めるのは困難である。ここでは、補正をしなければならない誤差が十分に小さく、$\mathrm{a_l}(\mathrm{v'_l}(\mathrm{t})) = \mathrm{a_l}(v_{\mathrm{l}}(\mathrm{t}))$と想定できるとし、

$$\mathrm{v'_l}(\mathrm{t}) = v_{\mathrm{l}}(\mathrm{t}) - \mathrm{a_r^{ex}{}_l}(\mathrm{v_l}(\mathrm{t}))$$
$$\mathrm{v'_r}(\mathrm{t}) = v_{\mathrm{r}}(\mathrm{t}) - \mathrm{a_r^{ex}}(\mathrm{v_r}(\mathrm{t}))$$

として補正を行った。ただし、$\mathrm{a_r^{ex}{}_l}(\mathrm{v_l}(\mathrm{t}))$、$\mathrm{a_r^{ex}}(\mathrm{v_r}(\mathrm{t}))$は左車輪、右車輪の移動距離の誤差について、速度がそれぞれ$\mathrm{v_l}(\mathrm{t})$、$\mathrm{v_r}(\mathrm{t})$に対応するモデルの平均である。

3.3. アーキテクチャ 2：ロボット側で制御

ロボット側で制御を行う場合は、コントローラ側でセンサー情報をもとに、ロボットの状態を求め、そのロボットの状態をタイムスタンプを付けて、ロボット側に送信する。ロボット側では、その情報をもとに、現在のロボットの状態を推定（予測）し、ロボットが実際に行う車輪の速度を計算し、制御を行う。図 2 に制御の流れを示す。

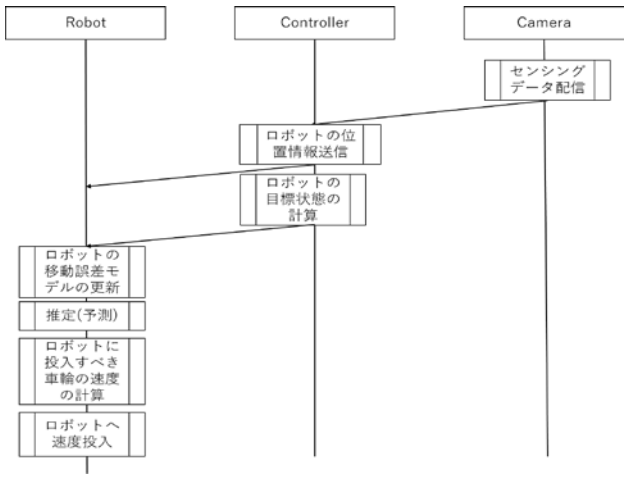

図 2　ロボット側で制御を行う際の流れ

ロボットの状態の把握：

本制御においては、コントローラにおいて、カメラ等のセンシング情報をもとに、当該センシングが行われた際のロボットの X 座標、Y 座標、角度 θ を得る。得られた情報は、ロボットが当該状況であった時刻のタイムスタンプを付け、ロボットに送る。

ロボットの移動誤差モデルの更新：

ロボット側では、受け取ったロボットの状態情報と、以前に受け取ったロボットの状態情報、その間に投入した制御をもとに、各車輪の移動距離の誤差を計算、誤差モデルの計算を行う。具体的な誤差の計算、誤差モデルの更新方法は、コントローラ側で制御を行う方法と同様である。

推定（予測）：

ロボット側では、車輪の回転速度の制御は、投入する回転速度を決定後、即座に行うことができる。その

ため、将来のロボットの状態を予測することは必要なく、現在のロボットの状況が把握できれば十分である。ただし、コントローラから送られてくるロボットの状態に関する情報は、遅延分前の情報である。そのため、コントローラから受信した情報をもとに、現在のロボットの状態を推定することが必要となる。この推定の手順は、コントローラ側で制御をする場合に、制御入力が投入される時点のロボットの状態を予測する手順と同様である。ただし、コントローラ側で制御する場合は、制御入力を伝えるパケットがロボットに到達すると考えられる時刻以降の将来の時刻における状態を予測する必要があるのに対して、ロボット側で制御をする場合は、将来ではなく、現在の状況を推定することとなる。そのため、遅延の揺らぎを考慮して、予測対象となる時刻を決める必要はなく、遅延が短い場合には、最近送られた情報をもとに正確な推定を行うことが可能となる。

ロボットの目標状態の計算：

ロボットの目標状態は、制御をロボット側で行う場合であっても、コントローラ側で行う。コントローラ側で目標状態をロボットに送り、その目標状態に近づくような制御をロボット側で行うことにより、コントローラで意図したとおりの制御をロボットにさせることが可能となる。ロボットの目標情報の計算の具体的な手順は、コントローラ側で制御を行う場合と同様である。

ロボットに投入する車輪速度の計算：

目標状態、ロボットの現在の状態の推定値をもとに、ロボットに投入する車輪速度を計算する。具体的な手順は、ロボット側で制御を行った場合と同様である。

4. シミュレーション結果

4.1. 評価環境

本評価において、距離は、ロボットの車輪間の距離を 14mm とする。また、移動速度に関するパラメータαは 271mm/s と設定した。また、ロボットの移動誤差は、表で定めた値と設定した。表 1 に本評価で生成した誤差のモデルを示す。ただし、本表は、1 秒間継続して、動作させた場合に観測される誤差として想定しているのに対して、本シミュレーションでは、1ms ごとにロボット位置を算出・記録しており、1ms ごとに誤差を生じる。そのため、シミュレーション上では、下記表の値をもとに、平均値を 1/1000 倍、標準偏差を$1/\sqrt{1000}$倍した正規分布に従う誤差を 1ms ごとに生じさせた。

コントローラからロボットへの通信周期は、通信レートを制御しない限り 10ms とした。つまり、コントローラから制御を行う場合の制御周期は 10ms となる。また、ロボット側で制御をする場合には、20ms 間隔で制御を行うものとした。また、カメラ・コントローラ間には固定で 133ms の遅延が入るものとし、コントローラ・ロボット間の遅延は、実測データに基づいた値に設定した。また、本評価に用いた経路を図 3 に示す。

表 1　誤差モデル

投入された車輪回転速度	右車輪の移動距離の誤差（1 秒間移動時）		左車輪の移動距離の誤差（1 秒間移動時）	
	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

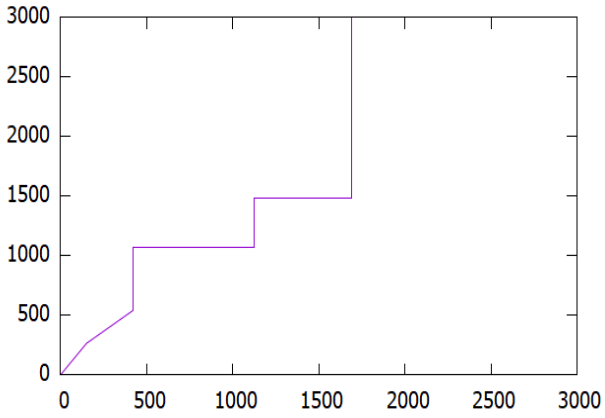

図 3　ロボットの目標軌跡

4.2. アーキテクチャ間の比較

まず、ロボット側で制御を行った場合とコントローラ側で制御を行った場合について、比較をする。制御をロボット側で行った場合と、コントローラ側で行った場合の大きな違いは、入力された制御を行う際の推定（予測）である。コントローラ側で制御を行う際には、当該制御入力が投入される時点でのロボット状態を予測するのに対して、ロボット側で制御を行う場合には、コントローラから送られてきた過去のロボットの情報から、現在のロボットの状況を推定する。そのため、(1) 推定や予測に用いる情報のうち、ロボットに実際に投入された各車輪の速度は、ロボット側制御であれば、すべて把握することができるのに対して、コントローラ側の制御では、計算した各車輪の速度がすべてロボットに投入されたものとして予測をせざるをえないものの、コントローラ・ロボット間の遅延により、入力時刻までに制御パケットが到達せず、破棄された制御入力も存在する、(2)コントローラ側で制御する際には、制御入力が実際に投入される時刻を遅延の揺らぎを考慮して、長めにとる必要がある（当該制御入力を行う時刻として制御コマンドに記載された時

刻までに制御コマンドがロボットに到達しない場合、当該制御コマンドは破棄されてしまう）のに対して、ロボット側制御では、現在のロボットの状況が推定できればよく、得られたロボットの状況からロボットの状態の予測（推定）が必要な現在時刻までの時間が短い。そのため、ロボット側制御の方が、より精度よく、現在の状況を推定することが可能であるといえる。

ロボット側制御とコントローラ側制御の違いを示すため、目標軌跡からの差を調べた。ここで、目標軌跡からの差は、各時刻において、目標軌跡とロボットの重心点の距離を調べたものである。

図 4 に目標軌跡からの差の累積捕分布を示す。本図を描くにあたり、実測された 6 パターンの遅延の時系列データを用い、各データについて、乱数のシードを変え、10 回評価を行い、その全シミュレーション結果の全時刻における目標軌跡とロボットの重心点間の距離を調べた。図より、ロボットで制御を行った場合の目標軌跡との差は、コントローラ側で制御を行った場合と比べ、極めて小さいことが分かる。これは、上述の通り、ロボット側で制御を行った方が制御時のロボットの状況を正確に推定することができるためである。

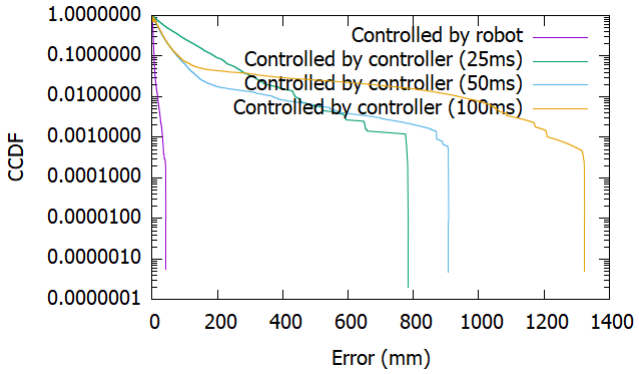

図 4　目標経路からの差の累積捕分布
(ロボット側制御とコントローラ側制御の比較)

4.3. 送信レート調整の効果

ロボット側で制御する際にも、コントローラからロボットに情報を送出する周期は重要であり、この周期が長くなると、ロボットは自身が推定した不正確な情報をもとに、行動を決定する必要がある。しかしながら、データの制御周期を短くすると、ネットワークに負荷をかけ、遅延が増大することが考えられる。この問題に対して、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方法[2]が提案されている。本評価では、文献[2]の手法を用い、情報送出周期を調整した場合と、固定のレート(10ms 間隔)で情報の送出を続けた場合について、比較を行う。本比較では、実際の無線ネットワーク環境下において、各制御周期で通信を行った場合の遅延を測定し、シミュレーションに用いた。

図 5 に、レート制御を行った上でコントローラからロボットに情報を提供し、ロボット側で制御を行った場合のロボットの軌跡、目標経路との差の累積捕分布の比較を示す。図より、レート制御を行いつつ、コントローラからロボットへの情報提供を行うことにより、固定のレートで情報提供を行った場合よりも、目標軌跡からの差を抑えることができていることが分かる。これは、上述の考察の通り、レート制御を行うことにより、ロボットに対して、最新の観測状況を伝えることができているためだと考えられる。

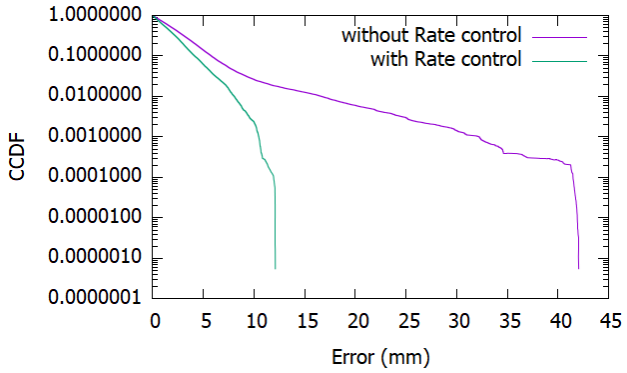

図 5　目標経路との差の累積捕分布
（レート制御のあり・なしの比較）

5. まとめ

本稿では、ロボットが動作する環境におけるロボットの状態を監視するセンサーが配置されている状況下において、無線ネットワーク経由でロボットを制御するためのアーキテクチャの検討を行った。その結果、環境センサーの値をもとにコントローラでロボットの状態を把握、ロボットに送出・ロボット側で、コントローラから受け取ったロボットの状態とその状態が観測された以後の自身の行動をもとに、現在のロボットの状態を推定し、行動を決定するという手法が、目標とする行動との誤差を抑えながら制御するのに有効であることが分かった。

本稿では、目標軌跡に合わせて移動をするという単純なタスクのみを扱ったが、今後は、より複雑なタスクについても検証を行う予定である。

文　献

[1] M.-F. R. Lee, et al, “Generalized predictive control in a wireless networked control system,” International Journal of Distributed Sensor Networks, vol. 9, p. 475730, Jan 2013.

[2] 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.