

FPGA を用いた後輪操舵ロボットの自動化 Team Robotics

滝田好宏(防衛大) ナチャノック・リアドプラトム(防衛大)

1. はじめに

FPGA は高集積化, 多機能化, 高速化が進み, また安価なデバイスが購入できるようになっているが, 知識と経験がないと目的とした性能が得られない難しさがある. FPGA デザインコンテストでは, 自動車の自動運転に適用可能な技術を FPGA で実現することを目標としており, 学生の教育と研究の方向が一致したので参加することにした.

2. ロボットの構成

図 1 に本大会のために開発した後輪操舵車両に ZYBO を搭載した FPGA カーの外観を示す. 車体は小回りがきくようにフォークリフトと同じ形式の前輪駆動の後輪操舵方式で, 200×200mm 内に収めるために後輪を 1 輪とした.

図 2 には制御システムの構成を示す. USB カメラは Logicool 製で 640×460 ピクセルであり, ホイールベースの距離の路面と信号の高さまで視野角を確保するように後方に設置している. モータ制御装置は SH7124 (ルネサスエレクトロニクス製) を用いて構成し, ZYBO と双方向でシリアル通信をおこなっている. SH7124 から走行距離を受信し, 走行速度, 操舵角を指令値として送信している.

3. 制御方法

図 3 にロボットに使用した車線追従の方法を示す. 図の下向きに移動すると後輪操舵車両で, 上方向に動くとき前輪操舵車両となり, 時定常円の軌道に追従方法として著者は SSM (Sensor Steering Mechanism) と名付けている. この方法を実現するために, カメラから取得した画面の最下位のラインを用いて路面の左側の白線を中心を検出する. 白線を中心は白線の幅のピクセル間隔で差分を求めると, 符号が入れ替わる値が得られたピクセル番号となる. この値は外乱光の影響を最小限にして検出することができる. 検出した白線を中心のピクセル番号から車体中心からのずれの大きさ L_s に変換して $\phi' = \tan^{-1}(L_s/L)$ を求めると, 近似式から操舵角 δ は $2\phi'$ となる. この値を操舵角に用いることにする.

制御プログラムは OpenCV ライブラリを用いているが, 画像取得のみで, それ以外のライブラリは処理の高速化のために使用していない. たとえば, 信号は赤のみのピクセルを検出するために $R > 230, G < 180, B < 180$ とした. 横断歩道の検出と歩行者の検出は注視するラインを足元と膝の高さの 2 ラインを選び, 隣の値との差分を計算するエッジ検出処理を行った結果, エッジの数の変化で求めることにした. 走行はコースのパネルを地図として入力し, 走行距離を用いて塗りつぶしを行うことで走行を達成する方式とした. これらのプログラムは C 言語で作成する.

4. まとめ

本チームはロボットの研究室に所属するが FPGA の初心者であり, まず Linux ベースで開発することを考えシステムを構築した. 当初は最新の ARTY Z7 とカメラ接続が可能な ZYBO Z7 を購入して Linux のインストールを試みたが達成できず, 何とか旧バージョンの ZYBO にインストールできたので参加が可能になった.

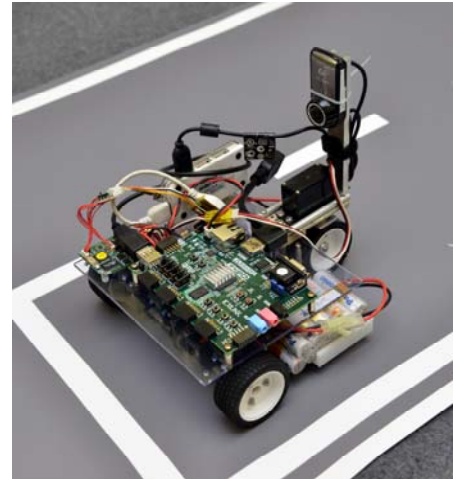


図 1 後輪操舵車両の外観

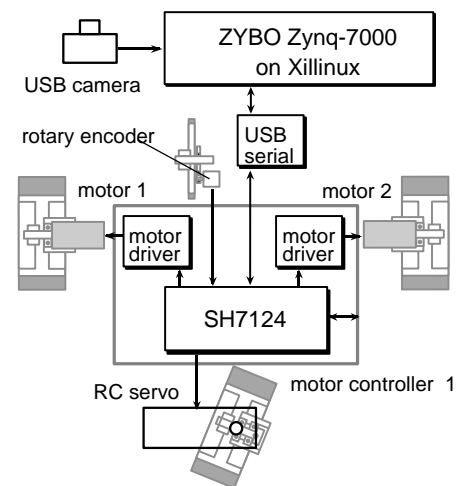


図 2 FPGA と制御装置の構成

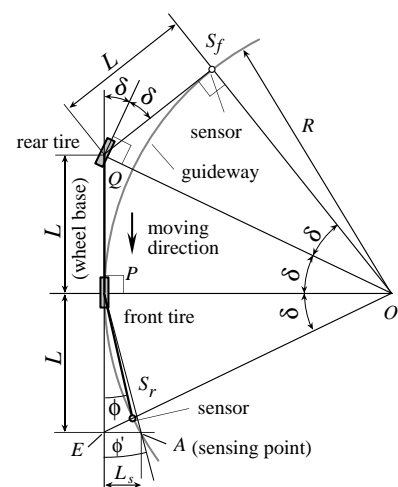


図 3 後輪操舵車両の軌道追従方法