

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の提案と実装

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。そのため、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。本報告では、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する。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、実機実験により有効性を評価する。評価結果より、タスクを実行しながら、観測された制御誤差の情報のみから、現在の環境において生じる誤差を推定して制御に用いる手法と比べ、目標軌跡からのずれと制御時間を削減できることを示す。

キーワード ネットワーク制御システム、遠隔制御機器、ベイズ推定、環境同定

Mobile Robot Control through Real-Time Identification of Physical/Network Environment by Bayesian Inference

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. In devices operated via a network, network delay is a serious problem. Therefore, it is necessary to predict the state of the robot ahead of the delay time. However, if the robot reaches the predicted state and control is performed after predicting the delayed state, the environmental fluctuation such as the slip of the wheel or fluctuation of the delay time of the network causes the error between the predicted state and the actual state. There is a problem that the control assumed in advance can not be performed. In this report, we propose an environment identification method using Bayesian Inference for remote control with an opposed two - wheeled robot as an example. By implementing Bayesian Inference, we implement a controller that controls while presuming errors occurring in the current environment, and evaluate the effectiveness by real machine experiment. From the evaluation results, we show that the deviation from the target trajectory can be reduced compared to the method using statistical estimation which estimates the error occurring in the current environment only from the observed control error information while executing the task.

Key words Network control system, remote control robot, Bayesian inference, Environment Identification

1. はじめに

ドローン、災害救助ロボット、無人搬送車などのロボットが開発され、広く利用されるようになってきている。このような機器の多くはネットワークを経由して制御されている [1]~[6]。一般的に遠隔制御機器はネットワークを介して接続されたコントローラー、ロボット、センサーで構成されている。ロボットの計算リソースは限られている一方、コントローラーは十分な計算リソースを持っており、コントローラーはセンサーデータを収集して分析し、ロボットへの入力を計算する。制御入力は、ネットワークの遅延時間分だけ後にロボットへ到達するため、コントローラーは、入力受信時のネットワーク遅延とロボットの状態を予測し、予測状態に対して入力を計算する。

ネットワークを介してロボットを制御する場合、ロボットの状態を予測することが重要であるが、ロボットの実際の状態が予測されたものと異なる場合、送信される入力は実際の状態での制御に適していない。ロボットの状態を予測する1つのアプローチとして、ロボットをモデル化することが挙げられる。モデル化することにより、ロボットの将来の状態を予測できるようになる。ただし、ネットワーク遅延の変動により、一部の入力が意図した時間とは異なる時間に実行されることがあるとロボットの実際の状態はモデルと異なる場合がある。また、移動ロボットの場合では床の表面や積載する荷物の重量などによる車輪のスリップなどが考えられる [7]。そのため、ロボットがネットワーク遅延の変動や車輪のスリップなどの環境を考慮して予測を行うことが重要となる。

ロボットが動作する環境に応じた誤差を考慮するためのアプローチとして制御とその制御の結果の対応を学習することが考えられる。[8]~[11]。制御とその結果をもとに学習を行い、その結果を利用することにより、コントローラーは将来のロボットの状態を正確に予測し、適切な入力を計算することができる。

ただし、動作する環境は変化する場合がある。たとえば、ロボットは、床の材質が変化、ネットワーク状態の変化やタスクの変更などが考えられる。コントローラーは、環境が変化してもすぐにその変化をとらえる必要があるが、機械学習に基づく方法では、十分な量の監視データが必要であり、現在の環境を識別するのに時間がかかる。[8]~[11]。

本研究では、上記のように遠隔制御装置での状態認識と予測での観測結果が少ない場合でも、ロボットがコントローラーの意図した状態に到達できる制御を可能にするフレームワークを確立する。文献 [12] は、生物は経験がほとんどなくても適切な認知を実行できるメカニズムを持つように進化し、進化の結果として、以前の分布が観察された情報に基づいてベイズ推定 [13] が実行され、起こりうる状況の認識に適した事前分布を持つ進化の可能性が示されている。そのため本研究では、機器の遠隔制御でも進化を通じて様々な環境で適切な制御を可能にする技術の確立を目指し、本論文では、進化を通じてさまざまな環境で適切な制御を可能にする認知能力を獲得できるかどうかを検証する。

2. 遠隔制御における環境同定手法

2.1 ベイズ推定にもとづく環境同定手法

2.1.1 遠隔制御の概要

本研究で想定するシステムでは、ロボットには限られた計算資源しかなく、コントローラーには十分な計算資源があると

する。図 1 は、本研究で想定する遠隔制御の概要を示す。 X_t は、ロボットが t 番目の入力を受信したときのロボットの状態、 A_t は、 X_t に対応するセンサーの観測値である。図のように、コントローラーはセンサーから時刻 t で観測値 A_{t-d_1} を得る。ここで、 d_1 はセンサーとコントローラー間のネットワーク遅延である。次に、コントローラーは時刻 $t-d_1$ でのロボット X_{t-d_1} の状態を推定する。ネットワーク遅延のため、時刻 t に計算された入力は、時刻 $t+d_2$ にロボットによって受信される。そのため、コントローラーはロボット X_{t+d_2} の将来の状態を予測する。次に、コントローラーは X_{t+d_2} に基づいて入力 B_{t+d_2} を計算し、ロボットに送信する。

$P(A_t|X_t)$ は、センサーとロボットの状態との関係によってモデル化でき、静的である。ただし、 $P(X_{t+1}|X_t, B_t)$ は環境に依存する。たとえば、床の表面やロボットが荷物を運ぶ場合の荷物の重量などによるスリップの量やネットワーク遅延などに依存する。したがって、現在の環境の $P(X_{t+1}|X_t, B_t)$ を識別する必要である。

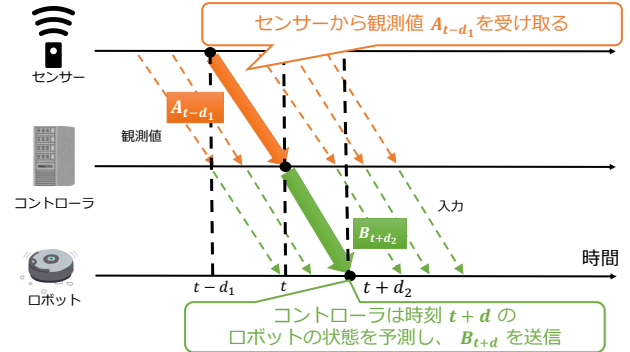


図 1: 想定する遠隔制御

上記環境下でコントローラーが行うロボットの状態の認知、将来の状態予測、制御方を以下に述べる。

2.1.2 現在の状況の認知

コントローラーでは、 A_t の情報を受信後、以下のようにベイズ推定により、時刻 t の状況を推定する。

$$P(X_t) = \alpha P(A_t|X_t) \hat{P}(X_t) \quad (1)$$

ただし、 $\hat{P}(X_t)$ は、前の時刻までに把握されていた状況 X_t の確率分布であり、 $P(A_t|X_t)$ 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、事前に与えられるものであるとする。また、 α は、 $\sum P(X_t) = 1$ となるような定数です。

2.1.3 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。

$$\hat{P}(X_{t+1}) = \int_{X_t, \epsilon_t} P(X_{t+1}|X_t, B_t, \epsilon_t) \hat{P}(X_t) P(\epsilon_t) \quad (2)$$

ただし、 $P(\epsilon_t)$ はコントローラが把握している誤差の分布、 $P(X_{t+1}|X_t, B_t, \epsilon_t)$ は、時刻 t におけるロボットの状態が X_t で、制御入力 B_t が与えられた際に誤差 ϵ_t が発生したときに次状態の分布を示す。

本報告における対向二輪型移動ロボットにおいては、 $P(X_{t+1}|X_t, B_t, \epsilon_t)$ は、モンテカルロ法 [14] を用いたシミュレーションにより以下のように求める。

(1) 時刻 t におけるロボットの状態 X_t となるエージェントを E 個生成する。

(2) 1. で得られたそれぞれのエージェントに対して $P(\epsilon)$ を用いて L 個の ϵ を生成し、入力 B_t を使用してそれぞれの ϵ が生じた場合に到達するロボットの状態 X_{t+1} を求める。

(3) 2. で得られた LE 個のロボットの状態を $\hat{P}(X_{t+1})$ として保存する。

2.1.4 制御入力の計算

本報告では、各時刻において、目標状態 X^{target} 与え、目標状態制御入力 B_{t+d} を与えることを続けることにより、目標軌道に沿ってロボットを移動させる。コントローラは入力 B_{t+d} を計算して、 X_{t+d+1} の状態を目的の状態 $X_{t+d+1}^{desired}$ にします X_{t+d+1} の確率分布の分散がしきい値 V^{th} より小さいという条件の下で。つまり、 B_{t+d} は、次の最適化問題を解くことによって取得されます。

$$\begin{aligned} & \text{minimize} |E(\hat{P}(X_{t+d+1}|B_{t+d}) - X_{t+d+1}^{target})| \\ & \text{s.t. } \text{Var}(P(X_{t+d+1}|B_{t+d})) \leq V^{th} \end{aligned}$$

ここで、 $E()$ および $\text{Var}()$ は期待値と確率分布の分散であり、 $\hat{P}(X_{t+d+1}|B_{t+d})$ は

$$\hat{P}(X_{t+d+1}|B_{t+d}) = \int_{X_{t+d}} P(X_{t+d+1}|X_{t+d}, B_{t+d}) P(X_{t+d})$$

2.2 ベイズ推定を用いた環境同定

コントローラは、現在の環境の $P(X_{t+1}|X_t, B_t)$ を正しく求めることが必要である。 $P(X_{t+1}|X_t, B_t)$ を識別する方法として機械学習が考えられるが、学習を行うには大量のデータが必要となる。そのため、ロボットが動作する環境が変化した場合にその環境を同定することが難しい。

本論文で提案する手法においては、ロボットのモデルを用いて環境同定を実行する。ロボットのダイナミクスは事前にモデル化されているとし、次式のように定義する。

$$X_{t+1} = f(X_t, B_t, \epsilon)$$

ここで、 f はロボットの次の状態を取得する関数である。 f には、ロボットの現在の状態 X_t 、現在の入力 B_t 、誤差項 ϵ の3つのパラメーターがあり、誤差 ϵ は環境を反映していると仮定する。したがって、 ϵ の確率分布を特定することにより、現在

の環境を特定する。

コントローラは、ロボット X_t および X_{t-1} の状態を取得することにより、現在の誤差項 ϵ_{t-1} を計算することができる。 ϵ_{t-1} は、以下の式を解くことによって得ることができる。

$$E(X_t) = f(E(X_{t-1}), B_{t-1}, \epsilon_{t-1})$$

取得した ϵ_t を使用して、 ϵ_t の分布 $P(\epsilon)$ を推定する。しかし、少数の ϵ_t しか取得しない場合、 ϵ_t の推定分布は不正確である場合が考えられる。不正確な推定を避けるために、事前分布 $P^{prior}(\epsilon)$ [13] を導入する。 $P(\epsilon)$ の分布は、

$$P(\epsilon) \propto P(\epsilon|\epsilon_0:t) P^{previous}(\epsilon)$$

ここで、 $P(\epsilon|\epsilon_0:t)$ は、同定対象となる ϵ の分布です。

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、誤差の観測結果が少ない、タスク実行の初期段階において、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。また、ベイズ推定による $P(\epsilon)$ の収束に必要な観測数も、事前分布の影響を受ける。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。[12]。そこで、本報告においても、遺伝的アルゴリズムにより、事前分布を定めるものとする。

本報告では、 $P^{prior}(\epsilon)$ 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： $P^{prior}(\epsilon)$ 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 $P^{prior}(\epsilon)$ 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. ケーススタディ：移動ロボットの場

3.1 移動ロボットのダイナミクス

まず、本報告において使用する対向二輪型移動ロボットのダイナミクスを示す。移動ロボットの状態は位置と角度で表される。簡単のため、位置 L_t と角度 θ_t で定義する

$$L_t = \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (3)$$

入力 B_t は次式のように左右の車輪の速度を指定する。

$$B_t = \begin{pmatrix} w_r \\ w_l \end{pmatrix} \quad (4)$$

ただし、 w_r は右車輪の速度、 w_l は左車輪の速度である。

以上の定義から、入力 B_{t-1} を用いた場合の位置と角度の変化は以下の式のようにして表される。

$$L_t = L_{t-1} + F(\theta_{t-1})B_{t-1} + \epsilon_{t-1}^{XY} \quad (5)$$

$$\theta_t = \theta_{t-1} + RB_{t-1} + \epsilon_t^\theta \quad (6)$$

$$F(\theta_{t-1}) = \begin{pmatrix} \frac{r}{2} \cos \theta_{t-1} & \frac{r}{2} \cos \theta_{t-1} \\ \frac{r}{2} \sin \theta_{t-1} & \frac{r}{2} \sin \theta_{t-1} \end{pmatrix} \quad (7)$$

$$R = \begin{pmatrix} \frac{r}{W} & -\frac{r}{W} \end{pmatrix} \quad (8)$$

ただし、 W は車輪間の距離、 ϵ_t^{XY} と ϵ_t^θ はそれぞれ位置と角度に発生する誤差である。

3.2 移動ロボットの誤差モデル

制御エラーの分布は、入力によって異なります。ロボットの動きが速くなると、エラー ϵ^{XY} も大きくなります。簡単にするために、速度に比例するように ϵ^{XY} をモデル化します。

$$\epsilon_t^{XY} = a_t^{XY} v_t \quad (9)$$

$$a_t^{XY} \sim N(\mu_{XY}, \sigma_{XY}) \quad (10)$$

ここで、 v_t はロボットの速度であり、 a_t はランダムな値です。このモデルでは、正規分布に従うように a を単純にモデル化します。同様に、 ϵ^{XY} は以下のようにモデル化する。

$$\epsilon_t^\theta = a_t^\theta w_t \quad (11)$$

$$a_t^\theta \sim N(\mu_\theta, \sigma_\theta) \quad (12)$$

w_t はロボットの角速度である。

3.3 移動ロボットにおける制御入力の計算

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、制御入力が入力される時刻 $t+d$ 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 $X_{t+d}^{ave} = (x_{t+d}^{ave}, y_{t+d}^{ave}, \theta_{t+d}^{ave})$ 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 $[x_{t+d}^{ave}, y_{t+d}^{ave}, \theta_{t+d}^{ave}]$ 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。

$$\begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} = \begin{pmatrix} \cos \theta_C & \sin \theta_C & 0 \\ -\sin \theta_C & \cos \theta_C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^{target} - x \\ y^{target} - y \\ \theta^{target} - \theta \end{pmatrix} \quad (13)$$

(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。

$$v = K_{t+d} \quad (14)$$

$$w = 2A_{t+d}K_{t+d} \quad (15)$$

ただし、

$$A_{t+d} = \text{sign}(e_x) \frac{e_y}{e_x^2} \quad (16)$$

$$K_{t+d} = \text{sign}(e_x) \frac{\alpha}{1 + |A_{t+d}|} \quad (17)$$

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。

$$w_{t+d}^{\text{right}} = \frac{v}{r} + \frac{Ww}{2r} \quad (18)$$

$$w_{t+d}^{\text{left}} = \frac{v}{r} - \frac{Ww}{2r} \quad (19)$$

制御入力 B'_{t+d} の決定後、2.1.4 節で示した方法により、コントローラが持つ事前分布をもとにロボットの速度を抑制する。

4. 実機実験

本章では、実機実験により、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。

4.1 評価環境

4.1.1 動作環境

評価環境においては、ロボットの位置、角度をカメラから取得する。コントローラは 30ms ごとにカメラからロボットの状態を受け取り、予測、制御入力の送信を行う。カメラは有線でコントローラと接続されており、カメラから得られるロボットの状態は 130ms 前のロボットの状態である。コントローラは無線ネットワークを経由してロボットに制御入力を送信し、遅延時間は一定ではない。

4.1.2 ロボットのタスク

本評価では、直線の往復運動をロボットのタスクとして指定する。実機実験では、スタートを点 (-35, -865) とし、点 (-145, 975) まで移動したのち折り返し、ゴールである点 (-35, -865) まで移動する。

タスク実行前にコントローラはキャリブレーションを行う。予め指定された入力を用いて回転運動を行い、予測される角度と実際に動作した後の角度の差を 10 回学習する。タスク実行時にはキャリブレーションで得られた誤差の情報をもとに予測・制御を行う。

4.1.3 ネットワーク環境

本評価では、大学院内の廊下において実験を行った。この環境では、車輪のスリップが生じにくいことが確認されている。ロボットとコントローラが使用する無線環境において生じるネットワーク環境を知るため、RTT を計測した。図 2 は各タイムスロットで送信した入力とロボットからの ACK をもとに計測した RTT を記録したものである。ほとんどの場合、RTT

は 100ms 以下であることがわかる。

本評価では、誤差が大きい環境かでも制御を行う。提案手法が正しく環境同定を実行できていることを確認するため、ネットワーク環境が悪化した場合を生成する。具体的には、コントローラが入力を送信する際 5% の確立で 400ms だけ待機して入力を送信する。図 3 は擬似的に遅延を生成した場合に各タイムスロットで生じた RTT を示した図である。

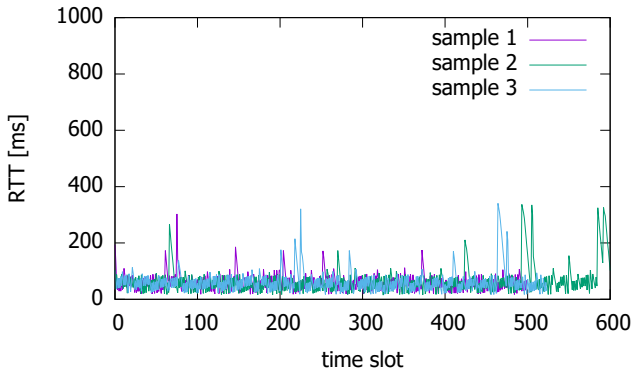


図 2: RTT (通常状態)

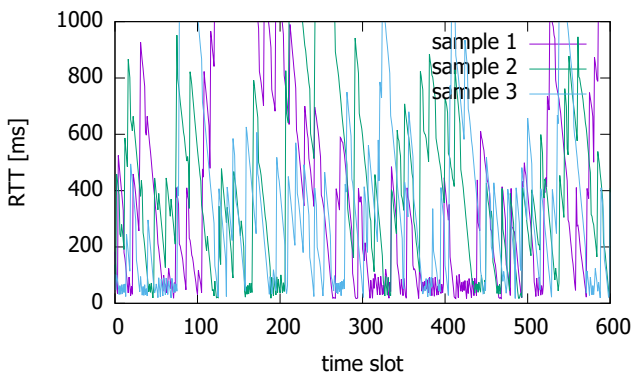


図 3: RTT (遅延生成時)

4.2 比較対象

本節では、以下の手法を比較する。

a) ベイズ的環境同定 (Bayesian)

提案手法を用いた制御である。コントローラは進化計算で得られた事前分布をもち、事前分布と観測された誤差を用いてベイズ推定を実行し、環境同定を行う。そしてコントローラは同定された環境をもとに入力の計算を実行する。

b) 統計的環境同定 (Statistic)

この手法は観測された誤差のみから環境同定を行う手法である。観測された誤差をもとに現在の環境を同定し制御入力を計算する。この手法との比較により、ベイズ推定を用いることの有効性を示す。

c) 固定誤差モデル (Fixed)

この手法は事前に定められた誤差モデルを使用する。評価においては、2つの固定誤差モデルを使用する。1つ目は通常状態での誤差のモデルを持つもの。もう一つは悪化したネットワーク環境の情報を持つものである。この手法との比較により、現在の環境を同定することの有効性を示す。

4.3 結果

図 5 と 4 は経路からの逸脱の最大値の累積補分布を示した

ものである。各手法を用いて 15 回ずつタスクを実行している。提案手法は現在の環境を正しく同定できているため、ネットワーク環境が悪化した場合でも、経路からの逸脱の最大値を抑えることができていることが示されている。環境同定が正しく実行できていることにより、ネットワーク環境が悪い場合は、大きな逸脱を回避するために速度を抑制している。一方、ネットワーク環境が悪い場合の事前情報を与えた固定誤差モデル以外の手法はネットワーク環境が悪化すると経路からの逸脱が大きくなっていることがわかる。これは、悪化したネットワーク環境を正しく同定できていないためであると考えられる。統計的環境同定は現在の環境を安定した環境であると認識する場合があるため、速度を上げてしまい、経路からのズレが大きくなっている。

図 7 と図 6 はタスク完了までの時間の累積補分布を示したものである。提案手法はどちらの場合においてもタスク完了までの時間を短く設定できていることがわかる。現在の環境を正しく同定することで、通常の状態では速度を上げて制御を実行する。また、ネットワーク環境が悪化すると速度を抑制して制御を実行する。他の手法では現在の環境を正しく同定できていない。通常のネットワーク状態について事前情報を与えた固定誤差モデルの手法では悪化したネットワーク環境でも速度を上げてしまい、目標地点への角度の調整に時間がかかっている。また、悪化したネットワークについての事前情報を与えた固定誤差モデルは必要以上に速度を抑制してしまい、タスク完了までの時間が大きくなっている。

5. おわりに

本論文では、少ない観測情報で現在の環境を同定することが可能なコントローラのフレームワークを提案した。提案手法は、事前分布とベイズ推定を使用して、現在の環境を特定できるように進化する生物のメカニズムをもとにしたものである。

本論文では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、コントローラで以下のような制御を実行する。ロボットから得られた観測値と送信した入力をもとに、コントローラは制御で生じた誤差を計算。観測された誤差をもとに、コントローラはベイズ推定を用いて現在の環境に関する誤差モデルを更新し、誤差モデルに基づいて制御を実行する。

提案手法において、誤差モデルの事前分布は、さまざまな環境下での進化によって取得していく。各世代で、コントローラによって行われたタスクの結果をもとに、事前分布を評価し、評価結果に基づいて事前分布のパラメータを進化を行う。

また、本論文では、対向 2 輪ロボットを用いてコントローラの実装を行った。実機実験を通して現在の環境を同定するコントローラを実装しロボットを適切に制御する。制御で得られた誤差のみから環境を同定する手法と比べ、目標経路からの逸脱とタスク完了までの時間を削減できることを示した。

本論文では、事前分布を得るにあたって、遺伝的アルゴリズムを用いたシミュレーションにより進化計算を行った。実際のロボットに適用する際には、動作テストの実行中に、実際の環

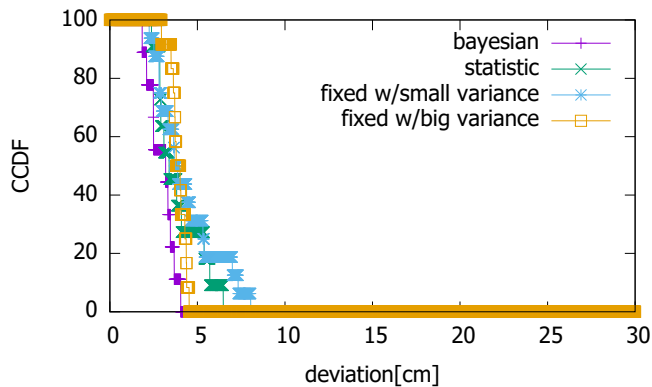


図 4: 通常時の経路からの逸脱の最大値

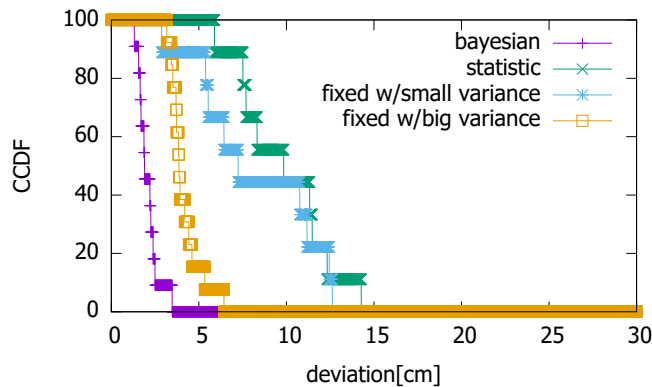


図 5: 遅延挿入時の経路からの逸脱の最大値

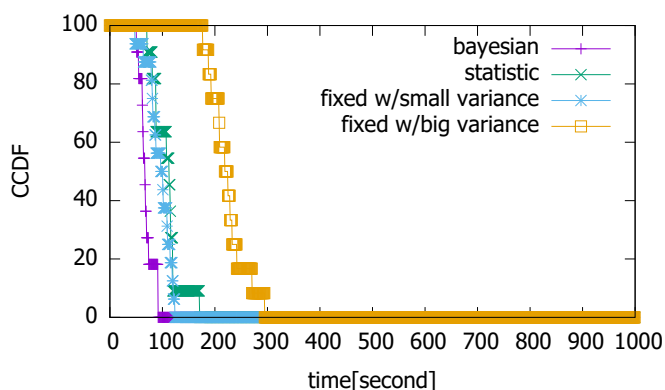


図 6: 通常時のタスク完了までの時間

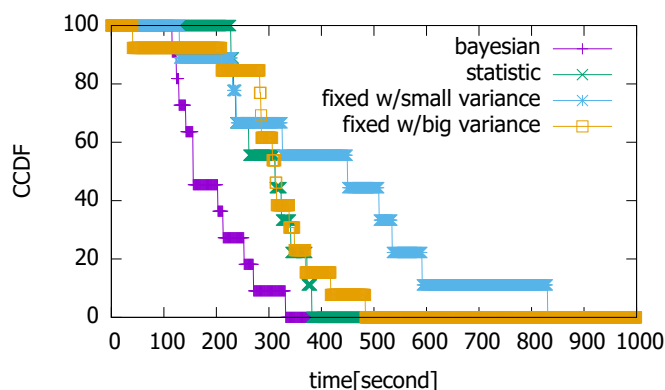


図 7: 遅延挿入時のタスク完了までの時間

境で発生した誤差に対処できる事前配布を取得することが必要である。このような方法を実現するためには、進化計算が実機実験によって実現できることを示すことが重要となる。

今後は提案手法を他のタスクや他のロボットに適用することを検討している。

文 献

- [1] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceeding of the IEEE*, vol.95, pp.138–162, Jan. 2007.
- [2] M.-F.R. Lee, F.-H.S. Chiu, H.-C. Huang, and C. Ivancsits, "Generalized predictive control in a wireless networked control system," *International Journal of Distributed Sensor Networks*, vol.9, no.12, p.475730, Jan. 2013.
- [3] M. Hoy, A.S. Matveev, and A.V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol.33, no.3, pp.463–497, March 2015.
- [4] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol.7, no.1, pp.1–17, 2019.
- [5] S. Yasuda and H. Yoshida, "Dynamic optimization of a remote control cycle for better responsiveness," *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society/IEEE*, pp.2986–2991 Oct. 2018.
- [6] Á. Cuenca, W. Zhan, J. Salt, J. Alcaina, C. Tang, and M. Tomizuka, "A remote control strategy for an autonomous vehicle with slow sensor using kalman filtering and dual-rate control," *Sensors*, vol.19, no.13, p.2983, 2019.
- [7] K. Nagatani, D. Endo, and K. Yoshida, "Improvement of the odometry accuracy of a crawler vehicle with consideration

of slippage," *Proceedings 2007 IEEE International Conference on Robotics and Automation/IEEE*, pp.2752–2757 April 2007.

- [8] J.-B. Mouret, "Micro-data learning: The other end of the spectrum," *ERCIM News*, vol.107, pp.18–19, Oct. 2016.
- [9] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol.104, no.1, pp.148–175, 2015.
- [10] R. Martinez-Cantin, N. De Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol.27, no.2, pp.93–103, 2009.
- [11] R. Martinez-Cantin, N. deFreitas, A. Doucet, and J.A. Castellanos, "Active policy learning for robot planning and exploration under uncertainty," *Robotics: Science and Systems*, vol.3, pp.321–328, 2007.
- [12] J.C. Ramírez and J.A. Marshall, "Can natural selection encode bayesian priors?," *Journal of Theoretical Biology*, vol.426, pp.57–66, Aug. 2017.
- [13] C.M. Bishop, *Pattern Recognition and Machine Learning*, C.M. Bishop, ed., NSpringer, 2006.
- [14] J. Hammersley, *Monte carlo methods*, Springer Science & Business Media, 2013.