

## ノート

# 組み込みシステム上で動作するRTミドルウェアによる運動制御 — 模型回転翼機への適用

佐々木 智典<sup>\*1)</sup> 島田 茂伸<sup>\*1)</sup>

## Motion control with robotic technology middleware on embedded system — application to a model rotorcraft

Akinori Sasaki<sup>\*1)</sup>, Shigenobu Shimada<sup>\*1)</sup>

キーワード：RTミドルウェア，クアッドロータ，飛行制御，組み込みLinux

Keywords：RT-middleware, Quadrotor, Flight control, Embedded Linux

### 1. はじめに

RTミドルウェアはロボットシステムを構築するためのソフトウェアプラットフォームであり，独立行政法人新エネルギー・産業技術総合開発機構（NEDO）のプロジェクトにおいて独立行政法人産業技術総合研究所，他により開発された<sup>(1)</sup>。RTミドルウェアを利用したシステム構築の機能的単位となるコンポーネントの仕様は国際標準化団体Object Management Group（OMG）による標準として策定されている<sup>(2)</sup>。RTミドルウェアの実際の標準実装であるOpenRTM-aistはオープンソースソフトウェアとして公開されている。本研究ではOpenRTM-aistを組み込みLinuxボード上で利用し，模型回転翼機（図1）の運動制御を行うシステムを構築した。



図1. 模型回転翼機

### 2. システム構成

2.1 ハードウェア OpenRTM-aistの利用のためには，同じくOMG標準のミドルウェアであるCORBA及びTCP/IPプロトコルスタックが必要となり，相応の計算機

性能が要求される。このため，RTミドルウェアの利用においてはPCベースのシステムの事例が多いものの，上記要求を満たしうる組み込みボードによるシステムの構築も行われている。本研究では，組み込みLinuxボードの一つであるArmadillo-440（株式会社アットマークテクノ）を用いる。Armadillo-440にはARM9コアを備えたSystem-on-Chip LSIであるi.MX25（Freescale Semiconductor, Inc.）が搭載されており，設定によりPWM出力機能4チャンネル，USBホストコントローラ機能1チャンネルがそれぞれ利用可能である。

図1の模型回転翼機F330（DJI Innovations Technology, Co. Ltd.）はクアッドロータ（あるいはクアッドコプタ）と呼ばれ，4個のロータを備える。その姿勢（ロール，ピッチ，ヨー角）を制御するには，姿勢に応じた4個のロータの回転速度の調節が必要となる。ロータには順ピッチと逆ピッチの2種類がそれぞれ1対あり，ヨー軸まわりに発生する力のモーメントを打ち消しあう様に配置する。

姿勢の計測には3軸ジャイロITG-3205（InvenSense Inc.），3軸加速度センサBMA180（Bosch Sensortec GmbH），3軸地磁気センサHMC5883L（Honeywell International Inc.）を利用する（これらを搭載したボードCRIUS MultiWii SEのファームウェアを書換えて利用）。ロータを駆動するブラシレスDCモータDJI 2212/920KVの回転速度制御にはRC模型用の速度制御器（DJI 18A OPTO ESC（Electric Speed Controller）），マイコン及びモータドライバ回路）を使用し，その目標速度をi.MX25のPWM出力により指令する。

推力，姿勢の目標値はユーザによるゲームパッド（またはジョイスティック）操作により指定される。ゲームパッドは別のホストPCに接続されており，操作に基づく指令は無線LAN（IEEE802.11g）によりクアッドロータに送信される。機体側ではUSB-無線LANアダプタWLI-UC-GNM2（株式会社バッファロー，Ralink RT3070搭載）により送受信を行う。

事業名 平成24年度 基盤研究  
\*1) 機械技術グループ

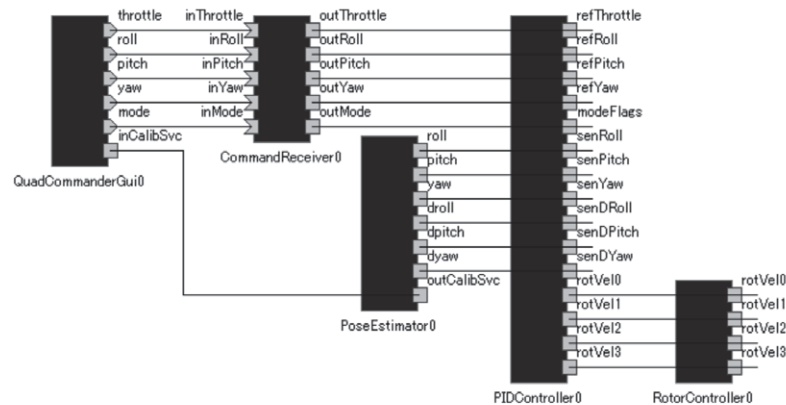


図2. RTコンポーネント間の接続

2.2 ソフトウェア 機体制御ソフトウェアはOpenRTM-aist 1.1.0を利用して構成され、Linux (カーネル2.6.26-at13)上で動作する。RTミドルウェアにより構成するシステムにおいて、2.1節に述べた計測、制御を複数の機能単位により分担して行う。この機能単位をRTコンポーネントと呼ぶ。RTコンポーネントの接続関係はRTミドルウェアのツール (OpenRTP RT System Editor) により図2の様に表示され、その状態のモニタリングが可能である。図2において矩形がRTコンポーネントを表し、その間の線は通信のための接続を表す。図2のRTコンポーネントそれぞれについて以下に述べる。以下のコンポーネントのうち、QuadCommanderGui以外はC++言語により実装され、クアッドロータ上で動作する。QuadCommanderGuiのみPython言語により実装され、クアッドロータと別のPC上で動作する。RTコンポーネント間の接続は言語に依存しない。

(1) RotorController 4個のロータの回転速度を制御する。実際上の制御はESCが行うので、このコンポーネントは入力の回転速度目標値に対応するPWM信号のパルス幅を計算し、そのPWM出力を実行する。

(2) PoseEstimator 3軸ジャイロ、3軸加速度センサ、3軸地磁気センサによる測定値を基に姿勢 (ロール、ピッチ、ヨー角) の推定を行う。この推定値は姿勢制御に使われる。

(3) PIDController 姿勢制御を行うため、PoseEstimatorの出力である推定姿勢とユーザ入力に基づいて、ロータ回転速度の目標値を計算する。対称にロータを配置した構成のクアッドロータが水平に近い姿勢であれば、ロール、ピッチ、ヨー角はそれぞれ独立に制御できる<sup>(3)(4)</sup>。

(4) QuadCommanderGui ユーザのゲームパッド (ジョイスティック) 操作に対応した指令を次のCommandReceiverに送信する。また、PC画面上のGUIによりスティック角度を表示しユーザにフィードバックする。

(5) CommandReceiver 外部のコンポーネントとのインタフェースとして働く。後述する様にクアッドロータ機体上で動作するRTコンポーネント間のデータ通信に共有メモリを利用している都合があり、このコンポーネントのみ標準のデータポートを配備し、外部との通信を行える様にする。

これらのコンポーネントの実行に関して、全てを同一

のプロセスにまとめて実行する方法 (複合コンポーネント機能の利用) が比較的に実装容易であるものの、本研究ではコンポーネント分割のケーススタディとして別々のプロセスとして実行する方法を採用した。この時、コンポーネント間の通信の遅延が問題となるため、これを低減する共有メモリを利用したコンポーネント間接続の方法を開発した。また、カーネルのプリエンブション機能を有効にする事により、制御周期の遅延の低減を図り制御を行う。

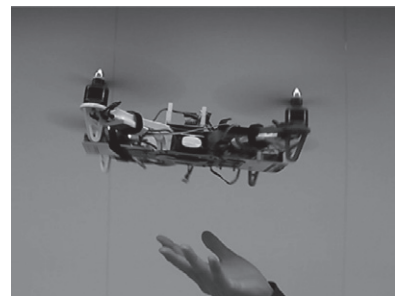


図3. 飛行中の模型回転翼機

以上のRTコンポーネントにより構成したシステムにおいて、10 msec前後の制御周期での姿勢制御を行える事を確認した。図3に、実験における飛行の様子を示す。

### 3. まとめ

RTミドルウェアを用いて複数のコンポーネントによる模型回転翼機 (クアッドロータ) の制御について試作開発を行った。今後、更に制御性能の改善等について検討する。

(平成25年7月16日受付)

### 文 献

- (1) 独立行政法人新エネルギー・産業技術総合開発機構研究評価委員会: 『『ロボットの開発基盤となるソフトウェア上の基盤整備』事後評価報告書』(2006)
- (2) Object Management Group: “Robotic Technology Component, version 1.1” (2012)
- (3) E. Altug, J. P. Ostrowski, and C. J. Taylor, “Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback”, The International Journal of Robotics Research, Vol.24, No.5, pp.329-341 (2005)
- (4) K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa. “Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles”. Springer (2010)