



# ROS/Movelt!のFAロボットの 組立工程動作教示への適用

2018/09/14

株式会社 日立製作所 研究開発本部  
デジタルテクノロジーイノベーションセンタ 知能情報研究部

石川忠明

# Contents

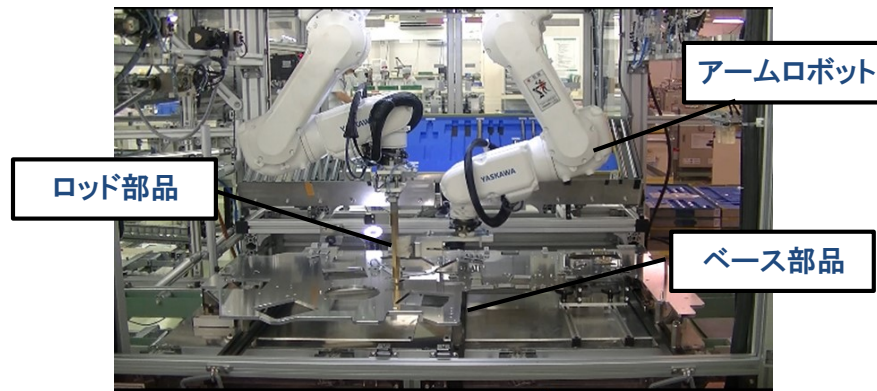
---

1. 背景と課題
2. 解決手法
3. まとめ

---

# 1. 背景と課題

- ・(株)日立ハイテクノロジーズでは医用分析装置のユニット組立工程に組み立てロボットセルを導入。
- ・垂直多関節型ロボット(安川電機:MOTOMAN-MH3F天吊り設置:6DOF)2体。
- ・可動台(1DOF)上のベース部品に、部品供給トレイからピックした部品を組付け。
- ・ツールチェンジャー有。作業内容によりハンド交換。(挿入、ねじ込み、把持等)
- ・両ロボットの可動域は一部交錯する。持ち替え等の共同作業有。(例:把持→ねじ込み)



・組み立て対象機種変更時、ロボット動作教示に要する負担が大きい。

・既製動作シミュレータによる動作教示

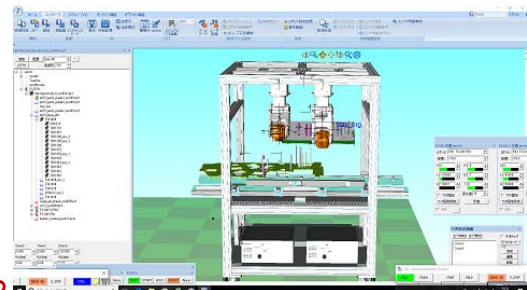
・作業者がシミュレータに慣れるまでに要する時間大。

・工程表を読み解き、動作を指定する部分は手入力。

・環境CADデータの精度、長尺部品の取り回し、等の問題。

→・人件費を含めた総作業コストが高くなる傾向。

・実機でのチェックが残るなら最初から手作業で。



・全手作業での動作教示

・動作経路の創出や各軸毎教示の手間が大きい。

・稼働中の設備のため、夜間の空き時間にしか作業時間が取れない。

→・教示作業者の負担大。



・教示作業の簡単化、低コスト化が必要。

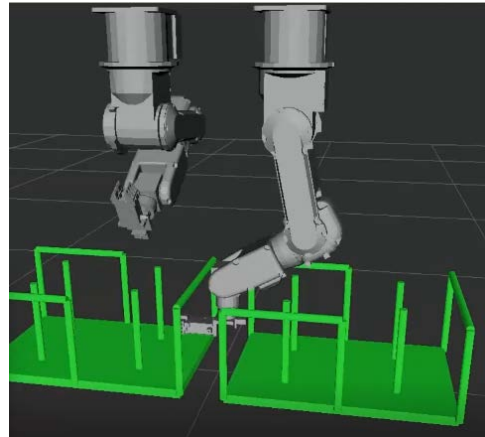
- ・教示作業の簡単化、低コスト化。
- ・既存機器の構成変更は最小に。(安価で実現)



- ・ROS/Movel!による動作経路作成
  - ・部署内でアームロボット動作経路生成に実績あり。(現時点では研究用途)
    - ・環境: Ubuntu14.04.5 (VirtualBOX上)  
ROS Indigo Igloo, Movel!
  - ・OSSの利用で短期間で実験、初期評価ができた。
    - ◆結果・複雑な環境でも比較的短時間で動作経路を提示。
      - ・提示される経路が効率的。(避けすぎない)



- ・組み立て動作教示への適用
  - ・バッチ処理で、工程表、環境CADデータ、部品リスト等から、工程ごとに始点終点を算出。ROS/Movel!でアーム動作経路を計画。ロボットコントローラ用ジョブファイルに変換。
  - ・教示作業者は実機でジョブファイルでの動作を確認し、微調整(パラメータ修正)を行う。



### ◆OSS(ubuntu,ROS,Movelt!)を利用した製品製造で、社内品証を通るか？



- ・ROS/Movelt!でアーム動作経路を生成し、その結果を基にロボットコントローラ用ジョブファイルを生成する。
- ・教示作業者はジョブファイルを実機で動作確認し、微調整(パラメータ修正)を行う。



- ・製品製造時は、**動作確認済みのジョブファイル**に従って、ロボットは動作する。

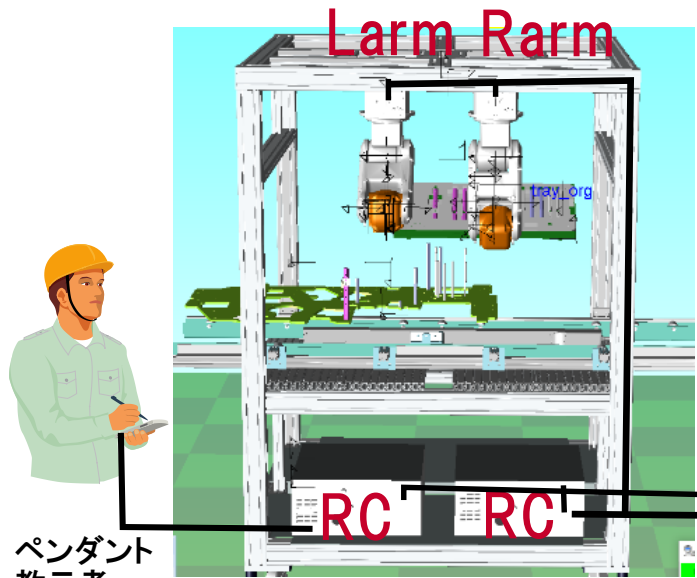
→社内品証をクリア可。

---

## 2. ジョブファイル作成手法



## 2-1 実機構成概略



ロボットセル概略図

ペンダント  
教示者

・現状のシステムでは各※RCはFA-PCからの信号により、  
ジョブファイル通りにそれぞれのアームを動かす。

・ジョブファイル中に記載された動作経路間は、コマンドに  
従って**自動補間計算された軌跡**となる。

ジョブファイル例  
(JBIファイル)

```

/JOB
//NAME 180320_76810631_02-03_L
//POS
///NPOS 7,0,0,0,0
///TOOL 10
///POSTYPE PULSE
///PULSE
C00000=225,-69012,-29402,484,-34674,-360
C00001=214,-69053,-29423,526,-34633,-344
C00002=212,-69727,-29872,520,-34249,-340
C00003=27,-128739,-69200,38,-711,-26
C00004=25,-129613,-69783,30,-214,-21
C00005=24,-129973,-70022,27,-10,-19
C00006=1,-129997,-70005,-8,-7,2
//INST
///DATE 2018/03/20 12:41
///COMM aa
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ C00000 VJ=60.00 PL=0
MOVJ C00001 VJ=60.00
MOVJ C00002 VJ=60.00
MOVJ C00003 VJ=60.00
MOVJ C00004 VJ=60.00
MOVJ C00005 VJ=60.00
MOVJ C00006 VJ=60.00
END
    
```

動作経路  
(中継点  
関節角)  
データ

PLCプログラム  
+  
動作計画  
ジョブファイル作成

動作  
コマンド

※RC:Robot Controller

## 2-2 データ間引き

- ・ジョブファイル中に記載された**動作経路間**は、**コマンドに従って自動補間計算される**。
- ・Movelt!の出力した動作経路(関節角形式点列データ)をそのまま列挙すると、ジョブファイルが大きくなるだけでなく、自動補間計算による動作遅延が大きくなる。
- ・**経路データの間引き**が必要となるが、適切に行わないと自動補間された経路とMovelt!の出力経路データとのずれが大きくなり、干渉が発生する可能性がある。
- ・データ間引きの方法を比較した。(MOTOSimにてジョブファイルを実行、評価)

間引き基準	間引き後データ量 (元データ=100%)	最大関節角度誤差の平均
全関節の角加速度が小さい要素	約9.4%	0.69°
全関節の角速度が小さい要素	約94.6%	0.91°

- ・結果：  
全関節の各加速度が小さい要素を間引く方が有利。

## 2-3 ジョブファイル作成手順まとめ

(1)前処理:CAD出力からモデル作成。

バッチ  
処理

(2)工程表、部品表等から1工程における動作始点終点抽出、ジョブファイルのひな型作成。

(3)ROS/Movelt!!による動作経路計画。

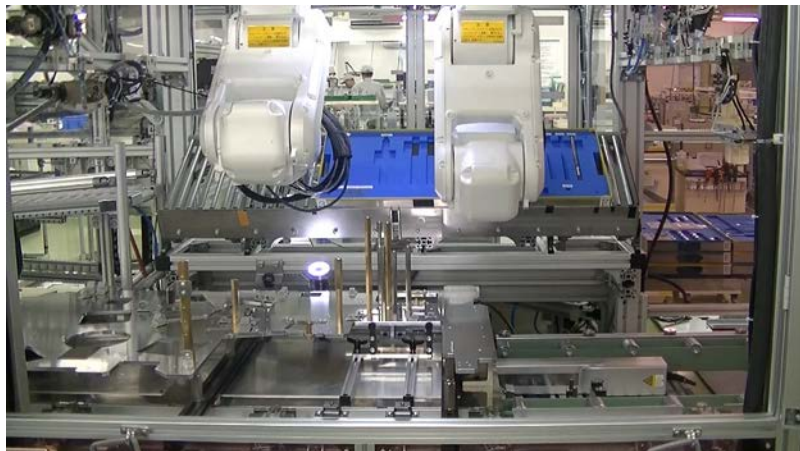
(4)経路データ出力ファイルから可動限界角チェック。(問題があったら再計画)

(5)後処理:経路データ出力を間引きして、ジョブファイルに埋め込み。

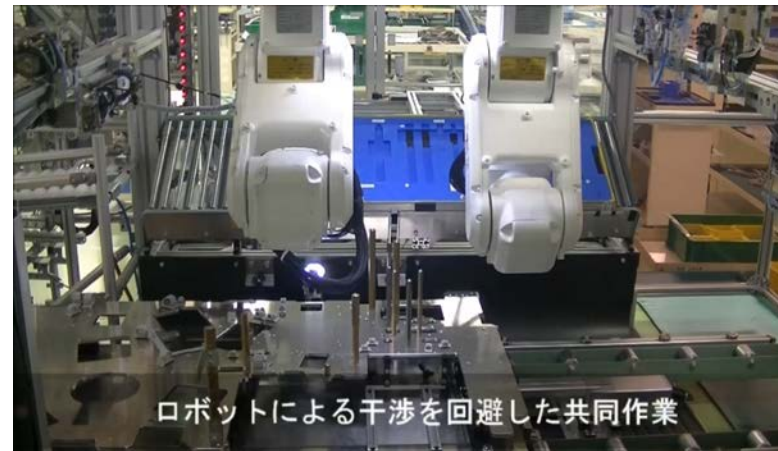
手作業 (6)調整:ジョブファイルを実機で動作チェックし、パラメータを調整し、完成。



- ・作業者の手作業は動作チェックおよび調整のみで約15分。(ロッド部材1本あたり)  
現場での作業時間は90分→15分に短縮。



手作業教示



本方式(ROS/MoveIt!利用)

---

### 3. まとめ

- ・組み立てロボットセルのアームロボットの動作教示作業における教示作業員の負担軽減のために、教示作業の簡単化を目指した。
- ・動作経路生成にROS/Movelt!を利用し、バッチ処理でジョブファイルまで自動生成するシステムを構築した。
- ・作業員は、ジョブファイルの動作チェックと微調整を行うだけでよく、現場での作業時間は90分から15分に短縮できた。(ロッド部材一本当たり)
- ・ROS/Movelt!で計画した経路での動作時間は、手作業で教示した動作での時間と遜色ない。
- ・今後、他の部材への適用実験を行い、全体として効率的なシステムに仕上げていく。

謝辞: 実験、撮影にご協力いただいた(株)日立ハイテクノロジーズ 小山 昌宏 主管技師  
小河原昭吾 技師に感謝いたします。

**END**

---

## ROS/Movelt!のFAロボットの組立工程動作教示への適用

2018/09/14

株式会社 日立製作所 研究開発本部  
デジタルテクノロジーイノベーションセンタ 知能情報研究部

石川忠明

**HITACHI**  
Inspire the Next 