

四脚型運搬補助ロボットの開発

伊藤 壮生, 浦池 隆文, 今岡 広一

Development of a Quadruped Robot for Conveyance Assistance

Soki ITO, Takafumi URAIKE, Koichi IMAOKA

抄 録

重量物の運搬作業には台車やリヤカーなどが使用されているが、段差や急傾斜の多い不整地ではこれらの器具を使用することが困難であることから、不整地での運搬作業は作業者にとって大きな負担となっている。一方、近年踏破性に優れた四脚ロボットの実用化が進んでおり、不整地での活用が期待されている。しかし、四脚ロボットは台車のように簡単な操作では運搬作業に活用できないため、既存の運搬機器を四脚ロボットに単純に置き換えることはできない。

本研究では、四脚ロボットを台車やリヤカーのように簡単に操作し、運搬補助に活用可能とすることを目的として、小型四脚ロボットの開発と、その制御技術の開発を行った。シミュレーションと開発した実機による動作検証の結果、四脚ロボットに押す・引くといった台車と同様の簡単な操作を行うことで、任意の方向に移動できることを確認した。

キーワード：四脚ロボット, シミュレーション, 運搬補助

Abstract

While carriages and handcarts are used to transport heavy loads, they are difficult to use on uneven terrain with many steps and steep inclines, placing a heavy burden on the operator. In contrast, quadruped robots which have been put to practical use in recent years, are expected to be utilized on uneven terrain because of their superior treading ability. However, they cannot be used for conveyance work as easily as carriages, so carriages and handcarts cannot be simply replaced by quadruped robots in conveyance work.

In this study, we developed a prototype of a small quadruped robot and control technology for the purpose of using it as a conveyance assistance by making it as easy to operate as a carriages and handcarts. As a result of operation verification using the simulation and actual prototype, it was confirmed that the robot can be moved in any direction by performing simple operations such as pushing and pulling on the quadruped robot.

KEY-WORDS : Quadruped Robot, Simulation, Conveyance Assistance

1. はじめに

重量物の運搬作業には台車やリヤカーなどが使用されているが、段差や急傾斜の多い不整地ではこれらの器具を使用することが困難であることから、不整地での運搬作業は作業者にとって大きな負担となっている。一方、近年踏破性に優れた四脚ロボットの実用化が進んでおり、車輪やクローラなどの一般的な移動機構を使用することが困難な不整地での活用

が期待されている。しかし、四脚ロボットは台車のように簡単な操作では運搬作業に活用できないため、既存の運搬機器を四脚ロボットに単純に置き換えることはできない。

ロボットで人間の補助を行うシステムとして、人とロボットの協調制御が提案されている¹⁾。このようなロボットシステムは自動化を目的とした場合と比較し、簡潔なシステム構成で実現できることが特徴であり、一般的にロボットや機械の導入が困難である不整地環境での利用に適していると考え

事業名：経常研究

課題名：不整地運搬補助のための人間協調四脚ロボットシステムの研究

られる。

本研究では四脚ロボットを活用し、台車やリヤカーのように簡単な操作で運搬補助に活用可能とすることを目的として、小型四脚ロボットと制御技術の開発、およびシミュレーションと実機による動作検証を行ったので報告する。

2. 開発したロボット

本研究では全長55cm、全幅44cm、全高36cm、重量12kgの小型四脚ロボットを開発した(図1)。1脚あたり3個の関節を有し、合計12個の関節駆動用モータを搭載することで前後左右への移動と旋回を可能にした。脚部を構成する部材は軽量化と低コスト化のために3Dプリンタを用いて製作した。ロボットの制御に必要な姿勢や速度などの計算は、モータに内蔵された角度センサと6軸センサ(SBG SYSTEM社製 Ellipse-E)を組み合わせて行った。電源(リチウムポリマーバッテリー)や制御プログラムを実行するためのシングルボードコンピュータ(Raspberry Pi4B)を本体に内蔵することにより、ロボット単体で歩行が可能な構成とした。

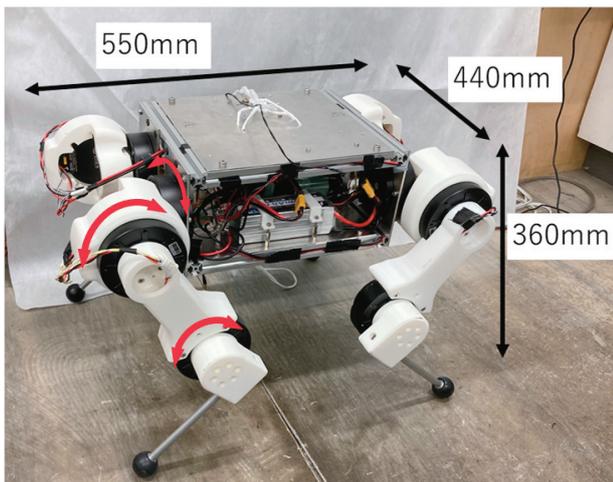


図1 開発した四脚ロボット

3. 歩行制御システム

四脚ロボットによる運搬補助を安全に行うためには、動作中の揺動を小さく抑えることや、押す・引くなどの外力に対しても安定性を維持する必要がある。このため、制御アルゴリズムにはKimらによるMPC(Model Predictive Control)とWBIC(Whole Body Impulse Control)のハイブリッド制御²⁾を採用した。この手法は、動力学にもとづいた各関節のトルク制御を行うことで、外力に対して安定性を維持したまま動作することが可能といった特徴がある。歩行制御システムの構成を図2に示す。本制御システムでは、目標となる速度、旋回速度、高さといった歩行指令を与えると、まず後述する歩容生成アルゴリズムにより脚の目標位置など歩行に

必要な脚の動作軌道の生成を行う。次にMPCとWBICによる制御アルゴリズムによって各モータへのトルク指令値等を計算する。また、制御に必要な胴体速度など、ロボットに搭載されているセンサから直接得られない状態を計算するため、カルマンフィルタによる状態推定を行う。歩行制御システムはROS(Robot Operating System)のパッケージとして開発することで、シミュレータの利用やセンサの追加などの改良を容易に行うことを可能とした。

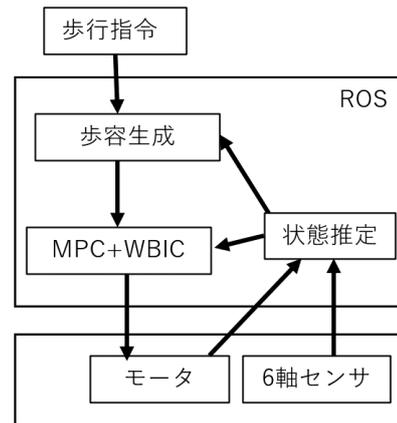


図2 歩行制御システム

3.1 歩容生成アルゴリズム

本研究で開発した歩行制御システムでは、事前に定めた脚の動作順序とタイミングにもとづいて歩容生成を行う。四足歩行の場合、ロボットの脚は地面に接触している支持脚と地面から離れている遊脚の2つの状態を一定の周期で繰り返す。このとき、支持脚期の割合や、脚ごとの遊脚期となるタイミングを変えることにより様々な歩容を生成することができる(図3)。本研究では、歩行速度と安定性のバランスに優れたトロット歩容を採用した。トロット歩容では対角にある脚のペアをもう片方のペアと半周期ずらして支持脚期と遊脚期を交互に繰り返す。歩行1周期あたりの時間は0.5秒とし、支持脚期と遊脚期の期間の比率を6:4とすることですべての脚が地面に接触する期間を確保し、歩行時の安定性向上

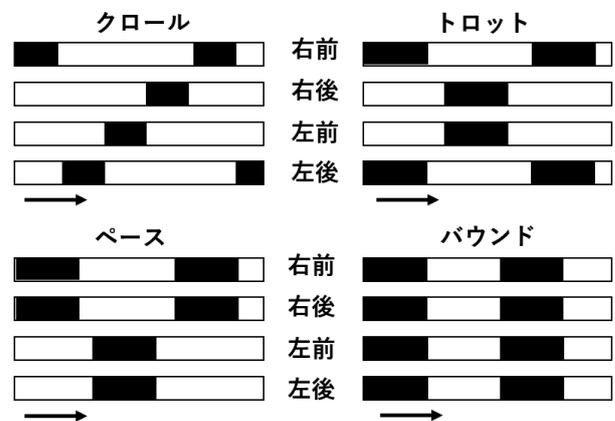


図3 歩容の種類による遊脚期のタイミングの違い
黒：遊脚期、白：支持脚期

を図った。また、遊脚の軌道は歩行指令とロボット胴体の速度から算出し、脚が地面から離れる瞬間と地面に接地する瞬間の速度が0となるようにすることで脚の着地による衝撃を小さくし、滑らかな歩行を可能とした。

3.2 トルク制御アルゴリズム

3.1節で生成した歩容からMPCとWBICを用いて関節駆動用モータへの指令値の計算を行う。MPCは未来の状態を予測しながら最適化計算を行うことで、目標値に追従するようにフィードバック制御を行う手法である³⁾。MPCを歩行制御に適用することで歩容に応じた最適な制御を実現することが可能である。一方でMPCは処理に時間がかかるため四脚ロボットの運動モデルを簡略化する必要があり、その結果、運動モデルと実物の運動には誤差が生じる。そこで、WBICによりMPCの計算時にモデルの簡略化によって生じる誤差を修正し、最終的なモータへの指令値を計算する。本節ではMPCとWBICのアルゴリズムについて簡単に説明する。

MPCでは計算を簡略化するため、ロボット脚の質量を無視した単一質点モデルとして扱う(図4)。単一質点モデルはロボットのすべての質量が胴体の重心に集中していると仮定したモデルであり、質点の運動と支持脚の位置のみを考慮する。歩行指令に従った動作を実現するためには、支持脚で地面を押し出すことで得られる地面反力を変化させる。単一質点モデルについて未来の状態を予測するため下記の離散状態方程式を考える。

$$x_{t+1} = A_t x_t + B_t u_t \quad \text{式(1)}$$

状態変数 x はロボットの運動状態を表すオイラー角、角速度、位置、速度からなるベクトルである。 A と B はロボットの姿勢と支持脚の位置を変数とした関数になっている。また、添え字の t は時間ステップを表す。未来の状態 x_{t+n} は現在の状態である x_t と地面反力 $u_t \cdots u_{t+n-1}$ を用いることで式(1)から再帰的に計算することが可能である。例えば $t+2$ のときの状態 x_{t+2} は下記のように計算する。

$$x_{t+2} = A_{t+1} A_t x_t + A_{t+1} B_t u_t + B_{t+1} u_{t+1} \quad \text{式(2)}$$

このように $t+1$ から $t+n$ までの未来の状態 $x_{t+1} \cdots x_{t+n}$ を計算し、歩行指令から求められるロボットの理想的な状態と可能な限り一致するように地面反力 $u_t \cdots u_{t+n-1}$ の最適化計算を行う。最適化計算を行う際に、地面反力の最大値や地面の摩擦係数を考慮して制約条件として与えることで、モータの定格トルクを超えない範囲で地面反力を計算することや、地面と脚が接触した際の滑りを防ぐことができる。

次にWBICのアルゴリズムについて説明する。WBICではMPCの計算時に無視した関節の状態などを考慮し、モータ

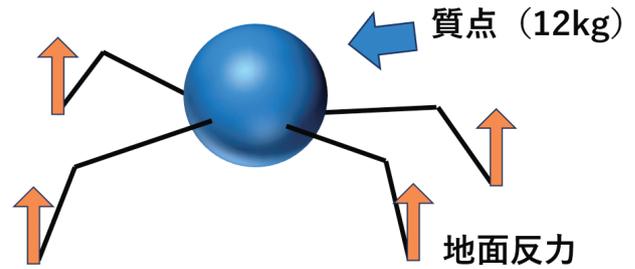


図4 単一質点モデルのイメージ
(単一質点と足の位置、地面反力のみを考慮する)

への駆動指令を求める。関節の状態を含む四脚ロボットの運動方程式は式(3)で表される。

$$M(q)\ddot{q} + b(q, \dot{q}) + g(q) = \tau + J_c(q)^T u \quad \text{式(3)}$$

q はロボット胴体の位置・姿勢、関節角度からなる一般化座標ベクトル、 M は慣性力、 b は遠心力とコリオリ力、 g は重力、 τ は関節トルク、 J_c はロボットと地面の接触点に関するヤコビ行列、 u は地面反力を表す。 $M(q)$ 、 $b(q, \dot{q})$ 、 $g(q)$ 、 $J_c(q)$ の各行列はロボットの状態により変化する関数であり、各行列の成分を高速に計算するためオープンソースで公開されているRBDL⁴⁾を使用した。式(3)について、関節トルク τ には胴体位置・姿勢に関する成分が含まれるが、それらは駆動軸ではないため常に0とする必要がある。よって胴体の運動のみに着目すると、四脚ロボットの運動は下記の制約条件を満たす必要がある。

$$[I_6 \ 0_{12}] (M(q)\ddot{q} + b(q, \dot{q}) + g(q)) = [I_6 \ 0_{12}] J_c(q)^T u \quad \text{式(4)}$$

I_6 は 6×6 の単位行列、 0_{12} は 12×12 の零行列である。

関節トルクを計算する手順としては、まず3.1節で生成した歩容に追従するようにPID制御則と逆運動学計算によって目標加速度 \ddot{q} を計算する。次に式(4)や摩擦などの制約条件を満たすよう、最適化計算により目標加速度 \ddot{q} とMPCで求めた地面反力 u を修正する。修正された目標加速度 \ddot{q} と地面反力 u を式(3)に代入することにより関節トルク τ を計算する。

MPCとWBICは計算量の問題から制御周期を十分に短くすることができず不安定になってしまう場合がある。そこで高速なフィードバックにより安定性を高めるため、サーボモータ内部でPD制御を行う。PD制御によって計算されたトルク出力をWBICのトルク出力に加算し、最終的な関節トルクとして出力する。PD制御はロボットの歩行指令と遊脚軌道から逆運動学計算によって計算した関節角度と関節角速度を目標値として行う。各指令値はCAN通信によってシングルボードコンピュータから関節駆動用モータへ送信する。

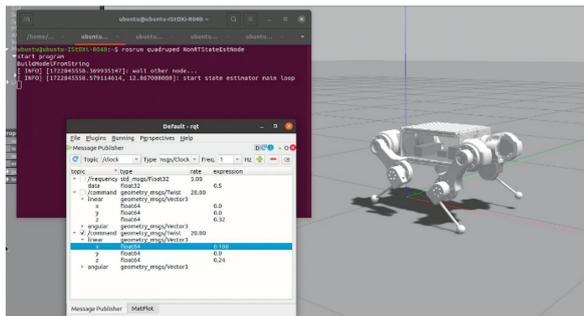


図5 Gazeboによるシミュレーションの様子



図6 段差を乗り越える様子

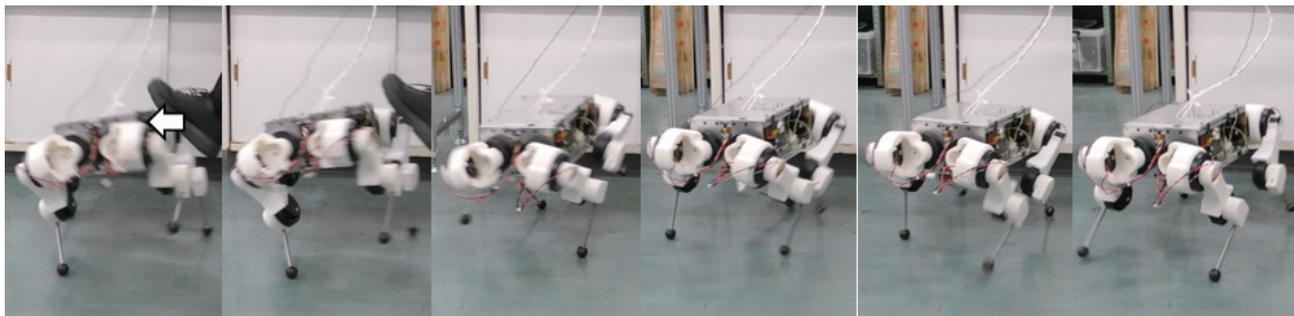


図7 衝撃に耐えて転倒を回避する様子

3.3 状態推定

3.2節で説明したトルク制御アルゴリズムを実行するためには関節の角度、角速度に加え、胴体の位置、速度、姿勢、角速度が必要となる。このうち関節の角度、角速度に関してはサーボモータに搭載されている角度センサで直接計測することが可能である。胴体の角速度についても6軸センサで直接計測することが可能であり、姿勢については6軸センサが有する機能により単体で取得することが可能になっている。しかし、ロボット胴体の位置と速度については直接測定することが困難であるため、Bledtらによる実装⁵⁾を参考にカルマンフィルタによる推定を行った。

カルマンフィルタは複数のセンサ等から得られる情報を組み合わせることで最適な推定値を得るアルゴリズムである。Bledtらの手法では胴体の運動に関して加速度を積分することによって得られる位置・速度の情報と関節角速度から支持脚の順運動学を計算することで得られる速度の情報を利用する。これらを組み合わせることでカルマンフィルタによる計算を行うことで、胴体の位置、速度について誤差が最も小さくなる最適な推定値を得ることが可能である。

3.4 歩行制御アルゴリズムの動作検証

実機の製作に先立ち、シミュレーションにより歩行制御アルゴリズムの動作検証、およびロボット各部の設計値と制御パラメータの調整を行った。シミュレーションはオープンソースのロボットシミュレータであるGazebo⁶⁾を使用して行った(図5)。製作する実機と同じ物理特性を有するモ

デルをシミュレータ上に構築し、3.1～3.3節の歩行制御システムによる歩行動作を行わせたとこ、前後左右への移動や旋回などの歩行指令に応じた動作が可能であることを確認した。シミュレーション結果をもとに実機を製作し、動作検証を行った。その結果、シミュレーションと同様に歩行指令に応じた動作が可能であることを確認するとともに、高さ50mm程度の段差の歩行(図6)や、側部からの強い衝撃に対して自動的に姿勢を保持して転倒を回避できることなどを確認した(図7)。

4. 操作力によるロボットの操作

台車など同様の操作でロボットを意図した方向へ移動させるための制御手法を検討し、シミュレーションによる動作検証を行った。この際、ロボットを移動させるために作業者がロボットに加える力を操作力と呼ぶこととする。本研究では図8に示すように作業者とロボットで荷物の両端を支え合

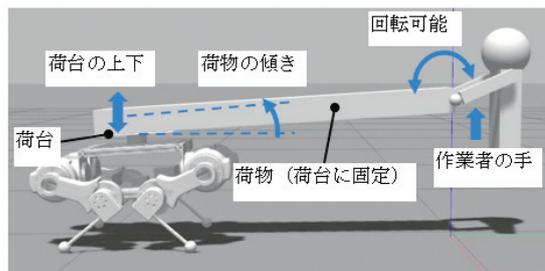


図8 シミュレーションによる荷台高さ調整実験の様子

うことで、ロボットに運搬作業を補助させる方法を想定した。シミュレーションの条件として、荷物はロボットに取り付けた荷台に固定され、かつ作業者と一定の距離を保ち、さらに図8に示すように作業者の手で回転支持されているものとした。また、作業者は前後と上下の移動のみ可能とした。

4.1 ロボットの高さの調整

作業者と荷台の高さが異なる場合は荷物が傾き（図8）、ロボットも同様に傾いた姿勢となる。荷物を水平に保ち安全に運搬作業を行うためには、ロボットの荷台高さを自動調整する必要がある。このため、一定以上の傾きを検知した際にロボットの姿勢に応じて荷台の高さを上下させることにより、荷物を水平に保つこととした。図9にシミュレーションによる動作検証の結果を示す。作業者が荷物を持つ「手の高さ」を変化させた際に、荷物が水平になるようにロボットの「荷台高さ」が自動的に調整されることを確認した。

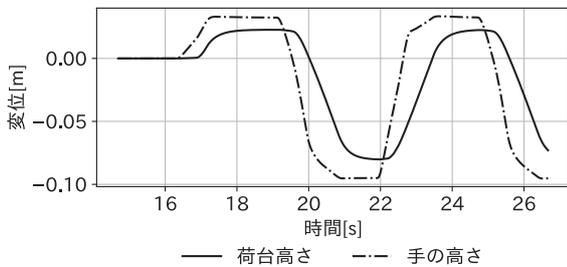


図9 高さの自動調整の実験結果

4.2 ロボットの前後の移動

四脚ロボットが外力に対して安定であることを利用して、台車やリヤカーなどと同様に、押す・引くなどの操作力に応じて四脚ロボットを受動的に移動させる動作の検証を行った。関節のトルク計算を行う際、水平方向の位置に関する目標値との誤差を無視するように制御パラメータを設定し、さらに速度指令を0とすることにより、操作力を加えた方向に受動的にロボットを歩行させることが可能である。また、ロボットの停止と歩行を操作力によって切り替えるため、足先の胴体からの相対位置を常に監視し、相対位置が変化してい

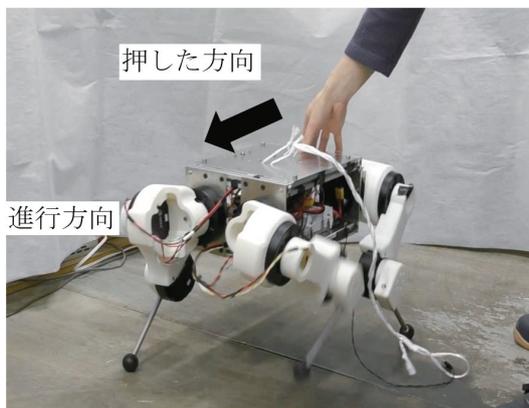


図10 ロボットを押して操作する様子

ない時は歩行を停止、作業者が操作力を加えることで相対位置が変化した時は歩行を開始する機能を実装した。シミュレーションによる検証を行い、作業者が移動することでロボットに操作力が加わると歩行を開始し、作業者が停止するとロボットも停止することを確認した。

実機を用いた実験においても図10のように操作力を加えたところ、ロボットを前後左右・旋回など任意の方向に操作できることを確認した。この際、前方へ移動させるために必要な操作力は約12Nであった。

4.3 操作力の大きさに関する検証

質量の大きな荷物を運搬する場合、荷物に合わせて四脚ロボットのサイズを大きくする必要があり、その際に操作力の増加が予想される。そこで、ロボットの大きさや質量と操作力の関係をシミュレーションで検証した。検証の結果、ロボットの操作に必要な操作力はロボットと荷物を合わせた質量に概ね比例して増加することがわかった（図11）。

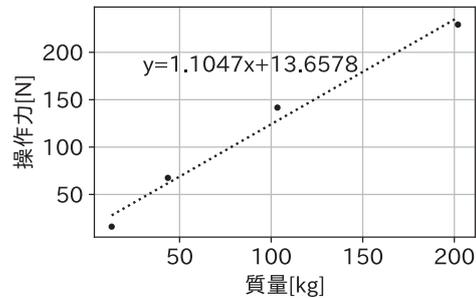


図11 ロボットと荷物の質量と操作力の関係

5. 操作力の低減

ロボットを移動させるための操作力はロボットと荷物を合わせた質量に比例することから、重量物を運搬するためにロボットを大型化すると、非常に大きな操作力が必要になることがわかった。このため、操作力を低減する仕組みが必要となる。

そこで図12に示すように、ロボットと荷台の間に操作力を計測するための力センサを設置し、計測した操作力をもとに能動的にロボットを歩行させる制御を検討した。操作力をもとにした歩行制御の指令値の計算には、慣性や粘性といった機械インピーダンスを制御するための手法であるアドミッタ

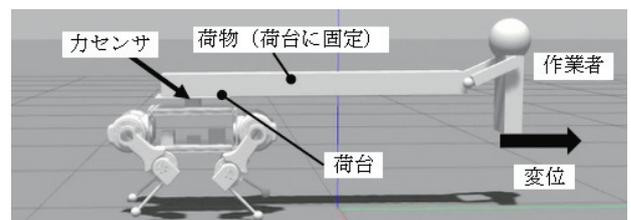


図12 操作力によるロボットの前後移動のシミュレーション

ンス制御を使用した。アドミッタンス制御では、式(4)の運動方程式により、目標とする特性を持つ仮想物体に力を加えた場合の運動を計算し、これと等しい運動をするように制御指令値を決定する。

$$M_d \ddot{x} + D_d \dot{x} + K_d x = F_{ext} \quad \text{式(5)}$$

ただし、 M_d , D_d , K_d は目標の慣性、粘性、剛性であり、 \ddot{x} , \dot{x} , x は仮想物体の加速度と速度、位置、 F_{ext} は外力を表す。

本研究 F_{ext} ではに力センサで計測した操作力を代入し、式(4)を計算することで得られる仮想物体の速度を歩行制御アルゴリズムへ目標速度として入力することでロボットを制御した。仮想物体の慣性および粘性を小さく設定するほど小さい操作力でロボットを動かすことが可能となるが、歩行の揺動による影響を受けやすくなる。そのため、ロボットの動作が不安定にならず、操作力を可能な限り小さくすることができるパラメータを試行錯誤によって決定した(表1)。

シミュレーションによる検証を行い、上記制御を組み込んだ際の操作力の変化を確認した。図13に示すように「力センサあり」の場合、ロボット操作に必要な力が「力センサなし」の場合と比較して平均で約1/3となることを確認した。

表1 アドミッタンス制御パラメータ

慣性 M_d	30kg
粘性 D_d	1.0N・s/m
剛性 K_d	0N/m

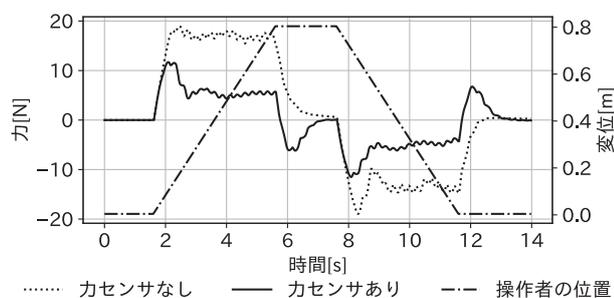


図13 カセンサの有無による操作力の違い

6. おわりに

本研究では台車やリヤカーの使用が困難な不整地において、簡単な操作で運搬作業の補助が可能な四脚ロボットの実現を目指し、小型四脚ロボットを開発するとともに、作業者の力によって簡便にロボットを操作するための制御技術の開発を行った。さらに、作業者がロボットに加える操作力を力センサで計測し、制御に活用することでロボットの操作に必要な操作力を小さく抑えることができることをシミュレーションによる動作検証を通して確認した。

今後は斜面を想定した際の操作性向上などについて検討を進める。また、本研究で得られたロボット制御技術やシミュレーション等を利用した開発手法について、技術支援等を通じて普及を図る予定である。

参考文献

- 1) 田窪朋仁・荒井裕彦 他:「人とロボットによる長尺物の協調運搬: 仮想非ホロノミック拘束による水平面内の制御手法」, 日本機械学会論文集C編, 66巻, 648号, pp.2677-2684, (2000)
- 2) KIM, Donghyun, et al., “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control.”, *arXiv preprint arXiv:1909.06586*, (2019)
- 3) 大嶋正裕, 小河守正:「モデル予測制御-I: 基礎編: 発展の歴史と現状, システム/制御/情報」, 46巻, 5号, pp.286-293, (2002)
- 4) Felis, M.L., “RBDL: an efficient rigid-body dynamics library using recursive algorithms.” *Auton Robot* 41, pp.495-511 (2017)
- 5) Bledt, Gerado et al. “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot.” *IEEE International Conference on Intelligent Robots and Systems*, October 2018, Madrid Spain, Institute of Electrical and Electronics Engineers, January, (2019)
- 6) N. Koenig, A.Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,”, in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2149-2154, (2004)