

# テレプレゼンスロボットののための遠隔ナビゲーション手法

A remote navigation method for telepresence robots

加藤 由花\*1

Yuka Kato

\*1東京女子大学

Tokyo Woman's Christian University

This paper proposes a remote navigation method for low-cost telepresence robots. We realize smooth navigation of such robots by implementing the intelligence functions into cloud computing environments. Here, we adopt two strategies for the method. One is operating robots remotely using a virtual space, and the other is integrating the effect of network delay into the localization model. This paper focuses on the latter point, which is the delay-conscious model, and explains the result of the model design.

## 1. はじめに

遠隔地に存在する簡易型移動ロボットをインターネット経由で操作するサービスを対象に、クラウド型の遠隔ナビゲーション手法の研究を進めている [Kato 15]. ここでは、遠隔操作の対象となる実環境の忠実なモデルを事前に仮想空間として構築しておき、仮想空間と実空間のマッピングにより遠隔操作を実現する手法を提案している. 本稿では、この手法を拡張し、ネットワーク遅延を考慮した制御モデルを提案する.

一般に、ネットワークを介したロボットの遠隔操作は、移動のためのロボットの知能 (計測・制御など) をどこに持たせるかにより3つの形態に分類できる. 自律度の高いロボットと環境センシングを組み合わせた制御 (実環境に知能を持たせる形態)、オペレータによるリモコン型の制御 (遠隔地のオペレータの知能を利用する形態)、そしてクラウド上に知能を持たせる形態である. それぞれ、多様な環境への適用の難しさ、操作性の低さ、ネットワークを介した通信におけるネットワーク遅延・リソース不足等の問題が存在し、既存手法では、仮想空間を用いることで、操作性の問題を解決してきた. 本稿では、「オペレータ (人間) はネットワーク遅延が存在する環境であっても慣れにより遅延に順応していく」ことに着目し、ネットワーク遅延を制御モデルに組み込む.

## 2. 関連研究

ネットワークロボット、クラウドロボティクスに関する研究は盛んに行われており、例えば、ロボット用クラウド環境である RoboEarth [Riazuelo 15] や、リソース利用のスケジューリング問題に取り組んだ研究 [Wang 15] などがある. 多くの研究は、クラウド環境への機能のオフローディングを考察したものであり、ネットワークを介した制御についてはほとんど考えられてない. ネットワーク品質を考慮した研究としては、ネットワーク転送のモデル化を行っている研究 [Salmeron-Garcia 15], 遅延を考慮したフィードバック制御 [Penizzotto 15], QoS (Quality of Service) モニタリングを利用した制御 [Blumenthal 15] などが提案されている. しかし、観測を基にネットワークリソースのスケジューリングを行う手法がほとんどで、制御モデルへの遅延の組み込み、学習機構等は考慮されていない.

## 3. 提案手法

### 3.1 システムの構成

対象とする遠隔操作システムの構成を図 1 に示す. システムは、仮想空間、実ロボット、サーバの3つの部分から構成され、事前に構築しておいた仮想空間を利用し、遠隔地に存在する簡易ロボットを操作する. 実環境におけるロボット制御が必要となる位置推定、経路計画等の処理は、全てクラウド上 (サーバ上) に実装する.

サーバ上に実装される機能は、i) 仮想空間での移動命令を実ロボットの移動命令に変換する機能、ii) 実ロボットから観測データを収集する機能、iii) 収集した観測データから実ロボットの状態を推定する機能、iv) 仮想ロボットと実ロボットの状態のズレを補正する機能の4つである. ネットワーク遅延は、このうちの ii), iii) の部分に組み込む. 以下、まず遅延を考慮しないモデルについて 3.2 で説明し、次に遅延を考慮したモデルについて 3.3 で説明する.

### 3.2 手法の概要

遠隔操作におけるナビゲーションは、仮想空間と実空間でのナビゲーションを、サーバ上でマッピングすることにより実現する. 機能モジュールの構成を図 2 に示す.

#### 3.2.1 仮想空間でのナビゲーション

まず、仮想空間でのナビゲーションを考える. 仮想ロボットの状態を、時刻  $t$  におけるロボットの二次元座標  $(x_t^*, y_t^*)$  と姿勢  $\theta_t^*$  を用いて  $r_t^* = (x_t^*, y_t^*, \theta_t^*)^T$  と表わし、仮想ロボットの移動を、移動速度  $v_t^*$  と角速度  $\omega_t^*$  を用いて  $a_t^* = (v_t^*, \omega_t^*)^T$  と表わすことにする. すると、仮想空間では  $a_t^*$  が入力として与えられることになるが、このとき、仮想ロボットの状態に関する情報は完全に観測可能であるため、これが  $r_t^*$  に反映され、仮想ロボットの移動が実現する.

#### 3.2.2 実空間でのナビゲーション

次に、実空間でのナビゲーションを考える. 実ロボットの状態を、時刻  $t$  におけるロボットの二次元座標  $(x_t, y_t)$  と姿勢  $\theta_t$  を用いて  $r_t = (x_t, y_t, \theta_t)^T$  と表わし、実ロボットの移動を、移動速度  $v_t$  と角速度  $\omega_t$  を用いて  $a_t = (v_t, \omega_t)^T$  と表わすことにする. また、実環境における制御の集合を  $a_{1:t} = \{a_1, a_2, \dots, a_t\}$ , 観測の集合を  $z_{1:t} = \{z_1, z_2, \dots, z_t\}$  とする.

実ロボットは、不確実性の伴う実環境内を、サーバから送信される移動コマンド ( $a_t^*$  をサーバ上で実ロボット用コマンド

連絡先: 加藤由花, 東京女子大学数理工学科,  
yuka@lab.twcu.ac.jp

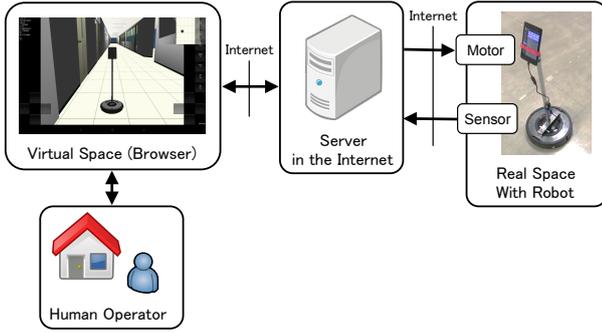


図 1: 対象とする遠隔操作システムの構成.

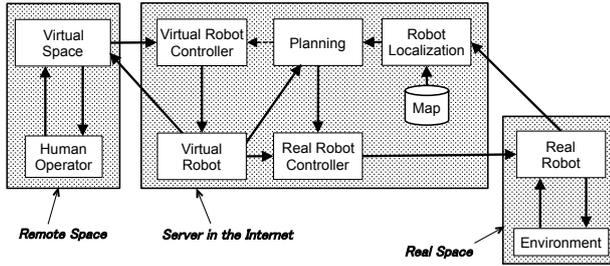


図 2: 機能モジュールの構成.

に変換したもの)に従って移動するが、実ロボットの真の状態を直接観測することはできない。そのため、移動誤差を考慮して、移動後の状態を推定することになる。今、サーバからの命令としての移動を  $\hat{a}_t = (\hat{v}_t, \hat{\omega}_t)^T$ ,  $\epsilon_t$  を移動の誤差 (正規分布等で与える) とすると,  $a_t = \hat{a}_t + \epsilon_t = (v_t, \omega_t)^T$  と書け, この  $a_t$  を使って  $r_t$  は以下の式で推定される.

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} v_t \cos \theta_{t-1} \\ v_t \sin \theta_{t-1} \\ \omega_t \end{pmatrix} \Delta t \quad (1)$$

これが、確率分布として定義される遷移モデル,  $p(r_t | r_{t-1}, a_t)$  になる。ここで、 $\Delta t$  は離散化した時刻の 1 ステップに相当する微小時間である。

### 3.2.3 マッピング機能

仮想ロボットは理想的な環境内を誤差なしに移動するが、実ロボットは不確実な環境内を移動するため、適切な補正を行わない限り、 $r_t$  と  $r_t^*$  のずれは拡大する。この補正を行うのがマッピング機能である。本システムでは、実空間のモデルとして仮想空間を事前に構築することから、これを位置推定のための地図として利用することを考える。ここでは、ランドマーク  $m_j$  の集合として地図  $m$  を定義し、 $m_j$  の位置を  $(m_{j,x}, m_{j,y})^T$  と表わす。そして、状態が  $r_t$  のとき  $z_t$  が観測される確率を、確率分布  $p(z_t | r_t, m)$  として与えておく (センシングの精度を正規分布等で与える)。これが、ここでの観測モデルになる。

この条件の下で、 $z_{1:t}$ ,  $a_{1:t}$ ,  $m$  から  $r_t$  を推定する。本稿では、実ロボットの初期状態は既知であり、状態の更新は再帰的に行われると仮定する。このとき、状態  $r_t$  の確率分布は、

$$\begin{aligned} p(r_t | z_{1:t}, a_{1:t}, m) &= \alpha p(z_t | r_t, m) \\ &\times \int p(r_t | r_{t-1}, a_t) p(r_{t-1} | z_{1:t-1}, a_{1:t-1}, m) dr_{t-1} \end{aligned} \quad (2)$$

となる。ここでベイズ理論とマルコフ性の仮定を用いた。その結果、 $r_t$  が逐次的に推定されることになる。

次に、これを  $r_t^*$  と比較し、差分を解消するための移動コマンドを生成し、実ロボットを移動させる。制御の振動を防ぐために、この補正は 1 回のみ行う。  $r_t$  と  $r_t^*$  の差が解消しない場合には、仮想ロボットの位置を更新する。

### 3.3 ネットワーク遅延の考慮

次に、前節で説明した位置推定モデルにネットワーク遅延を組み込む。このモデルでは、正確なランドマークは存在するが、観測はサーバ上で行うため、実際のランドマークの位置と観測時点での推定位置にずれが生じる。また、移動コマンドもサーバから送出されるため、送出時点と移動時点で時間のずれが生じる。オペレータは慣れによりこの補正を暗黙のうちに行っており、それまで経験した観測と制御のずれを考慮し、観測を遅らせ、制御を早めていると考えられる。

本稿ではこの状態を模擬するために、ping コマンド等を利用し、対象となるネットワークの遅延時間の平均値と分散をあらかじめ計測しておき、これを前節で定義した遷移モデルと観測モデルに組み込むことを考える。なお、ここでは平常時のネットワーク環境を前提とし、バースト的な性能劣化は考慮しない。また、クラウド環境の利用を前提に、遅延は最大でも平均 100ms 程度、計測結果は正規分布に従うと仮定する。これは、観測、制御の周期に対して、たかだか数周期程度のずれに相当する。

#### 3.3.1 遷移モデル

平均遅延時間に相当する、数ステップ前の移動命令から  $r_t$  を推定する。このとき、遅延時間は正規分布に従うと仮定し、各ステップごとの生起確率に応じた重み付き  $a_t$  の総和を、新たな  $a_t$  として (1) に代入する。  $t$  を連続時間と仮定すると、平均遅延時間を  $\mu$ , 分散を  $\sigma^2$  としたとき、

$$a_t = \frac{1}{\sqrt{2\pi\sigma}} \int \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot a_x dx \quad (3)$$

となる。ここで  $a_x$  は、時刻  $x$  における移動命令である。実際には時刻は離散化して実装する。離散化については後述する。

#### 3.3.2 観測モデル

平均遅延時間に相当する、数ステップ後の観測結果から  $r_t$  の事後確率を推定する。重み付けについては遷移モデルと同様に考え、重み付けされた新たな  $z_t$  により、 $p(z_t | r_t, m)$  を決定する。(3) と同様に、以下で求める。

$$z_t = \frac{1}{\sqrt{2\pi\sigma}} \int \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot z_x dx \quad (4)$$

#### 3.3.3 離散化

実装時には時間を離散化する必要がある。ここでは、平均時間に相当するステップ数を  $k$ , 分散に相当するステップ数を  $l$  とし、 $r_t$  を、 $\{a_{t-k-l}, a_{t-k-l+1}, \dots, a_{t-k}, \dots, a_{t-k+l}\}$  と、 $\{z_{t+k-l}, z_{t+k-l+1}, \dots, z_{t+k+l}\}$  を用いて推定する。ここで、 $k = \lfloor \mu/\Delta t \rfloor$ ,  $l = \lfloor 2\sigma/\Delta t \rfloor$  である。この結果、(2) の右辺は、 $p(r_t | z_{1:t+k+l}, a_{1:t-k+l}, m)$  に置き換えられ、これを逐次推定していくことになる。

## 4. まとめ

本稿では、簡易型移動ロボットの遠隔ナビゲーション手法として、ネットワーク遅延を組み込んだ新たな制御モデルを提案した。ここでは、遠隔操作時の遅延に慣れるオペレータの行動

---

を模擬し、事前に予測しておいた遅延モデルに従い、観測を遅らせ、制御を早める仕組みを組み込んだ。

今回、単純な遅延モデルを用いたが、各種パラメータの与え方、ステップ幅の決め方等、環境に適応したモデルの学習が必要であると考えられる。今後、学習機構を組み込んだモデルの構築、およびシミュレーション実験による手法の有効性検証を行っていく予定である。また、本稿では、遷移モデルと観測モデルへの遅延の影響を同等に考えているが、実際にこれらは非対称である（観測の方がデータ量が多い）。今後、これについても検討を行っていく予定である。

## 謝辞

本研究の一部は、JSPS 科研費 15K00137, 26330299 の助成を受けたものである。

## 参考文献

- [Blumenthal 15] Blumenthal, S., Hochgeschwender, N., Prassler, E., Voos, H., and Bruyninckx, H.: An Approach for a Distributed World Model with QoS-based Perception Algorithm Adaptation, in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS2015)*, pp. 1806–1811 (2015)
- [Kato 15] Kato, Y.: A Remote Navigation System for a Simple Tele-presence Robot with Virtual Reality, in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS2015)*, pp. 4524–4529 (2015)
- [Penizzotto 15] Penizzotto, F., Slawinski, E., Salinax, L. R., and Mut, V. A.: Human-centered control scheme for delayed bilateral teleoperation of mobile robots, *Advanced Robotics*, Vol. 29, No. 19, pp. 1253–1268 (2015)
- [Riazuelo 15] Riazuelo, L., et al.: RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach, *IEEE Trans. on Automation Science and Engineering*, Vol. 12, No. 2, pp. 432–443 (2015)
- [Salmeron-Garcia 15] Salmeron-Garcia, J., Inigo-Blasco, P., Rio, del F. D., and Cagigas-Muniz, D.: A Tradeoff Analysis of a Cloud-Based Robot Navigation Assistant Using Stereo Image Processing, *IEEE Trans. on Automation Science and Engineering*, Vol. 12, No. 2, pp. 444–454 (2015)
- [Wang 15] Wang, L., Liu, M., and Meng, M. Q.-H.: Real-Time Multisensor Data Retrieval for Cloud Robotics Systems, *IEEE Trans. on Automation Science and Engineering*, Vol. 12, No. 2, pp. 507–518 (2015)