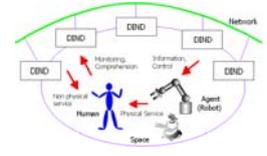


## 分散レーザレンジファインダの キャリブレーション支援 コンポーネント群

佐々木毅  
東京大学生産技術研究所  
橋本研究室

### 開発の背景

- 知能化空間
  - センサやアクチュエータを空間に分散配置し、ネットワーク化
- センサのキャリブレーションが必要
  - 各センサの座標系(ローカル座標系)において得られたデータを知能化空間の座標系(ワールド座標系)へ変換
  - 多数のセンサのキャリブレーションには多くの手間を要する



- **分散センサのキャリブレーション支援コンポーネント群を開発する**

### 開発の背景

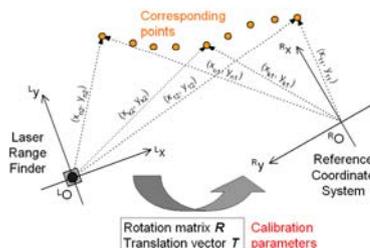
- レーザレンジファインダ(LRF)は主要なセンサの1つとして期待される
  - 設置が比較的容易
  - 知能化空間の基本機能の1つである物体トラッキングにおいても対象に特別なタグを持たせる必要がない
  - 空間の地図も獲得可能
- **本コンポーネント群では特にLRFの位置・姿勢のキャリブレーションに着目**

### 開発環境

- OS: Ubuntu Linux 7.0.4
- RTミドルウェア: OpenRTM-aist-0.4.1-RELEASE
- コンパイラ: gcc 4.1.2
- CORBA : omniORB 4.0.7
- ACE : ACE 5.4.7-12
- Eclipse : Eclipse 3.2
- Java実行環境: Sun Java 1.5.0-11-1

### LRFの位置・姿勢のキャリブレーション

- 環境中の2点以上の点について、それらのローカル座標系とワールド座標系の位置が既知ならばその対応から計算可能



### 開発したコンポーネント群

- LRFコンポーネント (LRFComponent)
  - 北陽電機(株)のLRF (URG-04LX)をRTコンポーネント化\*
- 移動体トラッキングコンポーネント (SimpleTracker)
  - LRFのスキャンデータから移動物体の位置を出力
- キャリブレーションコンポーネント (LRFCalibration)
  - 同一物体の2つの座標系での位置を入力として受け取り、その対応点から2つの座標系の位置・姿勢の関係を出力
- 座標変換コンポーネント (CoordTrans2D)
  - キャリブレーションパラメータに基づいてセンサ座標系から基準座標系への座標変換を実行
- 入力コンポーネント (ConsoleIn2)
  - コンソールから入力した値を順に2つのOutPortに出力

\*デバイス処理には商品付属のサンプルプログラムの一部を使用

### LRFCalibrationの使用例

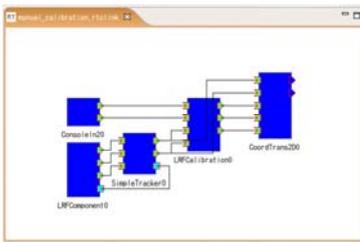
- キャリブレーションオブジェクトを用いた絶対的な位置・姿勢の手動キャリブレーション(※)
- 移動物体を用いたLRF間の相対的な位置・姿勢の自動キャリブレーション(※)
- 移動ロボットを用いた絶対的な位置・姿勢の自動キャリブレーション
- 天井カメラの位置・姿勢のキャリブレーション
- (※)は本コンポーネント群のみで実現可能な機能

### キャリブレーションオブジェクトを用いた絶対的な位置・姿勢の手動キャリブレーション

- レーザレンジファインダで検知しやすい物体を環境内の既知の位置に順番に置き、その位置情報に基づいてレーザレンジファインダの絶対的な位置・姿勢を計算
- 高精度なキャリブレーション
- 使用するコンポーネント
  - LRFCComponent ..... 1つ
  - SimpleTracker ..... 1つ
  - LRFCalibration ..... 1つ
  - ConsoleIn2 ..... 1つ
  - CoordTrans2D ..... 1つ

### システム構築

- コンポーネントの接続



- パラメータ設定
- アクティブ化

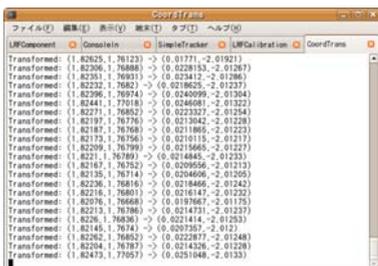
### キャリブレーション

- キャリブレーションオブジェクト(LRFで検知しやすい物体)を既知の位置に配置
- ConsoleIn2のコンソールからキャリブレーションオブジェクトの位置の座標を入力



### 結果の確認

- 物体を既知の位置に置き、結果を確認
  - (0, -2.0)に物体を置いた場合

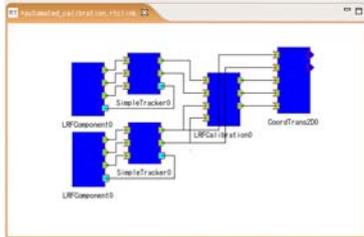


### 移動物体を用いたLRF間の相対的な位置・姿勢の自動キャリブレーション

- 2台のレーザレンジファインダの観測領域の重なりを利用し、一方のレーザレンジファインダを基準とした相対的な位置・姿勢を獲得
- 手動キャリブレーションと比べると精度は劣るが、簡単にキャリブレーションが可能
- 使用するコンポーネント
  - LRFCComponent ..... 2つ
  - SimpleTracker ..... 2つ
  - LRFCalibration ..... 1つ
  - CoordTrans2D ..... 1つ

## システム構築

- コンポーネントの接続



- パラメータ設定
- アクティブ化

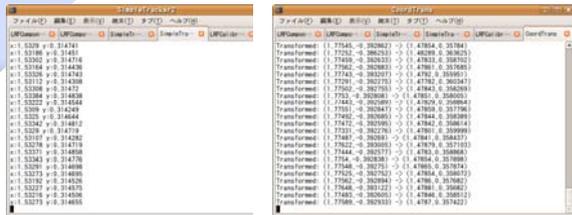
## キャリブレーション

- 2台のLRFの観測領域の重なり部分で、人間やロボットなどを移動させる



## 結果の確認

- 物体を適当な位置に置き結果を比較



基準となるLRFでの物体の位置

キャリブレーションを行ったLRFでの物体の位置

## その他の使用例

- 移動ロボットを用いた絶対的な位置・姿勢の自動キャリブレーション
  - 移動物体を用いた自動キャリブレーションでは、移動物体の絶対的な位置情報がわからないため、キャリブレーションも相対的
  - 基準となるLRFの代わりに移動ロボットの自己位置推定モジュールから得られる位置情報を利用すれば、絶対的な位置・姿勢を計算可能
- 天井カメラの位置・姿勢のキャリブレーション
  - LRFCalibrationはLRFのキャリブレーション用モジュールとして開発されたが、2次元平面上での位置を獲得するセンサであれば本コンポーネント群を同じように適用可能

## まとめ

- 分散レーザレンジファインダ (LRF) のキャリブレーション支援コンポーネント群の開発
  - 各機能要素のコンポーネント化
    - LRF(北陽電機URG04-LX)
    - 移動物体トラッキング
    - キャリブレーション
- コンポーネント群の使用例の提案と動作確認
  - キャリブレーションオブジェクトを用いた絶対的な位置・姿勢の自動キャリブレーション
  - 移動物体を用いたLRF間の相対的な位置・姿勢の自動キャリブレーション

Thank you for  
your attention!

知的制御システム 橋本研究室  
Intelligent Control System Laboratory - Hashimoto Lab.  
<http://dfs.iis.u-tokyo.ac.jp/>

### 謝辞

- レーザレンジファインダコンポーネント (LRFComponent) の開発に当たりましては、東京大学生産技術研究所橋本研究室の川路浩平氏、Drazen Brscic氏、佐々木毅が作成したC++ UrgLaserクラスのコードの一部を利用しております。川路浩平氏、Drazen Brscic氏の両名に感謝申し上げます。

### ライセンス(公開条件)について

- 著作権の放棄はしませんが、非商用利用であれば自由にご利用ください
- LRFコンポーネント (LRFComponent)について
  - URG-04LX付属のサンプルコードの一部を使用しているため、UrgLaserクラスはライブラリとして提供
  - 追記: 成果発表会の会場にて北陽電機様よりソースコード公開の許可をいただきました

### 参考資料 各コンポーネントのインタフェース概説

(詳細はマニュアルを参照)

### LRFComponent

- 北陽電機(株)のLRF (URG-04LX)をRTコンポーネント化

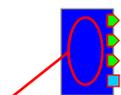


OutPort

名称	型	説明
ScanData	TimedShortSeq	センサが獲得した距離データ [mm]
StartPoint	TimedShort	スキャン開始点(ステップ)
EndPoint	TimedShort	スキャン終了点(ステップ)

### LRFComponent

- 北陽電機(株)のLRF (URG-04LX)をRTコンポーネント化



Configuration変数

名称	型	デフォルト値	説明
DeviceName	char*	"/dev/ttyACM0"	デバイスが接続されたポートの名前
ScanRate	int	19200	通信速度 [bps]
ScanStart	int	0	スキャン開始点(ステップ)
ScanEnd	int	768	スキャン終了点(ステップ)
ScanStep	int	1	ScanStartとScanEndの間のステップ数

### LRFComponent

- 北陽電機(株)のLRF (URG-04LX)をRTコンポーネント化

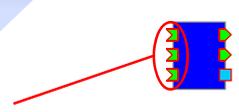


サービスポート (provider)

関数名	引数	戻り値の型	説明
getMaxRange	なし	short	センサの最大計測可能距離[mm]を返す
getMinAngle	なし	short	最小ステップに対応する角度[deg]を返す
getMaxAngle	なし	short	最大ステップに対応する角度[deg]を返す
getMaxStep	なし	short	最大ステップを返す

### SimpleTracker

- LRFのスキュンデータから移動物体の位置を出力

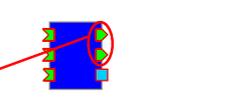


InPort

名称	型	説明
ScanData	TimedShortSeq	LRFCOMPONENT が出力した距離データ [mm]
StartPoint	TimedShort	スキュン開始点(ステップ)
EndPoint	TimedShort	スキュン終了点(ステップ)

### SimpleTracker

- LRFのスキュンデータから移動物体の位置を出力

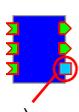


OutPort

名称	型	説明
X	TimedDouble	観測領域内で最大の大きさをもつ移動物体の位置のx座標 [m]
Y	TimedDouble	観測領域内で最大の大きさをもつ移動物体の位置のy座標 [m]

### SimpleTracker

- LRFのスキュンデータから移動物体の位置を出力

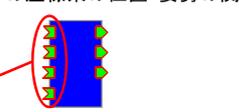


サービスポート (consumer)

提供される関数についてはレーザレンジファインダコンポーネント (LRFCOMPONENT) のサービスポートを参照

### LRFCalibration

- 同一物体の2つの座標系での位置を入力として受け取り、その対応点から2つの座標系の位置・姿勢の関係を出力

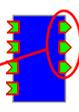


InPort

名称	型	説明
GlobalX	TimedDouble	物体の基準となる座標系での位置のx座標 [m]
GlobalY	TimedDouble	物体の基準となる座標系での位置のy座標 [m]
LocalX	TimedDouble	物体のセンサ座標系での位置のx座標 [m]
LocalY	TimedDouble	物体のセンサ座標系での位置のy座標 [m]

### LRFCalibration

- 同一物体の2つの座標系での位置を入力として受け取り、その対応点から2つの座標系の位置・姿勢の関係を出力

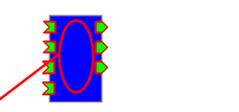


OutPort

名称	型	説明
Tx	TimedDouble	基準座標系に対するセンサ座標系のx方向の並進量 [m]
Ty	TimedDouble	基準座標系に対するセンサ座標系のy方向の並進量 [m]
Theta	TimedDouble	基準座標系に対するセンサ座標系の回転角度 [rad]

### LRFCalibration

- 同一物体の2つの座標系での位置を入力として受け取り、その対応点から2つの座標系の位置・姿勢の関係を出力

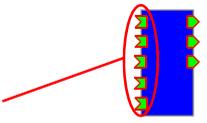


Configuration変数

名称	型	デフォルト値	説明
data_num	int	0	キャリブレーションに用いる対応点の数

### CoordTrans2D

- キャリブレーションパラメータに基づいてセンサ座標系から基準座標系への座標変換を実行

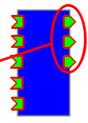


InPort

名称	型	説明
Xin	TimedDouble	センサ座標系での位置のx座標 [m]
Yin	TimedDouble	センサ座標系での位置のy座標 [m]
Tx	TimedDouble	基準座標系に対するセンサ座標系のx方向の並進量 [m]
Ty	TimedDouble	基準座標系に対するセンサ座標系のy方向の並進量 [m]
Theta	TimedDouble	基準座標系に対するセンサ座標系の回転角度 [rad]

### CoordTrans2D

- キャリブレーションパラメータに基づいてセンサ座標系から基準座標系への座標変換を実行

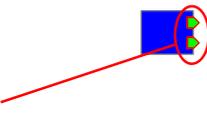


OutPort

名称	型	説明
Xout	TimedDouble	基準座標系に変換した位置のx座標 [m]
Yout	TimedDouble	基準座標系に変換した位置のy座標 [m]

### ConsoleIn2

- コンソールから入力した値を順に2つのOutPortIに出力



OutPort

名称	型	説明
out	TimedDouble	コンソールから最初に入力された数値
out2	TimedDouble	コンソールから2番目に入力された数値