

ZytleBot : ROS × FPGA を実現する ロボット開発プラットフォーム

田村爽, 新田泰大, 高瀬英希 (京都大学)

1 はじめに

自動運転を実現するにあたり, 京都大学「ZytleBot」チームは ROS(Robot Operating System) を採用した. ROS はロボットシステムの設計生産性の向上に資する開発支援フレームワークとして注目されている. ROS ではソフトウェア部品をノードとして表現し, 複数のノードを組み合わせることでロボットシステムを実現する. また, 実行ファイルと環境設定ファイルなどをまとめた ROS パッケージが数多く公開されている, 現実世界と同じ実行ファイルで動作させることが可能な 3D シミュレータなど, 効率的な開発のためのコンポーネントが用意されている.

我々が開発した「ZytleBot」は CMOS カメラからの入力のみかつ単体で動作する FPGA を活用した自動運転 ROS システムである. ターゲット環境は, FPGA 開発ボード: ZYBO Z7-20, OS: カスタム Ubuntu16.04, ROS のバージョン: kinetic, ロボット: TurtleBot3, カメラ: PCAM 5C である. 構築する ROS システムはそれ単体で動作するものとし, 他のホストコンピュータなどに接続はしないものとする.

Zybo に ROS を導入するための手順を簡単に記す. Zybo で Ubuntu を起動するためのブートローダを, Xilinx 社から提供されている Petalinux ツールを用いて作成する. FPGA と通信するためのデバイスドライバや uio の設定を含めてビルドすることで, Ubuntu から FPGA 上の IP を扱うことが可能となる. Zybo から Ubuntu が起動すれば, 通常の PC と同様に ROS をインストールすることが可能である.

2 ZytleBot の構造

FPGA 内に作成した画像取得 IP によってカメラからカメラからデータを受け取り, ARM に送信する.

ARM 上で実行させている ROS では三つのノードを作成する. 一つ目は, FPGA からのデータを受信し, 他の ROS ノードで処理できる形に変換して出版するためのカメラ画像取得ノードである. 二つ目は画像データを購読

し, ロボットの挙動を決定するための行動決定ノードである. 行動決定ノードでは, 購読した画像を元に, 車線検出, 物体検出および交差点等を認識しての自己位置更新を行う. これによりロボットを制御するための進行速度及び方向を決定して出版する. 三つ目のロボット制御ノードでは, 進行速度及び方向を購読し, openCR と通信する ROS のパッケージである rosserial を利用して, モータを制御する. 以上の構造を図 1 に示す.

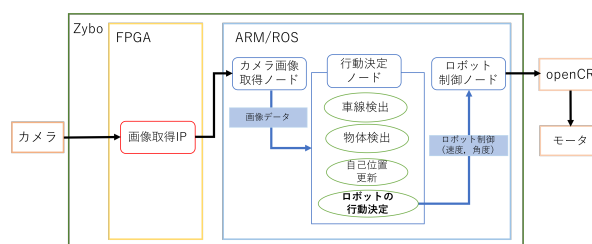


図 1 : ZytleBot の構造

3 今後の課題

コンテストの課題のうち, 信号機及び人の発見の精度に課題がある. また自動運転そのものに対しては, 急な飛び出し物に対して即座に反応し回避するような安全性が確保されていない.

これらの課題を解決するための方法として, FPGA を用いることが挙げられる. 例えば信号機や人を認識するためのモデルを機械学習で作成し, FPGA で推論を行わせることで, 精度の向上が期待できる. また, 急な飛び出しに対しても, FPGA で画像を高頻度で監視し, 危険物が発見された時には ROS を介さずに直接モーター制御することにより高速な反応が可能である. これにより, 安全性の高いロボットシステムを構築することが可能となる.