

再利用可能なマニピュレータ制御 RT コンポーネント群マニュアル

尹祐根（産総研, JST, CREST）

清水昌幸（静岡大, 産総研）

2008 年 12 月

1 はじめに

効率的なロボット開発ツールとして、ネットワーク分散コンポーネント化技術を利用した共通プラットフォームである RT ミドルウェアが開発されています。OpenRTM-aist は、RT ミドルウェア実装のひとつであり、現在は OpenRTM-aist-0.4.2 がリリースされています。

RT ミドルウェアでは、ロボットを構成する要素を RT コンポーネント(以下 RTC と呼ぶ)というモジュール単位で取扱うことによって、ロボットや動作環境に依存する部分と依存しない部分に切り分けることが可能となるため、RTC の再利用性を向上させることが可能となります。再利用性の高い RTC を開発するためには、他の RTC や動作環境への依存性、再利用のしやすさ、どのように再利用されるか等を十分に考慮し、システムのモジュール構成や個々の RTC の設計をする必要があります。

本マニュアルでは、再利用性を最大限に考慮して開発された、ロボットマニピュレータ制御用 RTC 群の概要や使用方法を解説します。今回開発したマニピュレータ制御 RTC 群は、以下の 4 つの RTC から構成されます。

- ・ サーボ制御 RTC
- ・ 順運動学 RTC
- ・ 逆運動学 RTC
- ・ PD 制御 RTC

それぞれの RTC は、再利用性を考慮に入れて設計されているため、それらの組み合わせによって多様な制御システムを構築することが可能です。例として、以下の制御システムを本マニュアルでは紹介します。

- ・ 関節角度制御システム
- ・ 関節速度制御システム
- ・ 手先位置制御システム
- ・ 手先速度制御システム

また、本 RTC 群を利用するための補助として、指令値生成 RTC のサンプルも紹介します。サンプルを元に、ユーザ独自の指令値生成 RTC を作成すれば、独自の制御アルゴリズムを容易に実現することが可能となります。

2 動作環境

本制御システムの検証済み動作環境は以下の通りです。

- OS : Ubuntu 8.04 [注 1, 注 2]
- RTM : OpenRTM-aist-0.4.2 (C++版)
- 商用ライブラリ [注 3] : ライフロボティクス社製 PA-10 高速制御用ソフトウェアライブラリ
- ロボットマニピュレータ [注 3] : 三菱重工業製 PA10 シリーズ (PA-10, PA10-7A, PA10-7C, PA10-6C)

[注 1] Ubuntu 8.04 のみでは、実時間制御はできません。実時間制御する場合には、Debian4.0 + ARTLinux2.6 カーネルの利用を推奨します。それ以外の実時間 OS での動作確認は行っておりません。

[注 2] サーボ制御 RTC 以外は OS に依存しないように実装されていますので、OpenRTM-aist-0.4.2 が動作する環境であればどの OS でも動作しますが、十分な検証は行っておりません。

[注 3] 本制御システムには、実機制御モードとシミュレーションモードがあり、シミュレーションモードの場合、ロボット本体や商用ライブラリが無くても動作します。

3 マニピュレータ制御 RT コンポーネント群の概要

本制御システムは、以下の RTC から構成されます。

- サーボ制御 RTC
- 順運動学 RTC
- 逆運動学 RTC
- PD 制御 RTC

以下では、各 RTC の概要、仕様、使用方法等を説明します。また、これらコンポーネント群を利用するための補助として、指令値生成 RTC についても説明します。

3.1 サーボ制御 RTC

3.1.1 概要

本 RTC は、マニピュレータのサーボドライバの制御を行います。対応マニピュレータは三菱重工業製の PA-10 シリーズ (PA-10, PA10-7A, PA10-7C, PA10-6C) です。マニピュレータの仕様に関しては、マニピュレータに付属のマニュアル等をご参照下さい。

PA-10 のサーボドライバは、関節の速度制御またはトルク制御が可能ですが、本 RTC は、関節速度制御のみに対応しています。

本 RTC は、シミュレーション用にも用いることができます。また、実機制御とシミュレーションの動作切替は RTC のコンフィグレーションで簡単に行うことができます。

なお、本 RTC を用いて実機の制御をする場合は、別途、商用ライブラリが必要となります。さらに、実時間制御を行う場合は、OS として ARTLinux が必要となります。

3.1.2 仕様

本 RTC の仕様を以下に示します。

【動作環境】

- OS : Linux, ARTLinux (実時間制御時のみ)
- RTM : OpenRTM-aist-0.4.2 (C++版)
- ライブラリ : ライフロボティクス社製 PA-10 高速制御用ソフトウェアライブラリ (実機制御時のみ)
- ロボット : 三菱重工業製ロボットアーム PA10 (PA-10/PA10-7A/PA10-7C/PA10-6C) (実機制御時のみ)

【主要諸元】

- 名前 : PA10Manipulator
- 言語 : C++ (ライブラリは C)

【データポート】

Inport			
名称	データ型	データ長	説明
JointVelocity	TimedDoubleSeq	7	目標関節速度 [rad/s]

Output			
名称	データ型	データ長	説明
CurrentJointAngle	TimedDoubleSeq	7	現在関節角度 [rad]

Duplicate CurrentJointAngle	TimedDoubleSeq	7	現在関節角度 [rad] ※ 一つの OutPort から複数の InPort へのデータ出力ができないため、CurrentJointAngle ポートのコピーポートを設けている。(暫定処置)
CurrentJointVelocity	TimedDoubleSeq	7	現在関節速度 [rad/s]

【コンフィグレーション】

Configuration			
名称	型	デフォルト値	説明
init_angles	std::vector<double>	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0	初期関節角度 [rad]
mode	string	sim	実機 (real) とシミュレーション (sim) の切替え. 設定ファイルで変更可能
sampling	double	0.001	シミュレーションのサンプリング周期 [sec]
tip_len	double	0.0	エンドエフェクタの長さ [m]

3.1.3 使用方法

- コンパイル方法

[Linux]

- サーボ制御 RTC ディレクトリ配下の driver ディレクトリへ移動.
- コンソールから”make”コマンドを実行.
- サーボ制御 RTC ディレクトリへ移動.
- コンソールから”make”コマンドを実行.

- 実行方法

[Linux]

PA10ManipulatorComp を実行.

3.1.4 使用条件

サーボ制御 RTC の使用に当たっては、本マニュアルの最後に記載されている特記事項に従ってください.

3.2 順運動学 RTC

3.2.1 概要

本 RTC は、マニピュレータの順運動学計算を行います。対応マニピュレータは、三菱重工業製の PA10 シリーズ (PA-10, PA10-7A, PA10-7C, PA10-6C), もしくは、それと同等の機構を持つものです。関節配置が同じであれば、異なるリンク長のマニピュレータにも対応可能です。

本 RTC が提供する機能は以下の通りです。

- ・ 関節角度から手先位置・姿勢を計算する。
- ・ 関節速度から手先速度を計算する。

運動学計算に際しては、マニピュレータのベースに固定した基準座標系および手先座標系を用います。座標系の定義については、マニピュレータに付属のマニュアル等をご参照下さい。PA-10/PA10-6C/PA10-7C に関しては、以下のアドレスからも入手可能です。(2008 年 11 月時点)

<http://www.mhi.co.jp/kobe/mhikobe/products/mechatronic/index.html>

3.2.2 仕様

本 RTC の仕様を以下に示します。

【動作環境】

- ・ OS : Any. (但し, Linux および Windows でのみ動作確認済)
- ・ RTM : OpenRTM-aist-0.4.2 (C++版)

【主要諸元】

- ・ 名前 : FwdKinematicsPA10
- ・ 言語 : C++

【データポート】

Inport			
名称	型	データ長	説明
Joint Angle	TimedDoubleSeq	7	関節角度 [rad]
Joint Velocity	TimedDoubleSeq	7	関節速度 [rad/s]

Output			
名称	型	データ長	説明
Tip Position	TimedDoubleSeq	6	基準座標系に関する手先位置・姿勢 [m rad] 3次元位置ベクトル + 姿勢3パラメータ 姿勢表現法はコンフィグレーションで指定

Tip Velocity	TimedDoubleSeq	6	基準座標系に関する手先速度 [m/s rad/s] 並進速度ベクトル + 角速度ベクトル
--------------	----------------	---	---

【コンフィグレーション】

Configuration			
名称	型	デフォルト値	説明
OrientationType	string	Roll-Pitch-Yaw	手先姿勢の表現方法の指定。 現在のところ、Roll-Pitch-Yawにのみ対応。 その他の姿勢表現方法も容易に追加可能。 追加方法はソースコードを参照。

3.2.3 使用方法

- コンパイル方法

[Linux]

1. 順運動学 RTC のソースディレクトリに移動.
2. コンソールから次のコマンドを入力.

```
$ make all
```

[Windows (Microsoft Visual C++ 2008 利用の場合)]

1. 順運動学 RTC のソースフォルダを開く.
2. FwdKinematicsPA10Comp_vc9.sln を開き、ソリューションのビルドを実行する.

- 実行方法

[Linux]

FwdKinematicsPA10Comp を実行.

[Windows]

FwdKinematicsPA10Comp.exe を実行.

3.2.4 注意事項

- PA10-6C のような 6 軸タイプのマニピュレータでは、仮想関節を第 3 関節 (S3) として追加し、7 軸分の関節角度・関節速度を入力してください。その際、仮想関節の角度および速度はゼロとしてください。
- 手先速度のみを計算する場合でも、関節角度の入力は必須です。手先位置のみを計算する場合は、関節速度の入力は必要ありません。

3.2.5 使用条件

順運動学 RTC (以下、本ソフトウェア) の使用に当たっては、本マニュアル最後に記載されて

いる特記事項に従ってください。本ソフトウェアに関しては、アカデミック用途に限り、ソースコードの改変およびその派生物の利用を許可します。ただし、いかなる目的であっても、改変したソースコードならびにその派生物（バイナリを含む）の配布は禁止します。また、本ソフトウェアまたはその派生物を用いた研究成果を公表する際は、以下の論文のいずれかを参考文献として引用してください。

[参考文献（日本語）]

「関節の可動範囲を考慮に入れた7自由度冗長マニピュレータの解析的逆運動学解法」, 清水昌幸, 角谷啓, 尹祐根, 北垣高成, 小菅一弘, 日本ロボット学会誌, vol. 25, no. 4, pp. 606-617, 2007.

[Reference (English)]

Masayuki Shimizu, Hiromu Kakuya, Woo-Keun Yoon, Kosei Kitagaki, Kazuhiro Kosuge, "Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution", IEEE Transactions on Robotics, Vol. 24, No. 5, pp. 1131-1142, 2008.

3.3 逆運動学 RTC

3.3.1 概要

本 RTC は、マニピュレータの逆運動学計算を行います。対応マニピュレータは、三菱重工業製の PA10 シリーズ (PA-10, PA10-7A, PA10-7C, PA10-6C), もしくは、それと同等の機構を持つものです。関節配置が同じであれば、異なるリンク長のマニピュレータにも対応可能です。

本 RTC の基本機能は以下の通りです。

- ・ 手先位置・姿勢から関節角度を計算する。
- ・ 手先速度から関節速度を計算する。

運動学計算に用いる座標系定義は、マニピュレータ付属のドキュメント等をご参照下さい。PA-10/PA10-6C/PA10-7C に関しては、以下のアドレスからも入手可能です。(2008 年 11 月現在)

<http://www.mhi.co.jp/kobe/mhikobe/products/mechatronic/index.html>

本 RTC における逆運動学計算は、6 軸 (非冗長) マニピュレータに対するものとなります。7 軸 (冗長) マニピュレータの場合、第 3 関節 (S3) の角度および速度が常にゼロとなります。

3.3.2 仕様

本 RTC の仕様を以下に示します。

【動作環境】

- ・ OS : Any. (但し, Linux および Windows でのみ動作確認済)
- ・ RTM : OpenRTM-aist-0.4.2 (C++版)

【主要諸元】

- ・ 名前 : InvKinematicsPA10
- ・ 言語 : C++

【データポート】

Inport			
名称	型	データ長	説明
Tip Position	TimedDoubleSeq	6	基準座標系に関する手先位置・姿勢 [m rad] 3次元位置ベクトル + 姿勢3パラメータ 姿勢表現法はコンフィグレーションで指定
Tip Velocity	TimedDoubleSeq	6	基準座標系に関する手先速度 [m/s rad/s] 並進速度ベクトル + 角速度ベクトル

Reference Joint Angle	TimedDoubleSeq	7	参照関節角度 [rad] 複数の逆運動学解の中から、参照関節角度に最も近い解を選択するために使われます。 この入力は必須ではありませんが、入力が無い場合、任意の解が選択されます。
-----------------------	----------------	---	---

Output			
名称	型	データ長	説明
Joint Angle	TimedDoubleSeq	7	関節角度 [rad]
Joint Velocity	TimedDoubleSeq	7	関節速度 [rad/s]

【コンフィグレーション】

Configuration			
名称	型	デフォルト値	説明
OrientationType	string	Roll-Pitch-Yaw	手先姿勢の表現方法の指定。 現在のところ、Roll-Pitch-Yaw にのみ対応。 その他の姿勢表現方法も容易に追加可能。 追加方法はソースコードを参照。

3.3.3 使用方法

・ コンパイル方法

[Linux]

1. 逆運動学 RTC のソースディレクトリに移動.
2. コンソールから次のコマンドを入力.
\$ make all

[Windows (Microsoft Visual C++ 2008 利用の場合)]

1. 逆運動学 RTC のソースフォルダを開く.
2. InvKinematicsPA10Comp_vc9.sln を開き、ソリューションのビルドを実行する.

・ 実行方法

[Linux]

InvKinematicsPA10Comp を実行.

[Windows]

InvKinematicsPA10Comp.exe を実行.

3.3.4 注意事項

- ・ PA10-6C のような 6 軸タイプのマニピュレータでも、関節角度・関節速度は 7 軸分出力されます。6 軸タイプの場合、第 3 関節 (S3) を仮想関節としてみなしてご使用ください。
- ・ マニピュレータの姿勢が特異姿勢もしくはその近傍になると、逆運動学計算が不能となるため RTC がエラー状態になります。一度エラー状態になると、リセットを実行しないと正常な状態に復帰できません。特異姿勢には十分注意してご使用ください。
- ・ 関節速度のみを計算する場合でも、手先位置・姿勢または参照関節角度の入力は必要です。手先位置・姿勢の入力が無く、参照関節角度が実際の状態と異なる場合、関節速度の出力は実際とは異なるものとなります。手先位置・姿勢と参照関節角度の両方を入力することにより、信頼性のある関節速度解が得られます。

3.3.5 使用条件

逆運動学 RTC (以下、本ソフトウェア) の使用に当たっては、本マニュアル最後に記載されている特記事項に従ってください。本ソフトウェアに関しては、アカデミック用途に限り、ソースコードの改変およびその派生物の利用を許可します。ただし、いかなる目的であっても、改変したソースコードならびにその派生物 (バイナリを含む) の配布は禁止します。また、本ソフトウェアまたはその派生物を用いた研究成果を公表する際は、以下の論文のいずれかを参考文献として引用してください。

[参考文献 (日本語)]

「関節の可動範囲を考慮に入れた 7 自由度冗長マニピュレータの解析的逆運動学解法」, 清水昌幸, 角谷啓, 尹祐根, 北垣高成, 小菅一弘, 日本ロボット学会誌, vol. 25, no. 4, pp. 606-617, 2007.

[Reference (English)]

Masayuki Shimizu, Hiromu Kakuya, Woo-Keun Yoon, Kosei Kitagaki, Kazuhiro Kosuge, "Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution", IEEE Transactions on Robotics, Vol. 24, No. 5, pp. 1131-1142, 2008.

3.4 PD 制御 RTC

3.4.1 概要

本 RTC は、任意次元の制御入力に対する PD 制御計算を行い、制御量を出力します。

本 RTC では、図 1 に示される制御系に基づいて、制御量を計算します。本 RTC を用いることにより、制御対象（図 1 では、ロボット）のある状態変数ベクトル x を目標状態変数ベクトル x_d に制御することが可能となります。制御対象への制御量 u は以下の式より計算されます。

$$u = K_p(x_d - x) - K_d\dot{x}$$

ここで、 K_p および K_d は、比例ゲイン行列および微分ゲイン行列です。

本 RTC は、マニピュレータ制御のみでなく、一般的な制御対象や状態変数に対しても用いることができ、再利用性が高い設計となっています。

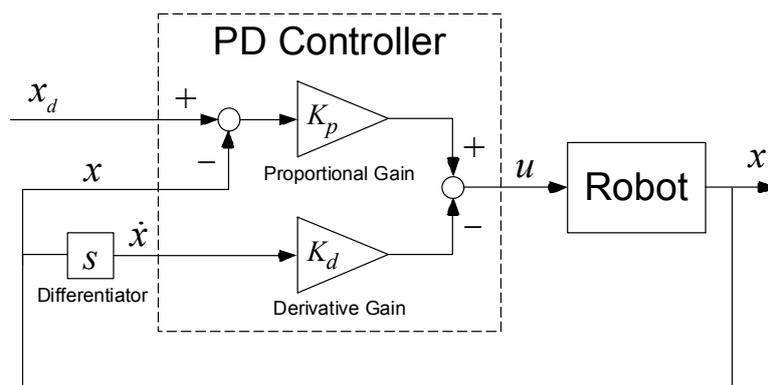


図 1. PD 制御系

3.4.2 仕様

本 RTC の仕様を以下に示します。

【動作環境】

- OS : Any. (但し、Linux および Windows でのみ動作確認済)
- RTM : OpenRTM-aist-0.4.2 (C++版)

【主要諸元】

- 名前 : PDController
- 言語 : C++

【データポート】

Inport			
名称	型	データ長	説明
Target State	TimedDoubleSeq	N (注)	目標状態変数ベクトル
Actual State	TimedDoubleSeq	N (注)	現在の状態変数ベクトル
Actual Differential State	TimedDoubleSeq	N (注)	現在の状態変数ベクトルの時間微分

Output			
名称	型	データ長	説明
Control Output	TimedDoubleSeq	N (注)	制御量

【サービスポート】

Provider port			
名称	サービス名	インタフェース名	説明
PD Gain Configuration	PDControlSvc0	PDControlService	比例および微分ゲイン行列を設定するためのインタフェース群. 詳細は PDControlService.idl を参照.

【コンフィグレーション】

Configuration			
名称	型	デフォルト値	説明
diag_Pgain	std::vector<double>	1.0, 1.0, 1.0, 1.0, 1.0, 1.0	比例ゲイン行列. 対角成分のみを設定可能.
diag_Dgain	std::vector<double>	1.0, 1.0, 1.0, 1.0, 1.0, 1.0	微分ゲイン行列. 対角成分のみを設定可能.

(注) データポートのデータサイズは、サービスポートまたはコンフィグレーションから設定されたゲイン行列の次元に自動的に合わせられます。すなわち、ゲイン行列の次元が $N \times N$ なら、データサイズは N となります。初期状態では、 $N = 6$ です。

3.4.3 使用方法

- ・ コンパイル方法

[Linux]

1. PD 制御 RTC のソースディレクトリに移動.

2. コンソールから次のコマンドを入力.

```
$ make all
```

[Windows (Microsoft Visual C++ 2008 利用の場合)]

1. PD 制御 RTC のソースフォルダを開く.
2. PDControlComp_vc9.sln を開き, ソリューションのビルドを実行する.

- 実行方法

[Linux]

PDControlComp を実行.

[Windows]

PDControlComp.exe を実行.

3.4.4 注意事項

- 比例ゲイン, 微分ゲイン, 入力変数のすべての次元が同じでなければエラーとなります.
- 比例ゲイン行列および微分ゲイン行列は正定行列である必要があります.

3.4.5 使用条件

PD 制御 RTC (以下, 本ソフトウェア) の使用に当たっては, 本マニュアル最後に記載されている特記事項に従ってください. 本ソフトウェアに関しては, アカデミック用途に限り, ソースコードの改変およびその派生物の利用を許可します. ただし, いかなる目的であっても, 改変したソースコードならびにその派生物 (バイナリを含む) の配布は禁止します.

3.5 指令値生成 RTC

3.5.1 概要

本 RTC は、制御指令値を出力するユーザコントローラのサンプルです。ユーザ独自の制御アルゴリズムをこのサンプルに従って実装すれば、簡単にロボット制御ができます。

このサンプル実装では、任意次元の状態変数ベクトルを初期値に+0.5 した目標値に等速で変化させます。目標到達時間は 10 秒に固定してあります。

3.5.2 仕様

本 RTC の仕様を以下に示します。

【動作環境】

- OS : Any. (但し, Linux および Windows でのみ動作確認済)
- RTM : OpenRTM-aist-0.4.2 (C++版)

【主要諸元】

- 名前 : UserController
- 言語 : C++

【データポート】

Inport			
名称	型	データ長	説明
Current Position	TimedDoubleSeq	N (注)	現在位置
Current Velocity	TimedDoubleSeq	N (注)	現在速度

Output			
名称	型	データ長	説明
Target Position	TimedDoubleSeq	N (注)	目標位置

【サービスポート】

Consumer port			
名称	サービス名	インタフェース名	説明
Configuration	PDControlSvc0	PDControlService	PD 制御 RTC のサービスコンシューマ. 詳細は PDControlService.idl を参照.

(注) データポートのデータサイズは、RTC の Activate 後に Current Position ポートに最初に
入力されたデータのサイズに合わせられます。また、その値は、初期値として利用されます。

3.5.3 使用方法

- ・ コンパイル方法

[Linux]

1. 指令値生成 RTC のソースディレクトリに移動.
2. コンソールから次のコマンドを入力.

```
$ make
```

[Windows (Microsoft Visual C++ 2008 利用の場合)]

1. 指令値生成 RTC のソースフォルダを開く.
2. PositionControllerComp_vc9.sln を開き、ソリューションのビルドを実行する.

- ・ 実行方法

[Linux]

PositionControllerComp を実行.

[Windows]

PositionControllerComp.exe を実行.

3.5.4 使用条件

指令値生成 RTC (以下、本ソフトウェア) の使用に当たっては、本マニュアル最後に記載されている特記事項に従ってください。本ソフトウェアに関しては、アカデミック用途に限り、ソースコードの改変ならびにその派生物の利用および再配布を許可します。

4 マニピュレータ制御系の構成

上述の RTC の組み合わせにより、様々なマニピュレータ制御系が簡単に構築できます。本章では、以下に示す 4 つの制御系の構築方法を説明します。

- ・ 関節角度制御システム
- ・ 関節速度制御システム
- ・ 手先位置制御システム
- ・ 手先速度制御システム

なお、以下では、RTC を用いたシステムの構成法のみを述べます。実際に制御系として用いる場合、各 RTC の処理やデータの流れを同期させる必要があります。しかし、現在リリースされている OpenRTM-aist-0.4.2 にはコンポーネント間の同期を取るためのフレームワークが提供されていないため、ここでは、処理やデータフローのタイミングについては考慮しません。

4.1 関節角度制御システム

サーボ制御 RTC、PD 制御 RTC を用いることにより、マニピュレータの関節角度を制御するシステムを構築することができます（図 2）。ポートの接続関係は以下の通りです。

Data port		
出力 (RTC:Port)	入力 (RTC:Port)	データ
PA10Manipulator: CurrentJointAngle	PDController: Actual State	現在関節角度 [rad]
PA10Manipulator: CurrentJointVelocity	PDController: Actual Differential State	現在関節速度 [rad/s]
PDController: Control Output	PA10Manipulator: JointVelocity	関節速度制御指令値 [rad/s]
UserController: Target Position	PDController: Target State	目標関節角度 [rad]

本システムを正常に起動させるためには、図 2 の右側の RTC から左へ向かって順に Activate する必要があります。また、PD 制御 RTC の比例ゲインおよび微分ゲインがあらかじめ正しく設定されている必要があります。PD 制御 RTC を起動する前にコンフィグレーションもしくはサービスポートからゲインの設定をしてください。

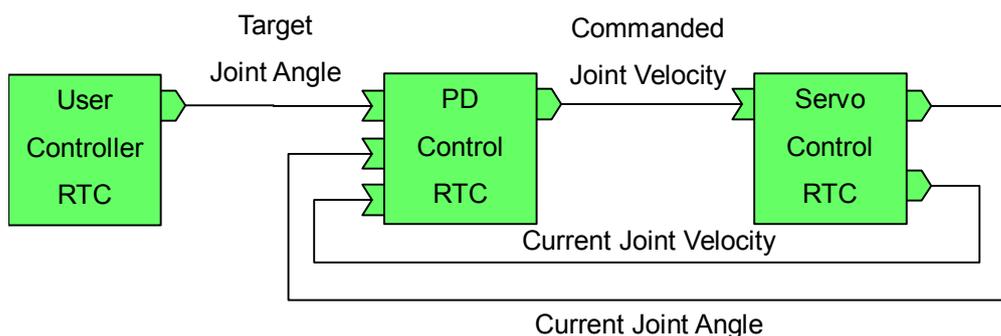


図 2. 関節角度制御システム

4.2 関節速度制御システム

サーボ制御 RTC を単独で用いることにより、マニピュレータの関節速度制御システムを構築することができます（図 3）。ポートの接続関係は以下の通りです。

Data port		
出力 (RTC:Port)	入力 (RTC:Port)	データ
UserController: Target	PA10Manipulator: Joint Velocity	目標関節速度 [rad/s]

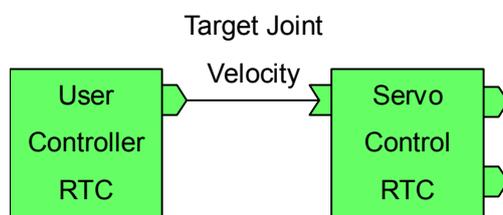


図 3. 関節速度制御システム

4.3 手先位置制御システム

サーボ制御 RTC, 順運動学 RTC, 逆運動学 RTC, PD 制御 RTC を用いることにより、マニピュレータの手先位置制御システムを構築することができます（図 4）。ポートの接続関係は以下の通りです。

Data port		
出力 (RTC:Port)	入力 (RTC:Port)	データ

PA10Manipulator: CurrentJointAngle	FwdKinematicsPA10: Joint Angle	現在関節角度 [rad]
PA10Manipulator: CurrentJointAngle	InvKinematicsPA10: Reference Joint Angle	現在関節角度 [rad]
PA10Manipulator: CurrentJointVelocity	FwdKinematicsPA10: Joint Velocity	現在関節速度 [rad/s]
FwdKinematicsPA10: Tip Position	PDController: Actual State	現在手先位置・姿勢 [m rad]
FwdKinematicsPA10: Tip Velocity	PDController: Actual Differential State	現在手先速度 [m/s rad/s]
PDController: Control Output	InvKinematicsPA10: Tip Velocity	手先速度制御指令値 [m/s rad/s]
InvKinematicsPA10: Joint Velocity	PA10Manipulator: JointVelocity	関節速度制御指令値 [rad/s]
UserController: Target Position	PDController: Target State	目標手先位置・姿勢 [m rad]

本制御システムを正常に起動させるためには、図4の右側のRTCから左へ向かって順にActivateする必要があります。また、PD制御RTCの比例ゲインおよび微分ゲインがあらかじめ正しく設定されている必要があります。PD制御RTCを起動する前にコンフィグレーションもしくはサービスポートからゲインの設定をしてください。

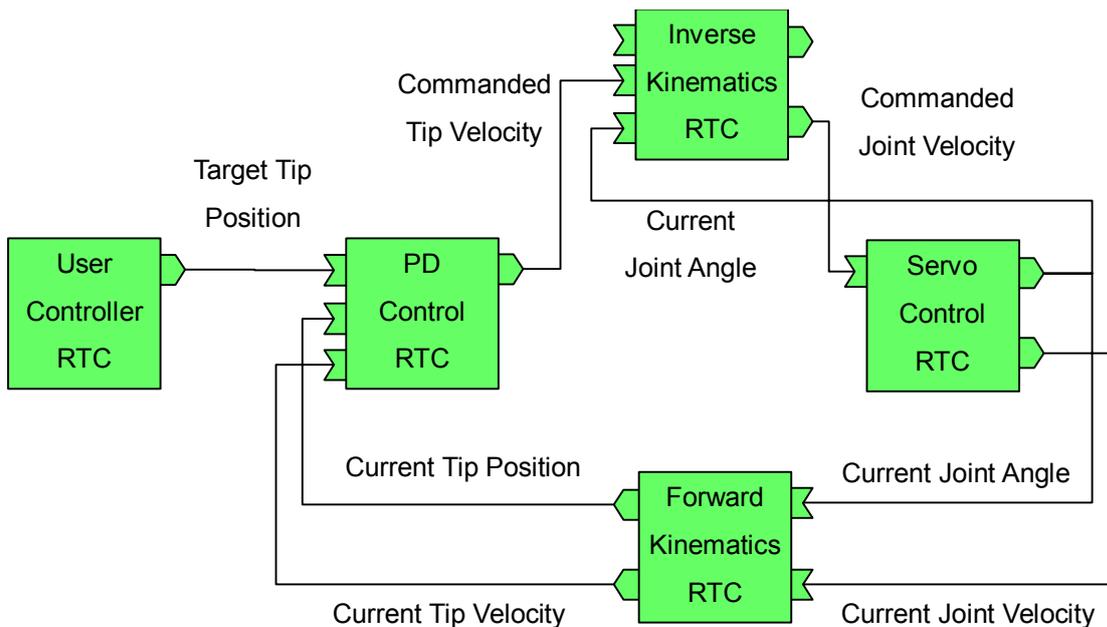


図4. 手先位置制御システム

4.4 手先速度制御システム

サーボ制御 RTC, 逆運動学 RTC を用いることにより, マニピュレータの手先速度制御システムを構築することができます (図 5). ポートの接続関係は以下の通りです.

Data port		
出力 (RTC:Port)	入力 (RTC:Port)	データ
PA10Manipulator: CurrentJointAngle	InvKinematicsPA10: Reference Joint Angle	現在関節角度 [rad]
InvKinematicsPA10: Joint Velocity	PA10Manipulator: JointVelocity	目標関節速度 [rad/s]
UserController: Target	InvKinematicsPA10: Tip Velocity	目標手先速度 [m/s rad/s]

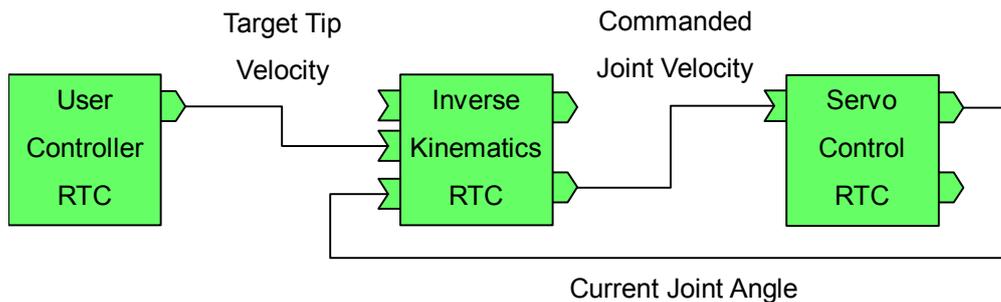


図 5. 手先速度制御システム

5 おわりに

本マニュアルでは, 再利用性を最大限に考慮して開発した, マニピュレータ制御 RTC 群の概要や使用方法を解説しました. マニピュレータ制御系を 4 つの RTC (サーボ制御 RTC, 順運動学 RTC, 逆運動学 RTC, PD 制御 RTC) に分割することによって, RTC の再利用性や制御システムの柔軟性が高くなることを示しました.

開発した RTC 群から構成されるマニピュレータ制御システムの例として, 関節角度制御システム, 関節速度制御システム, 手先位置制御システム, 手先速度制御システムを紹介しました.

6 特記事項

- ・ 本稿で紹介したソフトウェア（以下、当該ソフトウェア）の著作権は産総研に帰属します。
- ・ PA-10 高速制御用ソフトウェアライブラリは、ライフロボティクス株式会社の製品であり、権利はライフロボティクス株式会社に帰属します。取扱いにはご注意ください。
- ・ 当該ソフトウェアの使用に係る損害について、産総研、静岡大学、ライフロボティクス株式会社、およびプログラム開発者は、一切責任を負いません。
- ・ 当該ソフトウェアの使用は、アカデミック目的に限り、無償で許可します。ただし、ソースコードの改変および再配布は禁止します。
- ・ 当該ソフトウェアの全部または一部を商用目的で使用する場合は、産総研までご相談ください。
- ・ 当該ソフトウェアのサポートが必要な場合には、ライフロボティクス株式会社で有償にてご対応致します。お気軽にお問い合わせください。

【連絡先】

独立行政法人 産業技術総合研究所 知能システム研究部門

尹 祐根 (ゆん うぐん、Woo-Keun YOON)

Email: yoon at aist. go. jp (スパム対策のため、at を変換しています)

国立大学法人 静岡大学

清水 昌幸 (Masayuki Shimizu)

Email: tmsimiz at ipc. shizuoka. ac. jp (スパム対策のため、at を変換しています)

ライフロボティクス株式会社

〒305-8568 茨城県つくば市梅園1丁目1番地1

独立行政法人 産業技術総合研究所 つくば中央第2事業所内

TEL/FAX: 029-858-5666

Email: info at liferobotics. jp (スパム対策のため、at を変換しています)