

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

機能仕様書

オープンソース移動知能モジュール群
走行系モジュール編

V e r . 1 . 0

2011年6月30日

R T C 再利用技術研究センター

目次

1. はじめに	1
1. 1. 本書の適用範囲	1
1. 2. 関連文書	1
1. 3. 本書を読むにあたって	1
2. 機能仕様	2
2. 1. 機能概要	2
2. 2. モジュール構成	2
3. RTC 仕様	4
3. 1. RefHardRh2 (RH2 号機台車制御コンポーネント)	4
3. 1. 1. 機能概要	4
3. 1. 2. 動作環境	4
3. 1. 3. ポート情報	5
3. 1. 4. コンフィグレーション	6
3. 1. 5. 入出力データフォーマット	7
3. 1. 6. サービスインターフェース仕様	8
3. 1. 7. 設定ファイル	8
3. 2. Odometry (オドメトリ推定コンポーネント)	9
3. 2. 1. 機能概要	9
3. 2. 2. 動作環境	9
3. 2. 3. ポート情報	9
3. 2. 4. コンフィグレーション	10
3. 2. 5. 入出力データフォーマット	11
3. 2. 6. 設定ファイル	13
3. 3. SwitchInputRTC (自律／操作切り替えコンポーネント)	14
3. 3. 1. 機能概要	14
3. 3. 2. 動作環境	14
3. 3. 3. ポート情報	14
3. 3. 4. コンフィグレーション	15
3. 3. 5. 入出力データフォーマット	15
3. 3. 6. 設定ファイル	16
4. 特記事項	17

1. はじめに

1. 1. 本書の適用範囲

本書は、ロボット向けミドルウェア OpenRTM 上で車体速度を入力とする対向二輪型の移動ロボットを経路に沿って走行させる際の台車制御、およびオドメトリ推定機能に関する智能モジュールについて記述した文書である。

本書は「次世代ロボット知能化技術開発プロジェクト」における移動智能モジュール群として構成し、動作確認したものである。本構成以外の利用における有用性、汎用性について保証するものではない。

1. 2. 関連文書

本書は以下に示す移動智能モジュール関連文書の一部である。

表 1-1 関連文書

No.	文書名	備考
1	機能仕様書 自己位置姿勢推定モジュール	
2	機能仕様書 走行系	本書
3	機能仕様書 経路計画・軌道追従	
4	機能仕様書 障害物検知・衝突回避	
5	機能仕様書 オペレータ操作	

1. 3. 本書を読むにあたって

本書は RT ミドルウェア(以下 RTM)、RT コンポーネント(以下 RTC)を用いたロボットシステム開発者を対象に記述されており、RTM、RTC や関連ツールに関する一般的な知識を持つことを前提とする。RT ミドルウェア、RTC については下記を参照のこと。

OpenRTM-aist Official Website :

<http://www.openrtm.org/>

2. 機能仕様

本知能モジュールについての機能仕様、構成を以下に記述する。

2. 1. 機能概要

本知能モジュールは、対向二輪型の移動ロボットの走行を実現する際の差動駆動車輪型移動ロボット「リファレンスハードウェア」の自律移動制御を実現する。本知能モジュールで使用する移動ロボットは、前川製作所作成の「リファレンスハードウェア(2号機)」を対象としている。

本知能モジュールは、RefHardRh2 (RH2号機台車制御コンポーネント)、RefHardRh2 (RH2号機台車制御コンポーネント) および SwitchInputRTC (自律/操作切り替えコンポーネント) によって構成されている。

本知能モジュールの機能は、ハードウェアの制御およびオドメトリ推定であるが、自己位置姿勢モジュール、障害物検知および回避モジュールを利用することで、現在位置姿勢から目標位置姿勢までの自律移動、障害物を避けた安全な経路を生成、その経路に従う台車を移動実現することが可能である。

2. 2. モジュール構成

本知能モジュールは以下の構成で動作する。移動知能モジュール全体での位置付けは、移動知能モジュール群全体図を参照のこと。

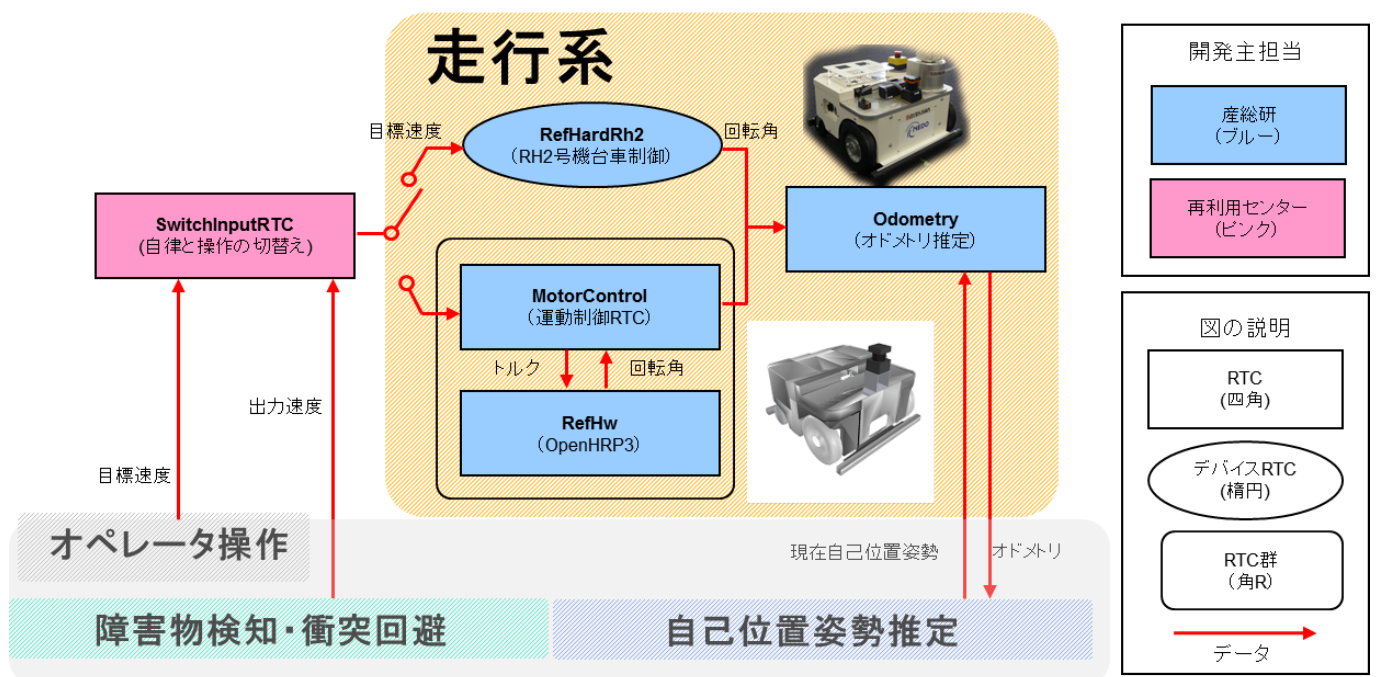


図 2-1 走行系モジュール構成

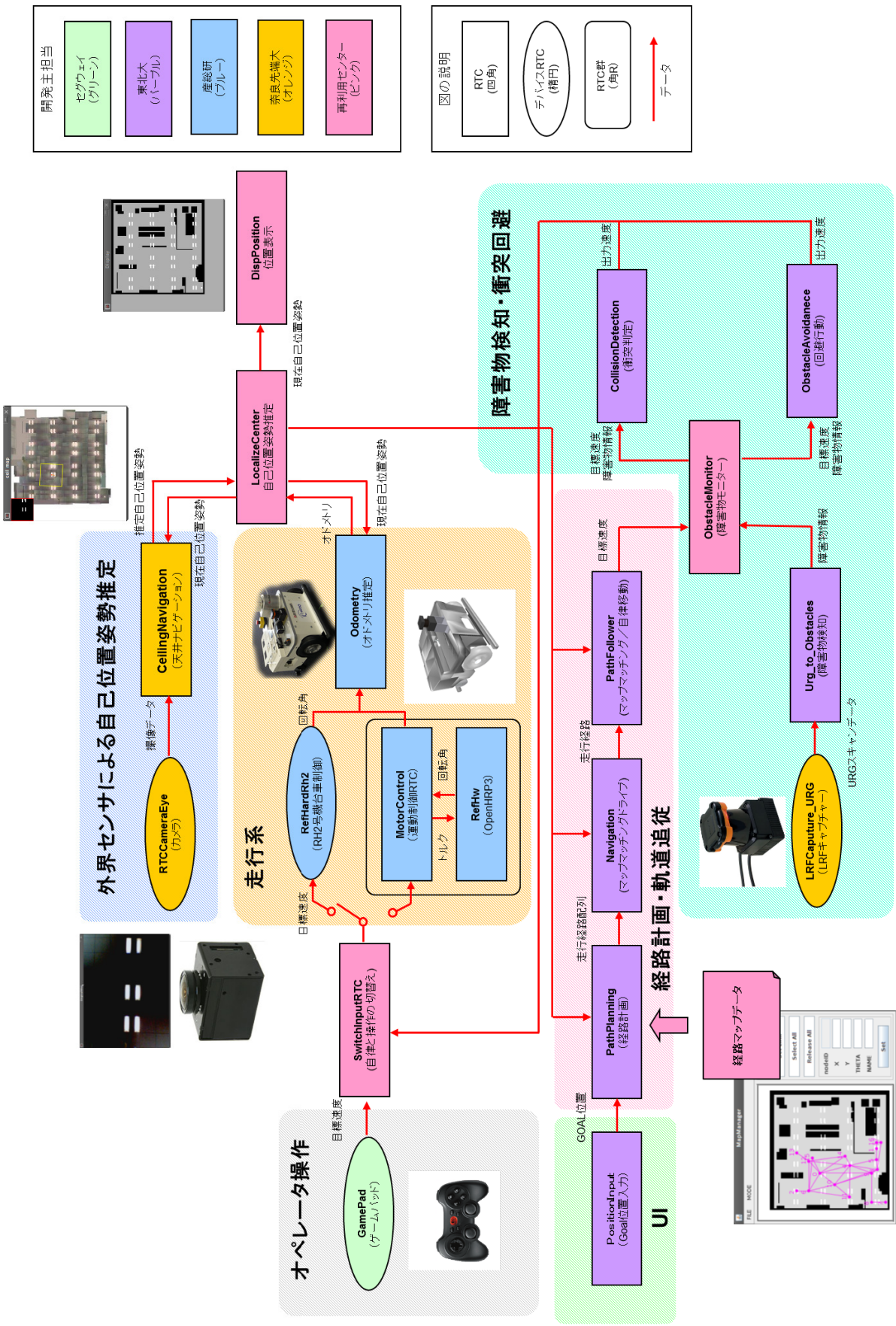


図 2-2 移動知能モジュール群全体図

3. RTC 仕様

3. 1. RefHardRh2（RH2 号機台車制御コンポーネント）

3. 1. 1. 機能概要

本 RTC は、リファレンスハードウェア 2 号機(RefHard2)の台車部の速度制御を行うコンポーネントである。RH2 号機台車制御コンポーネントは目標速度をデータポートより取得し車輪回転角度をデータポートより出力する。また、「駆動停止」「駆動開始」等の指令をサービスポートより取得する。

RefHard2 台車駆動用モータードライバとの通信は“svmlib”ライブラリを使用し RS485 シリアル通信にて行う。大まかなデータ流れのイメージ図を以下に記す。

本 RTC の利用には、現在位置と目的位置を入力するモジュールが必要である。

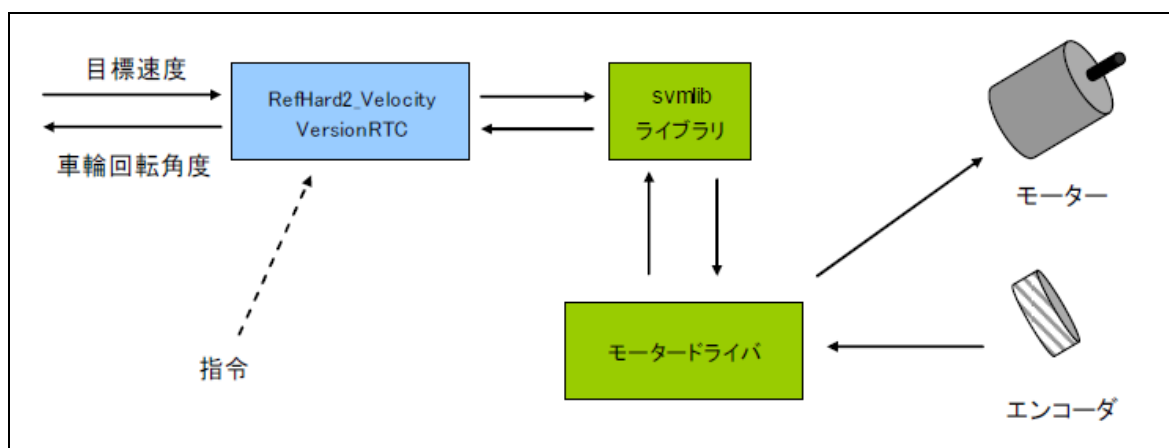


図 3-1 RH2 号機台車制御コンポーネントデータの流れ

3. 1. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux (Ubuntu10.04 LTS)
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

本 RTC は OpenHRP 3 と MotorControl（運動制御 RTC）によって、シミュレータ上での利用も可能であるが、本ドキュメントは実機走行についての記述のため、詳細については割愛する。シミュレータ詳細と使い方については OpenInvent サイトを参照のこと。

OpenInvent の Web ページ

<http://www.openrtp.jp/INVENT/>

3. 1. 3. ポート情報

A) データポート (InPort)

名称	型	説明
InTargetVelocity	IIS::TimedVelocity2D	並進と旋回の目標速度

B) データポート (OutPort)

名称	型	データ長	説明
OutWheelAngle	TimedDoubleSeq	2	エンコードカウンタから計算した 車輪回転角度

C) サービスポート (Provider)

サービス名	インターフェース名	説明
InventGUIRefHard	RefHard	走行／停止指令を取得する
BumpRefHard	RefHard	停止指令を取得する

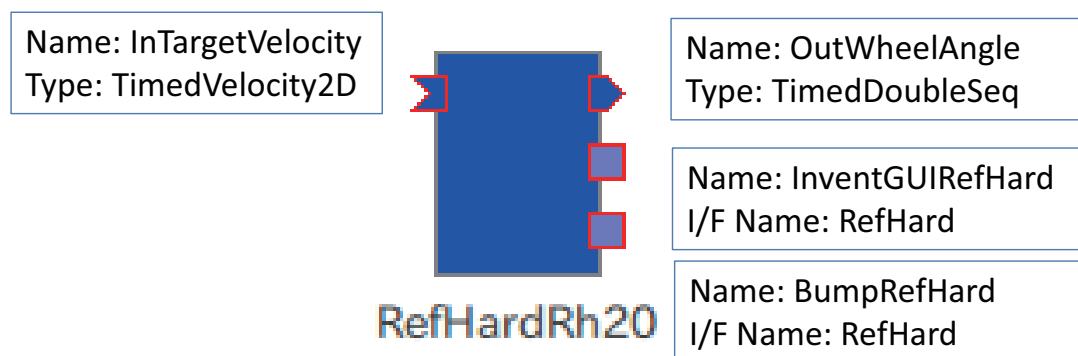


図 3-2 RefHardRh 単体モジュール構成図

3. 1. 4. コンフィグレーション

名称	型	デフォルト値	説明
DeviceFileName	string	/dev/ttyUSB0	RS485 シリアル通信のためのデバイス ファイル名
gain_P	int	10	駆動モーターの位置制御モードでの PID 制御パラメータ
gain_I	int	0	
gain_D	int	0	
gain_VP	int	50	駆動モーターの速度モードでの PID 制 御パラメータ
gain_VI	int	0	
gain_VD	int	0	
Velocity_limit	double	10.0	台車の移動速度制限値
Acceleration_limit	double	50.0	台車の移動加速度制限
WheelTread	double	0.441	車輪間の距離
WheelRadiusLeft	double	0.1	左右車輪の半径
WheelRadiusRight	double	0.1	
GearRatio	double	66.22	駆動モーターから車輪までのギア比
EncoderCount	int	2000	駆動モーター1 回転のエンコーダカウ ント値
EncoderConstant_L	double	1.0	左右車輪のエンコーダ
EncoderConstant_R	double	1.0	

3. 1. 5. 入出力データフォーマット

以下にデータ型および詳細について示す。移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

A) 入力 : InTargetVelocity

型 : IIS::TimedVelocity2D

```
struct TimedVelocity2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Velocity2D data;
    sequence<double> error;
};
```

要素	説明
data.vx	移動速度[m/s]
data.va	回転速度[rad/s]

B) 出力 : OutWheelAngle

型 : TimedDoubleSeq

```
struct TimedDoubleSeq{
    Time tm;
    sequence<double> data;
};
```

要素	説明
data	車輪回転角度[rad/s]

3. 1. 6. サービスインターフェース仕様

各サービスインターフェース仕様を本章に記述する。

3. 1. 6. 1. インターフェース名

A) 基本情報

インスタンス名	m_BumpProvPort, m_InventGUIProvPort
変数名	m_BumpRefHard, m_InventGUIRefHard
IDL ファイル	RefHardRhPro.idl
インターフェース型	RefHardSVC_impl

B) サービス関数一覧

No.	関数名	説明
1	Stop0	停止指令
2	Go0	走行指令

C) サービス関数詳細

関数名	Stop0	
引数	なし	
戻り値	型	説明
	void	
説明	Stop 指令 走行中の移動台車を停止させる	

関数名	Go0	
引数	なし	
戻り値	型	説明
	void	
説明	Go 指令 停止中の移動台車を走行させる	

3. 1. 7. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 2. Odometry（オドメトリ推定コンポーネント）

3. 2. 1. 機能概要

本 RTC は、実機 RH2 号機台車を制御する RefHardRh2（RH2 号機台車制御コンポーネント）から、現在の車輪角度などの情報を取り込み、ロボットの自己位置姿勢推定（オドメトリ）値を算出するコンポーネントである。算出に際して、自己位置姿勢推定モジュールからの現在位置姿勢推定情報を取得し、その後も常に更新された当情報を受け、オドメトリ値の補正等を行う。最終的に算出した自己位置姿勢推定（オドメトリ）値を自己位置姿勢推定モジュールに送信する。

なお、「差動駆動型移動台車ロボットの駆動輪を結ぶ車軸の中心」をオドメトリ値と定義している。

3. 2. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux（Ubuntu10.04 LTS）
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 2. 3. ポート情報

A) データポート（InPort）

名称	型	データ長	説明
CurrentWheelAngle	TimedDoubleSeq	2	車輪回転角度(絶対値) data[0]：左車輪回転角度[rad] data[1]：右車輪回転角度[rad]
LocalizedPosition	IIS::TimedPose2D	1	ローカライズ後の自己位置姿勢 x：X 位置[m] y：Y 位置[m] theta：姿勢角[rad] id：ID 用配列（未使用） error：エラーコード用配列（未使用） tm：タイムスタンプ[sec][nsec]

B) データポート (OutPort)

名称	型	データ長	説明
OdometryPosition	IIS::TimedPose2D	1	オドメトリによる自己位置姿勢 x : X 位置[m] y : Y 位置[m] theta : 姿勢角[rad] id : ID 用配列 (未使用) error:エラーコード用配列 (未使用) tm : タイムスタンプ[sec][nsec]

3. 2. 4. コンフィグレーション

a) デフォルト値

名称	型	デフォルト値	説明
leftWheelID	int	0	左駆動輪の ID 番号
rightWheelID	int	1	右駆動輪の ID 番号
radiusOfLeftWheel	double	0.1	左駆動輪の車輪半径[m]
radiusOfRightWheel	double	0.1	右駆動輪の車輪半径[m]
lengthOfAxle	double	0.441	左右駆動輪間の長さ[m]
radiusOfBodyArea	double	0.45	車体全体を円で囲んだ場合の半径[m] 安全を考慮したエリア定義用

b) 実機 Reference Hardware ver2 用 (Odometry.conf)

名称	型	デフォルト値	説明
leftWheelID	int	0	左駆動輪の ID 番号
rightWheelID	int	1	右駆動輪の ID 番号
radiusOfLeftWheel	double	0.0988	左駆動輪の車輪半径[m]
radiusOfRightWheel	double	0.0988	右駆動輪の車輪半径[m]
lengthOfAxle	double	0.456	左右駆動輪間の長さ[m]
radiusOfBodyArea	double	0.45	車体全体を円で囲んだ場合の半径[m] 安全を考慮したエリア定義用

3. 2. 5. 入出力データフォーマット

以下にデータ型詳細およびオドメトリ位置姿勢算出ロジックについて示す。移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

1) データ型詳細

A) 入力 : CurrentWheelAngle
型 : TimedDoubleSeq

```
struct TimedDoubleSeq{
    Time tm;
    sequence<double> data;
}
```

要素	説明
data[0]	左車輪回転角度[rad]
data[1]	右車輪回転角度[rad]
tm	タイムスタンプ[sec][nsec]

B) 入力 : LocalizedPosition
型 : IIS::TimedPose2D

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

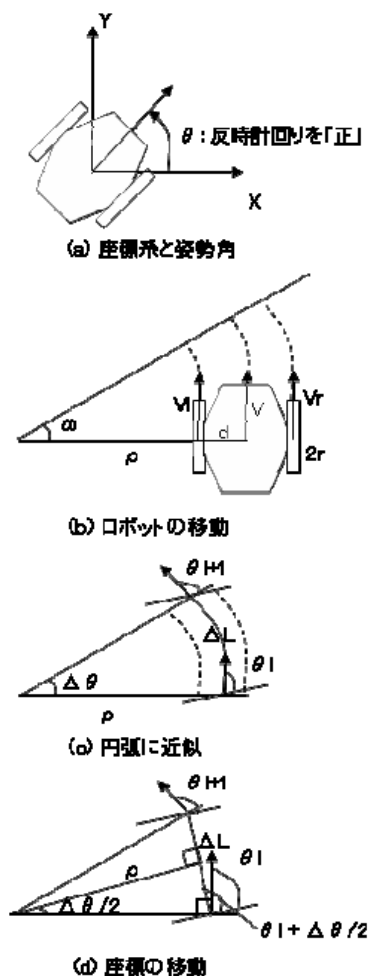
要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]
data.heading	姿勢角[rad]
tm	タイムスタンプ[sec][nsec]

C) 出力 : OdometryPosition
型 : IIS::TimedPose2D

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]
data.heading	姿勢角[rad]
tm	タイムスタンプ[sec][nsec]

2) オドメトリ位置姿勢算出ロジック



車輪の回転角速度を、 ϕ/d とした場合、車輪半径を r とすれば、両輪の速度は、
 $V_l = r * d\phi/dt = (\rho - d) * \omega$
 $V_r = r * d\phi/dt = (\rho + d) * \omega$
 となり、ロボット中心速度 V 、旋回速度 ω 、曲率半径 ρ は

$$V = (V_r + V_l)/2$$

$$\omega = (V_r - V_l)/2d$$

$$\rho = d * (V_r + V_l) / (V_r - V_l)$$

となる。

TimeStepを Δt とすると、
 $\Delta \theta = \omega * \Delta t$ より、現在の姿勢角度 θ_{i+1} は、
 $\theta_{i+1} = \Delta \theta + \theta_i$
 が得られ、

Δt 間に移動した距離 ΔL は、
 (1) ほぼ直線に近い時
 $\Delta L = \Delta t * (V_{i+1} + V_i)/2$
 (2) $\Delta \theta$ が微少な時
 $\Delta L = \rho * \Delta \theta$
 (3) $\Delta \theta$ が大きい時
 $\Delta L = 2 * \rho * \sin(\Delta \theta / 2)$
 となる。

現在位置 X_{i+1}, Y_{i+1} は
 $X_{i+1} = X_i + \Delta L * \cos(\Delta \theta / 2 + \theta_i)$
 $Y_{i+1} = Y_i + \Delta L * \sin(\Delta \theta / 2 + \theta_i)$
 となる。

参照元 : <http://www.mech.tohoku-gakuin.ac.jp/rde/contents/course/robotics/wheelrobot.html>

図 3-3 オドメトリ位置姿勢算出ロジック

3. 2. 6. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

3. 3. SwitchInputRTC（自律／操作切り替えコンポーネント）

3. 3. 1. 機能概要

本 RTC は、自律移動機能モジュールやオペレータ操作モジュールからの複数の目標速度の入力を走行系に出力するための、自律／操作切り替えコンポーネントである。

3. 3. 2. 動作環境

本 RTC の動作環境（動作 OS、RT ミドルウェア、開発環境など）について記述する。

動作 OS	Linux（Ubuntu10.04 LTS）
開発言語	C/C++
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE
依存パッケージ	なし

3. 3. 3. ポート情報

A) データポート（InPort）

名称	型	説明
inVelJoy	IIS::TimedVelocity2D	目標速度（ゲームパッド操作）
inVelAuto	IIS::TimedVelocity2D	目標速度（自律移動）

B) データポート（OutPort）

名称	型	説明
outVel	IIS::TimedVelocity2D	目標速度

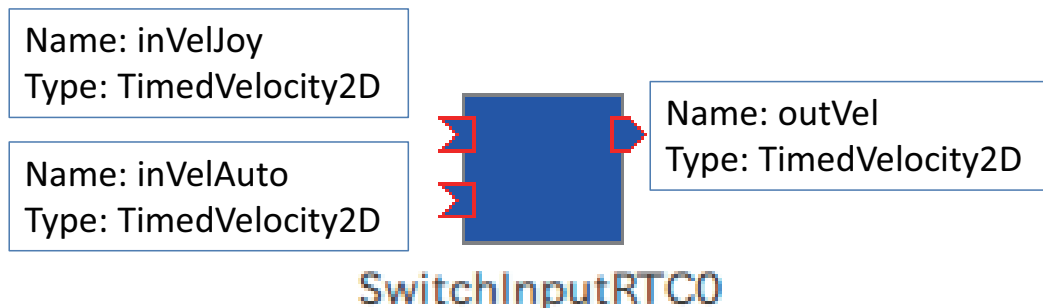


図 3-4 SwitchInput 単体モジュール構成図

3. 3. 4. コンフィグレーション

特に無し。

3. 3. 5. 入出力データフォーマット

以下にデータ型および詳細について示す。移動サブワーキンググループの共通インタフェース仕様については、「移動 SWG 共通 IF 案 101008.pdf」を参照のこと。

A) 入力 : inVelJoy

型 : IIS::TimedVelocity2D

```
struct TimedVelocity2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Velocity2D data;
    sequence<double> error;
};
```

要素	説明
data.vx	X 方向移動速度[m/s]
data.vy	Y 方向移動速度[m/s]
data.va	回転速度[rad/s]

B) 入力 : inVelAuto

型 : IIS::TimedVelocity2D

```
struct TimedPose2D {
    RTC::Time tm;
    sequence<long> id;
    RTC::Pose2D data;
    sequence<double> error;
};
```

要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]
data.heading	姿勢角[rad]
tm	タイムスタンプ[sec][nsec]

C) 出力 : outVel

型 : IIS::TimedVelocity2D

```
struct TimedPose2D {  
    RTC::Time tm;  
    sequence<long> id;  
    RTC::Pose2D data;  
    sequence<double> error;  
};
```

要素	説明
data.position.x	X 位置[m]
data.position.y	Y 位置[m]
data.heading	姿勢角[rad]
tm	タイムスタンプ[sec][nsec]

3. 3. 6. 設定ファイル

本モジュールでは、特殊なドライバ、ライブラリを使用しておらず、使用するにあたって、OpenRTM-aist-1.0.0-RELEASE・RTSystemEditor の標準的な設定で問題はなく、新たにドライバ、ライブラリなどのインストールは必要ない。

4. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構（NEDO 技術開発機構）の「次世代ロボット知能化技術開発プロジェクト」（平成 20 年～平成 23 年度）において、評価を目的として構成されたものである。
- RH2 号機台車制御コンポーネント、オドメトリ推定コンポーネント、自律／操作切り替えコンポーネントの著作権は産業技術総合研究所が所有しています。
- 本モジュールのライセンスについては各 RTC 付属のライセンス条項をご参照下さい。
- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

RTC 再利用技術研究センター

〒101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 1303 号室

Tel/Fax : 03-3256-6353 E-Mail : contact@rtc-center.jp