

JETBOT を利用した自律走行車の作成

S19545 TEVETTE DJIBY

1. はじめに

ロボットは、複雑な一連のアクションを自動的に実行できる機械であり、特にコンピューターによってプログラム可能である。JetBot は組込み型 AI デバイスである NVIDIA の Jetson Nano を利用した自律走行車である。JetBot を利用することで、ロボットの動きの速度変化や回りの物体の認識を行うことができる。

本研究では、JetBot の組み立てとプログラムの実行を行ない、ロボットの移動のコマンド、および、ロボットのセンサとカメラの役割を確認した。また、移動ロボットに関する関連研究の調査を行なった。

2. JetBot のセットアップ

本研究では、JetBot AI ROBOT KIT を利用した。これは、Jetson Nano 本体が付属されていないもので、NVIDIA Jetson Nano 開発者キット(4G のタイプ)を用意した。JetBot のサイト¹⁾を確認しながら部品の組み立てを行なった。図 1 は組み立て後の様子である。



図 1: 利用した JetBot

本研究で利用した JetBot には、モーターが二つ搭載されており、左右の車輪と前方のカメラ(Sony IMX219, 解像度 3280x264), また、無線 LAN (Wireless-AC8265) が搭載されている。それらの組み立てを行った上で JetBot のサイト¹⁾からビルド済みの JetBot SD カードイメージ (jetbot_image_v0) をパソコンにダウンロードする。このイメージファイルを microSD カードに書き込みを行い、JetBot に搭載した Jetson NANO のカードスロットに挿入し、JetBot 起動する。Linux の GUI である Ubuntu が起動するので、WiFi 接続の設定を実施することで、Web ブラウザから JetBot に接続することができる。JetBot の PiOLED ディスプレイの IP アドレスを読み取り、パソコンなどのブラウザを利用して、この IP アドレスに対して、「https://<IP アドレス>:8888/」のように接続を行う。

3. 障害物回避 (collision avoidance)

本研究では、JetBot のサンプルプログラムである障害物回避のプログラム (collision avoidance) の実行を行った。二つの異なるコースを利用して実験を行った。図 2 に示した実験コース1では、ロボットが黒い線に沿って走行する実験を行い、図 3 に示した実験コース2では、道路のようなコースを利用して実験を実施した。

障害物回避プログラムでは、カメラの映像から以下の2つの状態の分類を行うことで衝突回避タスクが実行される。

- フリー: ロボットが前進できる状態
- ブロック: ロボットが前進できない状態

ロボットは、カメラの映像からフリーの状態と認識し

た場合にはロボットは前進し、ブロックの状態と認識した場合には、左方向へ回転し、それでもフリーの状態を認識できなければロボットは停止する。

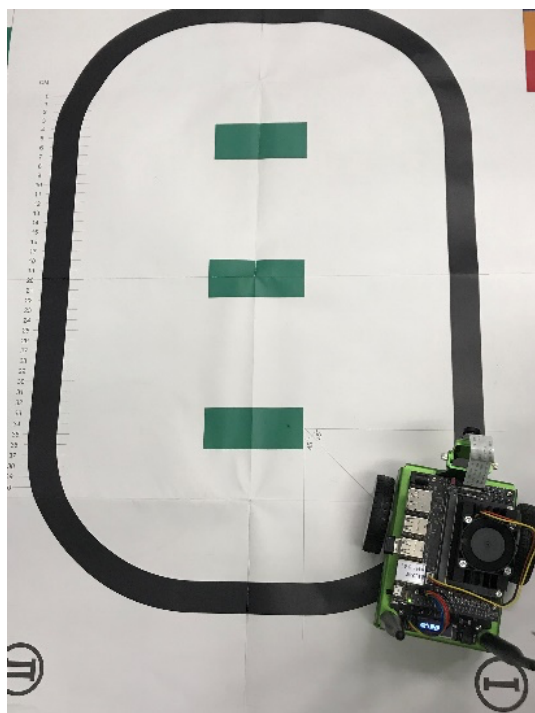


図2 実験コース1



図3 実験コース2

この障害物回避プログラムを実行するためには、最初に、データ収集プログラム(Data collection)を実行し、JetBot のカメラから取得した画像を元に状況を判断するモデルを作成する必要がある。ブロックの状態の画像とフリーな状態の画像は、それぞれ64枚まで収集可能である。

図2の実験コース1では、黒い線の画像をフリー状態、黒い線がない状態をブロックとして収集して実験を行った。その結果、JetBot のカメラが線を認識するとロボットは前進し、認識出来ない場合には、左方向へ回転し、それでも見つけられない場合は、ロボットは停止するように動いた。

また、図3の実験コース2では、コースの周りにおいた箱が映った画像をブロックの状態とし、そうでないものをフリーの状態として収集した。その結果、ロボットは移動中に、箱を認識して左側に回転することで、衝突を回避し、コースを周回することができた。

4. おわりに

本研究では、自律走行車として、Jetson NANOW お搭載した JetBot を対象として、障害物回避プログラムの実行実験を行った。このプログラムは、ロボットに搭載されたカメラを利用して、走行かのような状態と不可能な状態を識別することで、障害物回避を行いながら、自律的に走行するプログラムである。

本研究では、二種類のコースを利用して実験を行った。その結果、黒い線のコースでは、黒い線を認識した場合に前進し、黒い線がない場合には、黒い線を探すように左に回転するような動きが確認された。また、周囲に障害物を置いた道路のようなコースでは、周囲の障害物を認識し、それらを回避するように左へ回転し、道路を周回する動きが確認された。

参考文献

- 1) Home · NVIDIA-AI-IOT/jetbot Wiki · GitHub, <https://github.com/NVIDIA-AI-IOT/jetbot/wiki> (Access, 2023.1.20)