

平成 22 年度（経済危機対応・地域活性化予備費事業）

戦略的基盤技術高度化支援事業

「3次元ビジョンセンサを利用した

産業ロボットの動作制御プログラムの自動生成に関する研究開発」

研究開発成果等報告書

平成 23 年 1 1 月

委託者 近畿経済産業局

委託先 学校法人立命館 立命館大学



## 目次

第1章 研究開発の概要 .....	3
1-1 研究開発の背景・研究目的及び目標 .....	3
1-2 研究体制 .....	4
1-3 成果概要 .....	6
1-4 当該研究開発の連絡窓口 .....	7
第2章 本論 .....	8
2-1 現場のニーズ .....	8
2-2 ロボットおよびビジョンセンサの販売先・市場の動向 .....	9
2-3 本プロジェクトの目標 .....	11
2-4 実施内容別報告 .....	11
2-4-1 ビジョンセンサ+ロボットによる動作列定義：オフライン処理 .....	11
2-4-2 動作列の干渉シミュレーション：オフライン処理 .....	13
2-4-3 動作列決定アルゴリズム開発：オフライン処理 .....	22
2-4-4 各社ロボット別インライン制御の実装：インライン処理 .....	26
最終章 全体総括 .....	38
<参考文献一覧> .....	39

## 第1章 研究開発の概要

### 1-1 研究開発の背景・研究目的及び目標

#### (1) 研究開発の背景

##### 【産業ロボットの導入現場のニーズ】

自動車メーカー、精密機器メーカー、ロボットメーカー等では、3次元ビジョン搭載ロボットの登場により、ロボットの適用範囲は、従来、人にしかできなかった分野にも広がりつつある。例えば、バラ積みされたボルト・ナットのピッキングが可能となり、資材供給に関わる人的作業や装置の削減に寄与している。今後、溶接・組立などに展開することにより、国内ロボットメーカーの高付加価値化による競争力UP、ひいては国内製造業のコスト競争力UPが期待される。

##### 【3次元ビジョン搭載ロボットの課題点】

現在、3次元ビジョン搭載ロボット自体が黎明期であり、運用環境の整備が急務となっている。特に、制御プログラムの作成は熟練者の技能に依存しており、3次元ビジョン搭載ロボットを普及させる上で障害となっている。非熟練者でも制御プログラムを作成することができる支援ツールの開発が望まれる。

##### 【他社技術と課題】

現在、他社においても制御プログラムを手動で作成しているのが実情であり、同様の課題を抱えている。

#### (2) 研究の概要

3次元ビジョンセンサ搭載ロボットの登場により、バラ積みピッキングなどロボットの適用範囲が、従来、人にしかできなかった分野にも広がる中、ロボットを制御するプログラムの作成は未だ熟練技能者に依存しており、ロボット普及化の障害となっている。本提案では、非熟練者でも容易にプログラム作成できる運用環境を提供するため、①ビジョンセンサ制御プログラムシミュレータ、②制御プログラム自動生成の研究開発を行う。

#### (3) 研究の目的及び目標

##### 【目標とするシステム】

3次元ビジョンセンサを利用した産業ロボットの動作制御プログラムの作成作業において、①非熟練者が容易に制御プログラムを作成することができる環境を構築し、②熟練者のプログラム作成時間を従来と比較して80%以上削減する。

##### 【現在保有技術】

《3次元ビジョン搭載ロボット》（3次元ビジョンによる位置決め技術）

- ・ロボットが作業対象物の位置姿勢を認識できる。
- ・認識後の判断と動作をプログラムで柔軟に制御できる。
- ・制御プログラムの自動生成技術（特許取得完了）

##### 【解決方法の提案】

「3次元ビジョン搭載ロボット向け制御プログラム自動生成ツール」を研究開発し、熟練技能者に依存することなく、非熟練者が容易にプログラム作成できる環境を提供する。

《総称：3次元ビジョン搭載ロボット向け制御プログラム自動生成ツール》

1. ビジョンセンサ制御プログラムシミュレータ

オフライン処理で、ビジョンセンサ、ロボット、ハンドの動きを表現する動作列を定義、干渉シミュレーションを行うソフトウェアの開発。

【1】ビジョンセンサ+ロボットによる動作列定義：オフライン処理

【2】動作列の干渉シミュレーション：オフライン処理

【3】動作列決定アルゴリズム開発：オフライン処理

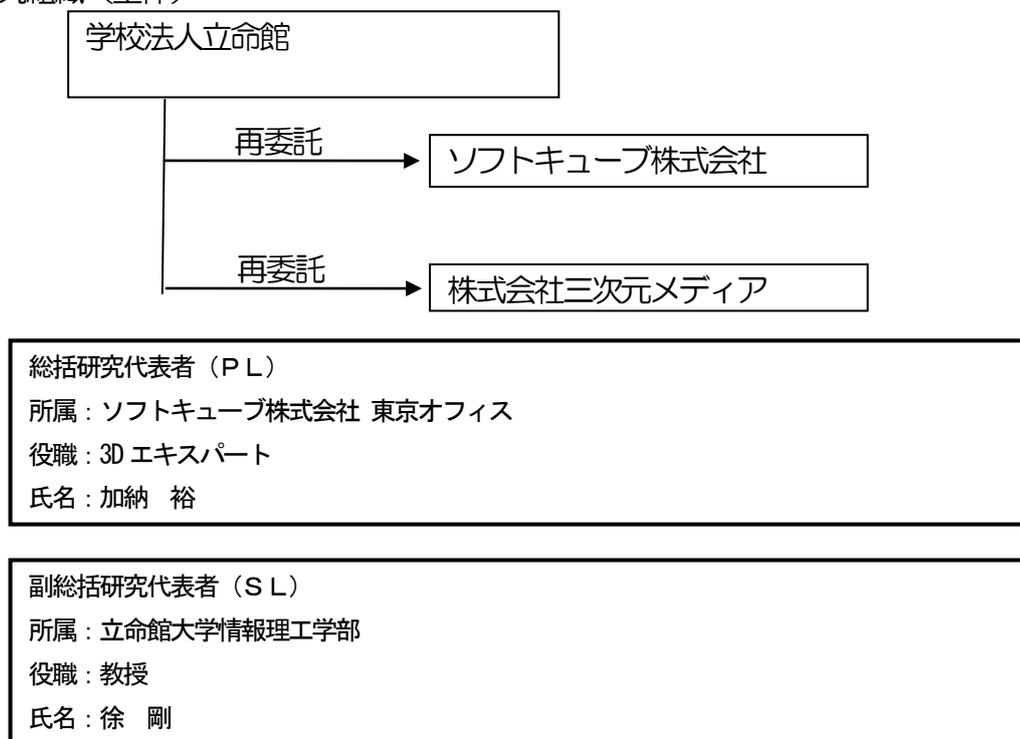
2. 制御プログラム自動生成

インライン処理で、ロボットハンドの位置姿勢を制御する、ロボット動作制御プログラムの自動生成

【4】各社ロボット別インライン制御の実装：インライン処理

1-2 研究体制

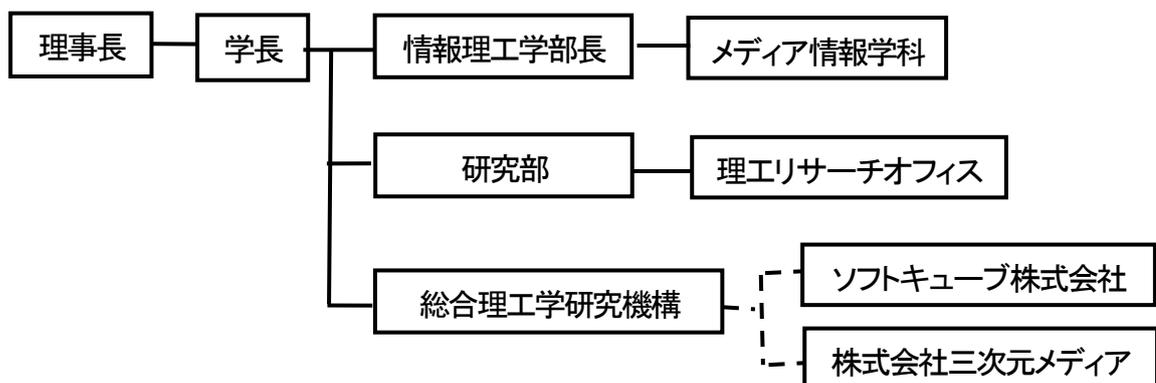
(1) 研究組織 (全体)



(2) 管理体制

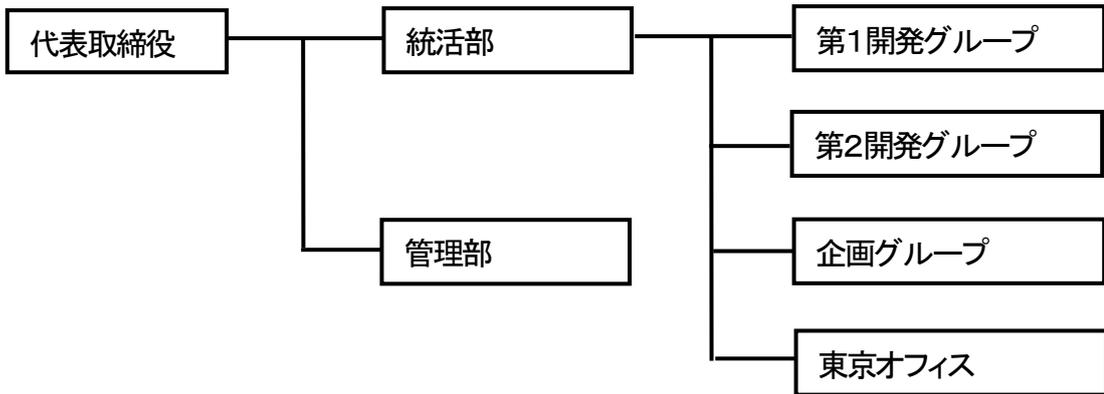
①事業管理機関

[学校法人立命館]

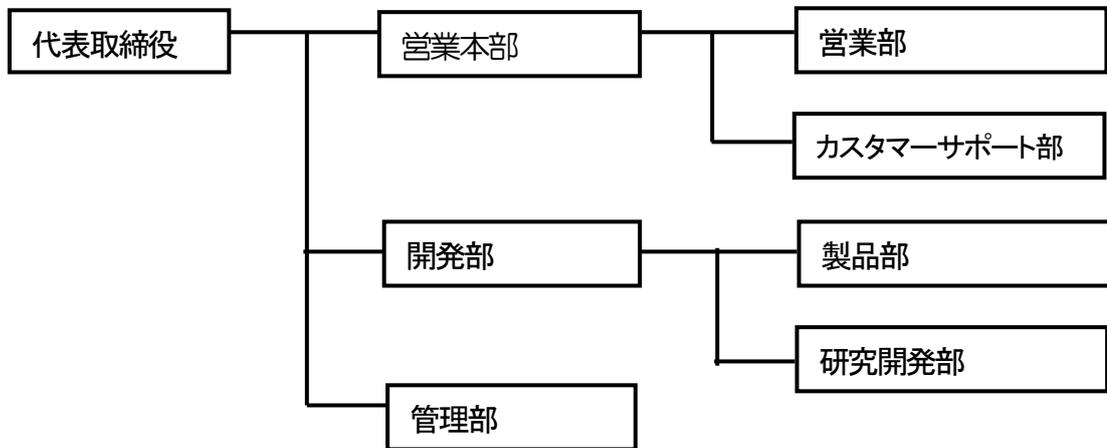


②再委託先

[ソフトキューブ株式会社]



[株式会社三次元メディア]



1-3 成果概要

研究開発項目	できたこと	できなかったこと
① [1-1]ビジョンセンサ+ロボットによる動作列定義の仕様設計	<b>【ビジョンセンサ+ロボットによる動作列定義：オフライン処理】</b> 1. ビジョンセンサと各社ロボットと動作列情報を通信するコマンド仕様設計を完成した。 2. 各社ロボットの標準通信仕様と拡張機能を調査し、各社の仕様の違いを吸収する方式を設計した。 3. 対象部品を把持する位置を教示する方法と、その位置を元に認識部品位置を取得する機能を実装した。	各ロボットメーカー毎に通信仕様や操作ペンダントが異なるため、統一された仕様とはならなかった。 ロボットをアクチュエータとして位置づけるようなミドルウェアを構築する事ができれば、各ロボットメーカーの差を吸収し、より高度な動作列を生成できる。
② [1-2]各社ロボット別動作列仕様理解		
③ [1-3]各社ロボット別の動作列取得および部品位置姿勢取得の実装		
④ [1-4]動作列ファイル入出力の実装		
⑤ [2-1]シミュレータの仕様設計	<b>【動作列の干渉シミュレーション：オフライン処理】</b> 1. シミュレータの仕様設計を完成した。 2. CADモデル、ロボットモデルの読み込みの実装。 3. シミュレータによるピックアップ教示のデータ生成の実装。 4. 干渉判定機能の実装。 5. 干渉判定モジュールとの連携。	本研究の当初の目論見に対して、できなかったことは特になかった。但し、本研究の目標を上回る新たなニーズが浮上しており、将来の研究テーマとなりえる。 例) シミュレータにより直接ロボットを駆動し、ロボットをアクチュエータとする統合システム。
⑥ [2-2]CADモデルの読み込みの実装		
⑦ [2-3]シミュレータによる動作列定義の実装		
⑧ [2-4]動作列干渉シミュレーションの開発		
⑨ [2-5]干渉回避アルゴリズムの開発		
⑩ [3-1]動作列テーブルの仕様設計	<b>【動作列決定アルゴリズム開発：オフライン処理】</b> 1. 動作列としてのアプローチの仕様の設計。 2. 干渉判定に必要なパラメータファイル入力の実装。 3. 入力されたワーク把持位置情報からの干渉判定による安全なパスの抽出アルゴリズムを開発した。	本研究の当初の目論見に対して、できなかったことは特になかった。但し、本研究の目標を上回る新たなニーズが浮上しており、将来の研究テーマとなりえる。 例) アプローチパスからの干渉判定ではなく、動的に干渉を回避するパスの生成。
⑪ [3-2]動作列テーブル入出力の実装		
⑬ [3-3]動作列決定アルゴリズムの開発		
⑫ [4-1]各社ロボット別制御プログラム仕様理解	<b>【各社ロボット別インライン制御の実装：インライン処理】</b> 1. 各ロボットメーカー毎の仕様を調査し、ロボットメーカー毎のビジョン連携プログラムの設計を行った。 対応ロボットメーカー ・デンソーウェーブ ・安川電機 ・川崎重工業 ・三菱電機 ・不二越 ・ABB 2. 3次元でのワーク（把持対象物）の位置姿勢を認識し、その位置姿勢をロボットのベース座標系変換するハンドアイ校正を行う手段を実装した。 3. ロボットメーカー毎に対応する通信ソフトの実装とインラインでのロボット動作定義プログラムとの連携ライブラリを実装した。 4. 干渉回避アルゴリズムによる最適なワークの選定機能と連携インターフェースを実装した。	主なロボットメーカーの内、下記メーカーは技術開示が得られず対象から外した。 ・ファナック ・KUKA このため、当初の計画のすべてのメーカーに対応できなかった。
⑭ [4-2]ロボットベース座標系への位置姿勢変換		
⑮ [4-3]各社ロボット別インライン制御の実装		
⑯ [4-4]各社ロボット別動作列通信ソフトの実装		
⑰ [5-1]総合検証および問題点の改良		

1-4 当該研究開発の連絡窓口

所属：〒525-8577 住所：滋賀県草津市野路東 1-1-1  
立命館大学 研究部 リサーチオフィス (BKC)

氏名：今村 真理子

電話：077-561-2802 FAX：077-561-2811

E-mail：mk-ima-a@st.ritsumei.ac.jp

## 第2章 本論

### 2-1 現場のニーズ

#### (1) 現場の現状

産業ロボットは主に日本の自動車産業で多く採用され、その中で進化、改良が進んできた。そのため、ティーチングプレイバック（教示と再生）を基本とした仕様で提供されている。決められた動作を、繰り返して実行動作して製品の搬送、加工することが主で、センサは条件分岐のon/off情報が主となっている。

ロボット技術者はこの教示を行う技術が主であり、プログラムによる操作列定義はあまり行われていない。このため、画像を用いたセンサの採用は少なく、ロボットメーカーがロボットと組み合わせたシステムとして提供しているものが主である。FAセンサメーカーがロボットに対応するセンサと提供しているものは、主に2次元画像センサである。2次元画像センサは対象ワークとカメラとの位置関係を固定して設置し、X、Y、θの情報をロボットに通知する仕様で、複数のFAセンサメーカーが商品化している。

ロボット用ビジョンセンサの形態として、画像処理ソフトウェアライブラリがある。画像処理ソフトウェアはライブラリの提供に留まり、実際のロボットの動作列の生成はロボットシステムインテグレータ（SI）が対応する必要がある、対応できる体制は少なく、実際には普及していない。

3次元ビジョンセンサによるロボット+ビジョンシステムはロボットメーカーが提供しているのみである。ファナック、安川電機、川崎重工業、三菱電機、不二越、ABBなどから発表されている。

これらのシステムは、メーカー技術者が個別に対応していて、ロボット技術者が自由にシステム構築できる仕様とはなっていない。

#### (2) 現場の課題

国内でのロボット技術者は、「ロボット教示」技術および各メーカー単位でのロボット制御言語プログラムのプログラミングの技術を保有し、画像処理プログラムやC言語でのプログラミングに対応できない。

この技術者の経験、スキルで対応できる3次元ビジョンセンサが提供され、干渉回避動作もこの構成で提供される必要がある。

これらの課題を解決しないと、ビジョンセンサ+ロボットのシステムの普及がなされない。

ロボットメーカー毎にロボットの仕様、位置・姿勢の定義、ペンダントの操作、制御プログラム言語は異なっている。この違いも普及の障害となっている。

## 2-2 ロボットおよびビジョンセンサの販売先・市場の動向

### (1) ロボットメーカーの動向

ビジョンセンサとロボットの連携システムについて、ロボットメーカーは独自のシステムを開発し、提供している。ビジョンセンサにはその方式によっていくつかに分類される。

- ①2次元ビジョン方式
- ②2次元ビジョン+距離測定方式
- ③3次元ビジョン方式

2次元ビジョン方式は、対象ワークの2次元情報(X、Y、 $\theta$ )を画像処理によって求め、ロボットのハンド位置決めを行うもので、すでに一般的技術として普及している。ロボットメーカーも対応したインターフェースを準備して対応をしている。この2次元ビジョンにレーザなどによる測長機能を組み合わせて、正対面しないワークにも対応するものがあり、一般的に2.5次元センサと言われ限定された条件で使用されている。

3次元ビジョン方式のセンサは各ロボットメーカーでも研究されているが、本格的な導入はなされていない。これは、3次元ビジョンの開発には高度な技術と開発期間が必要な事と、それに対するリソースの確保が難しいことによる。

このため、ロボットメーカーでは自前の開発のみでは対応が難しいため、画像技術専門のビジョンメーカーと協力して推進する方針をもっている。

今回の開発を推進するに当たり、ロボットメーカーの協力を得るための交渉を行い、以下のロボットメーカーの協力を得た。

- ①三菱電機
- ②ABB
- ③デンソーウェーブ
- ④安川電機
- ⑤川崎重工業
- ⑥不二越

各社はこの開発テーマの成果がロボットの普及に寄与するとして、技術開示、機材と場所の提供を承諾、協力して頂いた。

### (2) ビジョンセンサメーカーの動向

ビジョンセンサのメーカーには2つの業態が存在する。

- ①ビジョンセンサとして独立した装置を提供
- ②ビジョンシステムに必要な画像処理ライブラリを提供

センサの定義を狭義で行うならば①であるが、ロボットビジョンセンサと考える場合、②の影響も多く、ライブラリを使用してシステムインテグレータがシステム構築を行っている。ただし、現状3次元ビジョンでは成功例は見あたらない。

①のセンサメーカーもロボット用3次元ビジョンセンサを開発を試み、特定のエンド

ユーザと協力して推進をしているが、商品化されていない。

これは、センサメーカーはセンサのみの性能で商品化を行い、ロボットとの連携は敬遠している傾向があるからである。そのため、ロボット技術者が使いこなす事が難しく、この課題を解決できていない。当研究テーマもこの課題に対応するものである。

### (3) システムインテグレータの動向

ロボットの用途として、日本国内は自動車関連の生産工場での普及が先行していて、溶接、塗装、搬送が主な用途である。ロボットの普及につれて、加工、組立などの位置決め用途が増えている。前者のロボット動作は、ティーチングプレイバックが主体で、ロボット技術者の作業はティーチング（教示）である。技術者の技能もティーチング技術に向いていて、ロボットの制御やプログラミング技術は近年になって要求されてきた。これは後者の位置決め用途では重要な技術で、この技術を持ったシステムインテグレータが注目されるようになってきている。このシステムインテグレータはロボットメーカー毎に得意・不得意があるが、ロボットの選定はエンドユーザが決定する事もあり、複数のメーカーに対応している。

このようなシステムインテグレータはビジョンセンサを積極的に導入する動きがあり、すでに 2 次元のセンサを使いこなしている。このため、ビジョンセンサメーカーもシステムインテグレータが使いこなせるような仕様で商品化して、それに答えるようにシステムに導入されている。

このように、3 次元ビジョンもシステムインテグレータが使えるような仕様で提供する事が、普及を加速すると考えられる。しかし、3 次元ビジョンでは 3 次元に関する知識が必要であり、普及の障害となっている。このことが、大きな課題であり今回の研究テーマも解決手段として有用と考えている。

### (4) エンドユーザの動向

産業ロボットは自動車業界を牽引役として、国内で飛躍的に進化をしてきた。最近では多くの生産現場でのロボットの活用が進み始めている。その背景として次のような要求がある。

- ①安定的な熟練労働力の確保
- ②多品種に適應する生産ライン
- ③生産設備の立ち上げの短縮
- ④故障の少ない高信頼設備

これらの要求にはティーチングプレイバック型のロボットではなく、ビジョンを組み込んだ高機能ロボットで対応する必要がある。特に組立現場で多く使われているパーツフィーダの置き換えのニーズが高い。パーツフィーダを複数用いた生産システムから、ロボット 1 台で多機能工セル生産システムへの移行が求められている。

一方で、労働賃金の安価な海外生産に移行が進んでいたが、現在海外でのロボット化のニーズが高まってきている。これは、労働賃金の上昇のみでなく、流動的な労

働力で労働力の確保が難しい事と、作業者の熟練度が保てないため品質安定化が難しいためであり、中国などで需要が高まり、国内生産のロボットも多くが輸出されている。直接輸出に加えて、国内納品されたロボットが生産設備に組み込まれ輸出される事も多く、過半数を超える。

国内の中小の製造業でもコスト削減のためにロボットの導入の検討を進めている。特に中堅のメーカーでこの動きが顕著で、まだ少し余裕があるうちにロボット対応を進めている。ロボットに対する知識やシステムアップも社内でトライする企業が多く、3次元ビジョンセンサへの注目度も高い。このニーズに対応するためこの研究テーマは求められる技術である。

大手メーカーでも過去に対応できなかったテーマに再チャレンジする動向がみられる。この課題を解決する事により、ロボットとビジョンの利用が加速されると見られる。

## 2-3 本プロジェクトの目標

### 【目標とするシステム】

3次元ビジョンセンサを利用した産業ロボットの動作制御プログラムの作成作業において、

- ① 非熟練者が容易に制御プログラムを作成することができる環境を構築する。  
非熟練者とは初期ロボット講習を受講し、ティーチングと Move を主体としたロボットプログラムを作成できる者とし、その者が3次元ビジョンセンサを利用し、ワークをピックアップする制御プログラムを容易に作成できる環境を構築する。
- ② 熟練者のプログラム作成時間を従来と比較して80%以上削減する。  
熟練者とは、ロボットプログラムに精通し、外部 IO との連携やロボット言語の関数を利用し、条件判断をしてロボットが動作するプログラムを作成できる者とする。その者が3次元ビジョンセンサを利用したプログラムを作成することを想定し、作成時間を比較する。

## 2-4 実施内容別報告

### 2-4-1 ビジョンセンサ+ロボットによる動作列定義：オフライン処理

ビジョンセンサで部品を認識しその3次元位置姿勢を計算した上で、ロボットペンダントによりロボットハンドの動きを手動で制御し、掴む・離す動作の動作列を定義し、コンピュータに覚えさせる。ただし、部品の位置姿勢によっては掴む場所や離す方法を変える場合もあるので、複数の動作列をそれぞれ定義しておき、コンピュータに覚えさせる。この部品の3次元位置姿勢データと動作列データをシミュレータに出力する。

### 《計画》

#### 【1-1】 ビジョンセンサ+ロボットによる動作列定義の仕様設計

動作列及び部品の3次元位置姿勢の定義と、定義データを外部出力するファイルの仕様設計（ソフトキューブ株式会社、株式会社三次元メディア、学校法人立命館 立命館大学）《研究目標：各社別の動作列仕様を包含する統一仕様を策定》

【1-2】 各社ロボット別動作列仕様理解

ロボットメーカーごとの動作列の仕様理解（株式会社三次元メディア）《研究目標：各社別の動作列を分析し共通点と差異点を明確化》

【1-3】 各社ロボット別の動作列取得および部品位置姿勢取得の実装

ロボットメーカーごとに、ペンダントで定義した動作列を取得し、部品の3次元位置姿勢とともにファイルに書き込むコーディングとテスト（株式会社三次元メディア）  
《研究目標：各社別の動作列I/F、および3次元位置姿勢I/Fの実装》

【1-4】 動作列ファイル入出力の実装

複数の動作列をまとめた形でファイル書き込みおよび読み込みするコーディングとテスト（ソフトキューブ株式会社、株式会社三次元メディア）《研究目標：干渉シミュレーションとのI/Fの実装》

《実績》

【1-1】

ビジョンセンサ+ロボットによって、ロボット動作を決めるためのシステム仕様を策定した。

【1-2】

各ロボットメーカーのロボットコントローラと制御言語を調査し、ビジョンセンサとの連携方法を設計した。

【1-3】

ワーク把持位置の取得を行った。

【1-4】

動作列情報の入出力

6社のロボットメーカーに対応したロボット連携ライブラリを製作し、それぞれのロボットの仕様は違いに対応する。

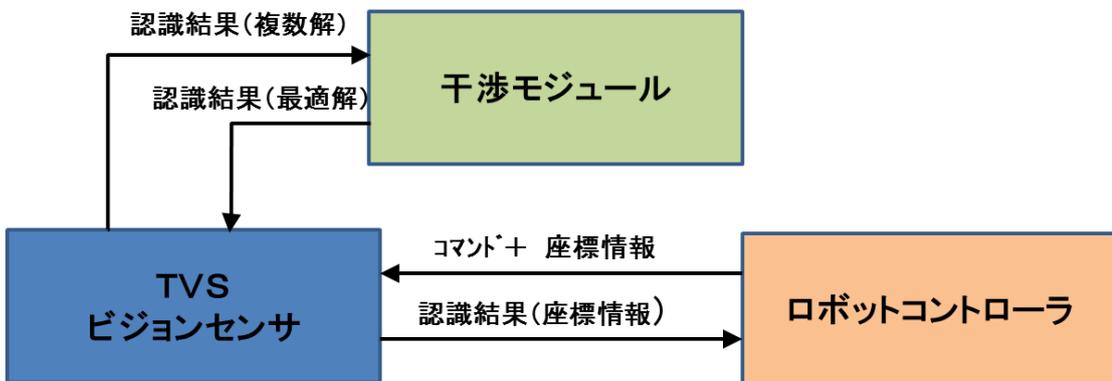
センサモジュールとロボットコントローラの通信コマンド（動作定義情報）は標準化しているため、ロボットが異なっても把持位置情報や動作定義は同じモジュールで対応する。

一方、統一された動作列の定義や干渉回避の処理のために、3次元認識処理モジュールの高速化と安定化が不可欠であり、そのための課題も明確となった。

非熟練者でもビジョンセンサ+ロボットのプログラミングを可能とするためには、3次元認識処理の各種パラメータの設定を自動化、最適化する事が必要でそのための改善も行った。

- ・OSを64ビットに変更し、処理の高速化を図る。
- ・CPUコアの複数化に対応し、並列処理による高速化を図る。
- ・SIMDなど、記述言語による高速化を図る。
- ・認識アルゴリズム改善による高速化、安定化を図る。
- ・ワークの特性に合わせたパラメータ設定の簡素化、可視化を図る。

また、干渉回避プログラムとの連携を図るために、以下のようなモジュール構成とした。



干渉モジュールを加えないシステムではビジョンセンサにおいて、把持すべきワークを1つ選定する。これに対し上記構成では把持対象ワークを複数認識して、その情報を干渉モジュールに渡し、干渉モジュールにより最適解を回答し、その情報をロボットコントローラに送る事とした。

#### 2-4-2 動作列の干渉シミュレーション：オフライン処理

ハンド、部品箱、部品の3次元形状を入力し、動作列が衝突を起すかどうかをシミュレーションする。部品箱のどこに部品がどういう姿勢で置かれているときに既存の動作列が部品箱との衝突を起すかが分かれば、新たな動作列を追加するようにオペレータに促す。

#### 《計画》

##### 【2-1】 シミュレータの仕様設計

ハンド、部品箱、部品の3次元形状モデルと【1】で出力された動作列ファイルを読み込み、動作列の干渉シミュレーションと干渉回避を行うアルゴリズムを開発するソフトウェアの仕様設計。シミュレータ内で動作列を定義するソフトウェアの仕様設計《研究目標：シミュレータの機能仕様の策定》

##### 【2-2】 CADモデルの読み込みの実装

ハンド、部品箱、部品の3次元形状CADファイルを読み込むコーディングとテスト《研究目標：STL形式の読み込みを実装》

##### 【2-3】 シミュレータによる動作列定義の実装

シミュレータ内のバーチャルなロボットハンド・アーム等进行操作することにより【1】と同様の動作列定義を行うコーディングとテスト《研究目標：各社別ロボットの動きをバーチャルに再現する機能の実装》

##### 【2-4】 動作列干渉シミュレーションの開発

動作列ごとに、部品箱における部品の3次元位置姿勢によって衝突を起すかどうかをシミュレーションするアルゴリズム開発・コーディングとテスト《研究目標：アルゴリズムの考案、および高分解能と処理高速化の追求》

##### 【2-5】 干渉回避アルゴリズムの開発

衝突を起す部品の 3 次元位置姿勢があれば、新たな動作列を追加するようにオペレータに促すアルゴリズム開発・コーディングとテスト《研究目標：アルゴリズムの考案、および高分解能と処理高速化の追求》

## 《実績》

### (1)シミュレータの仕様

#### ①概要

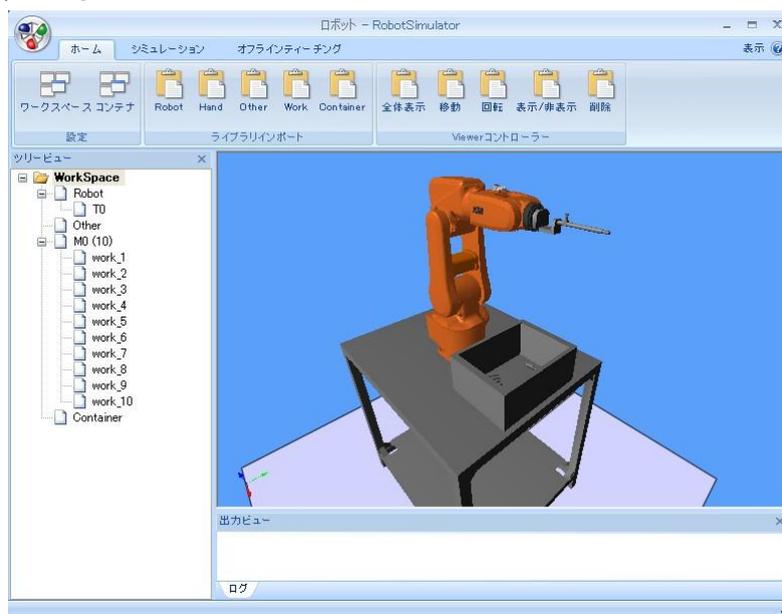
ロボット・ハンド・部品箱・部品などの3次元モデルを取り込み、動作シミュレーションを行うためのプログラムとする。

#### ②画面構成

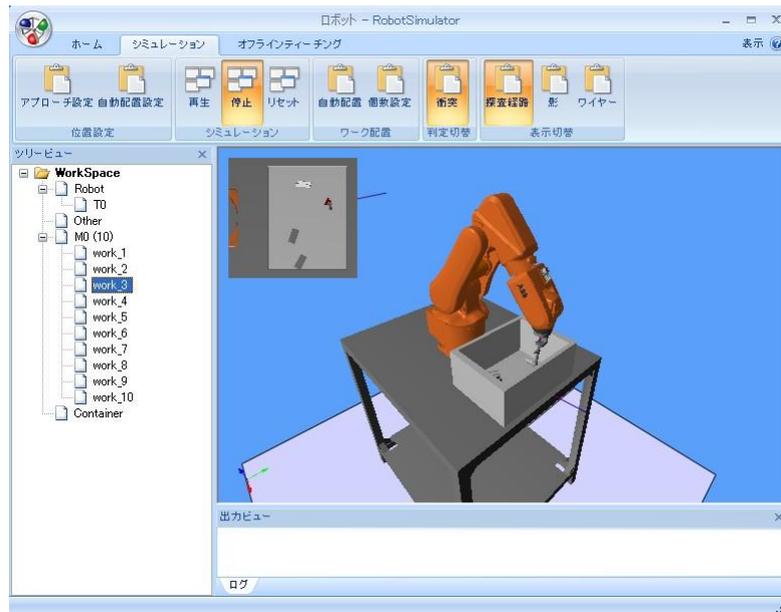
1)大項目として3つのタブ画面で構成される。

1. ホーム・・・モデルライブラリの読込、レイアウト変更、モデル操作などを行う。
2. シミュレーション・・・ワークの自動配置や TVS データ、モデルデータを使ったシミュレーションを行う。
3. オフラインティーチング・・・TVS のピッキング教示ファイルを読み込み、教示データの新規作成・編集・削除などを行う。

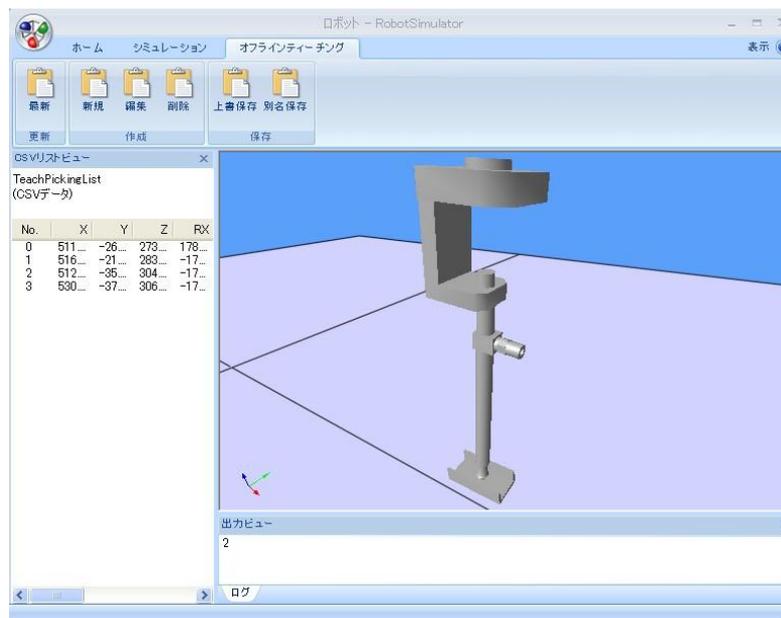
#### ③画面イメージ



1. ホーム画面



2. シミュレーション画面



2. オフラインティーチング画面

#### ④主な機能

##### 1) ホーム画面

##### 1. 設定

- ・ワークスペース、コンテナ

##### 2. ライブラリインポート

- ・Robot、Hand、Other、Work、Container

##### 3. Viewer コントローラー

- ・全体表示、移動、回転、表示/非表示、削除

##### 2) シミュレーション画面

1. 位置設定
  - ・アプローチ設定、自動配置設定
2. シミュレーション
  - ・再生、停止、リセット
3. ワーク配置
  - ・自動配置、個数設定
4. 判定切替
  - ・衝突
5. 表示切替
  - ・探査経路、影、ワイヤー

### ③オフラインティーチング画面

1. 更新
  - ・最新
2. 作成
  - ・新規、編集、削除
3. 保存
  - ・上書保存、別名保存

### ⑤シミュレーション動作の対象範囲について

- 1) モデルへのアプローチ開始地点からデパート終端地点までとする。
- 2) ロボット初期姿勢（ゼロ姿勢）からアプローチ開始地点やデパート終端地点以降の動作はシミュレータでは対象外とする。
- 3) 衝突判定については、コンテナ対ハンド、コンテナ対ワークの判定を行うものとする。

### ⑥動作環境について

- 1) OS・・・Microsoft Windows 7 64bit
- 2) 画面解像度・・・1280×1024 以上
- 3) メモリ・・・2G 以上推奨
- 4) グラフィックボード・・・NVIDIA 8000 シリーズ以上推奨  
(OpenGL3.3 以上)

## (2)CAD モデルの読み込み

### ①読み込み機能について

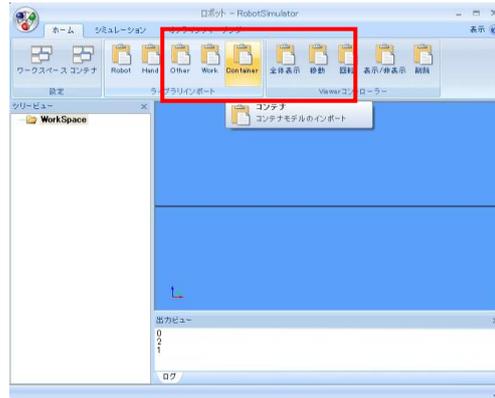
- 1) シミュレータ内で表示する CAD モデル（Hand、Other、Work、Container）の読み込みを行う。

### ②対応ファイルについて

- 1) インポート可能なファイル形式は以下の2種類とする。
  1. STL ファイル形式
  2. 3DS ファイル形式

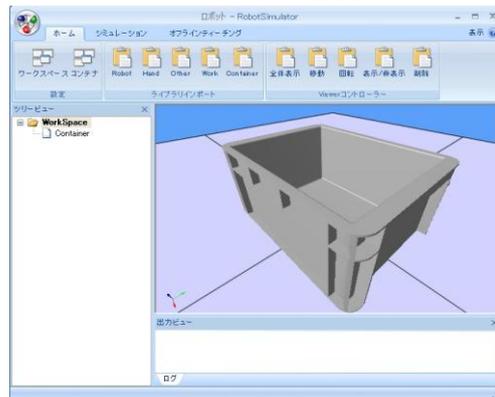
### ③動作について

- 1) リボン内のライブラリインポートボタン押下。



2) ファイルダイアログ表示→ライブラリを選択。

3) ツリービューと Viewer 上に読み込んだ CAD モデルが表示される。



### (3) ロボットモデル

ロボット形状については、市販されているモデリングソフト（3dsMax 等）で作成することを前提とし、3ds ファイルフォーマットでの読込を可能としている。但し、そのまま使用するのはではなく、変換用ツールを使い 3ds からロボット構成ファイルを作成して、シミュレータではこのファイルを使用する。

ロボット情報としては形状情報だけでなく、稼動範囲などの付加情報が必要であり、また同ファイルはシミュレータだけでなく、干渉処理でも使用しており、こちらでは詳細な形状情報ではなく、ボックス形状による簡易形状が必要であることから、ロボット構成ファイルにはその情報も保持している。

#### ① ロボット構成ファイルとは

ロボット形状を表示する為の形状情報（簡易・詳細）や各パーツ形状間の構成（親子）情報、動かす為の稼動制限情報を保有する。

本ファイルはバイナリ形式であり、拡張子を『.rbt』とする。

ファイルはヘッダー部とデータ部の構成となる。

ヘッダー部
データ部

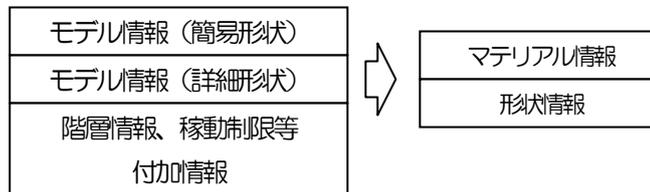
### ③ ヘッダー部

開始バイト	データ型	サイズ	説明
0	uint(※1)	4	ヘッダー部のバイト数
4	char	8	バージョン情報
12	char	32	ロボットメーカー名
44	char	32	ロボットの識別名称
76	-	52	予備

※1 uint = unsigned int

### ③ データ部

データ部はモデル情報（簡易形状と詳細形状）、階層情報及び稼動制限等付加情報等、以下の構成となる。



		データ型/サイズ	説明
モデル情報 (簡易形状)		char[文字列数]	パーツ形状の名称文字列
		uint	形状番号
		uint	形状タイプ[ボックス固定0]
		float[3]	ボックス形状サイズの縦横高さのハーフサイズ
		float[3]	形状原点からボックス中心点へのオフセット値
モデル情報 (詳細形状)	マテリアル情報	uint	マテリアル識別ID
		char[文字列数]	マテリアル名称文字列
		float[3]	環境光カラー値
		float[3]	拡散反射光カラー値
		float[3]	鏡面反射光カラー値
		float	非透過値
		float	輝度
	形状情報	uint	形状番号
		uint	頂点データ数
		float[3]x 頂点数	3次元頂点データ
		uint	法線データ数
		float[3]x 法線数	3次元法線データ
		uint	マテリアル識別ID (面に割り当てる) (※2)
		uint	マテリアル別面数
uint[3]x 面数	インデックス情報		
報 階層情報 付加情報	uint	形状番号	
	uchar[3](※3)	階層フラグ	
	short[2]	移動範囲角度[マイナス角度、プラス角度]	
	float[3]	階層間のオフセット移動量	

#### 1) 補足説明

1. モデル情報（簡易形状）  
パーツ形状分だけ繰り返し情報を格納している。
2. マテリアル情報

使用するマテリアル数分情報を格納している。

3. 形状情報  
パーツ形状数分情報を格納している。
4. マテリアル識別ID (※2)  
1パーツ形状で使用数分、ID・面数及びインデックス情報を格納している。
5. マテリアル別面数  
同一マテリアルを使用する面の数を指定している。
6. 階層情報、付加情報  
パーツ形状数分の情報を格納している。
7. uchar (※3)  
unsigned char を意味する。

#### (4)ワーク配置と物理エンジン

##### ①概要

ワークの位置姿勢をランダムに作りだすため、物理エンジン Bullet を使用し自由落下による自動配置機能を実装した。

##### ②自動配置におけるパラメータ

部品箱およびワークの位置姿勢・寸法、ワーク個数は読込モデルならびにシミュレーション上での配置位置・姿勢より設定する。

ワークをランダムに配置することが目的であるため、演算モデルにはワーク、部品箱のみのモデル化を行った。

落下開始位置については、自動配置設定ボタンより落下開始領域のサイズと領域中心の座標値 (Z は部品箱からの距離) の入力により設定を行う。

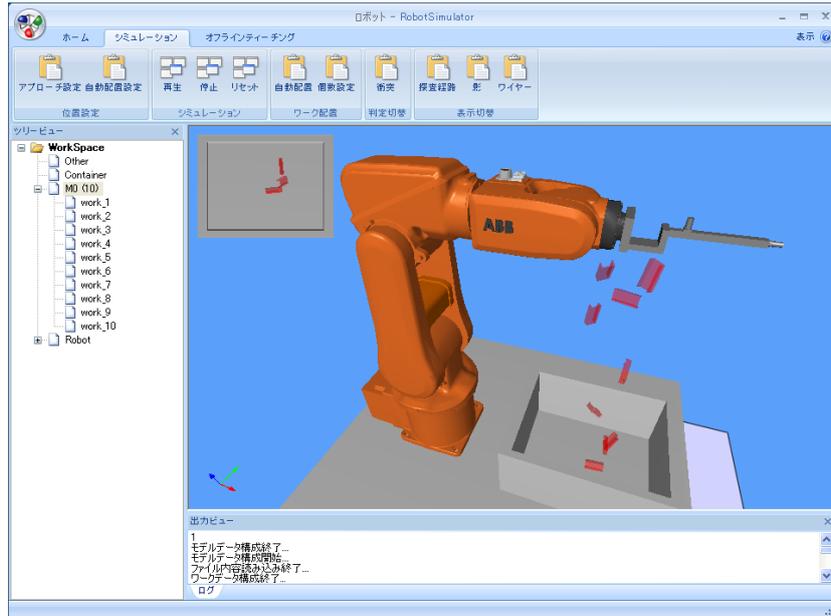
##### ③可視化

Bullet の DebugDraw 機能を利用し、物理演算モデルの可視化 (ワイヤーフレーム表示) を行った。

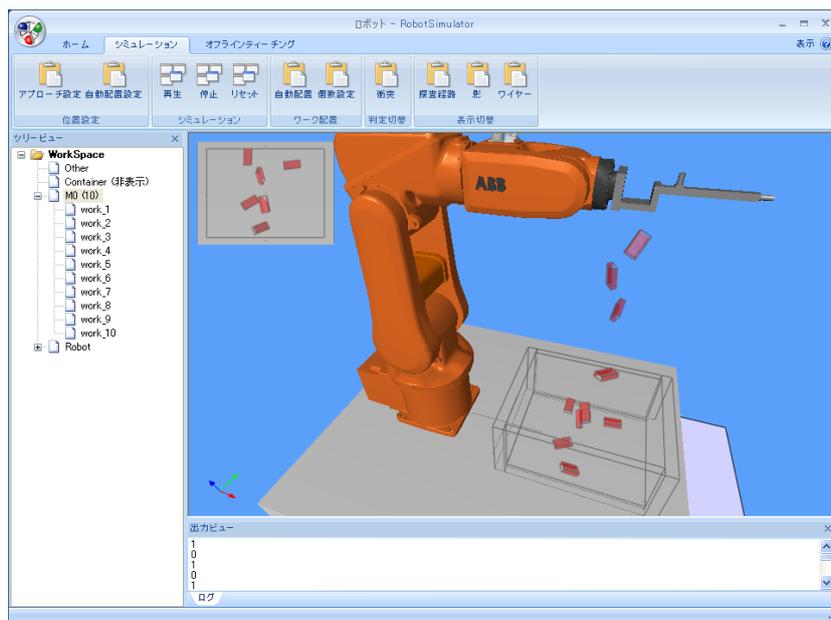
これにより、物理演算時に使用しているモデルとシミュレーション上のモデルを同時に確認することが可能となった。

また、部品箱の内形サイズはユーザーによる手入力設定のため、形状の確認をすることも可能となった。

##### ④画面イメージ



自動配置実行画面



自動配置実行画面（ワイヤーフレーム表示）

## (5) 干渉シミュレーション

### ① シミュレーション機能について

1) ピッキング教示ファイルで定義されている動作列を順番にシミュレータ内で干渉チェックを行いながらシミュレートを行う。

② 干渉チェックについては以下のチェックを行うものとする。

1) コンテナ対ハンド

2) コンテナ対ワーク

③ 動作列が複数定義されている場合は、定義順に干渉チェックを行い正しい動作列

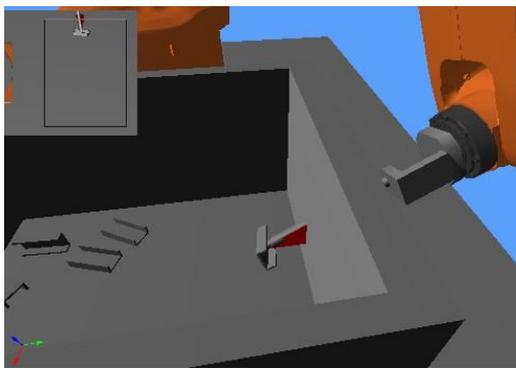
のみをシミュレーション動作として表現する。

④全ての動作列が干渉ありの場合は、エラーメッセージを表示しシミュレーション動作は行わないものとする。

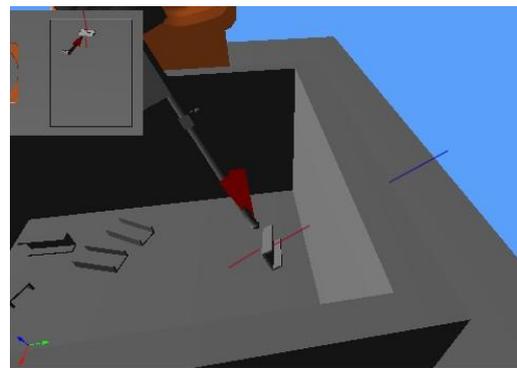
⑤シミュレーション動作範囲は、アプローチ設定により指定された位置から動作列で指定されているピッキング位置・姿勢までの往路と、ピック後のアプローチ設定位置までの復路を動作範囲とする。

⑥干渉については、衝突判定切替ボタンを OFF にすることで干渉チェックを行わず、ピッキング教示ファイルの最初に見つかった動作列を使ったシミュレーション動作を行うものとする。

⑦衝突判定 ON 時/OFF 時の動作について



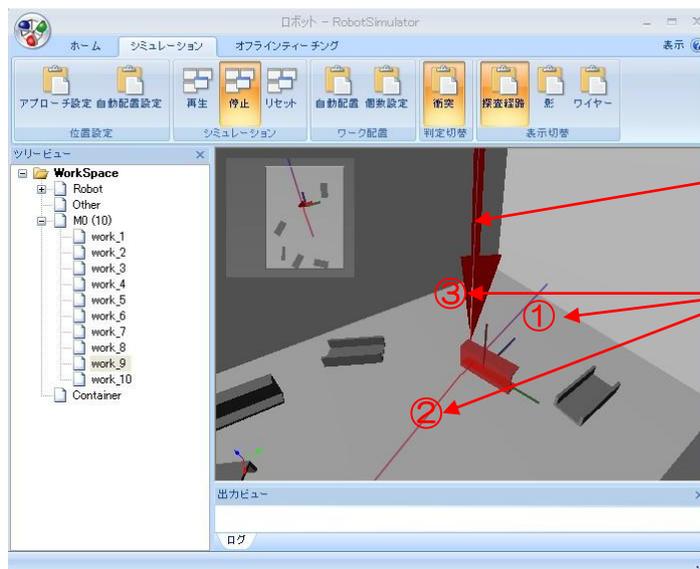
衝突判定 OFF



衝突判定 ON

## (6) 干渉の表現

①衝突判定 ON でシミュレーションを再生した場合、ピッキング教示ファイルに定義されている動作列のピック位置・姿勢を Viewer 上で判断できるように可視化を行う。



①～③の動作列が設定されている場合

①・・・部品箱との衝突がありピック不可。

②・・・IK からロボットの稼働範囲外でピック不可。

③・・・側面からはピック可能。

結果：③をアプローチ経路とし、シミュレーションが再生される。

### 2-4-3 動作列決定アルゴリズム開発：オフライン処理

ビジョンセンサで認識した部品箱と部品の 3 次元位置姿勢に応じて、定義された複数の動作列から最適な動作列を選択する。オフラインで作成されたルックアップテーブルをビジョンセンサに出力する。

#### 《計画》

##### 【3-1】 動作列テーブルの仕様設計

部品箱における部品の 3 次元位置姿勢ごとの動作列選択を記述するルックアップテーブルの仕様設計《研究目標：テーブル設計および高速化》

##### 【3-2】 動作列テーブル入出力の実装

部品箱における部品の 3 次元位置姿勢ごとの動作列選択をルックアップテーブルに記述し出力するコーディングとテスト《研究目標：テーブル I/F の実装および処理高速化》

##### 【3-3】 動作列決定アルゴリズムの開発

部品箱における部品の 3 次元位置姿勢に応じて、定義された複数の動作列から最適な動作列を選択するアルゴリズム開発・コーディングとテスト《研究目標：アルゴリズムの考案および処理高速化》

#### 《実績》

##### (1) 動作列をアプローチとして扱う。

アプローチとは最終的にロボットハンドが到達する地点（到達点）からロボットハンドにおける Z 軸方向をベクトルとして捉えた際の逆ベクトルに任意の距離だけ戻った地点をアプローチ開始位置（開始点）としたとき、開始点から到達点を直線で結ぶ経路を指す。

この経路をロボットハンドの先端が通過する際には、ロボットハンドの手先角度が変更されないものとする。

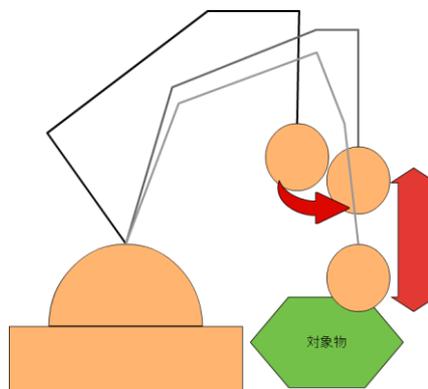


Figure 1 ロボットが対象物を把持するまでの想定している経路を表現している図。右側の大きな赤い矢印がアプローチを表現している。またそれ以外の矢印は安全な地点からアプローチの開始点への移動を表現している。本プロジェクトで取り扱うのは右の赤い矢印アプローチのみである。

(2) アプローチから干渉を判定するには

事前計算を行い、ルックアップテーブルを作成しておくのではなく、実際のロボット動作環境下でロボット動作を著しく阻害しない程度の時間内で位置姿勢の計算を行う。この際に衝突判定の対象となるモデルは、ロボットハンドつまりツールと把持対象物つまりワーク、ならびにコンテナである。アプローチ経路に沿ってロボットアームが変化することから、開始点から到達点への経路よりも到達点から開始点へと戻っていく状態変化に着目する。なぜならば、開始点から到達点への経路、到達点から開始点への経路において、ツールは共に移動物であるが、ワークは開始点から到達点においては不動物、到達点から開始点は移動物である。このことからアプローチに沿ってロボットアームが変化した場合に衝突判定計算が対象とするのは、アプローチの往路、復路のうち、復路に対してのみ計算を行う。またこの際にアプローチを計算速度、衝突判定対象モデルサイズを考慮に入れ、一定の距離で分割し、アプローチ上のそれぞれの点に対して、幾何学的に構築した逆運動学解決器を用い、ロボットアームが特異点付近を通過し、関節変位が不適切な変位を起こさないかを確認することも同時に行う。これによってロボットアームが特異点付近で動作を停止してしまう既存の問題を回避することができる。

(3) ロボットの姿勢をIKによって求める

IKとはInverse Kinematicsの略称であり、逆運動学のことである。逆運動学は順運動学に対する逆動作のことを指し、任意の地点における情報への各関節角の変位を求めるものである。この問題は解決するために利用できる方法は、ヤコビ行列を用いたもの、幾何学的な拘束によって求めるものがあるが、本プロジェクトでは、後者の幾何学的な拘束によって逆運動学の解を求めることにした。既知の問題として、逆運動学では特異点と呼ばれるものがあり、この地点での関節角が一意に解を決定することができないものがある。またロボットアームを実際に動作させる際には前述の特異点付近では、一部の回転角の変位が著しく大きくなり、ロボットアームを制御する上で不適切な動作を起こす可能性がある。

本プロジェクトで利用したABB社製ロボットアームを例にとる。このアームの構成は”Introduction to Robotics, Mechanics & Control” (John J. Craig 著、1986年 Addison-Wesley)での分類を用いれば、RPP-RPRというアーム構成になる。このアーム構成における幾何学的な逆運動学の解算出に関しては、『ロボット工学とその応用』(昭和59年 編集者 電子通信学会)を参考とし、逆運動学解決器を作成した。この時のロボットの幾何学的なモデルはハンドに設置されるツールの先端部がロボットアーム基準系における正面方向つまりは正のX軸方向をむく、各関節角が0度である状態を初期状態として採用している。これとは異なりツールの先端部がロボットアーム基準系における直上方向つまりは正のZ軸方向をむくモデルもあるが、今回は正のX軸方向を基準とするモデルを採用した。その理由は、関節変位がより直感的に理解できるものであることが大きい。

前提条件としてRの回転方向を時計回り、Pの回転方向を反時計回りを正とする。

下記の式を元にして計算を行い、一意な関節角を取得することによりロボットモデルの姿勢を計算した。

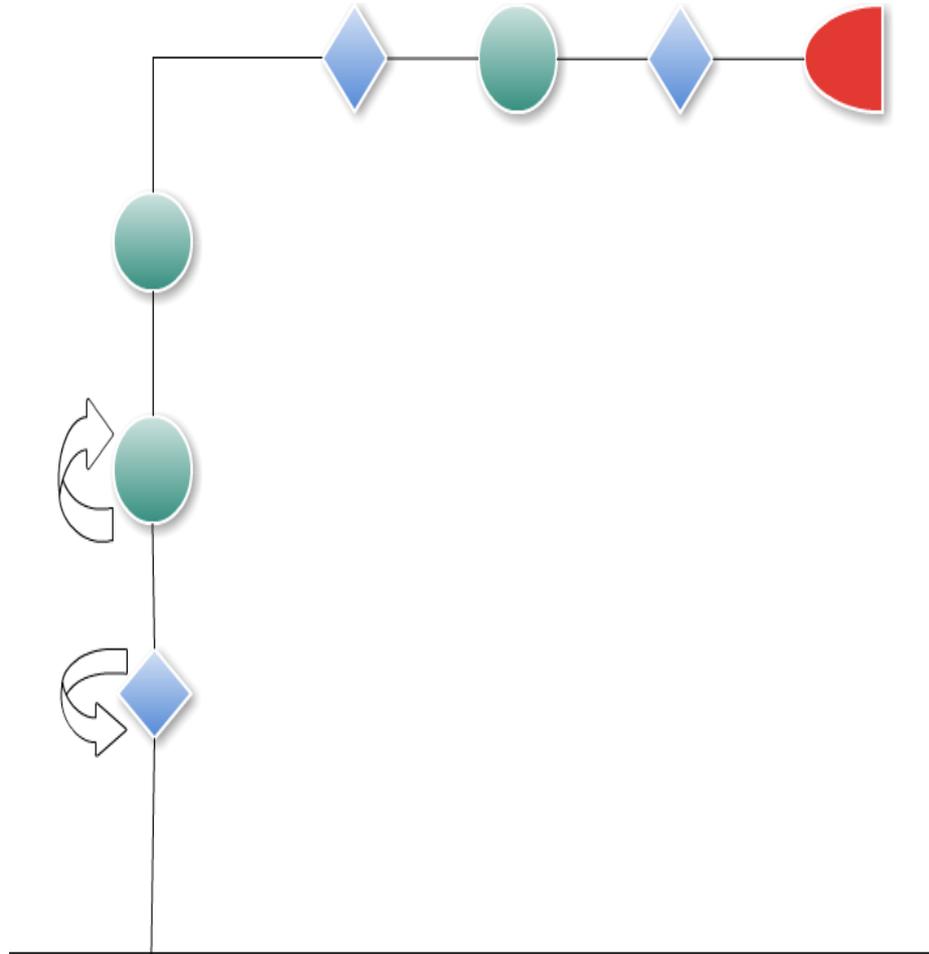


Figure 2 ABB のロボット構成のRPP-RPR モデル図。菱形がRを表現し、丸がPを表現している。この時のRとはZ軸での回転を表し、PはZ軸ではない他の軸での回転を表現している。この場合のPはY軸での回転を表す。

$$\phi_1 = \tan^{-1}(y_5/x_5) \text{ or } \phi_1 = \text{atan } 2(y_5, x_5)$$

$$R = |\vec{j}_5| = \sqrt{x_5^2 + y_5^2 + z_5^2}$$

$$\alpha = \text{atan } 2\left(z_5, \sqrt{x_5^2 + y_5^2}\right)$$

$$c_2 = \frac{l_2 + R^2 - (l_3 + l_4)^2}{2l_2R}$$

$$C_3 = \frac{(l_3 + l_4)^2 + R^2 - l_2^2}{2(l_3 + l_4)R}$$

$$\phi_2 = \alpha - \cos^{-1} C_2$$

$$\phi^3 = \alpha + \cos^{-1} C_3 - \phi_2$$

$$\phi_2 = \alpha + \cos^{-1} C_2$$

$$\phi^3 = \alpha - \cos^{-1} C_3 - \phi_2$$

$$c_4 = (e_{3y}, e_{5z})$$

$$s_4 = (e_{3y} \times e_{5y}, e_{3x})$$

$$\phi_4 = \text{atan} 2(s_4, c_4)$$

$$c_5 = (e_{3x}, e_{wx})$$

$$s_5 = (e_{3x} \times e_{wx}, -e_{5y})$$

$$\phi_5 = \text{atan} 2(s_5, c_5)$$

$$c_6 = (e_{5y}, e_{wy})$$

$$s_6 = (e_{5y} \times e_{wy}, e_{wx})$$

$$\phi_6 = \text{atan} 2(s_6, c_6)$$

Figure 3 上記式は逆運動学を幾何学的に解決するための計算式を表す。与えられた位置姿勢から第5軸の位置ベクトルを算出し、このベクトルから第1軸の角度変位を取得する。のちに余弦定理を用い第2軸、第3軸を求める。この段階で大まかな位置が計算されている。そして残りの第4、5、6軸の角度変位は、外積と内積を用い、それぞれの結果から atan2 を利用し解を得る。

#### (4) コンテナの定義

コンテナとは、ワークつまりは把持対象物を格納しておくための凹形の容器を指す。この形状は一般的な衝突判定では、取り扱いにくいものである。そのためこの形状を凸形の構成物として再構成した。構成は底面形状1個と長辺側面形状各2個ならびに短辺側面形状各2個の5つの衝突判定用境界領域を設定した。またこの形状の設定は外寸をモデリングツールによって作成した3次元モデルファイルを読み込み、包括矩形を計算し、別途設定を行う内寸値によって自動計算に

よって5つの衝突判定用境界領域を作成する機能を実現した。

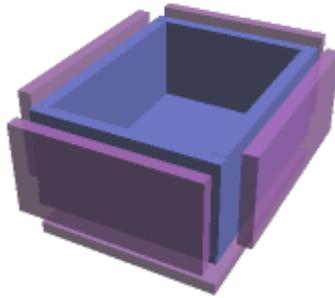


Figure 4 紫色の包括矩形モデルと赤紫色の衝突判定用に分割された包括矩形モデルの図

#### (5) 干渉判定

干渉判定は、その対象としてワークとハンド、そしてコンテナを取り、判定を行う範囲はアプローチ終端点からアプローチ開始点までの経路としている。

干渉判定の基礎的な部分は Bullet ライブラリによる実装を利用している。Bullet ライブラリは zlib ライセンスで提供されているオープンソースプロジェクトである。利用範囲は広く、ゲーム分野などでの利用もある。またマルチプラットフォームであり、移植性も高いことが特徴に挙げられる。

本プロジェクトでは、精度よりも高速に処理を行うことが求められており、また利用が容易であり、メンテナンスを行えることが求められていた。

Bullet はオープンソースであり、また公開されている API が利用しやすく、ライブラリのもつコンセプトも利用しやすく、多面体同士の衝突もこなせることが特徴に挙げられている。これらの点で他の物理エンジンと比較した結果、Bullet を採用した。

干渉判定の目的は、コンテナ内に配置されているワークを安全に把持することであり、この場合にコンテナとハンド、もしくはワークが衝突することを事前のシミュレーションで判定することである。

この目的を考慮に入れた場合、複雑なモデル形状による精密な衝突判定は、モデルと実形状の間の誤差によっては衝突を起こす危険性を含む。また高速な動作をさせる場合に、複雑な形状は処理を高コストにする原因であるため、本プロジェクトでは、包括矩形によるモデルの近似形状を利用して衝突判定を行っている。

実際のワークやハンドは凹形状であることもしばしばあり、実際の利用を鑑みると、動的に凹形状を分割する実装が望ましいと思われるが、実際にはワークとハンドの間で衝突判定を行わず、詳細なモデルでなければ通過できない経路を採用すること自体が安全な運用から逸脱しているため、ハンドならびにワークは近似凸形状でも問題がないと結論づけた。

#### 2-4-4 各社ロボット別インライン制御の実装：インライン処理

ビジョンセンサで認識した部品の 3 次元位置姿勢に合わせて選択された動作列に対して部品の 3 次元位置姿勢を付与しインラインでロボットを制御する。

## 《計画》

### 【4-1】 各社ロボット別制御プログラム仕様理解

ロボットメーカーごとのロボット制御プログラムの仕様理解（株式会社三次元メディア、ソフトキューブ株式会社）《研究目標：各社別の制御プログラムを分析し共通点と差異点を明確化》

### 【4-2】 ロボットベース座標系への位置姿勢変換

3次元ビジョンセンサの座標系とロボットベース座標系との変換を施し、ロボットベース座標系におけるハンドリング対象物の3次元位置姿勢を求めるコーディングとテスト（株式会社三次元メディア）《研究目標：各社別の座標系変換の実装》

### 【4-3】 各社ロボット別インライン制御の実装

ビジョンセンサで認識した部品の3次元位置姿勢に合わせて選択された動作列に対して部品の3次元位置姿勢を付与するコーディングとテスト（株式会社三次元メディア）《研究目標：各社別のインライン制御の実装》

### 【4-4】 各社ロボット別動作列通信ソフトの実装

部品の3次元位置姿勢を付与した動作列をロボットに送る通信ソフトのコーディングとテスト（株式会社三次元メディア）《研究目標：各社別の通信ソフトの実装》

## 《実績》

### 【4-1】

各社ロボットメ別の制御プログラムの仕様

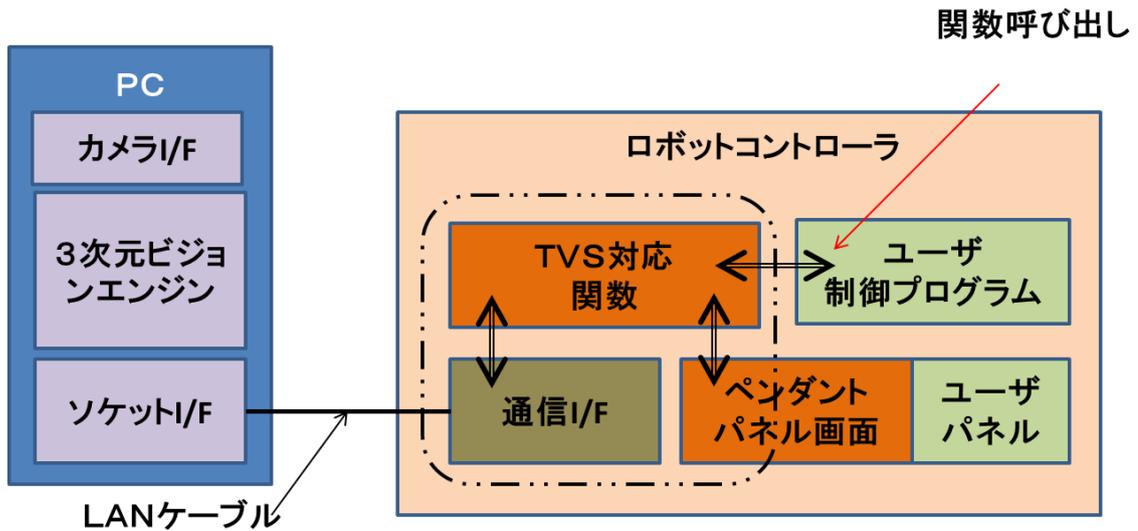
各社のロボット制御プログラムの仕様を調査し、最適なシステムを決定した。ロボット毎の仕様の違いは、ロボットメーカーの協力により情報開示して頂き検討した。また、ロボットの専門家としてのアドバイスや課題解決の糸口を頂いた。

ロボットメーカー毎の制御言語の違いにより、下記の2つのタイプに分かれる。

#### (1) ロボット言語サブルーチンモジュール型

下記のメーカーのロボットがこのタイプである。この仕様であれば、各ロボットメーカーの標準仕様または拡張機能の追加で対応可能である。

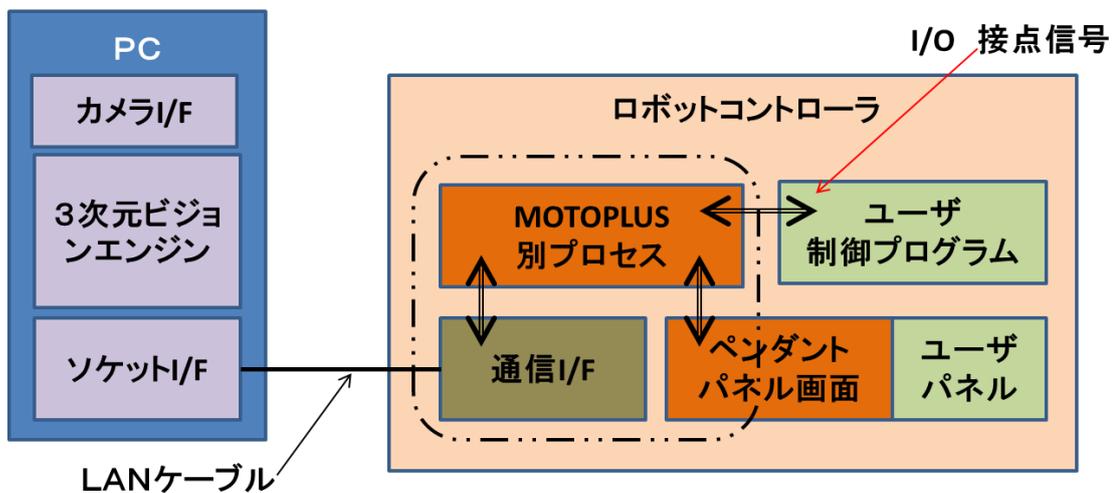
- ・デンソーウェーブ
- ・川崎重工業
- ・三菱電機
- ・ABB
- ・不二越



## (2) 別プロセス追加型

下記のメーカーのロボットがこのタイプである。この仕様では、ロボット制御言語とは別に製作された処理モジュール（プロセス）を追加する必要がある。ロボットユーザには別モジュールの提供とその動作環境をロボットメーカーに準備してもらう必要がある。

・安川電機



## 【4-2】

### ロボット座標系への位置姿勢情報の変換

ロボットがワークを把持するために必要な位置姿勢情報をビジョンセンサからもらうためには、事前にビジョンが認識する座標とロボットの座標へ変換する必要がある。すなわち、ビジョンセンサとロボットが交換しあう座標はすべてロボットのベース座標系で行う。

画像によりロボットの座標系で位置姿勢を求めるには、以下のような3つの校正

(キャリブレーション) を事前に行う。

①カメラ校正

カメラ、レンズなどの正確な寸法情報取得やレンズのゆがみを補正する。

②ステレオ校正

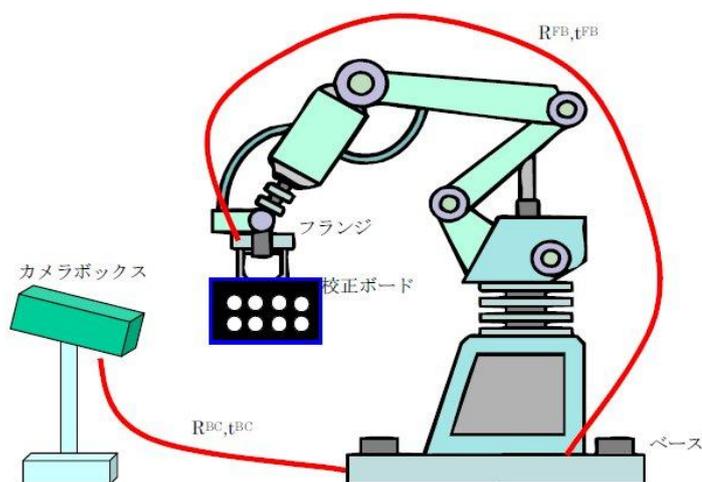
2台のカメラ位置を正確に求め、3角測量に必要なパラメータを求める。

③ハンドアイ校正

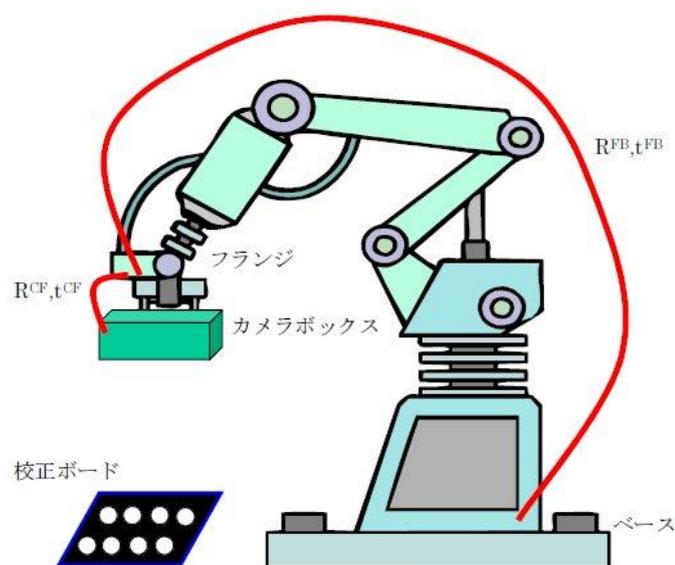
ビジョンで求める3次元座標をロボットのベース座標系に変換する。

ロボットを使用した生産設備において、簡単にこれらの校正を行う手段を設計し、実装した。校正はビジョンとロボットコントローラが通信によりコマンド制御し、自動的にを行うようにした。

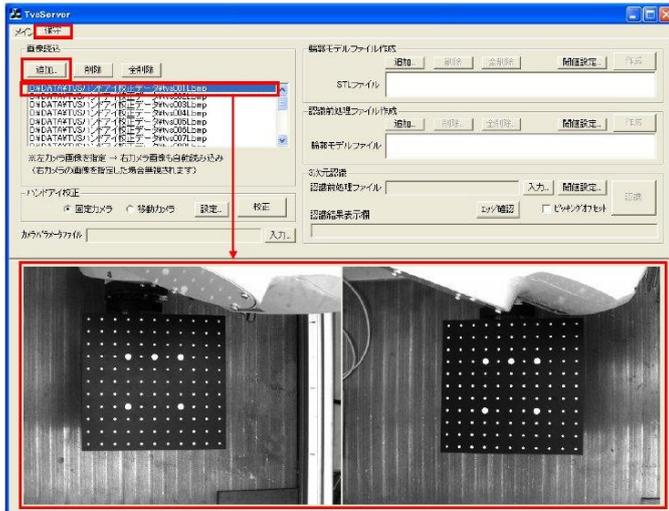
校正方法は、カメラを固定配置した場合と、ロボットのハンドに取り付けた場合を想定し、両方に対応している。



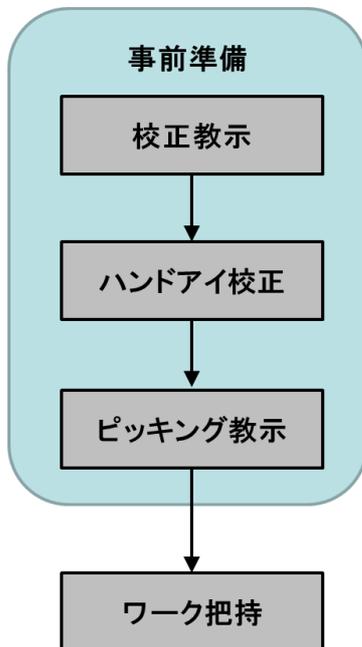
固定カメラによる校正



移動カメラによる校正



校正作業の設定画面



このように、ビジョンセンサ+ロボットによるピッキングシステムを非熟練者でも短時間で校正できるように、校正作業、ピッキング教示を簡単にできるモジュールとした。  
事前準備作業は左図のような手順で実施する。

#### 【4-3】

各社ロボット別の位置姿勢制御プログラムを実装した。

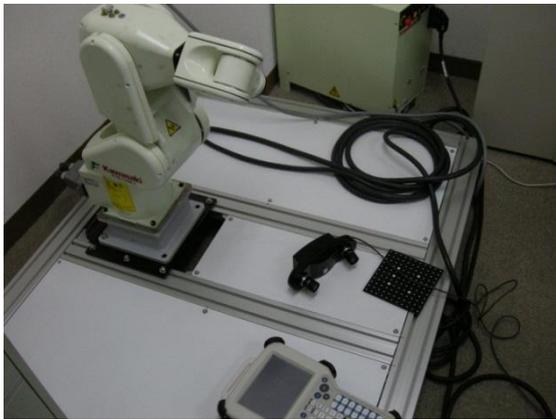
(1)デンソーウェーブ



ロボットのフランジに移動カメラを取り付け、ワークの位置認識をおこない、把持ポイントを示す事で、動作確認を行った。  
また、ワーク把持のための事前準備を非熟練作業でも行えるように、ロボットの操作用ペンダントに情報表示と操作ボタンを配置した。



(2)川崎重工業



川崎重工業のロボットもフランジにカメラを取り付けて、評価を行った。ペンダントの操作画面は制御盤のパネルの代用としての設計コンセプトで、デンソーウェーブのように自由度は高くないが、基本機能を組み込む事ができた。



### (3) 安川電機



標準の通信機能では、統一されたビジョンセンサとのコマンドをする事ができないため、安川電機から別途提供されているロボット制御用プログラムSDK「MOTOPLUS」を用いて別モジュールとしてロボットコントローラにインストールする。これは、制御プログラム（ロボット動作制御言語）とは全く別タスクとして機能するため、ユーザは意識することなくビジョンセンサと通信が可能である。制御プログラムとの連携は仮想的なI/Oポートで行うため、PLCプログラマなども理解しやすい。

ることなくビジョンセンサと通信が可能である。制御プログラムとの連携は仮想的なI/Oポートで行うため、PLCプログラマなども理解しやすい。

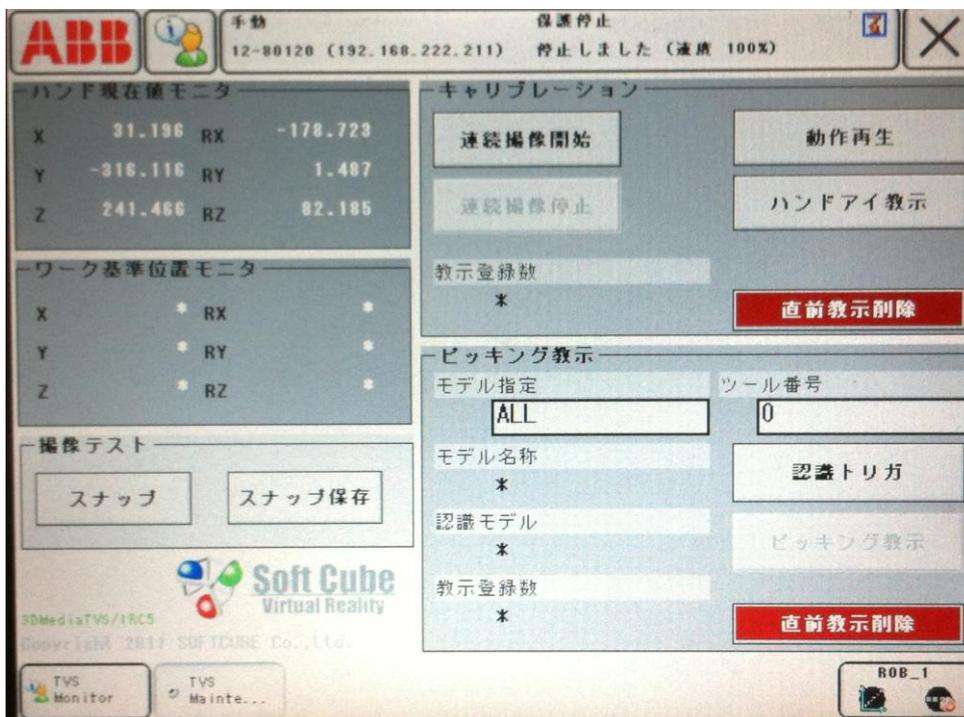
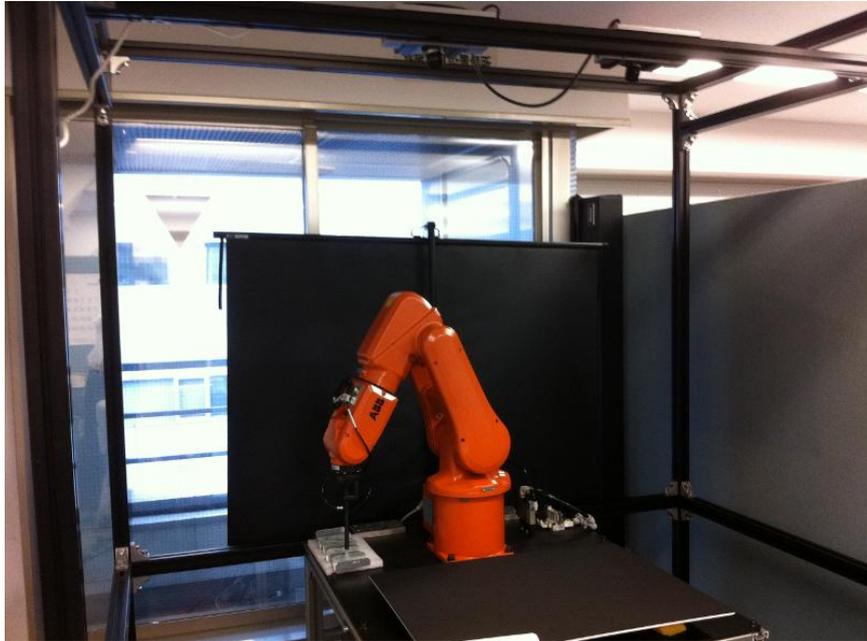


### (4) 三菱電機

三菱電機のロボットはPLCなどとの親和性が高い。また操作用ペンダントもOSの乗ったターミナル構造である。



ABB ロボットでは、フレーム上部に固定カメラを取り付けてワークの認識を行っている。操作ペンダントは三菱と同じくOSの乗ったターミナル構造である。



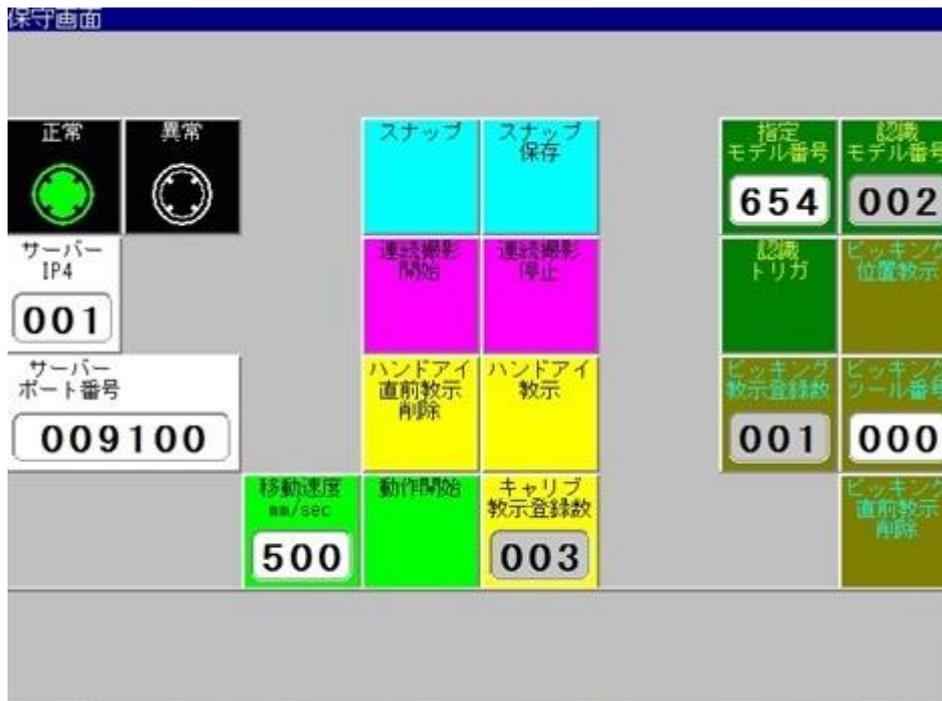


(6) 不二越

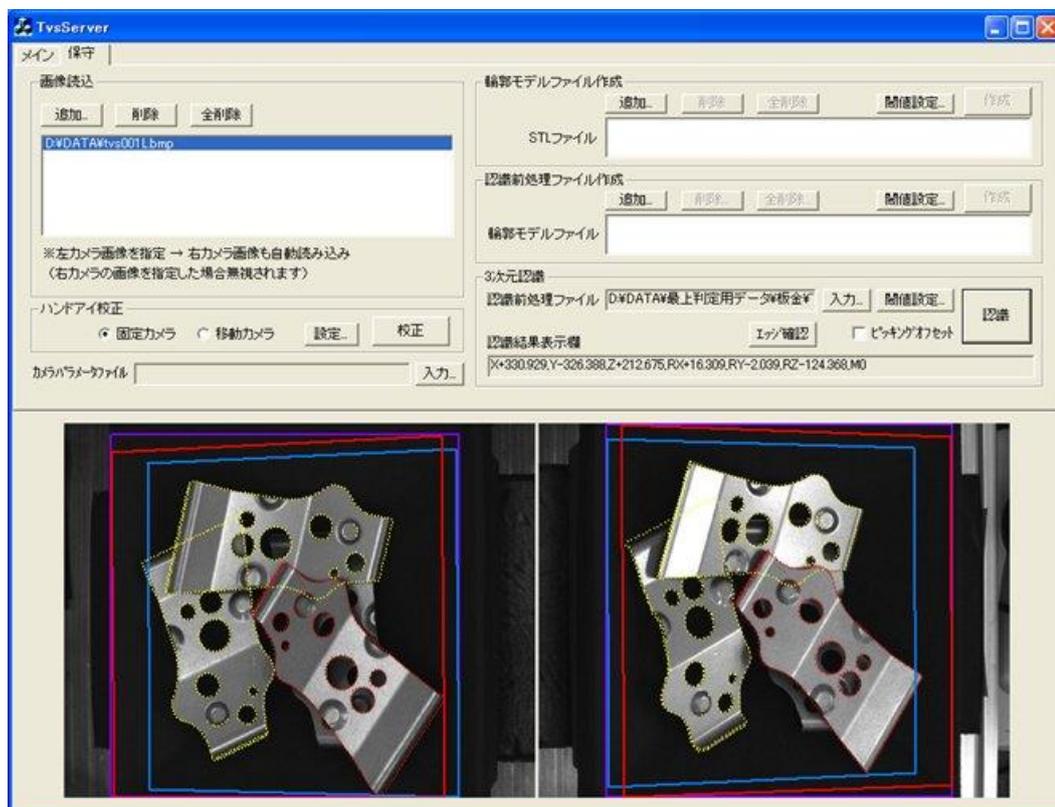


不二越は比較的大型のロボットが多く、フランジにカメラを取り付けるツールの寸法が合わなかったために、変換アダプタを用いて取り付けた。通信コマンドの送受において、一部制御コードが異なるため、他のメーカーのロボットとの互換性を保ちながら、センサー側の処理を変更し対応した。

操作画面は、照光 SW 式操作ボタンをレイアウトする方式で作っている。基本的操作は他社のロボットと同様に仕上げています。



標準ワークを認識した場合のビジョンセンサの画面を下記に示す。この画面は、各社のロボットと連携しても同じ画面で対応可能である。



【4-4】  
各ロボット用通信制御の実装

前項のように、各社のロボットと通信はすべてLAN接続のソケット通信で構成する事ができた。一部安川電機のように通信機能が限定仕様であったために、特別なモジュールを製作したが、基本的コマンドを統一して使用できる事が確認できた。

このコマンドは、初期の設計方針より可視化ができる文字列で定義されているため、今回対応しなかったロボットとの連携も可能と考えられる。また高度な動作を実現させるために、PCでの処理を追加する場合も、このコマンド仕様であれば対応できる事が確認できた。

また、課題であった3次元認識の処理の高速化・安定化も進め一定の成果があった。

#### (1)高速化

- ・3次元認識の並列化：3～4倍の高速化ができた。
- ・64ビットOS対応：1.5倍程度の高速化ができた
- ・アルゴリズムの改善、SIMD仕様：2倍程度の高速化ができた

#### (2)安定化

認識処理の安定化のためアルゴリズムの改善を行い、認識解の再現性の向上ができた。ただし、処理量が増えたため処理速度が低下したが、高速化の成果により、トータルの高速化は得ている。

成果としては、試験標準ワークでの認識は、従来1個約4秒かかっていたが、現在0.4秒まで改善する事ができた。

## 最終章 全体総括

本研究開発においては、以下の成果を得た。

三次元ビジョンセンサにより認識されたワークの位置姿勢に対して、事前にティーチングにより定義された複数の動作列から、〈最も安全にピックアップ可能なもの〉を選択し、ピックアップを実行させるための一連の（ニインライン及びオフライン双方）ソフトウェアの研究開発を行った。ここで、〈最も安全にピックアップ可能なもの〉とは、通常ワークが格納されることになる部品箱に、ロボットアーム及びハンドが干渉しないこと、及び選択されたワークがロボット可動範囲に存在する、ということである。このソフトウェア研究開発により、三次元ビジョンセンサを搭載した産業ロボットの応用範囲が、飛躍的に拡大されることが期待できる。

研究開発後の課題・事業化展開については、以下のとおりである。

課題については、干渉計算の正確性に関することがある。今回は、ロボットの逆運動学計算として、一般的な六軸ロボットに関する、いわゆる教科書的な逆運動学計算を用いた。しかるに、近年のロボットにおいては、ロボットの動力学的側面を考慮し、それを逆運動学計算に反映させる試みが行われている。この場合、干渉計算を行う前提となるロボットの軌道が、計算と実際とで異なる可能性がある。この不一致を回避するためには、ロボットの逆運動学計算方法を、各ロボットメーカーから提供を受けられるのが望まれる。

もうひとつの干渉計算の正確性に関する課題としては、部品箱の実際の配置が、シミュレータに入力された配置と異なる場合への対処である。現在の実装では、これらが正確に一致していなければ、干渉計算に不都合が生じる。これを避けるためには、インライン時に、何かしらの方法で、部品箱の実際の配置を認識する必要があるが、これは未解決の問題である。

事業化展開については、今回の研究開発予算で購入したロボットについては、既に製品化が視野に入っている。しかるに、それ以外のロボットについては、逆運動学計算の実装、及びロボット形状の取得が必要不可欠である。研究開発されたソフトウェアについては、そのようなロボット依存データを容易に組み込めるようには設計されているものの、実際の実装及びテストは、今回の研究開発スケジュールにおいてはなされていない。これが今後の事業化展開における課題となっている。

一方では、3次元ビジョンセンサを用いたロボットシステムは、市場では漸く導入期に入ったところであり、新市場の開拓に当たって、本研究の成果が起爆剤となることを期待している。よって、サポイン事業での研究開発はひとまず終了となるが、事業化展開は、ここで手を止めることなく、引き続き、計画を進める予定である。ロボット依存データについては、ロボットメーカー各社の各機種毎の対応が必要であるため、その調達費用が課題となっているが、今後は、具体的な顧客ユーザへの対応の中でさまざまな提供方法を模索していくことになる。

以上

<参考文献一覧>

- 1) 高野 政晴、“詳説 ロボットの運動学”、オーム社、2004.
- 2) 辻 三郎、江尻 正員、“ロボット工学とその応用”、電子通信学会、1984.
- 3) 広瀬茂雄、“ロボット工学”、裳華房、1987.
- 4) Tomas Akenine-Möller, Eric Haines, Naty Hoffman, “Real-Time Rendering 3rd edition”, AK Peters, 2008.
- 5) Randi J. Rost, Bill Licea-Kane, “OpenGL Shading Language 3rd edition”, Addison-Wesley, 2010.
- 6) Christer Ericson, “Real-Time Collision Detection”, Morgan-Kaufmann, 2005.