1 ROS

1.1 ROS とは [1][2]

ROS(Robot Operating System) とは Open Source Robotics Foundation によって管理されているソフトウェア開発者のロボット・アプリケーション作成を支援するフレームワークである．具体的には，ハードウェア抽象化，デバイスドライバ，ライブラリ，視覚化ツール，メッセージ通信，パッケージ管理などが提供されている．つまり ROS は汎用コンピュータ向けの OS ではなく，汎用コンピュータ向け OS 上で動作するメタ OS として捉えることができる．

図 1 に示すように ROS ではプロセス (実行プログラム) はノードという単位で扱い，ノード間の通信はトピックと呼ばれる “Publisher/Subscriber” モデルで実現される．これにより，プログラミング言語や通信相手さえ意識することなく簡単にプロセス間通信を実現できる．これは各ノード間のインタフェース，即ちトピックの名前と型さえ決定すればノードごとに独立して開発を行うことができるという利点でもある．

以上の利点を鑑み，本研究室では ROS がインストール可能なマイコンボードである RaspberryPi3 Model B 上に ROS をインストールして開発を進めていくこととした．

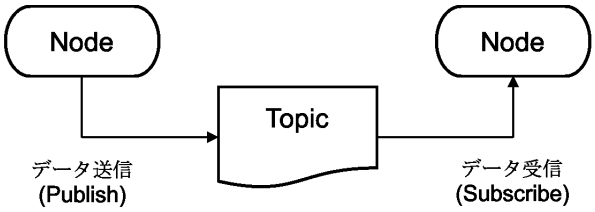

図 1: ROS ノードとトピックの概念

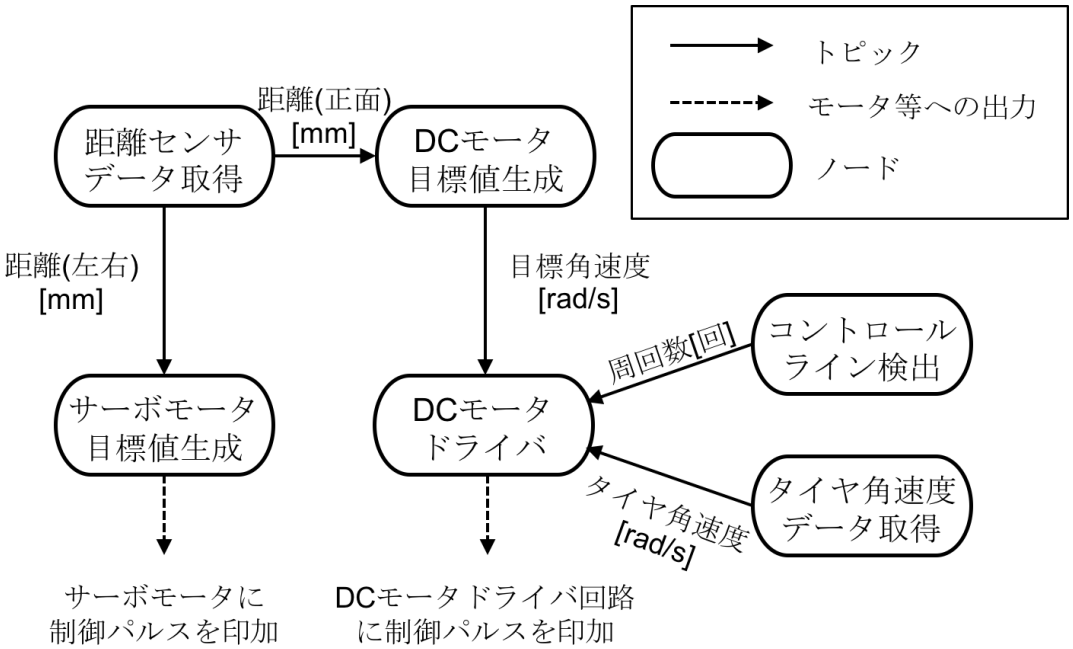

図 2: ROS ノードとトピックの構成

1.2 ROS ノードとトピックの構成

図 2 に開発する ROS ノードとトピックの構成を示す．各ノードの役割は次の通りである．

距離センサデータ取得

機体前方及び両側面に設置した距離センサからシリアルバス規格の一つである I^2C を介して距離データを [mm] 単位で取得し外れ値処理や正規化を施した後に Publish する．

コントロールライン検出

機体後方下部に設置したフォトリフレクタによってコントロールラインを通過した回数をカウントし Publish する．

タイヤ角速度データ取得

回転方向は考慮しないため，ロータリーエンコーダの A 相のパルスのみをカウントし，ロータリーエンコーダの一周あたりの出力パルス数 (500 パルス), サンプリング周期 (0.01 [s])，ロータリーエンコーダの軸に取り付けたピニオンギアとドライブシャフトに取り付けたギア間のギア比 (2.74) を考慮して k 時点のタイヤの角速度 $\omega(k)$ を [rad/s] 単位で算出する．また，ノイズ対策として算出した角速度を式（1.1）で表されるデジタルフィルタ (LPF に相当) に通した結果を Publish するようにしている．ただし，α は 0〜1 の範囲で定める定数である．

$$\omega(k) = \alpha\omega(k) + (1-\alpha)\,\omega(k-1) \tag{1.1}$$

DC モータ目標値生成

機体前方方向の距離データを Subscribe し，それをもとに DC モータに与える目標値を生成し Publish する．

DC モータドライバ

DC モータに与える目標値とタイヤの角速度データを Subscribe し，PI 制御則に基づき DC モータを駆動する．

サーボモータ目標値生成

機体前方の距離データを Subscribe し，それをもとにサーボモータに与える目標値を生成しその後直接サーボモータを駆動する．サーボモータドライバノードが存在しないのはサーボモータの軸にロータリーエンコーダがついていないため角度フィードバックが出来ないためである．

参考文献

[1] 表允, 倉爪亮, 渡邊裕太, “詳説 ROS ロボットプログラミング-導入から SLAM・Gazebo・MoveIt まで-”, Kurazume Laboratory, pp.15-18, (2015).

[2] 小倉崇, “ROS ではじめるロボットプログラミング”, 工学社, pp.8-10, (2015).