

通信状況が不安定な環境下における予測に基づく 複数制御コマンド同時送信によるロボットの遠隔制御

大下 裕一[†] 安田 真也[‡] 熊谷 太一[‡] 吉田 裕志[‡] 金友 大[‡] 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,

[‡] {shinya-yasuda@, t_kumagai@, yoshida@, d-kanetomo@}nec.com

あらまし 本稿では、通信状況が不安定な無線ネットワークを経由してロボットを制御する方法について議論する。通信状況が不安定なネットワーク環境においては、遅延が急増したり、パケットがロスしたりすることがある。その場合、ロボットの制御コマンドが意図したタイミングで到達せず、意図とことなる状況で制御を投入してしまう。本稿では、上記の問題に対応するため、ロボットの状況を予測しながら、将来にわたる複数の制御コマンドを同時送信する手法について議論する。

キーワード 無線ネットワーク, ロボット, 制御

Remote control of a robot by sending multiple control commands based on prediction under unstable communication environment

Yuichi OHSITA[†] Shinya YASUDA[‡] Taichi KUMAGAI[‡] Hiroshi YOSHIDA[‡]
Dai KANETOMO[‡] 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,

[‡] {shinya-yasuda@, t_kumagai@, yoshida@, d-kanetomo@}nec.com

Abstract In this paper, we discuss how to control a robot via a wireless network with unstable communication conditions. In a network environment where communication conditions are unstable, delays may increase rapidly or packets may be lost. In that case, the control command of the robot does not arrive at the intended timing, and the control is applied in a situation different from the intention. In this paper, we discuss a method to simultaneously transmit multiple control commands by predicting the future states of the robot in order to address the above problems.

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。そのため、我々は、環境センサーを用いて、ロボットの無線ネットワーク経由での遠隔制御について研究してきた[2,3]。しかし、ロボットの低コスト化のためには、ロボットに搭載するコンピュータについても考える必要がある。高度な処理を行うためのコンピュータをロボット側に

搭載すると、当該コンピュータのコストのみならず、コンピュータを動作させるための大容量のバッテリーも必要となり、ロボット側のコストが高くなる。

そこで、本稿では、ロボットを制御するためのコマンドの計算も、ロボット側ではなく、制御コントローラが行うことを考える。制御コントローラは、環境センサーから情報を収集し、各ロボットが次にとるべき行動を計算し、ネットワーク経由でロボットにコマンドを送る。ただし、特に無線ネットワークでは、ネットワークの遅延が大きく変動しうる。この場合、コマンドが意図した時間にロボットに届かない。特に、多数の無線ネットワークが設置されている環境などでは、通信状況が不安定となり、遅延の急増やパケットロスが発生しやすい。

本稿では、このような通信状況が不安定な無線ネットワーク経由でも、ロボットを適切に制御する手法について検討する。

2. 無線ネットワークで発生する遅延

WiFiなどの無線ネットワークでは、アクセスポイントから通信相手となる機器までフレームを送信する。データフレームに対するACKフレームを受信することにより、通信相手にデータが届いたことを確認する。ACKフレームを受信できない場合、データフレームを再送する。複数の無線ネットワークが混在する環境下などで、フレームの衝突が多発する場合、データフレームを正しく受信できず、上述の再送が必要となる。これが無線ネットワークの遅延が変動する理由となる。図1に、我々の研究室内のWiFiで測定したRTTの時間変動を示す。図に示されるように、ネットワークの状況が悪化した際に、急激にスパイク状の遅延の増大が発生し、その後、遅延の増加が徐々に解消されるということが分かる。

そこで、本稿では、このような遅延の変動が発生しても、適切にロボットを制御することができる手法について検討する。

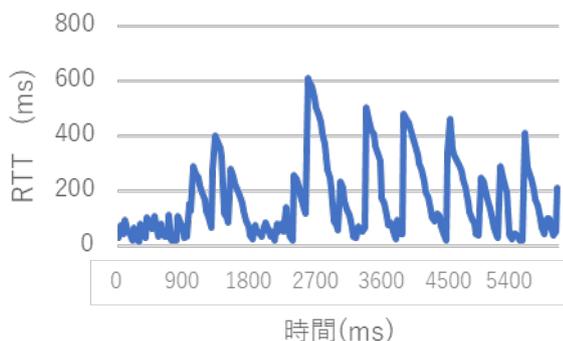


図1 RTTの変動（実測値）

3. 不安定なネットワーク経由でロボットを制御する手法

スパイク状に急増する遅延の影響としては、投入すべきタイミングでコマンドが到達しないという影響が考えられる。本稿では、上記の影響を緩和して、ロボットの制御を行う。

遅延の急激な増加時に、投入すべきタイミングでコマンドが到達しないことへの対応としては、あらかじめ、複数時刻分の制御コマンドを送っておくという方針が考えられる。これにより、遅延が急増してコマンドが到達しなかったとしても、以前に送られたコマンドを順に実行することにより、ロボットが意図しない状態に陥ることを防ぐことができる。

ただし、複数時刻分の制御コマンドをあらかじめ計算するためには、将来のロボットの状態を予測して、計算を行うことが考えられる。しかしながら、ロボットの状態は、予測対象時間が先になれば、より不確実になる。例えば、移動ロボットの場合、最初のロボットの向きに計測誤差が含まれていると、その誤差はわずかであっても、長時間直線移動をした結果、ロボットは大きく異なる位置に移動してしまう。

この問題に対して、我々は、ロボットの将来の状態を複数のエージェントを用いたシミュレーションにより求めた上で、ロボットの状況が不確実にならないような制御入力を定めるという方針をとる。このような制御は、ロボットの入力を求める際に、当該入力を与えた際に各エージェントが陥る状況の差が一定以内となるような制御入力を求めることにより達成できる。具体的な定め方は、移動ロボットを例にして4節で述べる。

4. Case Study：移動ロボットの制御

4.1. 移動ロボット

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 $v_1(t)$ 、左車輪の速度 $v_r(t)$ 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 $(x(t), y(t))$ 、角度 $\theta(t)$ を変化させる。 $v_1(t)$ 、 $v_r(t)$ 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。

$$x(t+1) = x(t) + \frac{v_1(t) + v_r(t)}{2} \cos\left(\theta(t) + \frac{v_r(t) - v_1(t)}{2W}\right)$$

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

また、角度は以下のように変化する。

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

ただし、 W は車輪間の距離である。そのため、 $v_l(t)$ 、 $v_r(t)$ 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。

4.2. 制御方式の実装

4.2.1. 構成

本稿の実験は、図2に示すように、移動ロボットと、ロボットの位置を把握するためのカメラ、制御コントローラで構成される。カメラと制御コントローラは有線で接続されており、コントローラは、安定した遅延でロボットが動作する環境の映像を取得し、ロボットの位置、向きを把握することができる。それに対して、コントローラ、ロボット間には不安定な無線ネットワークで接続されており、コントローラ・ロボット間の遅延は急増することがある。

4.2.2. パケットの構造

コントローラからロボットに送出するパケットには、シーケンス番号に加え、右車輪の速度、左車輪の速度を同時に送信するコマンド数分列挙した情報が含まれる。

それに対して、ロボットからコントローラへは、ACKパケットしか送らない。ACKパケットには、受け取ったパケットのシーケンス番号を記載する。ACKを受け取ることにより、コントローラは当該パケットを送出した時刻とACKを受け取った時刻を把握することができ、RTTを計測することができる。

4.2.3. ロボット側の動作

ロボット側で行う動作は、定期的な動作と、コマンド受信時の動作の2種類が存在する。

4.2.3.1. コマンド受信時の動作

制御コマンドを受信した際には、以下の動作を行う。

1. 到着したコマンドが受け入れ可能かを確認
パケットに付与されている有効期限を確認し、当該コマンドが受け入れ可能かを確認する。受け入れ可能な場合は、以降の処理を行い、受け入れ不可な場合は、パケットを破棄する。
2. 制御コマンドキューの更新
ロボットが持つ制御コマンドのキューを受信した複数のコマンドで置き換える。
3. ACKを返す。

4.2.3.2. 定期的な動作

ロボット側では、定期的に制御コマンドキューの先頭

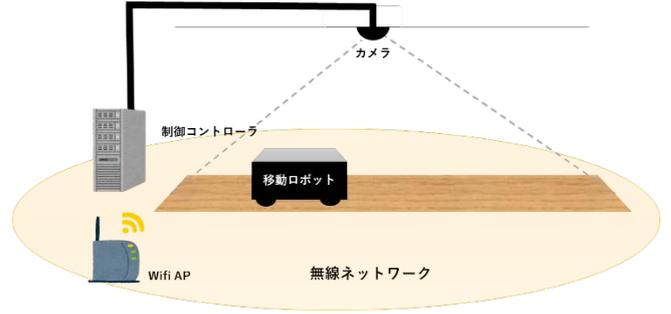


図2 実験の構成

のコマンドを取り出し、自身に投入する。本稿の実験では、この制御は33ms周期で行う。

4.2.4. コントローラ側の動作

コントローラ側では、定期的に、以下の動作を行うことにより、制御コマンドを計算し、ロボット側に送出する。

1. カメラ映像を取得し、ロボットに対応するエージェントの位置情報を更新する
2. これまでに取得したRTTをもとに、遅延を予測する
3. 複数タイムスロット分の制御コマンドを計算し、送出する制御コマンドを得る。
4. 制御コマンドパケットを送信する。

以降、上述の制御コマンドの計算の詳細について述べる。

4.2.4.1. 遅延の予測

コントローラ側では、ACKを受信する度にコントローラ・ロボット間のRTTを計測し、遅延の予測分布を更新する。本稿では、遅延の分布は、簡単のため、正規分布でモデル化する。そして、遅延の平均、二乗平均を観測されたRTTから加重移動平均で与える。すなわち、遅延は、正規分布 $N(\mu, \sigma)$ で与えられるものとし、そのパラメータ μ 、 σ は以下の手順で更新する。本手順では、新たにACKを受け取る度に、当該ACKに対応するコマンドを送出した時刻を確認し、ACKが返ってくるまでの時間 T を計算する。そして、

$$E_T = (1 - \alpha)E'_T + \alpha T$$

$$E_{T^2} = (1 - \alpha)E'_{T^2} + \alpha T^2$$

として、 T の平均 E_T 、 T^2 の平均 E_{T^2} を更新する。ただし、 E'_T 、 E'_{T^2} は更新前の E_T 、 E_{T^2} であり、 α は新たに観測したRTTに対する重みを表すパラメータである。そして、 $\mu = E_T$ 、 $\sigma = \sqrt{E_{T^2} - (E_T)^2}$ として正規分布のパラメータを更新する。これにより、ACKを受信しRTTが得られるたびに、遅延予測モデルを更新することができる。

4.2.4.2. ロボット位置の予測

本稿では、ロボットに対応するエージェントを複数配置し、各エージェントを動作させることにより、将来

のロボットの状態を予測する。各エージェントは、遅延予測モデルに従った遅延を生成し、生成された遅延時間後にコマンドを受信させる。これにより、遅延の揺らぎを考慮した様々な状態のロボットに対応するエージェントを準備できる。この各エージェントを動作させることにより、遅延の揺らぎを考慮したロボットがとりうる状態を把握することができる。

4.2.4.3. 制御コマンドの計算

制御コマンドを計算する際には、制御コマンドが到達する時刻の分布を遅延予測により得、その際のロボットの状況をエージェントシミュレーションより複数得る。得られたロボットの位置、向き of 平均値に対して、次の目標地点まで移動するために必要な左右の車輪速度を計算する。その際、ロボットが各タイムスロットで動作する速度の上限を定め、急な加減速を避けるとともに、ロボットが意図しない状態となることを避ける。

ここで、ロボットが意図しない状況とにならないための速度について検討する。ここでは、簡単のため、ロボットの位置を原点、ロボットが次に向かうべき点が $(v, 0)$ であるとして検討する。また、コマンドが投入されるタイミングのずれにより、ロボットの角度は θ ずれ、位置は (d_x, d_y) と原点からずれているものとする。この場合、ロボットが v 進んだ際の座標は、 $(v \cos \theta + d_x, v \sin \theta + d_y)$ となる。そのため、目標とする点 $(v, 0)$ からのずれを x 方向、 y 方向ともに D 以下にするためには、移動距離 v を

$$v \leq \min \left(\left| \frac{D + d_x}{1 - \cos \theta} \right|, \left| \frac{D - d_y}{\sin \theta} \right| \right)$$

とすることが必要となる。以降、 $v'_0 = \min \left(\left| \frac{D + d_x}{1 - \cos \theta} \right|, \left| \frac{D - d_y}{\sin \theta} \right| \right)$ とする。

各時刻の速度は、以下の条件を満たすように定める。

- 前の時刻のロボットの速度と比べた変化が閾値 a 以下
- a 以下の減速により、移動距離 d 以内で停止できる速度。ただし、 $d = \min(v'_0 - d^{tmp}, d^{target})$ であり、 d^{tmp} は、今回送出する最初のコマンドの到達時から現在計算中のコマンド到着までのロボットの移動距離、 d^{target} はロボットが停止する必要がある地点までの距離である。

これにより、急減速、急加速を防ぐことができるのみならず、同時に送信するコマンド列により、ロボットを v'_0 以上移動させることを防ぐことができ、ロボットが意図しない位置に移動してしまうことを防ぐことができる。

5. 評価

5.1. 概要

本稿では、4 節で述べた、移動ロボット、コントローラを実装して評価した。コントローラとロボットの状況を把握するためのカメラの間は 130ms の固定の遅延が入る。また、コントローラ、ロボット間は Wifi で接続されるものとした。ただし、ロボットの実験を行う環境でのコントローラ・ロボット間の遅延は安定していたため、より、ネットワークが不安定な環境下での評価を行うため、間に人為的に遅延を挿入するプログラムを挿入した。なお、挿入する遅延は、たくさんのアクセスポイントが設置され、ネットワークの状況が不安定な展示会会場で実測した値を用いた。

5.2. 比較対象

本稿では、以下の手法を比較する。

提案手法（同時送出コマンド数=50）：4 節で述べた手法。本評価では、同時に送出するコマンド数を 50 とし、 $D = 30 \text{ mm}$ として動作させた。

提案手法（同時送出コマンド数=3）：上述の手法と同じ手法であるが、同時と送出コマンド数を 3 として動作させた。

提案手法（同時送出コマンド数=50、速度抑制なし）：4 節で述べた手法において、ロボットの予測した状態の分布をもとにした速度の抑制を行わない手法。本手法と比較することにより、速度を抑制する効果を明らかにすることができる。

単一コマンド送出：各時刻において、提案手法と同様の方法で、コマンド到着時のロボットの状況を予測し、制御コマンドを投入する手法。ただし、毎回、制御コマンドは 1 つのみ送出する。

5.3. 実行したタスク

本稿では、楕円軌道を 10 周するというタスクを行った。この軌道を与えるにあたり、本実験では、楕円軌道上の 12 個の点の位置を目標点として与え、順に点の位置をなぞることにより、楕円軌道をたどる。ただし、本実験では、目標地点から半径 20cm 以内に到着した時点で、当該目標地点に到着したものとして、次の目標地点を定めて移動を開始するものとする。

本タスクでは、同一の経路を 10 周というタスクであるため、制御を正しく行うことができれば、10 周すべての経路が同じコースを通る。しかしながら、ロボットが意図と外れた制御結果となってしまう場合、周によって異なる経路を通る。

5.4. 評価結果

図 4 に評価結果を示す。図 4 より、提案手法では、

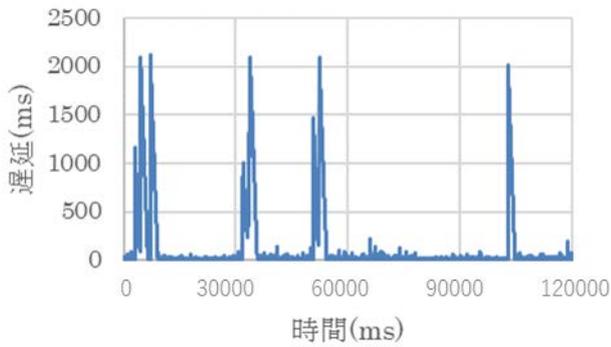


図3 実験に用いた RTT 計測値

楕円軌道を通っているものの、目標点よりも内側を通っていることが分かる。これは、本実験で与えたタスクが目標点に向かい、目標点から 20cm 以内の位置に到達すると、次の目標点に向かうことを繰り返すことにより楕円軌道を通るといったものであったためであり、目標地点の 20cm 手前で次の目標地点に向かうためにロボットの向きを変えるという挙動を取った結果、目標地点よりも内側の地点を経由した楕円軌道となっている。そのため、本実験においては、目標地点からの距離をもとにロボットの精度に関する考察を行うことは適切ではない。

そこで、以降では、ロボットの軌跡の目標地点からのずれではなく、ロボットの各週の軌跡間を比較することにより行う。コントローラ・ロボット間の遅延は時々刻々変動するため、各周によって異なる遅延が生じる。そのため、遅延の変動の影響を受ける制御方法であれば、各周によって異なる軌跡を取る。それに対して、遅延の変動の影響を受けない制御方式では、各周ともに同一の軌跡を取ると考えられる。

図 4(d)より、単一コマンドを送出した場合には、各週の軌跡が大きく異なり、楕円からもはずれた軌跡を通ることが分かる。これは、遅延が急増した際に、投入されるべきコマンドが到達しないことに起因する。本実験で用いた移動ロボットは、コマンドが投入されなければ、左右の車輪の速度を変化させることはない。その結果、意図した地点よりも大きく行き過ぎてしまう。結果として、楕円軌道から大きく外れた軌跡をたどる周回が生じてしまう、

図 4(a)から図 4(c)は、図 4(d)と比べると、楕円軌道から大きく外れることはなく、各周とも近い軌跡を描いている。これは、複数コマンドを同時に送出することにより、遅延が瞬間的に増大したとしても、その後実行すべきコマンドをロボット側が保持しており、目標とする経路から大きく逸脱することなく、ロボットを動作させることができるためである。

各週の軌跡のずれについて、より明確に評価するため

に、10 周の周回のうち平均的な軌跡を調べ、平均的な軌跡との距離について調べた。図 5 に平均的な軌跡、平均的な軌跡との差の求め方について示す。本評価では、カメラにより計測・記録されたロボットの各観測時刻における場所について、目標軌跡の楕円の中心となる点から当該地点の向きを計算する。そして、計算された向きが同一の θ となるロボット経路地点の座標の平均値 (x_{θ}, y_{θ}) を得、その値を平均的な軌跡において、中心地点からの向きが θ となった際に経路する場所として考える。ロボットが点 p に到達した場合の平均的な軌跡のずれは、その点の楕円の中心からの向き θ_p を求め、軌跡の平均となる点 $(x_{\theta_p}, y_{\theta_p})$ からの距離として定めた。

図 6 に軌跡の平均からの距離の累積補分布を示す。図 6 より、各週の軌跡のばらつきは、同時送出コマンド数が 3 の場合、同時送出コマンド数が 50 の場合と比べて大きくなっていることが分かる。これは、本実験で加えた遅延は最大 2 秒であり、同時に送出するコマンドが 3 つ (99ms 分のコマンド) のみでは、遅延増大時に蓄えたコマンド数が不十分であるためである。このため、複数のコマンドを送出する手法を用いる場合には、十分な数のコマンドを送出することが重要となる。

また、図 6 より、コントローラにおいて、予測された

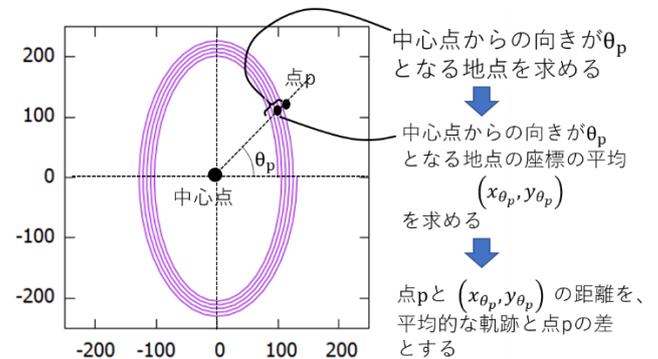


図5 平均的な軌跡との差の求め方

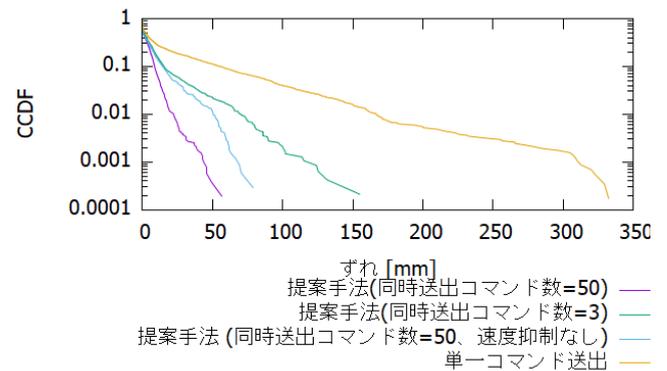


図6 平均的な軌跡との差の分布

エージェントの向きのばらつきをもとにした速度抑制を行った手法では、速度抑制を行っていない手法よりも、軌跡のずれがすくないことが分かる。これは、ロボットの向きの不確かさが大きい場合には、速度を抑制することが、ロボットが意図した軌跡からはずれることの防止につながるためである。つまり、経路を安定させるためには、予測されたロボットの状況にあわせて、移動速度を抑制することが重要であることが分かる。

6. まとめ

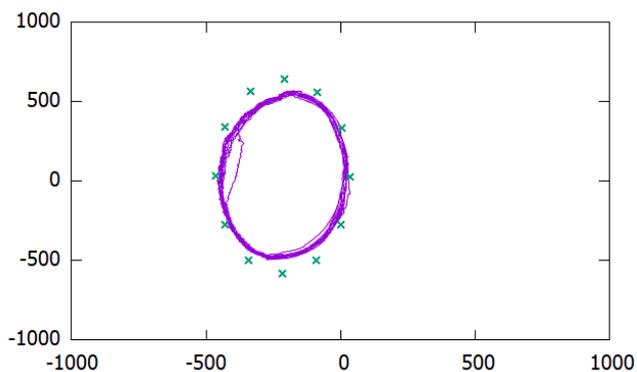
本稿では、通信状況が不安定な無線ネットワークを経由してロボットを制御する方法について議論した。通信状況が不安定なネットワーク環境においては、遅延が急増したり、パケットがロスしたりすることがある。その場合、ロボットの制御コマンドが意図したタイミングで到達せず、意図とことなる状況で制御を投入してしまう。本稿では、上記の問題に対応するため、ロボットの状況を予測しながら、将来にわたる複数の制御コマンドを同時送信する手法について議論した。そ

して、実機の実験を通して、複数のコマンドを送ること、速度を抑制することが高い精度でロボットを制御するのに有用であることを示した。

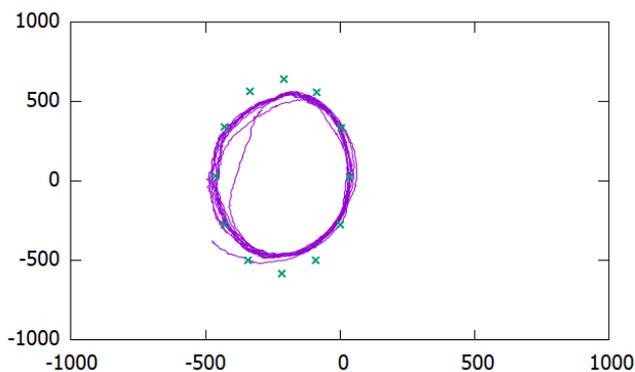
本稿では、目標軌跡に合わせて移動をするという単純なタスクのみを扱ったが、今後は、複数台のロボットが連携して動作するという、より複雑なタスクについても検証を行う予定である。

文 献

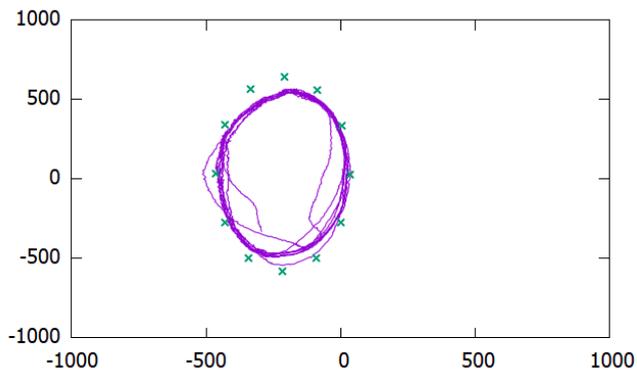
- [1] M.-F. R. Lee, et al, "Generalized predictive control in a wireless networked control system," *International Journal of Distributed Sensor Networks*, vol. 9, p. 475730, Jan 2013.
- [2] 安田真也, 大下裕一, 熊谷太一, 吉田裕志, 里田浩三, 村田正幸, "環境センサーを用いたロボットの無線遠隔制御の実機実験," *電子情報通信学会技術研究報告(IN2018-15)*, pp. 15-20, Aug. 2018.
- [3] 大下裕一, 安田真也, 熊谷太一, 吉田裕志, 里田浩三, 村田正幸, "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," *電子情報通信学会技術研究報告(IN2018-14)*, pp. 9-14, Aug. 2018.



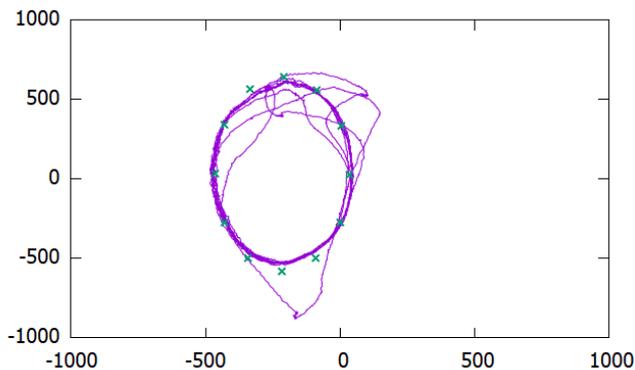
(a) 提案手法 (同時送出コマンド数=50)



(c) 提案手法(同時送出コマンド数=50, 速度抑制なし)



(b) 提案手法 (同時送出コマンド数=3)



(d) 単一コマンド送出

図4 移動ロボットの軌跡