

資 料

6 軸アーム作業セルによるメカトロシステムの開発

大峠 忍*・辰野 恭市**

1. 目 的

ユーザの仕様に沿った多種多様な製品を作れるような生産システムを構築しようとする動きがインダストリー4.0に代表されるIoTシステムの流れにある。多種少量生産のためには製品の仕様変更に対応できる作業セルの実現が不可欠である。このような作業セルで作業をするロボットシステムでは、仕様変更のたびに教示に時間を割くことは生産効率を大きく下げる要因になる。そのため、タスクレベルの指示で、例えば「ボルトを掴め」などの指示で動作し、ごく簡単な教示のみで作業を行えるようにする必要がある。

本開発の目的はタスクレベルの指示で、かつごく簡単な教示で作業可能な作業セル用ロボットコントローラの構築例を示すことである。作業セルの例として、射出成型取り出しロボットのLinear Motionガイド（以下LMガイド）のボルト締めを取り上げる。辰野ら名城大学グループによる研究ではタスクレベルで動くロボットコントローラのアーキテクチャが提案されている¹⁾。コントローラはタスクレベルの指示で作業を実行できるようにするものである。タスクレベルの指示では、作業手順をC言語のコマンド関数で記述する。この手順を記述したプログラムを作業プログラムと呼ぶことにする。次に作業開始位置を、ハンドアイカメラを作業対象物に向けてラフに対象物の位置を教示する。曖昧な教示位置をハンドアイカメラによるボルトヘッド及びボルト穴の検出によるビジュアルフィードバックで補正する。このロボットコントローラのアーキテクチャを用いてボルト締め作業セルを構築し、作業できることを検証したので、その結果について報告する。

2. 方 法

2.1 LMガイドボルト締め作業システム

2.1.1 作業内容

この作業セルでは人とロボットアームが協働で作業する。作業台上に、ボルトを挿入したパレット（図1）とブロック・レール（図2）を人がラフに置き、ブロックにレール

をボルトで4カ所固定する。作業者が「ボルトを締めろ」と指示すると、ロボットアームは先端に装着されているボルト締め工具でパレット上のボルトを真空吸着によって把持し、LMガイドのレールのボルト穴と把持したボルトの位置合わせを行い、ボルト締めを行う。

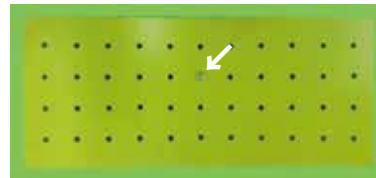


図1 ボルト（矢印位置）を挿したパレット



図2 LMガイドブロックとレール

2.1.2 ハードウェア構成

図3にボルト締め作業をおこなう作業セルを示す。作業台の上には、ロボットアーム、ボルトをセットしたパレット、LMガイドがある。ロボットアームの先端には、力覚センサ・工具・ハンドアイカメラがついている。工具は六角レンチであり、真空吸着によりボルトを把持することが

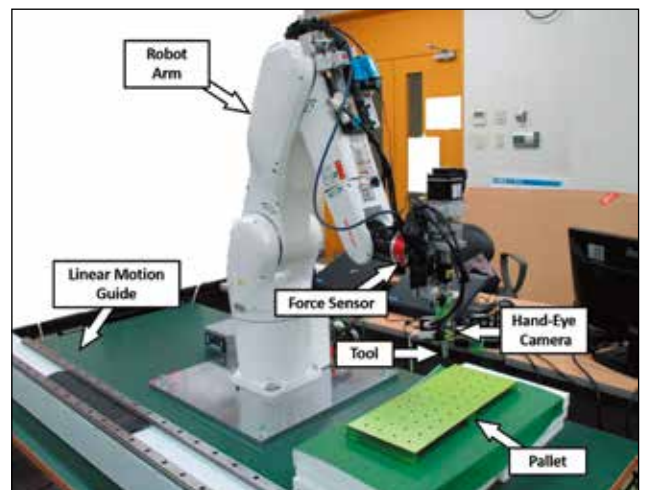


図3 ボルト締め作業セルの機構

*次世代パワーエレクトロニクス技術開発プロジェクトチーム,**名城大学（現：島根県産業技術センター）

でき、トルクセンサにより一定トルクで締め付けることが可能である。

2.1.3 コントローラーアーキテクチャ

コントローラーアーキテクチャを図4に示す。このコントローラーはHuman Robot Interface (HRI) とTask Planner (TP), Arm Controller (AC), Vision System (VS), Computer Graphics (CG) の5つのソケット通信により接続されたサブシステムにより構成される。

HRIは操作画面をGUIで表示し、操作者は操作画面からタスクを指示するとHRIはTPへメッセージを送信する。

TPは指示されたタスクに応じたタスクプログラムを実行し、ロボットアームを動かすための軌道生成などの計算やアームを動作させるなどのコマンドをその他のサブシステムへ送信する。

ACはTPからの移動命令を受信し、その命令に応じてロボットアームを指定した関節角や位置へ移動を行う。

VSはカメラ画像を入力し、ボルトなどの対象物の認識や位置計測を行う。

CGは作業対象物の位置や現在のロボットアームの位置など、その他のサブシステムのパラメータを表示する。

HRIから「パレット上のボルトを掴め」と作業指示されると、HRIはTPにメッセージ“GraspBoltOnPallet”を送る、TPがメッセージを受け取ると、作業手順を記述したプログラム“GraspBoltOnPallet”を起動する。作業プログラムはロボット言語のようなコマンド関数を用いて記述されている。コマンド関数がコールされるとAC, VSサブシステムにそれぞれメッセージ“AC_MoveArmUnderVelocityControl”, “VS_StartVisualTrackingBoltOnPalletInHandeye”を送ってAC, VSがアームを動かしボルトなどの作業対象物を認識する。

2.2 パレット上のボルトの把持

2.2.1 作業プログラム

パレット上のボルトを把持するための作業プログラムをリスト1に示す。作業を分析し、C言語のコマンド関数（ほかのサブシステムに送る命令）を用いて記述したものである。

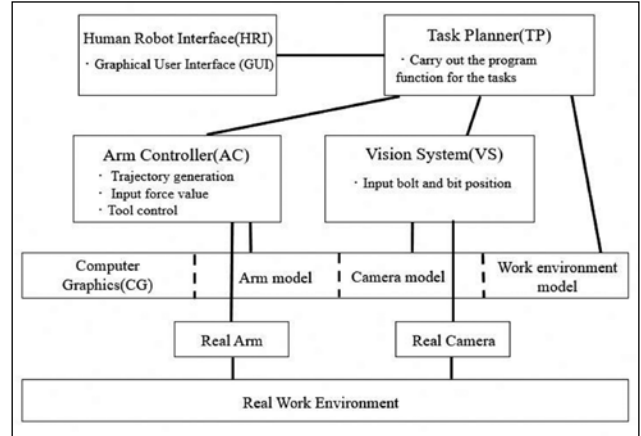


図4 作業セルのコントローラーアーキテクチャ

リスト1 ソースプログラム“GraspBoltOnPallet”

```
void GraspBoltOnPallet(void)
{
  //(1) Open a robot controller RC8.
  AC_OpenRC8();
  //(2) Servo on the robot arm.
  AC_ServoOnArm();
  //(3) Point the hand eye camera at the pallet.
  AC_PointHandeyeAtPallet();
  //(4) Approach to bolt by visual feedback.
  AC_ApproachBoltUnderVisualFeedback();
  //(5) Align the tool tip to the bolt by teaching.
  AC_MoveToBoltHeadByTeaching ();
  //(6) Contact with bolt by force control.
  AC_ContactToBoltUnderForceControl();
  //(7) Bolt gripping by vacuum suction.
  AC_VacuumChuckOn();
  //(8) Pull up the arm tip.
  AC_PullArmUp(100);
  //(9) Servo off the robot arm.
  AC_ServoOffArm();
  //(10) Close robot controller RC8.
  AC_CloseRC8();
}
```

2.2.2 用意するパラメータ

「GraspBoltOnPallet」を実行するには、作業開始のアーム位置（図5）とボルトヘッド認識のための画像テンプレート（図6）、アプローチ終了点からボルトヘッドまでの位置偏差（図7）、力一定制御する際の力の大きさの4つのパラメータをセットする必要がある。

2.2.3 ビジュアルフィードバックによるアプローチ

認識したボルトの位置U, Vおよびボルトの大きさからアームの速度を生成する。図8のようにボルトの現在位置

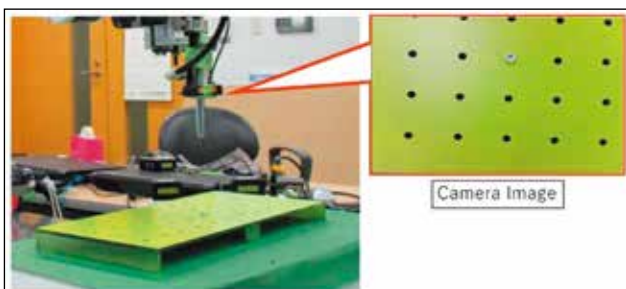


図5 ボルト把持開始のアーム位置

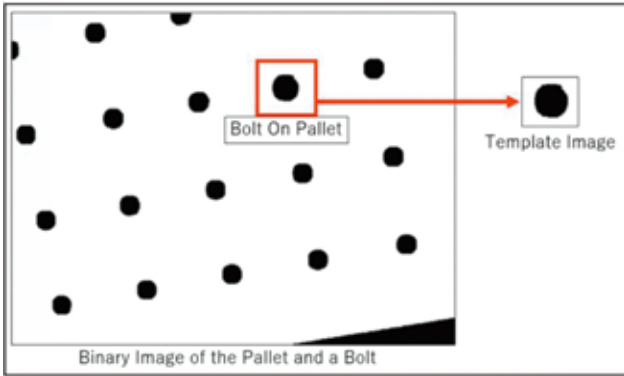


図6 ボルトヘッド認識のための画像テンプレート

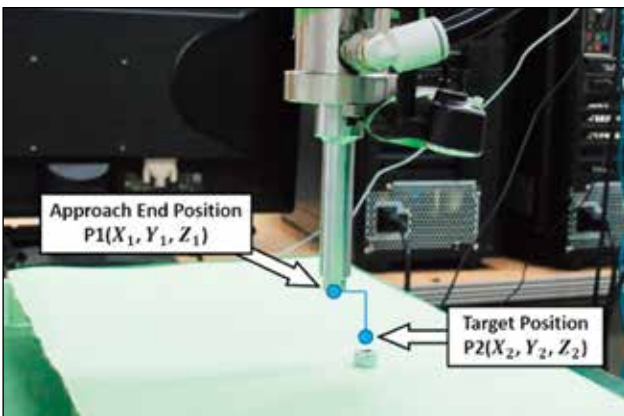


図7 ボルトと工具先端の位置偏差

と目標位置（画像の中心）から画像上のU, Vの偏差を計算し、(1)式によりアームの速度を計算する。ここで、 (V_x, V_y, V_z) はアーム速度であり、 (K_x, K_y, K_z) が速度調整ゲイン、TargetSizeがボルトの大きさの目標値（図6右枠）、BoltSizeが現在のボルトの大きさ（図6赤枠）である。ボルトのトラッキングはボルトサイズが一定以上目標サイズに近づいたら終了する。

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} K_x \Delta U \\ -K_y (\text{TargetSize} - \text{BoltSize}) \\ K_z \Delta V \end{pmatrix} \dots (1)$$

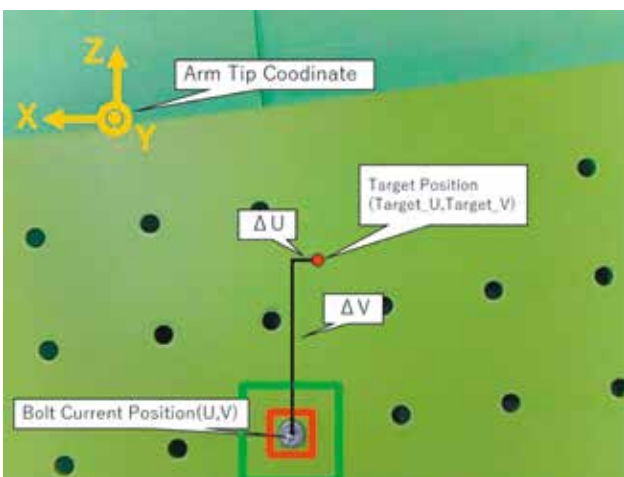


図8 ボルト現在位置と目標位置の偏差

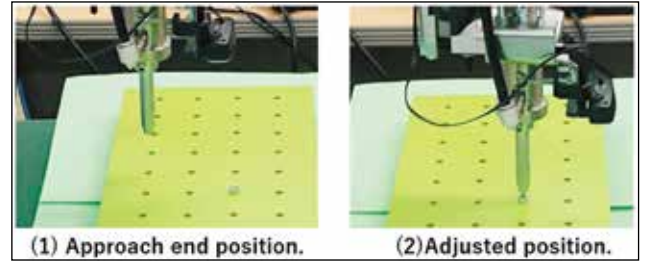


図9 ボルト把持開始のアーム位置

2.2.4 教示を用いたボルトヘッドへのアジャスト

次に工具とボルトヘッドの位置合わせを行う。ビジュアルフィードバックにより、カメラ画像上の中心に接近した状態では、ボルトヘッドと工具の位置合わせができていない。(図9(1)) アプローチが終了した点(P1)からボルトヘッド(P2)までの変位 $\Delta P = P2 - P1 = (X2 - X1, Y2 - Y1, Z2 - Z1)$ を教示しておき、ロボットアームの位置制御によりボルトヘッド直上まで移動する。(図9(2))

2.2.5 力一定制御を用いたボルトヘッドの接触

ボルトヘッドのアジャストが完了すると、工具はボルトヘッドの真上で停止した状態である。そのため、工具を鉛直下向きに下げることによって、ボルトを把持する。この時、工具がボルトに強く当たってしまうのを防ぐために力一定制御を用いる。力一定制御とは、2.2.2で設定した力の大きさで、力一定になるように動作する力制御である。

2.3 LMガイドのボルト締め

2.3.1 作業プログラム

把持したボルトをLMガイドのボルト穴へ締め付けるための作業プログラムをリスト2に示す。これも作業を分析し、C言語のコマンド関数を用いて記述する。

リスト2の(3)、(4)、(5)は2.2.3および2.2.4と同じ手法で行っている。そのため、(6)の力制御によるボルト締め付手法について2.3.2および2.3.3で述べる。

リスト2 ソースプログラム“TightenBoltToLMGuide()”

```
void TightenBoltToLMGuide()
{
  //(1)Open a robot controller RC8.
  AC_OpenRC8();
  //(2)Servo on the robot arm.
  AC_ServoOnArm();
  //(3)Point the hand eye camera at the LM Guide.
  AC_PointHandeyeAtLMGuide();
  //(4)Approach to bolt hole by visual feedback.
  AC_ApproachBoltHoleUnderVisualFeedback();
  //(5)Align the bolt to the bolt hole by teaching.
  AC_MoveToBoltHoleByTeaching();
  //(6)Tighten bolt by force control and arm position.
  AC_TightenBoltUnderForceControl();
}
```



```

//(7)Release the bolt by turn off vacuum suction.
AC_VacumChuckoff();
//(8)Pull up the arm tip
AC_PullArmUp();
//(9)Servo off the robot arm
AC_ServoOffArm();
//(10)Close robot controller RC8.
AC_CloseRC8();
AC_CloseRC8();
}
    
```

2.3.2 用意するパラメータ

「LMガイドのボルト締め付け」を実行するにはボルト締め開始のアーム位置と、ボルト穴認識のための画像テンプレート、アプローチ点からボルト穴までの変位、ボルト締め時の力制御における力の大きさの4つのパラメータをセットする必要がある。

2.3.3 力制御を用いたボルト締付

工具とボルト穴の位置合わせを行った後、工具を鉛直下向きに動かし、ボルトとボルト穴を嵌め合わせて締め付けを行う。このとき、鉛直下向きの力一定制御だけでは、ボルトとボルト穴の位置合わせにずれが生じた際に嵌め合わせることができない。(図10)そのため、ずれが生じた場合、力制御(ハイブリッド制御)を用いた修正を行い、ボルトとボルト穴を一致させて締め付けを行うようにする。

ハイブリッド制御では、Y軸負方向に力一定制御による押しつけを行い、X-Z軸方向にインピーダンス制御を用いる。インピーダンス制御では式(2)の運動方程式で示す、指定したインピーダンスをもった仮想物体を想定し、外部環境からアーム先に力Fが加わった時の応答をその仮想物体の応答と一致させる。

$$M \frac{d^2 \Delta r}{dt^2} + D \frac{d \Delta r}{dt} + K \Delta r = F \quad \dots(2)$$

(2)式では、Mが仮想慣性、Dが仮想粘性、Kが仮想バネ、 Δr が中性点からの変位、Fは外部環境(ボルト嵌め合い位置からのズレなど)によりアームが受ける外力である。このような制御によりアームが変位して中性点(ボルトの嵌め合い位置)に到達すると外力がなくなるので、それ以

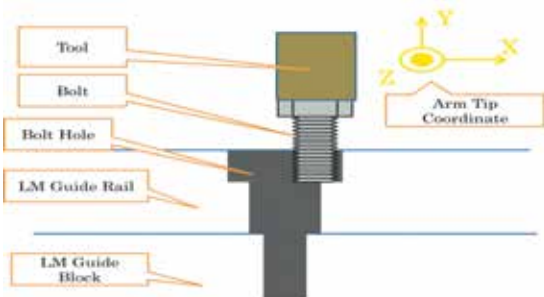


図10 ボルト締め付け時のハイブリッド制御

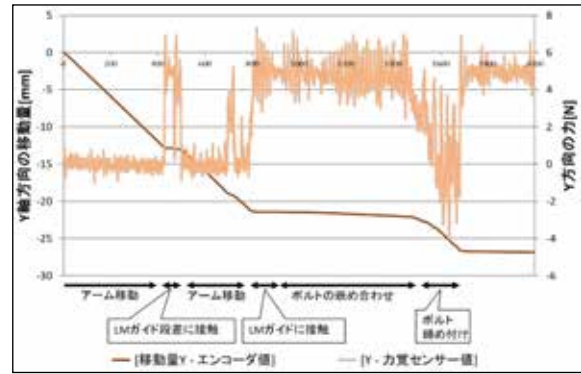


図11 ボルト締め時の力と変位(Y軸)

降アームはX-Z平面内では変位しないこととなる。

アームを変位させながらY軸負方向へ押しつけを行い、定められた時間Y軸負方向の力が力一定制御の設定値を継続し、かつアームの変位がない場合に、ボルトの嵌め合わせが完了したと判定する。ハイブリッド制御により嵌め合わせが完了した際の鉛直下向きの座標を取得し、鉛直下向きの変位とボルト穴深さを勘案し、工具の回転速度を変化させ、締め付け中盤のみ高速でおこなう。

3. 結果

ボルト締め作業セルプログラムを用いてボルト締め作業を行い、ボルトの把持および締め付けを行う事ができた。

図11にボルト締め時のアーム手先位置と力のデータ(Y軸)を示す。ビジュアルフィードバックとティーチングにより、ボルト穴直上まで移動し(リスト2の(4),(5)および図11の横軸0~約400の領域)、力制御を用いてボルト締めを行った。ビジュアルフィードバックによるズレにより、ボルト穴の段差に接触した(図11の横軸450付近)が、力制御による修正によって(図11の横軸800付近)嵌め合わせができ、ボルトを締めることができたことがわかる。

タスクレベルの指示で、且つ、ごく簡単な教示で作業可能な作業セル用ロボットコントローラの構築例を示すことができた。

今後は課題解決、ドキュメント等の更なる環境整備を行い、システムインテグレータがシステムを検討するための作業実験が可能な状態に整備する。

文献

- 1) Koichi Kurabe; Yukiko Kato; Motoki Koike; Kouji Jinno; Kyohei Yamashita; Kazumasa Kito; Mohammed Tahri Sqalli; Kyoichi Tatsuno. A Robot Controller for Power Distribution Line Maintenance Robot Working by Task-Level Command. 2016 IEEE/SICE International Symposium on System Integration.2016.p.441-446.