

脚型ロボットによるつくばチャレンジ 2020 への取り組み

千葉工業大学

入江清 鈴木太郎 原祥堯 吉田智章 友納正裕 西村健志 大和秀彰 清水正晴

A Legged Approach towards the Tsukuba Challenge 2020

Kiyoshi Irie, Taro Suzuki, Yoshitaka Hara, Tomoaki Yoshida, Masahiro Tomono, Takeshi Nishimura,
Hideaki Yamato, Masaharu Shimizu (Chiba Institute of Technology)

Abstract : This paper presents an autonomous navigation system based on Laikago, a commercially available quadruped robot. We mounted a 3D Lidar onto the robot and ported our software suite developed for wheeled robots. The robot system successfully completed autonomous runs over several hundred meters in our university campus.

1. はじめに

つくばチャレンジ 2020 は、当初計画されていた、つくば市研究学園駅付近での実験走行会は中止となり、各チームが独自の取り組みを行うこととなった [1]。筆者らのチーム fuRo アウトドア部（以下、本チームと表記する）は、脚型ロボットによる大学キャンパス内での自律走行システムの実現を目標として研究開発に取り組んでいる。本稿では、執筆時点（2020 年 10 月）までの成果と今後の計画について述べる。



Fig. 1: 開発中のロボットシステムの外観

2. システム概要

開発したロボットシステムの外観を図 1 に示す。本チームでは Unitree 社の Laikago を土台とし、センサと自律走行のためのソフトウェア群を追加で搭載している。

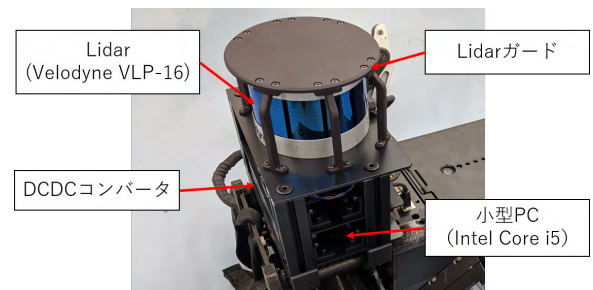


Fig. 2: センサ搭載部の拡大図

2.1 Laikago の仕様

Laikago は全長 55cm、全幅 35cm、高さ 60cm であり、重量は約 25kg、最大過半重量は約 9kg である。最大 0.8m/s で歩行することができ、連続動作時間は約 2 時間である。ロボット内部には、各脚を制御するための組み込みコンピュータ、IMU が内蔵されており、外部から通信が可能である。

2.2 ハードウェア改変点

Laikago に地図作成・自己位置推定・障害物検出用として 3D Lidar (Velodyne VLP-16) を追加で搭載した。センサ搭載部の拡大写真を、図 2 に示す。ロボットの前部にアルミフレームで治具を作成し、小型 PC (CPU: Intel Core i5、Laikago の付属品) とセンサ駆動用の DCDC コンバータな

どを追加で搭載した。センサおよび PC の電源は Laikago 本体から供給される 19V 電源を利用している。

現在の搭載機器の重量は約 2kg である。脚型ロボットは車輪型ロボットと比較すると転倒のリスクが非常に高いため、ロボット上部に取り付けた Lidar の保護のために、3D プリントで作成した Lidar ガードを搭載している。この Lidar ガードは幅 5mm ほどのピラーを Lidar の周囲に 6 本配置した構造である。このピラーにより Lidar の視野が若干制限されるが、実際にデータを取得したところ、地図作成・自己位置推定などには大きな影響を与えないことを確認している。

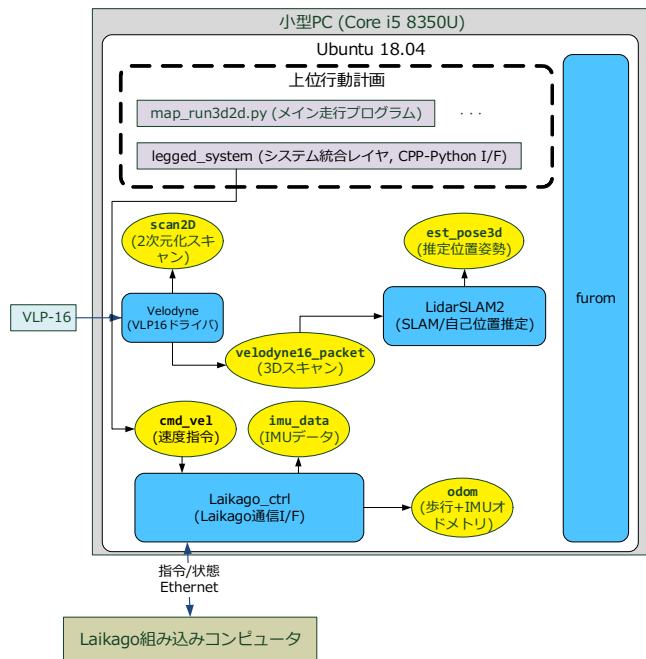


Fig. 3: システム構成図

2.3 システム全体構成

システム構成図を図 3 示す。Laikago のコンピュータシステムは 2 層で構成され、ロボット本体に内蔵されている組み込みコンピュータが歩行など下位の制御を、ロボット上部に搭載している小型 PC にて認識・計画など上位のプログラムを実行する。上位 PC と組み込みコンピュータはイーサネットによって接続されコマンド方式で通信を行う。

Laikago では 2 種類のコマンド体系が提供されており、それぞれ

- ・ 高レベル: 抽象的な 2 次元速度指令（前後左右旋回）
- ・ 低レベル: 関節単位的位置速度の指令

を与えて動かすことができる。高レベルコマンドでは、メーカーによる歩行制御の実装を用い、低レベルコマンドではユーザが当該部分を実装することになる。

本チームではこれまでのところ、高レベルコマンドを用いてシステムを構築している。高レベルコマンドを車体の速度・角速度指令とみなせば、上位ソフトウェアの多くが車輪型ロボット向けのものが流用できる。ただし、段差乗り越えのような三次元的な移動は低レベルコマンドを駆使しなければ実現できない。

2.4 ソフトウェア

2.4.1 ミドルウェアとアーキテクチャ

ソフトウェアシステムは本チームが 2017 年のつくばチャレンジにて使用したものを土台として開発した [2]。上位 PC は Linux で動作し、各機能モジュールを独立したプロセスとして実装し、全体を統合してプランニングを行う部分を Python スクリプトで記述している。モジュール間の通信には共有メモリを利用した pub-sub フレームワーク fuRom [3] を用いる。

組み込みコンピュータと PC とのコマンドやりとりには UDP 通信が用いられているが、メーカーが提供する SDK の C++ API を利用すれば通信のバックエンドについては意識する必要がない。

2.4.2 地図構築と自己位置推定

点群地図と 3D Lidar 観測を入力とし、ICP により 3 次元 6 自由度のロボット位置および姿勢を推定する。現在の実装では自己位置推定にオドメトリは使用していない。

地図構築には、事前にロボットをマニュアル走行させ、観測されたセンサデータを用いてオフラインで SLAM を行う。SLAM の手法としては、文献 [4] に述べた Lidar による逐次移動量推定とループ閉じ込みを用い、さらに地図の傾きや反りを補正するため、IMU 観測による重力方向を追加拘束としてポーズグラフに与えるよう拡張した。ポーズグラフの最適化は文献 [5] に示したものを改良して用いた。

2.4.3 走行経路の設定と制御

走行戦略としては、経路を事前に線分列の形で与えておき、各線分に追従するよう前述の指令を発することにより走行する。現状では経路への追従精度が（従来の車輪型ロボットに比べて）低いという課題がある。これは歩行制御に与える指令が物理量に対応しているわけではなく、適切なフィードバックゲインなどの設計が難しいためである。

走行経路の設定には、図 4 に示す経路エディタを用い、点群地図や事前に走行した軌跡を参考にしながら人手で設定する。3 次元 6 自由度の経路を手で直接設定することは容易ではないため、ユーザが指定した位置周辺の点群から地面を検出することで、高さや姿勢を自動的に設定することで経路設定の手間を軽減している。

障害物は Lidar にて検出し、走行経路付近に障害物が検出された場合は減速・停止する。障害物の回避は今後実装予定である。

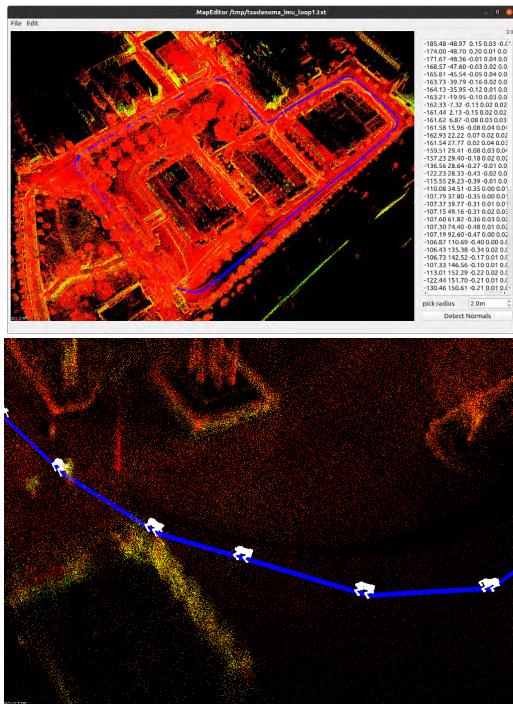


Fig. 4: 経路エディタの編集画面。上: 点群地図 (Lidar 受光強度による色付け) と設定した経路 (青線)。下: 拡大図。

2.5 シミュレータ

ナビゲーションシステム開発支援を目的に、移動ロボットシミュレータ VTC [6] に Laikago モデルを実装した。ナビゲーションシステムは Laikago の高レベルコマンドのみを使用することを想定し、歩行に関する動力学シミュレーションは行わないこととした。

実装した VTC のスクリーンショットを図 5 に示す。Laikago モデルは前後左右旋回の 3 自由度のコマンドを受けて移動する。他の物体と干渉したり凹凸のある地面を移動しても転倒しないが、一定の高さ (実機の観察により 8cm と設定) の段差までしか通過できない。またランダムに加えられる誤差により足踏みしていても一か所にとどまることはなく、一定の速度指令を与えただけでは直進できない。モデルの各部は事前に定義した歩行モーションによって動かし、歩行モーションによる胴体の揺動は搭載しているセンサ (Lidar および IMU) の観測にも影響する。歩行モーションの足の運びや歩幅は速度指令値に応じて変化するが、ロボットは歩行によってではなく与えた速度指令値に基づいて運動する。

3. 実験

2020 年 8 月 6 日に千葉工業大学 津田沼キャンパスにて地図構築のためのデータ収集を行った。データ収集には車

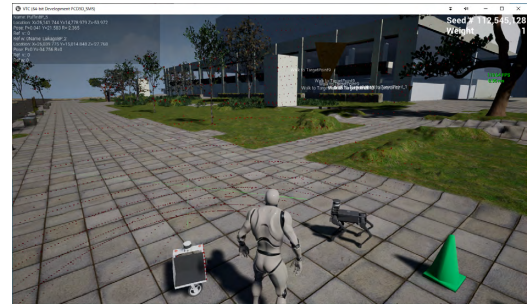


Fig. 5: シミュレータのスクリーンショット。Lidar を搭載する Laikago (右) と Puffin (左) が動作している。赤い点群は Lidar の反射点を可視化したもの。

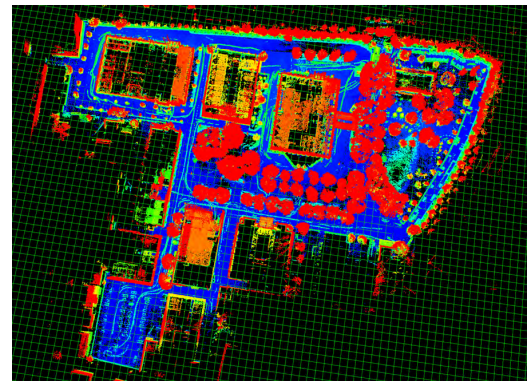


Fig. 6: 構築した千葉工業大学 津田沼キャンパスの地図 (高さによる色付け)

輪型移動ロボット Puffin [7] を用いた。収集した Lidar および IMU データから地図を構築した。得られた点群地図を図 6 に示す。この地図を航空写真と見比べ、特段の破綻がなく良好であることを確認した。

2020 年 9 月 30 日に同キャンパスにて自律走行実験を行った。走行させた経路はキャンパス内を一周する約 600m の平坦な周回路である (図 4 上参照)。走行の様子を図 7 に示す。安全のために犬用のリードをロボットに接続しているが、安定して自律で走行可能なことを確認した。この経路を 2 回自律走行させ、いずれも人手による介入なく走行を完了できた。

4. 脚型ロボット特有の課題

ここでは筆者らが過去に扱ってきた車輪型ロボット [7] などと比べた、脚型ロボットに特有の課題について論じる。

4.1 安全の確保

メーカーが提供する歩行制御は多少の外乱にも耐えられるようであるが、段差や滑りやすい床面では転倒すること

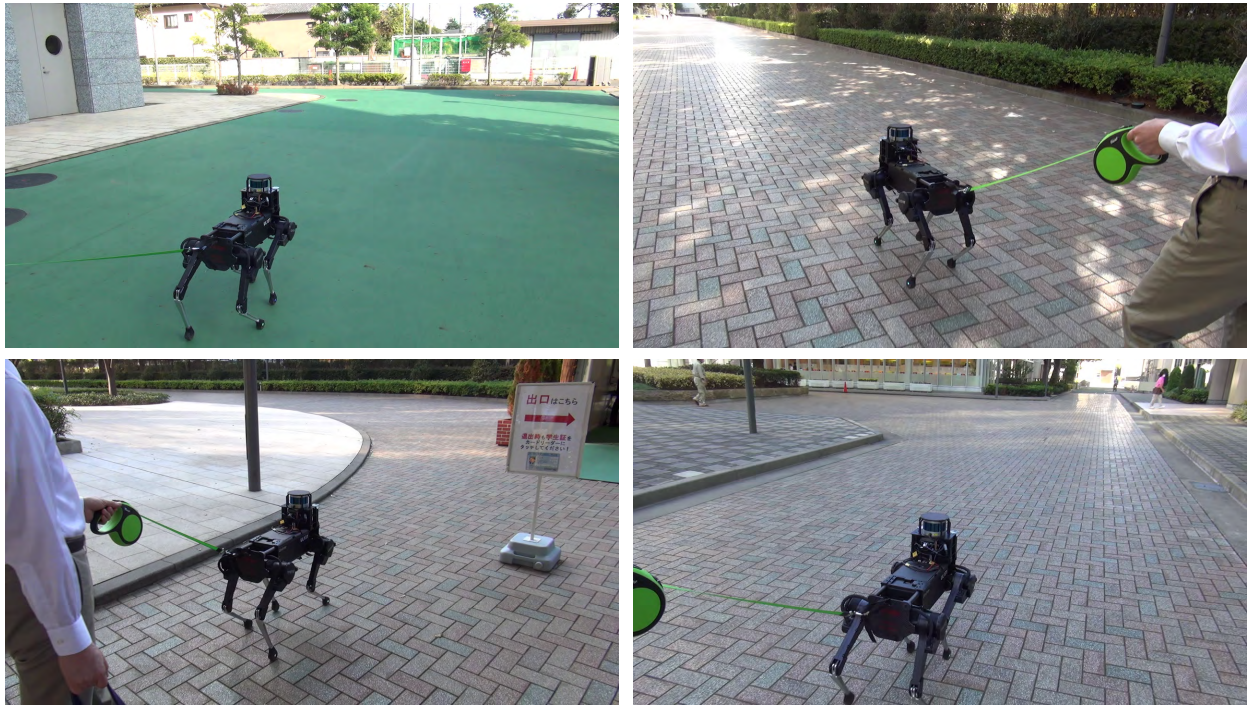


Fig. 7: 千葉工業大学 津田沼キャンパスにおける自律走行実験の様子

がある。実験時には安全の確保のため、周囲の環境により深い注意を払う必要がある。

ロボットの脚関節部には挟み込みの危険があるため、外装によって隠すなどの対処を検討している。非常停止の実現も課題であり、ロボットは無線スイッチによる緊急停止が可能であるものの、停止時に関節を現在角度でロックするため、それが原因でロボットが転倒してしまう恐れもある。今後より安全な停止方法を模索していきたい。

4.2 実験にかかる手間とコスト

実験の時間・コスト効率が低いことも課題である。上述の安全確保やロボットの運搬などのため、実験を補助する人手が多く必要になる。車輪型のようにサーボフリーにして押すことはできないため、運搬時には二人で持ち上げて移動させることになる。現状ではロボットに画面やキーボードなどのインタフェースを備えないため、ソフトウェアのデバッグも無線経由となり手間がかかったりトラブルの元となり得る。対策としては、シミュレータを用いて事前にソフトウェアを可能な限りデバッグしておくことや、補助電源や可動式クレーンなど実験補助機材を整備していくことなどが挙げられる。

5. まとめと今後の課題

市販の四脚ロボットに 3D Lidar とナビゲーションソフトウェアを搭載し、ナビゲーションを行うシステムを開発

した。大学キャンパス内に設定した数百メートルの平坦な経路において実験を行い、自律走行に成功した。

今後は、経路追従部分の改良、歩道・車道間のスロープを含めた走行、未知領域のナビゲーションなどの課題に取り組む予定である。また安全面の対策として、巻き込みや挟み込みを防ぐための外装の作成、非常停止スイッチの実装などを進める予定である。

参考文献

- [1] つくばチャレンジ 2020. <https://tsukubachallenge.jp/2020/>. [Accessed: 05- Oct- 2020].
- [2] 原祥亮, 入江清, 吉田智章, 西村健志, 大和秀彰, 友納正裕. furo 内製技術の統合による自律移動ロボットの開発と実証実験. つくばチャレンジ 2017 参加レポート, 2018.
- [3] 入江清. ROS との相互運用性に配慮した共有メモリによる低遅延プロセス間通信フレームワーク. 第 35 回 日本ロボット学会学術講演会 (RSJ2017) 2B2-01, 2017.
- [4] M. Tomono. Loop detection for 3D LiDAR SLAM using segment-group matching. *Advanced Robotics*, 2020. <https://doi.org/10.1080/01691864.2020.1824809>.
- [5] 入江清, 友納正裕. 姿勢表現に回転ベクトルを用いた三次元グラフベース SLAM. 第 18 回 計測自動制御学会 システムインテグレーション部門講演会 (SI2017) 3E3-06, 2017.
- [6] 吉田智章, 入江清, 原祥亮, 西村健志. Unreal engine4 を利用したつくばチャレンジシミュレータ. 日本機械学会 ロボティクス・メカトロニクス講演会 (ROBOMECH2019) 1P1-S09, 2019.
- [7] 原祥亮, 西村健志, 入江清, 吉田智章, 大和秀彰, 友納正裕. 屋外 3 次元地形を走行する自律移動ロボットの開発. 第 19 回 計測自動制御学会 システムインテグレーション部門講演会 (SI2018) 1E3-10, 2018.