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

信学技報
TECHNICAL REPORT OF IEICE.

無線ネットワークを経由した移動ロボットの遠隔操作におけるベイズ推定を用いた環境同定手法

松田　拓己† 　大下　裕一† 　村田　正幸†

† 大阪大学大学院 情報科学研究科
E-mail: †{t-matsuda,y-ohsita,murata}@ist.osaka-u.ac.jp

あらまし　近年、ドローン、災害救助ロボット、無人搬送車など、ネットワーク経由で制御可能なロボットが注目を集めており、様々な手法が研究されている。ネットワーク経由で操作する機器においては、ネットワークの遅延が大きな問題となる。ネットワーク経由で機器を操作する際には、遅延時間分先のロボットの状態の予測が必要となる。しかし、予測した状態にロボットが達するとして、遅延分先の状態を予測したうえで制御を行うと、車輪のスリップやネットワークの遅延時間の揺らぎといった環境変動により、予測した状態と実際の状態の誤差が大きく異なり、 事前に想定した制御ができなくなるという問題が生じる。本稿では、遠隔制御のための環境同定をベイズ推定を用いて行う手法を提案する。本手法では、事前分布と観測により得られた情報をもとにベイズ推定により、ロボット制御により生じる誤差を推定し、推定結果を考慮してロボットの制御を行う。さらに、上記の事前分布を、ロボットを制御した結果を評価値として用いた遺伝的アルゴリズムにより定める。本稿では、提案手法の有効性をシミュレーションにより評価し、提案手法が制御目標からのずれを抑えながらロボットを動作させるのに、有効であることを示す。さらに、実機実験を通し、提案手法により実際の遠隔制御ロボットを制御可能であることを示す。
キーワード　ネットワーク制御システム、遠隔操作ロボット、ベイズ推定

State identification method using Bayesian inference for wireless remote control of mobile robot

Takumi MATSUDA†, Yuichi OHSITA†, and Masayuki MURATA†

† Graduate School of Information Science and Technology, Osaka University
E-mail: †{t-matsuda,y-ohsita,murata}@ist.osaka-u.ac.jp

Abstract　In recent years, robots that can be controlled using the network, such as drones, disaster rescue robots, and UAV, have attracted attention and various methods are being studied. Network delay causes a serious problem for the robots controled via network. A controller predict the state of the robot when the robot receives the command from the controller. But the predicted states may be significantly different from the actual state due to slip of the wheel and so on. As a result, the control via the network cannot achieve the suitable performance. In this paper, we propose an environment identification method using Bayesian inference for the remoto control of robots. In this method, the error of the robot control is infered by the Bayesian inference, whose prior probability distribution is optimized through a genetic algorithm. We evavluate our method by simulation to demonstrate that our method properly control the roobots, avoiding a large error. In addition, we implemented our method and demonstrate that our method can be used to control the actual robots.
Key words　Network control system, remote control robot, Bayesian inference

1. はじめに

近年、建設機械から搬送車まで、ネットワーク経由で制御可能なロボットが注目を集めており、ネットワークを経由することにより、作業を行う領域に人が立ち入ることなく、作業が可能となり、安全性の向上や作業効率の向上が期待されている [1]。

ネットワーク経由で機器を制御する際には、ネットワークの遅延を考慮して制御することが必要となる。すなわち、コントローラにおいて、制御コマンドが到達する将来のロボットの状況を予測し、予測した将来のロボットに対して適した制御コマ

ンドを計算し、制御コマンドをロボットに向かって送出する。この手順においては、将来のロボットの状況の予測精度が、制御性能に大きな影響を及ぼす。将来のロボットの予測は、それまでに送出した制御コマンドをもとに、その制御コマンドにより制御が行われた結果、ロボットが到達する状態を予測する形で行われるが、車輪がスリップする、遅延のゆらぎにより制御コマンドの到達が送れる等により、将来のロボットの状態が意図通りの状態に到達しているとは限らない。その結果、予測した状態と実際の状態の誤差が大きく異なり、意図通りの制御ができない。

本稿では、上記のようなネットワーク・ロボットの動作ともに不安定で、制御に誤差が生じる環境下において、ロボットの環境を同定し、制御に用いる手法について検討する。本稿では、不安定な環境下におけるロボットの状況の推定は、確率モデルで表す。この確率モデルは、遠隔制御ロボットを制御しながら、行った制御とその制御後に遠隔制御ロボットが到達する状態から求めることができる。しかし、この方法では、行った制御、制御後に到達したロボットの状態に関する情報が少ない、タスク実行初期の段階では、適切な環境同定ができないといった問題や、ロボットを制御する上で未経験の突発的な大きな誤差を考慮することができず、適切な制御を行うことができないという可能性がある。

本稿では、上記のような遠隔操作ロボットに対して、得られた観測結果が少ないタスク実行初期の段階であっても、適切な制御が可能となるような環境同定手法を提案する。文献 [2] では、 生物が少ない経験であっても、適切な認知を行うことができる仕組みを持つように進化をすることが示唆されており、進化の結果、自身が持つ事前分布をもとに観測情報を用いてベイズ推定を行い、発生しうる状況を認知するのに適した事前分布を持つような進化が行われる可能性を明らかにしている。そこで、機器の遠隔制御においても、ベイズ推定の認知を行うとともに、その事前分布を遺伝的アルゴリズムにより得ることにより、少ない観測情報しか得られていないタスク実行初期の段階であっても、現在の環境下で生じる誤差を適切に予測し、制御を行うことが期待できる。

提案する手法では、コントローラにおいて、(1) ロボットから得られる観測情報と、ロボットへの制御コマンドをもとに、当該コマンドに該当する制御を行った際に生じた誤差を計算、(2) 得られた誤差をもとに、事前分布を用いてベイズ推定により、現在の環境において生じる誤差を把握する。そして、把握された誤差を考慮し、制御コマンド到達時のロボットの状況を予測するとともに、ロボットに対して送る制御コマンドを計算する。遺伝的アルゴリズムにより求めた事前分布からベイズ推定を行うことにより、現在の環境で生じる誤差を推定しながら制御を行うコントローラを実装し、シミュレーションに より有効性を評価するとともに、実機実験を通し、実際のロボットを制御可能であることを示す。

本稿の構成は、以下の通りである。2. 章でネットワーク制御における、遠隔操作機器の環境同定手法を提案する。3. 章で本稿で使用した移動ロボットのモデルについて説明した後、3.2 章でシミュレーションによる評価の結果を示す。さらに、**??**章で実際のロボットで確認できた内容について述べ、4. 章でまとめと今後の課題について述べる。

2. ネットワーク制御における環境同定手法

2.1 遠隔制御の概要

本稿では、制御対象のロボットは、ロボットの状態 X_t の際に、制御入力 B_t を与えることにより、以下のように、次の制御コマンド到達時のロボットの状態が X_{t+1} に変化するようなロボットを考える。

$$X_{t+1} = F(X_t, B_t, \epsilon_t) \tag{1}$$

ただし、$F()$ は制御コマンドの影響の表す関数であり、ϵ_t は誤差項を表す。

上記のロボットを制御する際には、時刻 t において、ロボットが自身のセンサーや周囲のセンサー等から情報を取得し、観測値 A_t を得、A_t をコントローラに送出する。コントローラでは、片方向遅延 d_1 後の時刻 $t+d_1$ に観測値 A_t を受け取り、A_t から時刻 t におけるロボットの状態 X_t を推定し、遅延分先の時刻 $t+d_1+d_2$ の状態を予測する。そして、予測した状態に対する制御入力 $B_{t+d_1+d_2}$ を決定し、ロボットに送出する。この制御を繰り返すことにより、遠隔地にある機器の制御を行う。本節では、上記環境下で、ベイズ推定に基づいてコントローラが行う状況の認知と、その認知された状況にもとづく制御方法、ベイズ推定のための事前分布の定め方について述べる。

2.2 ベイズ推定にもとづく環境同定と環境同定にもとづく手法

2.2.1 環境同定

本稿では、遠隔制御ロボットを用いたタスクを実行しながら、行った制御とその制御後に遠隔制御ロボットが到達する状態をもとに $P(\epsilon)$ を推定することにより、現在、ロボットが動作する環境の同定を行う。タスクを実行しながら、$P(\epsilon)$ の推定を行う場合、これまでの観測結果には含まれないような大きな誤差が生じる可能性についても考慮することが求められ、このような場面においても、ロボットに対して、誤差の影響が大きくなってしまうような不適切な制御入力を与えることは防ぐ必要がある。

本稿では、ベイズ推定により、$P(\epsilon)$ の推定を行う。これにより、事前分布を適切に定めることができれば、観測結果が不十分な場合であっても、不適切な制御入力を与えてしまうことを避けることができる。

本稿における、コントローラが $P(\epsilon)$ を逐次推定する手順を以下に示す。

（1） ロボットから受信した観測値 A_t から、X_t を推定する
（2） X_t、X_{t-1}、B_{t-1} から、式 (1) より、ϵ_{t-1} を得る
（3） 直近 K タイムスロットの $\epsilon_{t-K:t-1}$ をもとに、$P(\epsilon|A_{t-K:t-1})$ をベイズ推定により得る。

$$P(\epsilon|A_{t-K:t-1}) = \alpha P(\epsilon_{t-K:t-1}|\epsilon)P^{\text{prior}}(\epsilon)$$

ただし、$P^{\text{prior}}(\epsilon)$ は事前に定めた事前分布である。

2.2.2 同定した環境を考慮した制御

a） 現在の状況の認知

コントローラでは、A_t の情報を受信後、以下のようにベイズ推定を行うことにより、時刻 t の状況を推定する。

$$P(X_t) = \alpha P(A_t|X_t)\hat{P}(X_t) \tag{2}$$

ただし、$\hat{P}(X_t)$ は、事前に定められた状況 X_t の事前確率分布であり、$P(A_t|X_t)$ は状況 X_t の場合に得られる観測値 A_t の確

率分布であり、式 (1) をもとに定義される。また、α は定数である。

b） 将来の状況の予測

時刻 $t+d$ における制御入力を計算するためには、ロボットが制御入力を受け取る時刻 $t+d$ における状態を予測する必要がある。

時刻 $t+1$ のロボットの状態の確率分布 $\hat{P}(X_{t+1})$ は以下のように予測できる。

$$\hat{P}(X_{t+1}) = \int_{X_t,\epsilon_t} P(X_{t+1}|X_t, B_t, \epsilon_t)\hat{P}(X_t)P(\epsilon_t|A_{t-K:t-1}) \quad (3)$$

これを繰り返すことにより、時刻 $t+d$ の状態の予測結果 $\hat{P}(X_{t+1})$ を得る。

c） 制　　御

時刻 $t+d$ の状態の予測結果 $\hat{P}(X_{t+d})$ を得たのちに、式 (3) の予測により、$\hat{P}(X_{t+d+1}|B_{t+d})$ の関係を得る。この関係をもとに、

$$P(|X_{t+d+1} - X^{\text{target}}| > X^{\text{threshold}}) < \lambda$$

を満たす制御入力 B_{t+d} を計算し、制御入力とする。

2.3 進化計算による事前分布算出

ベイズ推定を用いた環境同定では、事前分布の影響が大きく、事前分布を適切に定めていないと、誤差の影響が大きくなるような不適切な制御入力を与えてしまうことも考えられる。そのため、タスクを実行しながら、ベイズ推定により環境の同定を行う手法においては、適切な事前分布を定めることが必要となる。

生物が進化の過程でベイズ推定における適切な事前分布を獲得した可能性が示唆されている [2]。そこで、本稿においても、遺伝的アルゴリズム [3], [4] により、事前分布を定めるものとする。

本稿では、$P^{\text{prior}}(\epsilon)$ の分布の形状は既知であるとし、そのパラメータを進化させる。具体的な手順は、以下の通りである。

（1） 初期化： $P^{\text{prior}}(\epsilon)$ の分布を定めるパラメータをランダムに設定与えた個体を N 個生成する。

（2） 評価： 各個体について、当該個体が保持するパラメータの $P^{\text{prior}}(\epsilon)$ を事前分布として持つコントローラにより、移動ロボットを遠隔制御するタスクを想定しうる環境下で実行した場合の挙動をシミュレーションにより得る。そして、そのタスクの正確性、速度といった指標で、各個体を評価する。

（3） 選択・交叉・突然変異： 評価値をもとに選択、交叉、突然変異を行い、新たな個体を生成する。

（4） 2 へ戻る

上記の手順では、適切にタスクを実行するのに有用な事前分布を持つように選択・交叉が行われる。そのため、上記の手順による進化を行うことで、適切な事前分布を得ることができると考えられる。

3. 対向二輪移動ロボットへの適用

3.1 移動ロボットのモデル

3.1.1 移動ロボットのダイナミクス

本ロボットの車輪の半径を r、2 つの車輪の間隔を W とする。移動ロボットの状態はロボットの重心が位置する 2 次元の座標と向きで表される。すなわち、時刻 t におけるロボットの状態を X_t は以下のように定義される [5]。

$$X_t = \begin{pmatrix} x_t \\ y_y \\ \theta_y \end{pmatrix} \quad (4)$$

ただし、x_t、y_t はロボットの重心の座標であり、θ_t はロボットの進行方向の X 軸方向に対する角度である。

対向二輪型移動ロボットには、2 つの駆動輪の速度を入力として与え、制御入力 B_t は次式のように定義される。

$$B_t = \begin{pmatrix} w_t^{\text{right}} \\ w_t^{\text{left}} \end{pmatrix} \quad (5)$$

ただし、w_t^{right} は右車輪の速度、w_t^{left} は左車輪の速度である。

ロボットが状態 X_t で、制御入力 B_t を受け取った際、制御誤差が生じない場合は、次のロボットの状態 X_{t+1} は、以下のようになる。

$$X_{t+1} = X_t + \begin{pmatrix} v_t \cos(\theta + \frac{w_t}{2}) \\ v_t \sin(\theta + \frac{w_t}{2}) \\ w_t \end{pmatrix} \quad (6)$$

ただし、

$$\begin{pmatrix} v_t \\ w_t \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{W} & -\frac{r}{W} \end{pmatrix} \begin{pmatrix} w_t^{\text{right}} \\ w_t^{\text{left}} \end{pmatrix} \quad (7)$$

3.1.2 移動ロボットの誤差モデル

ロボットの車輪はスリップ等により、意図した状態にまで到達できない。また、ネットワークの遅延時間の揺らぎにより意図した位置まで到達しない、もしくは意図した状態を行きすぎるといったことが起こりうると考えられる。そこで本稿では、右車輪、左車輪が、それぞれ、意図した移動距離の $1-\epsilon_t^{\text{right}}$ 倍、$1-\epsilon_t^{\text{left}}$ 倍となる誤差が生じるものとし、$\epsilon_t^{\text{right}}$、$\epsilon_t^{\text{left}}$ は正規分布に従うものとした。

この場合、ロボットが状態 X_t で、制御入力 B_t を受け取った際、次のロボットの状態 X_{t+1} は、以下のようになる。

$$X_{t+1} = X_t + \begin{pmatrix} v'_t \cos(\theta + \frac{w_t}{2}) \\ v\text{'}_t \sin(\theta + \frac{w_t}{2}) \\ w\text{'}_t \end{pmatrix} \quad (8)$$

ただし、

$$\begin{pmatrix} v'_t \\ w\text{'}_t \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{W} & -\frac{r}{W} \end{pmatrix} \begin{pmatrix} w_t^{\text{right}}(1-\epsilon_t^{\text{right}}) \\ w_t^{\text{left}}(1-\epsilon_t^{\text{left}}) \end{pmatrix} \quad (9)$$

この予測を $t+1$ から逐次的に行うことにより、$\hat{P}(X_{t+d})$ を得ることができる。

3.1.3 移動ロボットにおける制御入力の計算

目標状態へ到達するための制御入力の計算

本節では、移動ロボットの場合における制御入力の導出方法について説明する。まず、制御入力が到達する時刻 $t+d$ のロボットの状態の予測結果をもとに、時刻 $t+d+1$ に目標状態 X^{target} に到達することができる入力 B_{t+d} を求める。ここで、予測された時刻 $t+d$ の状態の期待値を $X_{t+d}^{\text{ave}} = (x_{t+d}^{\text{ave}}, y_{t+d}^{\text{ave}}, \theta_{t+d}^{\text{ave}})$ とする。B_{t+d} は以下のように計算できる。

（1） 時刻 $t+d$ のロボットの状態の期待値 $[x_{t+d}^{\text{ave}}, y_{t+d}^{\text{ave}}, \theta_{t+d}^{\text{ave}}]$ と目標状態 $[x^{target}, y^{target}, \theta^{target}]$ との差 $[e_x, e_y, e_\theta]$ を計算する。

$$\begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} = \begin{pmatrix} \cos\theta_C & \sin\theta_C & 0 \\ -\sin\theta_C & \cos\theta_C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^{target} - x \\ y^{target} - y \\ \theta^{target} - \theta \end{pmatrix} \quad (10)$$

（2） 目標地点に到達するために、時刻 $t+d$ のロボットが出すべき速度と角速度 (v, w) を導出する。

$$v = K_{(t+d)} \quad (11)$$

$$w = 2A_{(t+d)}K_{(t+d)} \quad (12)$$

ただし、

$$A_{(t+d)} = sign(e_x)\frac{e_y}{e_x^2} \quad (13)$$

$$K_{(t+d)} = sign(e_x)\frac{\alpha}{1+|A_{(t)}|} \quad (14)$$

ただし、α は正の定数であり、値が大きくなるほど ロボットが速く移動する。

（3） 速度と角速度から入力 $B'_{t+d} = (w_{t+d}^{\text{left}}, w_{t+d}^{\text{right}})$ を計算する。計算式は次のようになる。

$$w_{t+d}^{\text{right}} = \frac{v}{r} + \frac{Ww}{2r} \quad (15)$$

$$w_{t+d}^{\text{left}} = \frac{v}{r} - \frac{Ww}{2r} \quad (16)$$

誤差を考慮した速度の抑制

時刻 $t+d$ に入力 B_{t+d} が与えられた際、時刻 $t+d+1$ のロボットは、式 (8) により求まる。入力 B_{t+d} の絶対値が大きくなればなるほど、式 (8) における $\epsilon^{\text{left}}, \epsilon^{\text{right}}$ の影響が大きくなる。本稿では、時刻 $t+d+1$ のロボットの状態の取りうる範囲が、一定以下となるように、上記の手順で計算された入力 B'_{t+d} をスケールすることにより、$\epsilon^{\text{left}}, \epsilon^{\text{right}}$ の影響が大きくなることを防ぐ。すなわち、変数 s を導入し、以下の最適化問題を解き、入力 $B_{t+d} = sB'_{t+d}$ を定める。

目的関数：

$$minimize|s-1|$$

制約条件：

$$\forall X_{t+d+1}(sB_{t+d}) \in \{X_{t+d+1}(sB_{t+d}) | P\left(X_{t+d+1}(sB_{t+d})\right) < P^{\text{threshold}}\},$$
$$|X_{t+d+1}(sB_{t+d}) \breve{} E[X_{t+d+1}(sB_{t+d})]| \leq X^{\text{threshold}}$$

ただし、$P^{\text{threshold}}$、$X^{\text{threshold}}$ は閾値であり、$P\left(X_{t+d+1}(sB_{t+d})\right)$ は、

$$P\left(X_{t+d+1}(sB_{t+d})\right) = \int_{X_{t+d},\epsilon_t} P(X_{t+d+1}|X_{t+d}, sB_{t+d}, \epsilon_{t+d})\hat{P}(X_{t+d})P(\epsilon_{t+d})$$

である。

これにより、誤差が大きく、大きい ϵ_{t+d} に対する $P(\epsilon_{t+d})$ が大きな環境下においては、s が小さな値に設定され、速度を抑制することが可能となる。また、$\hat{P}(X_{t+d})$ の分布が大きい場合には、s を 0 として、ロボットの状態の不確実性がさらに高まることを防止することも可能となる。

3.1.4 環境同定

移動ロボットにおいて、ロボットを制御した際に生じる誤差 $\epsilon = (\epsilon^{\text{left}}, \epsilon^{\text{right}})$ の確率モデルを同定することにより、現在の環境を同定する。本稿では、簡単のため、$P(\epsilon)$ は平均 0 の正規分布に従うとし、事前分布として、以下の分布を与えた。

$$\epsilon_i \sim N(0, \sigma)$$

$$\frac{1}{\sigma^2} \sim Gamma(\alpha, \beta)$$

また、本稿では、遺伝的アルゴリズムにより、α と β を定める。

遺伝的アルゴリズムでは各世代 g で存在する事前分布 N 個から適応度に応じて選択を行う。選択後、交叉、再生、突然変異の動作を行い次世代 $g+1$ における事前分布を N 個生成する。これを繰り返すことにより、適切な事前分布のパラメータ α、β を求める。

本稿では、適応度は、各個体に対応する事前分布を持つコントローラにより、遠隔操作のタスクを実行し、その際に目標となる移動軌跡からのずれと、タスク完了までの時間の双方を考慮して以下のように定めた。

$$f_i = \alpha_T \frac{1}{\frac{T_i - M^T}{M^T} + 1} + \alpha_E \frac{1}{\frac{E_i}{M^E} + 1} \quad (17)$$

ただし、M^T はタスクを環境するのに最低限必要となる制御時間、M^E は許容する目標軌跡からのずれ、T_i は個体 i のタスク完了に要した時間、E_i は個体 i の目標とする移動軌跡からのずれの最大値である。また、α_T、α_E は、タスク実行時間、目標軌跡からのずれに対する重みである。本評価値では、許容するタスク実行時間を超える、あるいは、許容する目標軌跡からのずれを超えると、0 となる。本稿では、許容する目標軌跡からのずれを 20 とし、それが達成できる限りは、タスク実行時間を重視するように、α_T を α_E に対して十分大きな値に設定した。

また、本稿では、以下の手順で選択、交叉、再生、突然変異を発生させた。

- 選択

本稿では、個体の選択方法はルーレット選択を使用する。各世代の N 個の個体に対して、適応度を計算し、それをもとに計算した以下の確率 p_i に基づいて個体を選択する。

$$p_i = \frac{f_i}{\Sigma_{k=1}^{N} f_k} \quad (18)$$

- 交叉

本稿では、上述の方法で、親個体を 2 体選択する。そして、選択した個体の子個体の事前分布のパラメータ (α_c, β_c) を次のように生成する。

$$\alpha_{\text{c}} = \text{random}(\min(\alpha_1, \alpha_2) - aI_\alpha, \max(\alpha_1, \alpha_2) + aI_\alpha) \quad (19)$$

$$\beta_{\text{c}} = \text{random}(\min(\beta_1, \beta_2) - aI_\beta, \max(\beta_1, \beta_2) + aI_\beta) \quad (20)$$

$$I_\alpha = |\alpha_1 - \alpha_2| \quad (21)$$

$$I_\beta = |\beta_1 - \beta_2| \quad (22)$$

ただし、2 体の親個体の事前分布のパラメータをそれぞれ $(\alpha_1, \beta_1), (\alpha_2, \beta_2)$ とする。また、a は定数であり、$\text{random}(a, b)$ は区間 $[a, b]$ の一様乱数を表している。

- 突然変異

事前分布のパラメータをランダムに設定し、新たに生成する。新たに生成する事前分布のパラメータを (α, β) とするとパラメータは以下に示す式のようにして設定する。

$$\alpha = \text{random}(0, MAX_c) \quad (23)$$

$$\beta = \text{random}(0, MAX_c) \quad (24)$$

ただし、MAX_c は定数であり、$\text{random}(a, b)$ は区間 $[a, b]$ における乱数である。

• 再生

次世代 N における事前分布を導出する際に、現世代 $N-1$ に存在する事前分布のパラメータ (α, β) を 1 つ選択し、次世代にそのまま受け継ぐ。

3.2 シミュレーション評価

本章では、シミュレーションにより、遺伝的アルゴリズムにより導出された事前分布を用いたベイズ推定により $P(\epsilon)$ を推定し、制御に用いる手法が、移動ロボットを動作させうる様々な環境下においても、目標軌道から大きくずれることなく、適切にタスクを完了することができることを示す。

3.3 評 価 環 境

3.3.1 制御を行う環境

本シミュレーションにおいては、10ms ごとにコントローラはロボットに送出する指示を計算、ロボットは自身の座標、向きの情報をコントローラに送出するものとする。また、簡単のため、ロボットが把握している自身の位置は正確であるものとする。ロボット・コントローラ間には、往復遅延が存在するものとし、遅延時間は平均 100ms, 分散 0.05 および 0.25 の正規分布に従う環境で動作させた。また、本シミュレーションにおいて、各車輪が進む速度は、スリップ等の影響により、制御で投入された速度に、平均 1、標準偏差が 0.05 から 0.2 の正規分布に従う値を乗じた速度になるという環境下で動作するものとした。

3.3.2 ロボットのタスク

本シミュレーションでは、座標位置 (0,0) から (5000,0) までの直線を移動するタスクを課した。本シミュレーションの環境では、ロボットの左右の車輪の移動距離に誤差が生じる。そのため、大きな誤差が生じた際には、ロボットの角度がずれ、意図した直線から離れた箇所を経由する。そのため、本シミュレーションでは、本タスクにおいて、実際にロボットが経由した軌跡の Y 軸からのずれに注目した。

3.3.3 比 較 対 象

本節では、以下の手法を比較する。

a） 提 案 手 法

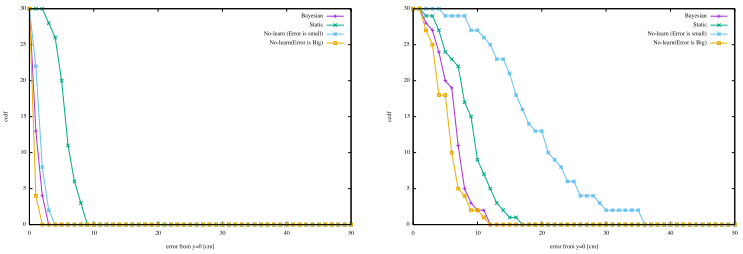
遺伝的アルゴリズムにより求めたパラメータを持つ事前分布と、直近 20 タイムスロットの誤差の観測地をもとに、ベイズ推定により誤差を把握する。本評価においては、各世代 50 個体を生成、50 世代進化を行った結果得られたパラメータをもつ事前分布を初期事前分布として与えた。

b） 統計的手法

事前分布を用いずに、制御中に得られた直近 20 タイムスロットの誤差を用いて誤差を学習し、制御に用いる。本手法では、制御開始時点では、誤差についての情報を持たず、制御を行いながら、制御結果をもとに ϵ^{left}、ϵ^{right} を計算し、その平均 μ、分散 σ^2 を計算する。そして、ϵ^{left}、ϵ^{right} が平均 μ、分散 σ^2 の正規分布に従う想定して、ロボットの状態の予測・制御を行う。本手法と比較することにより、事前分布を用いて、逐次ベイズ推定により現在の環境で生じる誤差を推定することの効果を確認する。

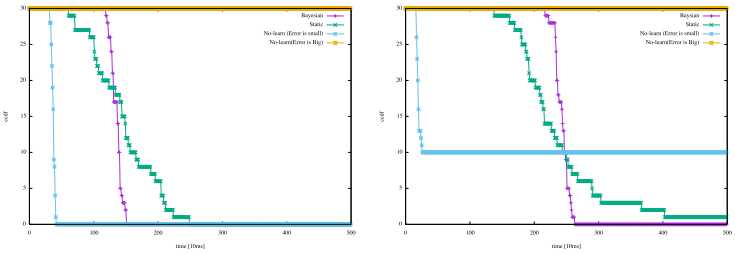
c） 固定の誤差モデルを用いる手法

制御開始前に、ロボットの移動時に生じる誤差のモデルに関するパラメータをすべて固定値として与える。具体的には、ϵ^{left}、ϵ^{right} が従う分布を平均 0、分散 σ^2 の正規分布（ただし、σ はパラメータとして事前に与える）としてモデル化し、このモデルを用い、ロボットの状態の予測・制御する。本評価では、$\sigma = 0.05$、$\sigma = 1.0$ の 2 種類のモデルを用いた。本手法との比較により、誤差モデルのパラメータを環境に合わせて学習することの効果を確認する。



(a) 遅延の分散、車輪速度の誤差が小　　(b) 遅延の分散、車輪速度の誤差が大

図 1: 経路からのずれの最大値の累積補分布



(a) 遅延の分散、車輪速度の誤差が小　　(b) 遅延の分散、車輪速度の誤差が大

図 2: 制御の時間の累積補分布

3.3.4 結　　果

提案手法でロボットを制御した際と、比較対象の手法でロボットを制御した際の動作について比較する。本評価では、車輪の移動距離の誤差の分散値がそれぞれ 0.025、0.04 の環境において、各制御手法でロボットを制御するタスクを 30 回行い、各タスクの試行において、目標軌跡からのずれの最大値を調べた。図 1a, 1b に、その累積補分布を示す。

図より、いずれの環境下においても、ベイズ推定により誤差を推定する手法は、目標軌跡からのずれを小さく抑えることができていることが分かる。それに対して、$\sigma = 0.05$ の固定の誤差モデルを用いた手法や、誤差が大きい環境（車輪の移動距離の誤差の分散値が 0.04 の環境）下において、目標軌跡からのずれが大きくなっている。これは、コントローラが想定する以上の誤差が生じているためである。同様に、観測した誤差のみから誤差を推定する手法でも、多くの場合、誤差が大きい環境（車輪の移動距離の誤差の分散値が 0.04 の環境）下での目標軌跡からのずれが大きい。これは、得られた直近の観測結果のみからは、誤って誤差が小さい環境と認識し、高速なロボットの移動を行うような制御入力を行ってしまう場合が存在することが原因である。それに対して、ベイズ推定を行う手法では、十分な観測結果が得られていない時点では、大きな誤差も想定し、速度を抑制した制御入力を与える。その結果、いずれの環境においても、目標軌跡からのずれを抑えた制御が可能となる。

図 2a、2b に、各手法を用いて制御を行った際のタスク完了までにかかる時間の累積補分布を示す。横軸はタスク完了までにかかる時間を表しており、縦軸はロボットの累積補分布を表す。図より、$\sigma = 1$ の固定の誤差モデルを用いた手法では、車輪の移動距離の誤差の分散が 0.0025 と、小さな誤差しか生じない環境下においても、誤差が大きな環境下と同じく、長い時

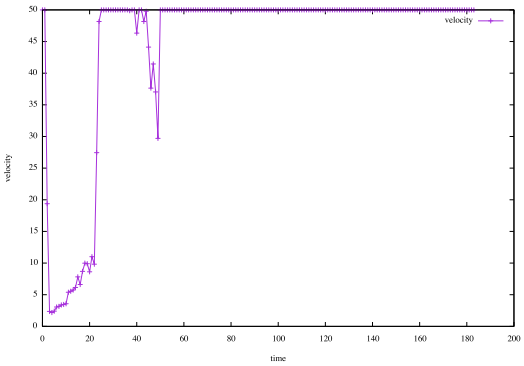


図 3: ロボットの速度の時間変化

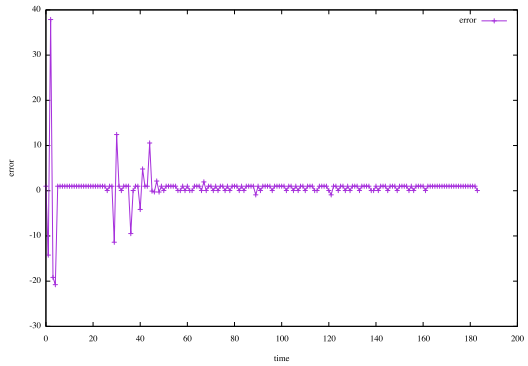


図 4: ϵ の時間変化

間を要していることが分かる。これは、固定の誤差モデルを用いた手法では、環境によらず、当該モデルに従い、誤差の影響を抑えるように速度を抑制するためである。また $\sigma = 0.0$ の固定の誤差モデルを用いた手法においては、誤差の大小に関わらず、速度を抑制できていないため、タスク完了までの時間が短くなっている。しかし、車輪の誤差、遅延時間の揺らぎが大きい環境では半数ほどの制御において経路からのずれが大きくなりすぎてしまい、次の目標点まで移動するための制御入力を導出できず、コントローラがロボットを停止させているため、制御時間が無限にかかる個体もある。つまり、固定の誤差モデルを用いた手法では、そのモデルがタスクを実行する環境と合致していれば、適切な制御を行うことができるものの、タスクを実行する環境が保持している誤差モデルと異なる場合には、適切な制御を実行することができない。それに対して、ベイズ推定を行う手法によるタスク完了までの時間は、車輪の移動距離の誤差の分散値が小さい場合には、短い時間でタスクを完了することができることが分かる。これは、ベイズ推定を行う手法では、タスクを実行しながら観測された誤差から、現在の環境で発生しうる誤差が小さいと認識できた場合には、ロボットの移動速度を速めることができているためである。

つまり、ベイズ推定を行うことにより、各環境の誤差を推定し、その環境にあった制御が可能であるといえる。

3.4 実機実験での動作確認

本節では、上記の手法が、実機において動作することを確認した。実験には、対向移動二輪ロボットとして、Khepera IV を用いた [6]。本稿では、Khepera IV を無線 LAN に接続した。コントローラのプログラムは、PC で動作するプログラムとして実装し、同一無線 LAN 配下のノート PC で動作させた。制御の際には、Khepera に搭載された超音波センサーをもとに壁からの距離を測位して、始点から決められた終点まで直線移動をするタスクを実行した。制御周期は 10ms とした。

実験により、本稿の環境同定手法を導入した場合であっても、10ms 間隔での制御を実現できることを確認した。また、本実験の結果、コントローラにおいて、$\epsilon^{\mathrm{right}}$ の値を図 4 に示し、図 3 では各制御周期におけるロボットの速度を示す。これらの図より、$\epsilon^{\mathrm{right}}$ が大きくなったことを検知し次第、ロボットの移動速度を下げ、誤差の影響を抑えるような制御を行うという、状況に合わせた速度抑制が実現できていることが確認できる。

4. おわりに

本稿では、対向二輪ロボットを例として、遠隔制御のための、ベイズ推定を用いた環境同定手法を提案した。提案手法では、遺伝的アルゴリズムにより求めた事前分布と実際に観測された誤差をもとに、ベイズ推定を行うことにより、現在の環境で生じる誤差の確率モデルを推定しながら制御を行う。本稿では、提案手法の有効性をシミュレーションにより示すとともに、実機実験を通し、提案手法が実際のロボットの制御に適用できることを示した。

本稿の実機実験では、実機での動作確認にとどまっているが、今後は、実機実験を通して、実際の環境下、特に時間帯や移動先の状態によりロボットの動作環境が大きく異なる場合における提案手法の有効性を検証する予定である。

謝辞 本研究の一部は、文部科学省科学究費補助金基盤研究 (C) 16K00125 によっている。ここに記して謝意を表す。

文　献

[1] 古屋弘，“建設会社が目指す ict の活用とロボット化，” 計測と制御，vol.55，no.6，pp.489–494，2016.

[2] J.C. Ramírez and J.A. Marshall, “Can natural selection encode bayesian priors?,” Journal of Theoretical Biology, pp.57–66, Aug. 2017.

[3] H. Kitano, “Genetic algorithm,” Journal of Japanese Society for Artificial Intelligence, vol.7, no.1, pp.26–37, 1992.

[4] I. Ono, H. Satoh, and S. Kobayashi, “A real-coded genetic algorithm for function optimization using the unimodal normal distribution crossove,” Journal of Japanese Society for Artificial Intelligence, vol.14, no.6, pp.1146–1155, nov 1999. https://ci.nii.ac.jp/naid/110002808247/

[5] C. Lozoya, P. Martı, M. Velasco, and J. Fuertes, “Effective real-time wireless control of an autonomous guided vehicle,” Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on, pp.2876–2881, Nov. 2007.

[6] K-term-Corporation, “khepera4,” https://www.k-team.com/mobile-robotics-products/khepera-iv.