

データ抽象化を用いた制御アルゴリズム単純化によるロボットの多種ワーク高速ハンドリングの実現

阿部 将佳, 川上 真司, 金井 聡庸, 齊藤 ゆみ

近年、日本をはじめとする製造業からサービス業へのシフトが進んだ国において、質の高い労働者の人件費が高騰しており、人手が中心となっていた中品種中量生産現場におけるロボットの自動化ニーズが高まっている。そこで我々は、中品種中量生産の組立前工程で必須となるキitting工程に着目し、生産性を落とさず人の置き換えが可能となるよう、(1) 人同等以上の速度で、(2) 位置姿勢を合わせながら、(3) 多様な部品を決められた場所に並べられるロボット技術の実現性を検討した。

既存技術で課題となる多様な部品への対応のうち、ハードウェア面の課題は我々の既存ハンドを改良することで対応した。本稿では、部品の品種毎に対応が必要となりプログラム工数が増大するソフトウェア課題に対して、ロボットのタスク実行に必要な抽象化データと、ロボットが取得するセンサデータを自動で抽象化データに変換し制御に用いるアーキテクチャを定義し、プログラム工数の増大なくタスクを完遂できるロボットによる高速ハンドリング手法を紹介する。また、本手法を用いてキittingの実システムを構築し、(1)~(3) を達成したことも報告する。

Realization of the High-Speed Robot Manipulation for Variety Parts by Data Abstraction and Simplified Control

ABE Masayoshi, KAWAKAMI Shinji, KANAI Akinobu and SAITOH Yumi

In recent years, the labor cost of high-quality workers has been rising in Japan and other countries that have shifted from manufacturing to services, and there is a growing need to automate robots in medium- to low-volume production, which has been done mainly by hand. Therefore, we focused on the kitting process, which is essential in the pre-assembly process of medium- and low-volume production, and examined the feasibility of a robot technology that can (1) perform at a speed equivalent to or faster than that of a human, (2) adjust its position and orientation, and (3) arrange various parts in a predetermined location, so that it can replace human workers without reducing productivity.

Among the issues with the existing technology, the hardware issues were addressed by improving our existing hand. In this paper, we define the abstract data required for robot task execution and an architecture that automatically converts the sensor data acquired by the robot into abstract data for control, in order to solve the software problem that requires support for various parts and increases the manpower required for programming. This paper introduces a high-speed handling method using a robot that can complete tasks without increasing the programming effort. We also report the construction of an actual kitting system using our method and the achievement of (1) to (3).

1. はじめに

従来、産業ロボットは、自動車業界の塗装や溶接など、精度と効率を要求される繰り返し作業を中心に活用されて

きた。一方、様々な種類の物を扱い、位置誤差の補正など環境に合わせて作業することが要求される現場では人が主に従事してきた。近年、日本をはじめとする製造業からサービス業へのシフトが進んだ国においては、質の高い労働者の人件費が高騰しており、出荷現場における完成品の

Contact : ABE Masayoshi masayoshi.abe@omron.com

箱詰めや食品工場における食品の整列など、形状が似た製品を扱う少品種多量生産現場の工程で産業ロボットの導入が進み始めている。今後は、ロボットの中品種中量生産現場への適用が期待されている。そこで、本研究では中品種中量生産のデジタル機器の組立工程の前工程として必須となるキッティングに着目し、自動化を検討した。人からロボットに置き換えて、生産性を維持するためには、(1) 人同等以上の速度で、(2) (部品の) 位置姿勢を合わせながら、(3) 多様な部品を決められた場所 (以後、「プレイス場所」と呼ぶ) に並べられる能力がロボットに求められる。

従来、部品 (ワーク) を人同等以上の速度でハンドリングするために、ワークに応じた治具の設計で対応してきた¹⁾。また、ロボットが作業に応じて自律的に位置姿勢を合わせる (補正する) ために、センサを用いたフィードバック制御が検討されてきた²⁾。しかし、いずれの方法でも、人同等以上の速度による位置姿勢誤差の補正という (1) (2) を両立したワークのハンドリングができていない。これに対し、環境との接触を積極的に活用するソフトロボティクスという研究領域があり、この領域における、環境との接触を活用して位置姿勢誤差を推定および補正する手法を利用すると、人同等以上の速度で組立タスクを実行できることを Karako らが実証している³⁾。しかし、この手法はリング状のワークのシャフト挿入タスクのみに対応しており、ワークやプレイス方法の多様性には対応出来ない。

本研究では、人同等以上の速度で位置姿勢誤差を推定し補正する Karako らの研究³⁾ を先行研究とし、多種ワーク対応性の向上を本研究の目的とする。多種ワーク対応において課題となるのは、(a) 多種ワークに対応したハードウェアの実現と、(b) ワーク個別に制御プログラムを実装することに伴い増加する人的コストの削減であるが、(a) の課題に対しては、先行研究³⁾ で開発したハンドを多種ワーク対応向けに改良して対応した。(b) の課題に対しては、キッティング中に得られるセンサデータを階層化し、キッティングタスクに共通で、かつ、ワークや姿勢に依存しない抽象度の高いデータを出力するアーキテクチャを定義し、このワークや姿勢に依存しない抽象データからアーム・ハンドへの指令データを生成することで対応した。また、(a)、(b) の解決手法を用いたキッティングシステムを構築し、人同等以上の速度で、位置姿勢を合わせながら、多様な部品を決められた場所に置くことも確認した。

本稿は、課題 (b) の解決手法を中心に報告する。2章で多種ワークに対応して人同等以上の速度で位置姿勢ずれを自律的に補正する制御系を実現するセンサデータの抽象化手法と、この抽象化したデータを用いた制御系について記述する。3章でキッティングシステムを構築するにあつ

て前提としたハードウェアと動作戦略、そして、実際に実装した抽象化したデータと、システムについて記述した後、4章で構築したシステムによるキッティング作業の評価結果を示す。そして5章でまとめと今後の展望について述べる。

2. 提案する手法

本研究では、キッティングの要件である人同等以上の速度で位置姿勢誤差を補正するために、Karako らの環境を活用して位置姿勢誤差を推定し補正する手法³⁾を採用し、残る課題となる多種ワーク対応性向上の課題に取り組む。様々なワークを扱う課題は、様々なワークを把持可能なハードウェア面の課題と、ワークやプレイス場所ごとに制御プログラムを実装し調整するソフトウェア面の課題に分けられるが、前者のハードウェアに対しては、開発済みのハンドに機能追加したハンドを用意して対応した (詳細は3章を参照)。本章では、後者の制御ソフトウェアの実装と調整について記述する。

2.1 方針

先行研究³⁾においてワークやプレイス場所ごとに制御プログラムを実装することになる原因は、ワークやプレイス場所に依存したデータを用いているためである。そこで我々は、ワークやプレイス場所が変わっても同じ制御プログラムでキッティング作業を実行できるように、ロボットに装着した様々なセンサデータや、アームやハンドのモータの角度データといった未加工データを直接扱わず、キッティングの自動実行に必要で、かつ、ワーク依存性・作業依存性のない抽象データに変換し、この抽象データで制御することにする。

2.2 抽象データの要件定義

2.1節の方針で重要となる抽象データの要件を、キッティングの自動化に求められる以下の要件から定義する。

- 要件 1. アームは、ワークやプレイス場所に合わせて自律的に位置姿勢を補正すること
- 要件 2. ハンドは、早い作業でもワークを落とさないように安定して把持すること
- 要件 3. 人同等の時間でキッティングすること

2.2.1 要件 1 から導く抽象データ

図 1 に示すように、(a) ピック時において、上位システムから与えられたハンド把持位置 (必ずしもハンドの中心に一致しない) でワークを把持できるように、「ハンド把持位置とワーク想定把持位置」の位置姿勢を合わせるようにアームを制御する。プレイス時においては、「ワークを把持した状態のハンドを、ワークを含めて一体的なハンド

とみなし、把持したワークの面や辺、角あるいは重心」と「プレイス場所のガイドや印」の位置姿勢を合わせるようにアームを制御する。このように制御することで、ピック時とプレイス時のそれぞれで個別のプログラムを作成することなく、同じ制御で位置姿勢誤差を補正できるようになる。また、(b) 指を動かした際に指の高さが変わるため、アームはその高さの変化に追従する必要がある。

(a) のピックとプレイス時における位置姿勢を合わせるために、上位システムやセンサから得られる位置姿勢に関する情報を利用する。センサは、「2 対象間の位置姿勢誤差相当値 $\Delta x (\in R^6)$ 」を測定可能なセンサ（たとえばビジョンセンサ）や「ハンドが受ける外力相当値 $f^{hnd} (\in R^6)$ 」を測定可能なセンサ（例えば力覚センサ）を利用する。ハンドが受ける外力を測定するセンサは、ワークにハンドが接触したことや、把持したワークがガイドに接触したことを感知することで位置姿勢を合わせる用途で利用する。位置姿勢誤差を推定し補正する場合、 Δx は最終的に誤差量が 0 になるよう制御するので、数十%のスケール誤差が含まれていても問題なく機能する。また、 f^{hnd} は、接触していない時は 0 で、接触力に合わせて増減すれば良いので、数十%のスケール誤差が含まれていても問題なく機能する。このため、これらセンサには、厳密なキャリブレーションは不要である。これら 2 つの「2 対象間の位置姿勢誤差相当値 Δx 」と「ハンドが受ける外力相当値 f^{hnd} 」は、ワークや作業に依存しないため、それぞれ抽象データの 1 つとする。

次に (b) の指の動きに合わせてアームの高さを追従するために、「指の角度情報 q_i^{act} 」を利用する。この角度情報はワークや作業に依存しないため、抽象データの 1 つとする。

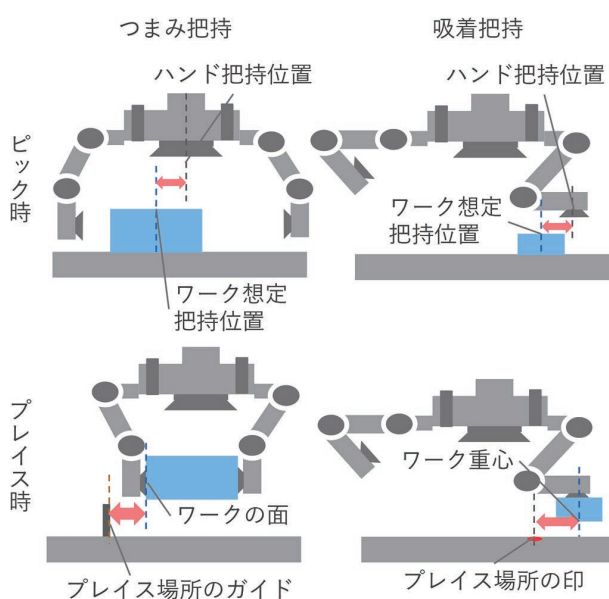


図 1 各把持方法におけるピック・プレイス時の位置誤差

2.2.2 要件 2 から導く抽象データ

ワークを複数の指で把持する場合は、指がワークから受ける外力が、その接触点/面において法線方向に生じている状態が、その指配置における最も安定性の高い状態として必要である⁴⁾。また、吸着など指単独で把持する場合は、「上位システムから指定された把持位置でワークの想定把持位置を吸着把持している状態」が安定した把持状態である。どちらの場合も、安定して把持できていないと、指に対してワークが傾き、各指について指の面の部位（例えば、指先と指根本）によって受ける力に偏りが出る。

このため、ハンドがワークを想定通りに把持できているかどうかを把握するため、「指 i の指の面における力の偏り」を観測する。この値は偏りがなときは 0 となり、偏りがあるときはその偏りの大きさに応じて増加する値である。この「指 i の指の面における力の偏り相当値 Δf_i^{fns} 」は、ワークや作業に依存しないため、抽象データの 1 つとする。

本節の最初に記載したように、複数の指で把持する場合は、「それぞれの指に一定以上の力がかかっていること」を感知する必要がある。この「指 i が受ける外力」は、指が受ける外力に応じて増減し、外力がないときは 0 になる値であればよい。また、0 を少し超えた値を保つように制御するので、推定している力に数十%のスケール誤差を含んでも機能し、センサの厳密なキャリブレーションは不要である。この「指 i が受ける外力相当値 Δf_i^{fns} 」は、ワークや作業に依存しないため、抽象データの 1 つとする。

2.2.3 要件 3 から導く抽象データ

人は、腕を動かしながら独立して指を動かし、作業によっては腕と指を連携して動かすことで、効率的に作業する。本システムも、同様の仕組みを採用する。このため、アームとハンドには、独立に動作できるように指令値を個別に入力する。また、必要に応じてお互いの作業の達成を待ってから、連携して次の作業を実行開始できるように、それぞれの「作業の達成度合い」を出力する。アームもハンドも、現在値が制御の目標値に近づくと速度を落とすことに着目し、「作業の達成度合い」として、作業依存性のない「アーム速度の絶対値 $|v^{arm}|$ 」と「ハンド各指の角速度の絶対値の平均 $|q^{act}|$ 」を抽象データの 1 つとして採用する。

2.3 提案方針に基づく制御系の構成

2.3.1 タスクの実現

2.2 節に記載の要件 1、要件 2 の抽象データを用いてフィードバック制御をおこなう。この制御はワークや作業に依存せず、目標状態に達するとアームとハンドの動きが

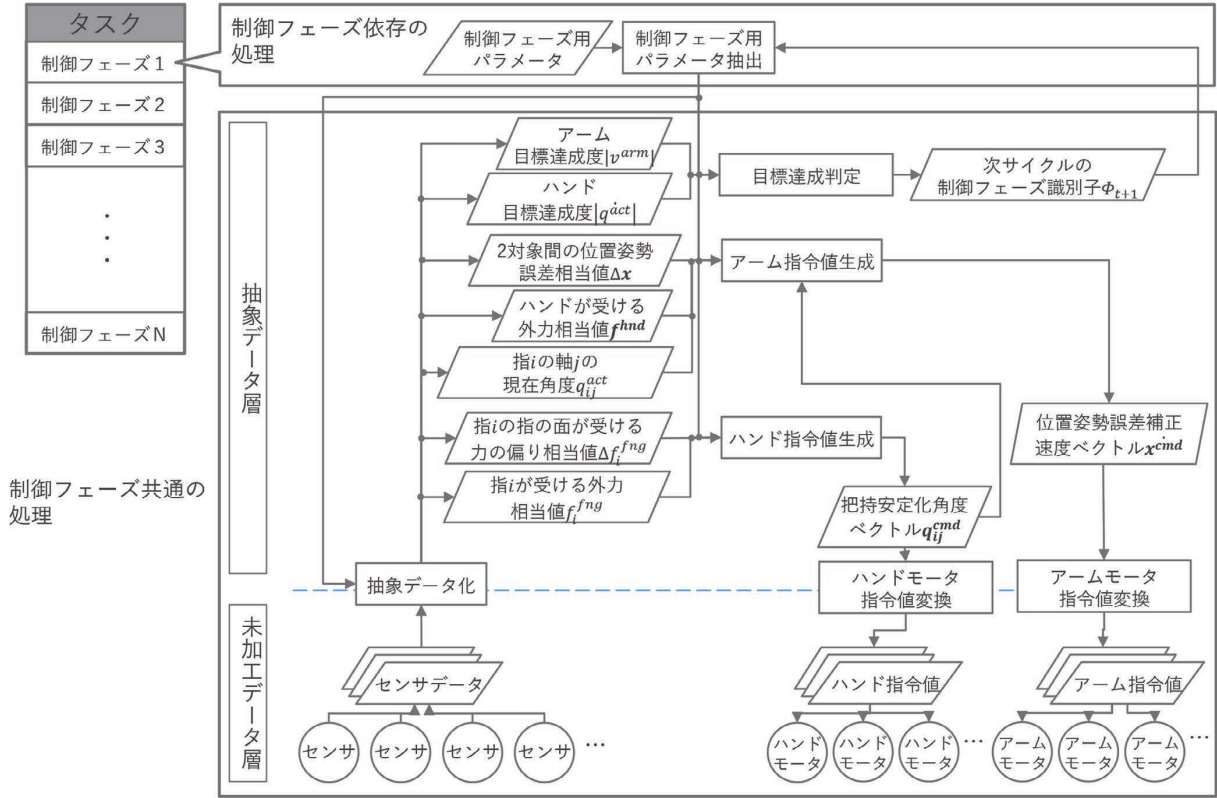


図2 提案方針に基づく制御系の構成

止まる。この動きが止まることは、要件3の抽象データが「0」となることに等しいため、目標状態に達したことの判定は、要件3の抽象データにより行う。この制御により、たとえば「ワークの場所にアームを移動する」、「ワークを把持する」といった、キッキングを構成する1つ1つの動作が実現できる。この1つ1つの動作を本稿では「制御フェーズ」と呼ぶ。この「制御フェーズ」を複数用意し、順に実行することで、キッキングなどの「タスク」を実現する。(図2の左側)

図2の右側に示すように、各制御フェーズの処理は、その制御フェーズに依存するパラメータ(2.3.4節で詳述)を、すべての制御フェーズで共通の処理に代入することで実現する。制御フェーズ共通の処理では、2.1節の方針に基づき、センサが出力する未加工データ層のデータを一旦、抽象データ層のデータに変換し、抽象データ層で指令値を生成して、再び、アームおよびハンドのモータへの未加工データ層の指令値に変換する構成となっている。「未加工データ層」はセンサや、アームおよびハンドのモータが扱うデータを処理し、「抽象データ層」はワークや作業に依存しないデータを処理する。

2.3.2 指令値の生成

「制御フェーズ」では、センサが出力する未加工のセンサデータは、「抽象データ化」機能により、2.2節に記載の

要件を満たす抽象データに変換する。「抽象データ化」機能には、センサデータから2.2節に記載の要件を満たす抽象データを算出する式を定義している(本研究で作成したキッキングシステムにおける抽象データの定義は3.3節を参照)。

アームへの指令値である「位置姿勢誤差補正速度ベクトル x^{cmd} 」は、「アーム指令値生成」機能で、抽象データ f^{hnd} 、 Δx 、 q_i^{act} を用いて以下のように生成する。

$$x^{cmd} = r_{\Phi}^d + K_{\Phi}^{fh} f^{hnd} + K_{\Phi}^{xa} \Delta x - is_{\Phi}^{hv} \text{ave}_{i=L, n_f} (J(q_i^{act}) q_i^{cmd}) \quad (1)$$

ここで、 r_{Φ}^d はフェーズごとに設定するアームの目標速度、 K_{Φ}^{fh} と K_{Φ}^{xa} はフェーズごとに設定する単位換算行列 ($\in R^{6 \times 6}$)、と is_{Φ}^{hv} はその項を実行するか否かのフラグ(実行する場合は1、実行しない場合は0)、 n_f はハンドの指の数、 $\text{ave}()$ は引数となるベクトルの要素ごとに平均値を計算する関数、すなわち、 a_i を任意のベクトルとしたときに以下の式で定義される関数である。

$$\text{ave}_{i=L, n_f} (a_i) = 1/n_f \sum_{i=1}^{n_f} a_i \quad (2)$$

q_i^{act} は指 i の各軸 j の角度 q_{ij}^{act} からなる角度のベクトル、 $J(q_i^{act})$ は指 i のハンド各軸の角速度からハンド各指の先端速度への対応を表すヤコビ行列、 q_i^{cmd} は指 i の各軸 j の指令角速度 q_{ij}^{cmd} からなる角速度のベクトルである。

\mathbf{x}^{cmd} の生成式は、先行研究³⁾におけるアーム指令値を定義した以下の式の $\mathbf{J}^{-1}(\boldsymbol{\theta})$ にかかる括弧内の式を多種ワーク対応のキッキング用に拡張したものである。

$$\dot{\boldsymbol{\theta}}_c = \mathbf{J}^{-1}(\boldsymbol{\theta}) \cdot (\dot{\mathbf{r}}_d + \mathbf{G}\Delta\hat{\mathbf{p}} - \mathbf{V}_c\hat{\mathbf{e}}_0) \quad (3)$$

我々が提案する \mathbf{x}^{cmd} の生成式において、位置姿勢誤差を認識できている場合は $\dot{\mathbf{r}}_d = \mathbf{0}$ 、 $\mathbf{K}_\Phi^h = \mathbf{0}$ とし、 $\mathbf{K}_\Phi^a\Delta\mathbf{x}$ の項で誤差補正を行うための指令値を生成する。また $ave()$ の項は各指 i の指先速度の平均値を表しており、指による摘まみ上げ動作による位置変化に追従することで指先の位置を一定に保つための項である。指先を動かしたときに位置を一定に保つ必要がない場合は $is_\Phi^{hv} = 0$ とする。

次に、ハンドの各軸への指令値である「把持安定化角度ベクトル \mathbf{q}_i^{cmd} 」は、「ハンド指令値生成」機能で、抽象データ Δf_i^{fng} 、 f_i^{fng} を用いて以下のように生成する。

$$q_{ij}^{cmd} = q_{ij\Phi}^d - k_{ij\Phi}^{f\Delta} \int_0^t \Delta f_i^{fng} dt + k_{ij\Phi}^{ff} \int_0^t f_i^{fng} dt \quad (4)$$

ここで、 $q_{ij\Phi}^d$ は指 i の軸 j のフェーズごとの目標角度、 $k_{ij\Phi}^{f\Delta}$ と $k_{ij\Phi}^{ff}$ はフェーズごとに設定する単位換算係数である。

q_{ij}^{cmd} の生成式は、先行研究の手法³⁾におけるハンドの目標角度 q_{dj} の算出に相当する。 $q_{ij\Phi}^d$ の位置に指を動かした後、複数の指で把持する場合は項 $k_{ij\Phi}^{f\Delta} \int_0^t \Delta f_i^{fng} dt$ で片側の指が押し勝たない位置になるよう調整し、項 $k_{ij\Phi}^{ff} \int_0^t f_i^{fng} dt$ で一定の力で把持するように角度を調整している。吸着など単独の指で把持する場合は $k_{ij\Phi}^{f\Delta} = k_{ij\Phi}^{ff} = 0$ とする。

アームやハンドの各軸モータを動作させるために必要な指令値の単位には、これらの指令値ベクトルから実際に使用しているモータドライブのインタフェースに合わせて変換する。

2.3.3 アームとハンドの連携実行の実現

アームとハンドを独立して動かすことが求められるため、アームとハンドへの指令値は同時に出力する。一方で、必要ときにはアームとハンドが連携して次の動作を同時に開始できるように、「抽象データ化」機能は、アームとハンドそれぞれの「作業の達成度合い」である「アーム目標達成度 $|v^{arm}|$ 」、「ハンド目標達成度 $|q^{act}|$ 」を出力している。「目標達成判定」機能が、これらの目標達成度と「許容誤差 $\boldsymbol{\varepsilon}_\Phi^{arm}$ 、 $\boldsymbol{\varepsilon}_\Phi^{hnd}$ 」を参照して、次の「制御フェーズ」に進めるかどうかを決める。アームとハンドを同じタイミングで動作させたい制御フェーズにおいて、アームとハンドの目標達成度のどちらか一方が「許容誤差 $\boldsymbol{\varepsilon}_\Phi^{arm}$ 、 $\boldsymbol{\varepsilon}_\Phi^{hnd}$ 」を超えている場合、「目標達成判定」機能は、現在の「制御フェーズ」の識別子を出力し、両方の目標達成度が目標を達成する場合、次の「制御フェーズ」の識別子を出力し、制御全体は次の「制御フェーズ」に移行する。

2.3.4 制御フェーズごとのパラメータ

「制御フェーズ」を、「ワークの場所にアームを移動する」、「ワークを把持する」といった動作を実行できるものに特徴づけるデータが「制御フェーズ用パラメータ」である。本手法において、ユーザもしくは上位系が設定するのはこのパラメータのみである。このパラメータには、前述の目標達成を判断するための「目標値 r_Φ^d 、 $q_{ij\Phi}^d$ 」や「許容誤差 $\boldsymbol{\varepsilon}_\Phi^{arm}$ 、 $\boldsymbol{\varepsilon}_\Phi^{hnd}$ 」と、2.2節に記載の要件を満たす抽象データから指令値を生成する際の単位変換係数 \mathbf{K}_Φ^h 、 \mathbf{K}_Φ^a 、 is_Φ^{hv} 、 $k_{ij\Phi}^{f\Delta}$ 、 $k_{ij\Phi}^{ff}$ が存在し、「制御フェーズ」ごとに固有の値を設定する。「制御フェーズ」が実現する動作は、基本的な動作になるため、こうしたパラメータは、一度設定すれば様々なタスクで流用できる。

「制御フェーズパラメータ抽出」機能が、現在の「制御フェーズ」に基づき、該当する「制御フェーズ用パラメータ」を、「抽象データ化」や「目標達成判定」などの各機能に出力する。

3. システムの実現

本研究では、2章で提案した制御系を実装し、人同等速度で動作するキッキングシステムを構築した。本章ではこのシステムについて記述する。

3.1 前提としたハードウェア

本節では、システム実現のために利用した機器について記述する。

3.1.1 吸着機構と複数センサを備えた柔軟ハンド

人同等速度で動作するキッキングシステムを構築するにあたり、先行研究³⁾で開発したハンドに対し、以下の機能要件を追加した改良ハンドを作成した。

- 1つのハンドで、対象とする様々なワークを把持できること
- 2.2節の要件を満たす抽象データを算出できるように、把持/非把持の各状態で、ハンドに対するワークの相対的な位置姿勢誤差を認識できること

まず1つ目の要件に対応するため、各指に大小の吸着パッドによる吸着機構を取り付け、4種類の把持方法で図3に示す範囲のワークを把持できるようにした。

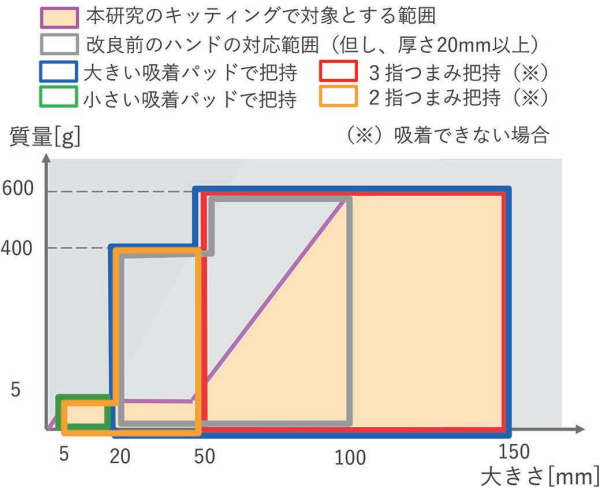


図3 対象ワークの把持方法

次に、2つ目の要件を満たすように、掌の中心にビジョンセンサ用カメラを、各指の吸着パッドに、その押込量1軸、姿勢2軸の合計3自由度の変形量を静電容量変化により取得可能な吸着パッドセンサ⁵⁾を取り付けた(図4)。これにより、非把持状態、把持状態のそれぞれでワークとの位置姿勢誤差を取得できるようにした。

3.1.2 ms オーダーで処理可能なシステム構成

先行研究³⁾が提案する手法では、システムの高速度性を確保するため、制御周期を1msとしている。我々が今回作成するシステムも、この方針に倣い、制御周期の目標値をms オーダーとした。構築したシステムを図5に示す。

このシステムの特徴は以下の3点である。

1. ビジョンセンサ用カメラからの画像データはデータ量が多いので、一旦、USB3.0で接続した専用PCで処理して必要な特徴量だけを1ms以内で抽出し、その特徴量のみをリアルタイムコントローラに送信し、トータルでレイテンシー含め2ms以内で処理

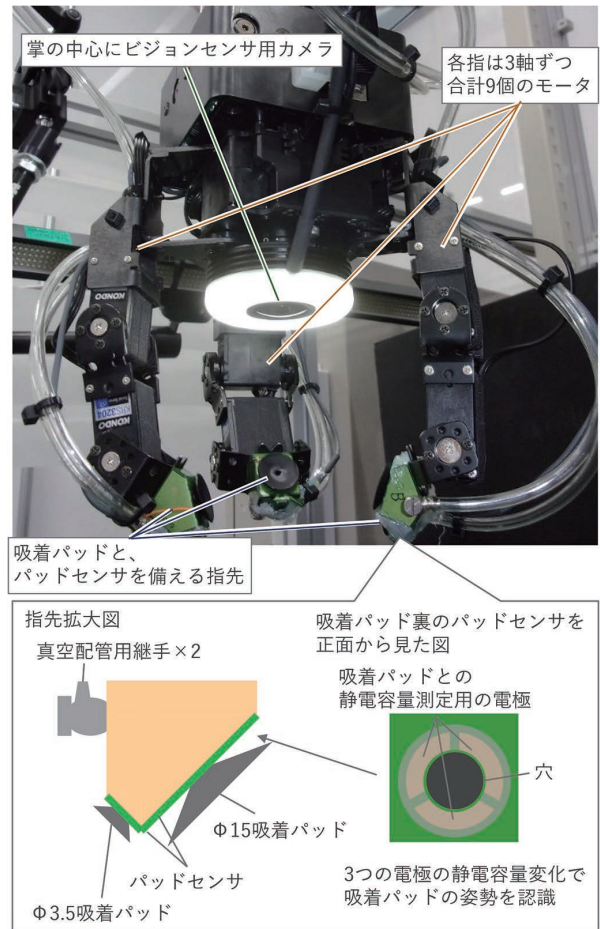


図4 本研究で用いたハンド

2. コントローラやマイコン間は100BASE-TのEthernetで接続し、各プロセスが最大のスループットを發揮できるようにデータ送受信時のトランスポート層のプロトコルをコネクションレス型のUDPにする
3. ハンドマイコンの処理を可能な限り早くするために、モータへの指令値出力後のモータ内処理とセンサからのデータ受信によるハンドマイコン処理を並

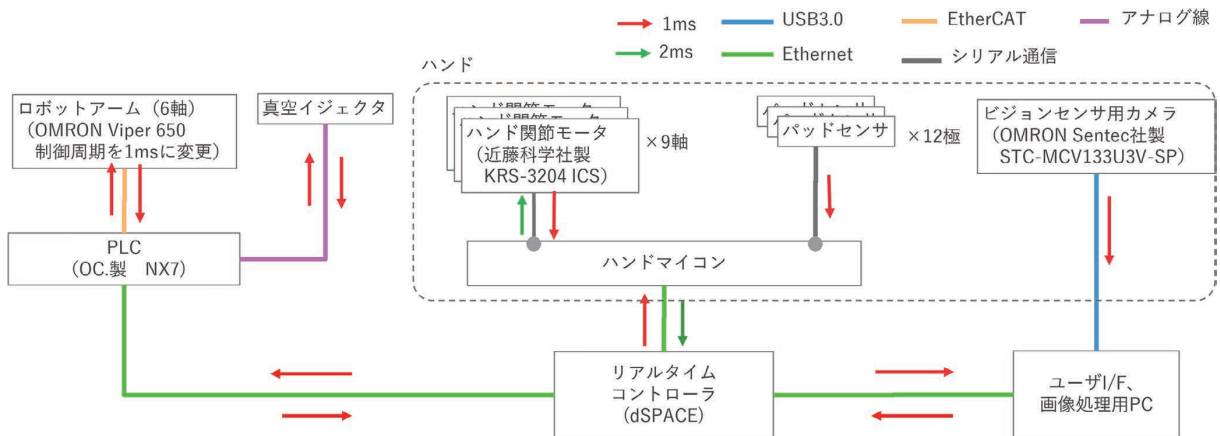


図5 本研究で構築したシステムの構成図

列に処理できるようにピンアサイン。またセンサデータを必要な精度に落とすことで処理を高速化。以上により、1 サイクル 2ms 周期での処理を担保

3.2 キッキング制御プログラムの作成と動作戦略

図2の「タスク」における、各「制御フェーズ」は、例えば、「ワークの上へ移動しながらワークに当たるまでアームを下ろす」や「吸着パッドによりワークを吸着保持する」といった、1つの目標を達成する動作である。こうした動作を実現できる「制御フェーズ用パラメータ」を設定し、「制御フェーズ」の実行順序をタスクに設定することで、キッキングを実行できる制御プログラムを作成した。

この「制御フェーズ」の実行順序を設定することは、ロボットの動作戦略を策定することにあたる。今回、この動作戦略の策定にあたり、先行研究³⁾による環境を活用して位置姿勢誤差を推定し高速に補正する考え方を採用する。すなわち、①「ロボット側に受動変形要素を持たせることによる、指先とワーク、ワークと環境との高速接触時の衝撃緩和」と、②「ワークと環境の幾何学的拘束と受動要素との力のつり合いによって現れる受動要素の、変形量観測に基づくワークと環境間の幾何学的関係性の推定」と、③「①②を活用した高速位置姿勢合わせ制御」を行う。今回のハンドでは、受動要素は吸着パッドであり、受動要素の変形量計測は吸着パッドセンサで行う。

これらを利用すると、ワークの高さ情報が不確実な中で把持も、吸着パッドの受動変形に頼ることで高速にハンドをワークに接触させることが可能となる。また、把持したワークをガイドに合わせて置く際も、高速にワークをガイドに接触させることが可能となり、事前にわかっているワークとガイドの幾何学情報を用いることで、ワークとガイド間の相対的な位置姿勢ズレ関係の推定や、それに基づくズレ解消制御が可能となる。

3.3 抽象データの定義

本節では今回作成したキッキングシステムにおける各抽象データの定義を示す。

3.3.1 2対象間の位置姿勢誤差相当値 Δx

2対象間の位置姿勢誤差相当値 Δx は、以下のように定義し、6次元ベクトルである。

$$\Delta x = (1 - i_{\Phi}^{VFB}) \Delta X + i_{\Phi}^{VFB} K^{CAM} R^{arm} \Delta cam \quad (5)$$

ここで、 i_{Φ}^{VFB} は実行中のフェーズにおいてカメラによる位置姿勢誤差補正 (Visual feedback) を実行するか否かのフラグ (実行する場合は1、実行しない場合は0)、 $\Delta X (\in R^6)$ はそのフェーズにおける目標位置姿勢と現在位置姿勢の差分、 $K^{cam} (\in R^{6 \times 6})$ はセンサに対する単位変換行

列、 $R^{cam} (\in R^6)$ はロボットの姿勢に応じて決まり、カメラの座標系をロボットの座標系に合わせる回転行列、 Δcam はカメラによる Visual feedback 実行時のセンサ値であり、ピック時は「ハンド把持位置とワークの想定把持位置」、プレイス時は「把持したワークの重心とプレイス場所の印」に関する、カメラ上の画素における2点間の距離誤差と把持姿勢に対する対象の姿勢誤差 (水平方向2軸と鉛直軸回りの回転方向の3自由度のセンサ値を含み、残る3自由度の値は0) である。

3.3.2 ハンドが受ける外力相当値 f^{hnd}

ハンドが受ける外力相当値 $f^{hnd} (\in R^6)$ は、以下のように定義したベクトルである。

$$f^{hnd} = \text{ave}_{i=1..n_f} (k^p n_i^p p_i) \quad (6)$$

ここで、 k^p は吸着パッドの変形量から力に単位変換するための係数、 $n_i^p (\in R^6)$ は指 i の吸着パッドの面に対する法線単位ベクトル (把持安定化ベクトル計算の都合4-6要素目を0にした6次元ベクトル)、 p_i は指 i の吸着パッドセンサが測定した吸着パッドの変形量である。

3.3.3 指 i の指の面が受ける力の偏り相当値 Δf_i^{fng}

指 i の指の面が受ける外力の偏り相当値 Δf_i^{fng} はスカラー量であり、以下のように定義する。

$$\Delta f_i^{fng} = k^{hnd} h_i \cdot \Delta x_i^{hnd} | \text{ave}_{i=1..n_f} (\Delta p_i) | \quad (7)$$

ここで、 k^{hnd} は吸着パッドの変形量から力に単位変換する係数、 $h_i (\in R^2)$ は指 i の可動面における水平方向のみの単位ベクトル、 $\Delta x_i^{hnd} (\in R^2)$ は指 i の可動面における目標位置と実際の指先の位置の誤差ベクトル、 Δp_i は指 i のパッドセンサの指先の値と、指の根本の値の差分、 \cdot はベクトルの内積演算である。

複数の指で把持する場合で、押し勝っている指があると、その指が目標位置よりもより内側に入り込み、位置誤差 Δx_i^{hnd} とパッドの歪み Δp_i となって表れる。この指 i の方向成分だけ取り出したものを外力の偏り相当値として定義した。

3.3.4 その他の抽象データ

- ・指 i が受ける外力相当値 f_i^{fng} : スカラー量。吸着パッドセンサが測定した吸着パッドの変形量 p_i を用いる
- ・各指の現在角度 q_{ij}^{act} : スカラー量。ハンドの各軸モータのエンコーダから取得する
- ・アーム目標達成度 $|v^{arm}|$: スカラー量。アームの手先速度であり、アームのヤコビ行列から算出する
- ・ハンド目標達成度 $|q^{act}|$: スカラー量。ハンド各軸の角速度の絶対値の平均値であり、モータから取得した角速度から算出する

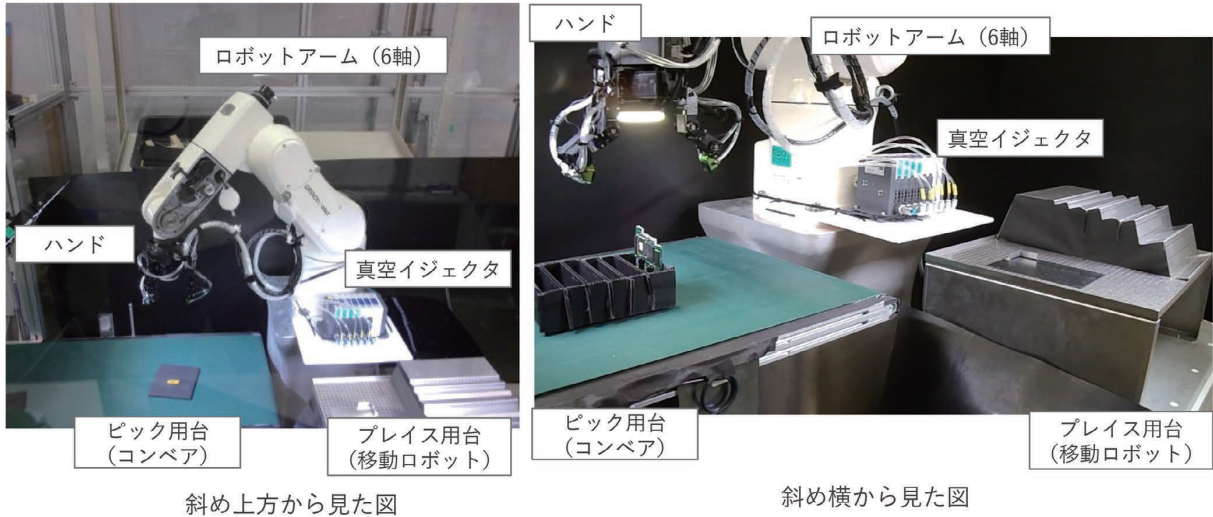


図6 提案手法により構築したキッティングシステム

3.4 構築したキッティングシステム

本研究では、3.3節で定義した抽象データ算出の式を、図2の「抽象データ化」機能に実装し、3.2節で作成した制御プログラムを3.1.2節に記載のシステムのリアルタイムコントローラで実行できるようにし、3.1.1節に記載のハンドを使用したキッティングシステムを構築した。そしてこのシステムの周りに、ピックするワークを配置するコンベアとブレイス先となるワーク置場を備えた移動ロボットを配置した(図6)。

4. 評価

本章では3章で構築したシステムに対して実施した評価実験について記述する。評価実験では、キッティング工程の要件として設定していた、(1) 人同等以上の速度で、(2) 位置姿勢を合わせながら、(3) 多様な部品をブレイス場所に、並べる事ができることを確認する。評価で用いるワークは、社内の生産現場での活用を想定して、オムロン製品であるPLC (Programmable Logic Controller) とサーボドライバを構成する部品とした。評価における目標値を表1に示す。位置姿勢誤差の対応性の目標値である±10mmは、本システム外に設けられた部品検出用のカメラから先見的に与えられる部品位置と真の部品位置の最大誤差として設定し、1.5deg. は、水平方向に20cm移動した際に高さが5mm変わるような傾斜であり、アルミフレームなどで作業台を作成した場合に生じる最大の角度誤差として設定した。位置誤差補正の目標値は、次の組立工程で問題にならないと想定される1mm以内とした。高速性に関する目標値は、著者が同じ環境で実際に実行した時の時間が平均3.8秒、最大4.6秒であったので、同等程度として4秒を設定した。

表1 評価項目に対する評価実験の目標値

評価項目	目標
1. 高速性	1ワークあたり4秒以内にキッティングできること
2. 位置姿勢誤差対応性	ワークとブレイス場所として与えられた位置姿勢情報と真の情報に誤差(±10mm、±1.5deg.)があっても、誤差を自律的に1mm以内に補正し、キッティングできること
3. 多種ワーク対応性	図7の赤枠内のワークに対応

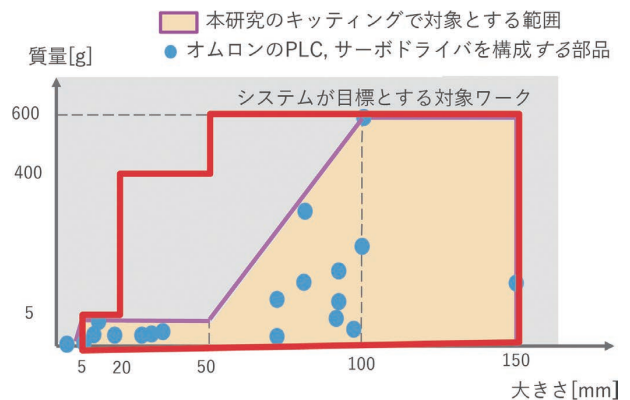


図7 構築したキッティングシステムの目標対象ワーク

本研究では、各評価項目を確認するため、以下の2つの評価実験を実施した。

実験1. 形状が異なるが、把持方法が同じであるワークを、1つの制御プログラムで把持できることを確認する実験

実験2. 異なる把持方法のワークに対して、先見的に与えられた位置姿勢情報に $\pm 10\text{mm}$ 、 $\pm 1.5\text{deg}$.の誤差があっても自動的に位置姿勢を補正して4秒以内の時間でキッキングできることを確認する実験

4.1 実験1

今回のハンドでは4種類の把持方法がある。このうち吸着による把持方法の2種類は実験2で評価する。また、図8に示すワークで、2つの指でつまんで把持するワークは存在しない。このため、実験1では、3つの指によるつまみ把持で1つの制御プログラムでの把持可否について評価する。

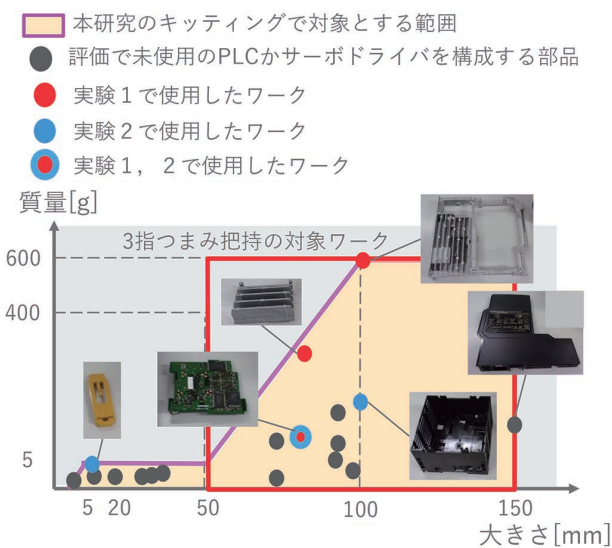


図8 実験1, 2で用いたワーク

評価で用いるワークとして、最も重くかつ重心が偏っているものと、密度が大きくかつ質量が大きいものと、薄く凸凹した形状のもので最も重いものという、安定把持が難しい3つのワークを選定した(図8)。これら3つを把持できれば、他は同様の大きさで、より軽いものになるため、いずれも把持できると想定している。なお、図8には赤枠の境界に、今回選定していない150mmの大きさのワークが存在しているが、これはつまみ把持では把持困難な薄いワークであり、吸着により把持するものであるため、対象としていない。

実験の結果、図9のように、3つのワークすべてで把持できることを確認した。

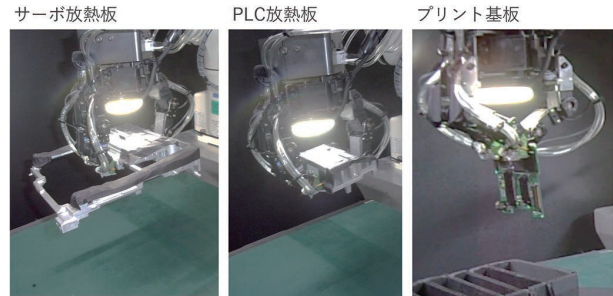


図9 実験1の実行結果

4.2 実験2

今回用意したワークの中に2指つまみ把持により把持するワークが存在しないため、2指つまみ把持を除く3種類の把持方法でキッキングの評価実験を実行した。各把持方法で特に把持が難しいと想定されるワークでキッキングが成功すれば、他のワークでもキッキングが成功すると考え、図10のワークを選定した。「10cmの樹脂ケース」は大きいパッドで吸着により把持するが、嵩高いため、吸着部が慣性による影響を受けやすく、把持安定性が悪い。「幅1cmの樹脂部品」は、吸着できる部位が重心からずれているため、吸着部位に重力によるモーメントがかかり、把持安定性が悪い。「プリント基板」は、表面実装部品により吸着面が存在しないため、つまみ把持しかできず、また、2つの指を平行にして残る1つの指と対峙する形に配置してつまみ把持するため、2本の指の方が押し勝ち、力の制御が必要になり、ロボストに安定して把持する難易度が高い。図10に示すように、各ワークは実際のキッキングを想定したピック位置に置き、プレイス時も次の組立工程を想定して、ガイドに沿って配置する。このキッキングを実行する際、ピック位置とプレイス位置の目標座標について、誤差補正量が最大となるように、実際にワークが置かれている位置と実際にガイドが存在している位置からピック位置の目標座標について $+10\text{mm}$ 、 $+1.5\text{deg}$.、プレイス位置の目標座標について -10mm 、 -1.5deg .の誤差がある情報を与えた。この条件下において、各ワークで5回ずつピック&プレイスを実行し、プレイス位置に対する誤差の絶対量と、ピックからプレイスまでの実行時間を計測した。

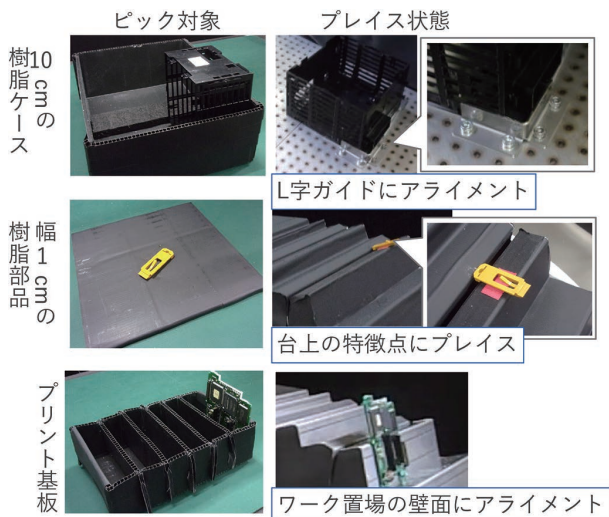


図 10 評価で実施した各ワークに対するキッキング

図 11、図 12 にプリント基板に対する実験 2 の様子を載せる。図 11 に示したように、ピック時は実行前に与えていた位置と姿勢の誤差をハンドに装着したビジョンセンサにより自律的に補正してワークを把持している。また図 12 に示したように、ブレイス時は基板がガイドの下面に接したことを検知するまでアームを下ろし、その後、ガイドの側面に接したことを検知するまでアームを手前に引くことで、ガイドの位置と姿勢の誤差を自律的に補正している。またガイドは 1.5deg. の角度をつけて置いていたが、基板がガイドの面に接すると、ハンドの指先のやわらかさにより、そのガイド面と基板の面が接するように姿勢の誤差が補正され、意図した場所に意図した姿勢でブレイスされる。

「10cm の樹脂ケース」、「幅 1cm の樹脂部品」もピック時はカメラにより位置姿勢の誤差を補正して、ワークを把持した。ブレイス時は、「10cm の樹脂ケース」は基板同様、ガイドに接したことを検知するまでアームを引くことで位置姿勢の誤差を補正してブレイスした。「幅 1cm の樹脂部品」はブレイスしながらハンドに装着したビジョンセンサでガイド上の特徴点に合わせることで位置姿勢を補正した。

この評価実験の結果を表 2 に示す。実験により、「10cm の樹脂ケース」では実行時間を 4 秒以内に収めることができたが、位置誤差は目標である 1mm 以内に収めることができなかった。そのほかのワークではすべてのワークで実行時間を 4 秒以内に収め、かつ、誤差を 1mm 以内に補正して、キッキングタスクを実行できることを確認できた。

目標を達成できなかった「10cm の樹脂ケース」においては、ブレイス時の速度や許容値のパラメータを変更して、実行時間が最大 4.2 秒程度かかるような値にすることで位置誤差の最大を 1mm にすることができている。このことから、おおよそ人同等の性能を達成していると考えられる。

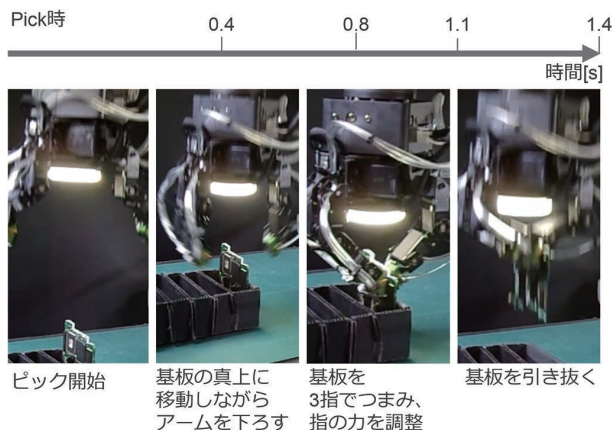


図 11 プリント基板に対する実験 2 のピック時

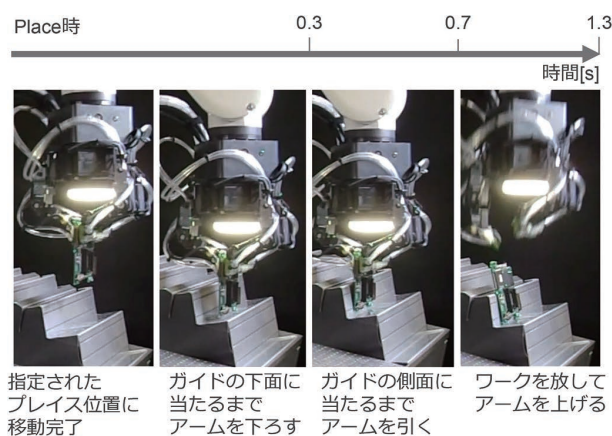


図 12 プリント基板に対する実験 2 のブレイス時

表 2 キッキング評価の結果

ワーク	評価観点	実行結果
10cm の樹脂ケース	位置誤差平均	2.9 mm
	位置誤差最大	4.0 mm
	実行時間平均	3.8 秒
	実行時間最大	4.0 秒
幅 1cm の樹脂部品	位置誤差平均	1.0 mm
	位置誤差最大	1.0 mm
	実行時間平均	3.6 秒
	実行時間最大	3.7 秒
プリント基板	位置誤差平均	0.0 mm
	位置誤差最大	0.0 mm
	実行時間平均	3.6 秒
	実行時間最大	3.9 秒

5. むすび

キッティング工程の自動化において、既存技術で課題となる部品毎に発生するプログラム工数が増大するという課題を解決するため、本稿では、センサデータをロボットが自動でワークや作業に依存しない抽象化データに変換し、制御に用いるアーキテクチャを提案した。このアーキテクチャを元に作成したキッティングシステムにより、ワークが変わってもプログラムの変更や調整が不要で、かつ、人同等の速度で位置姿勢ずれを補正してキッティングを自動実行するシステムを実現した。本稿では、キッティング工程の自動化を対象としたが、本研究の手法は、組立工程における嵌合作業の自動化にも容易に応用できると考える。

ただし実用化に向けては、自律化とハンドの品質の面で課題を残している。特に自律化においては、「環境への接触を活用する動作戦略」の自動生成や「制御フェーズ用パラメータ」内の各パラメータの自動設定が課題である。また、位置姿勢誤差対応性や高速性はトレードオフの関係にあるため、これらをバランスさせる場合にはパラメータ調整が必要となる。その調整の単純化も今後の課題である。今後、こうした課題解決に取り組み、実用化を目指していきたい。

なお、この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP16007）の結果得られたものである。

参考文献

- 1) 原田研介. 産業用ロボットによる組み立て作業の自動化に関する研究動向. 精密工学会誌. 2018, Vol.84, No.4, p.299-302.
- 2) 小山佳祐, 鈴木陽介, 明愛国, 下条誠. 指先に近接覚センサを備えたロボットハンドによるブリグラスブ制御. 日本ロボット学会誌. 2015, Vol.33, No.9, p.712-722.
- 3) Karako, Y.; Kawakami, S.; Koyama, K.; Shimojo, M.; Senoo, T.; Ishikawa, M. "High-Speed Ring Insertion by Dynamic Observable Contact Hand". IEEE International Conference on Robotics and Automation (ICRA). 2019, p.2744-2750.
- 4) 辻徳生, 原田研介, 金子健二. 摩擦円すいの楕円体近似を用いた把持安定性の拘束評価. 日本ロボット学会誌. 2011, Vol.29, No.3, p.278-287.
- 5) Doi, S.; Koga, H.; Seki, T.; Okuno, Y. "Novel Proximity Sensor for Realizing Tactile Sense in Suction Cups". 2020 IEEE International Conference on Robotics and Automation (ICRA). 2020, p.638-643.

執筆者紹介



阿部 将佳 ABE Masayoshi
技術・知財本部 コアテクノロジーセンタ
専門：ソフトウェア工学



川上 真司 KAWAKAMI Shinji
技術・知財本部 テクノロジープロデュースセンタ
専門：制御、ロボティクス
所属学会：システム制御情報学会



金井 聡庸 KANAI Akinobu
技術・知財本部 コアテクノロジーセンタ
専門：電気電子工学



齊藤 ゆみ SAITOH Yumi
技術・知財本部 コアテクノロジーセンタ
専門：制御工学
所属学会：システム制御情報学会、電子情報通信学会、電気学会

本文に掲載の商品の名称は、各社が商標としている場合があります。