

設備全体の仮想化を実現する物理シミュレーションを用いたカスタムメカニカル機構

長谷川 直人, 岩村 慎太郎, 島川 はる奈, 白田 聖人

近年は市場ニーズの変化が速く、製品ライフサイクルの短命化が進んでいる。このような市場動向に対応するため、より短い期間での生産設備の立ち上げが必要となる。これまでにオムロンはFA統合開発環境 Sysmac Studio において仮想化技術を導入し、3Dシミュレーションによる設備の事前検証を可能にした。

設備はロボットや周辺機器など、様々な要素で構成されるが、一つでも仮想環境上で再現不可の機器がある場合は設備全体を対象とした3Dシミュレーションができないため、事前検証することができない。3Dシミュレーションは各機構の仮想化モデルの組み合わせにより実現するが、従来は汎用の機構に対してのみサポートしていた。従来技術でサポート外の機構がある場合、その機構の仮想化モデルを作成できず、3Dシミュレーションに組み込むことが出来なかった。そこで、ユーザが任意に可動部品や結合点の動作を簡略に定義できるカスタムメカニカル機構の仮想化モデルを開発した。これにより、これまで検証できなかった機構を含む設備の事前検証が可能となり、生産設備の立ち上げ時間の削減を実現した。

Custom Mechanics to Realize Virtualization of Whole Facility Using Physical Simulation

HASEGAWA Naoto, IWAMURA Shintaro, SHIMAKAWA Haruna and SHIRATA Seito

In recent years, rapidly changing market needs has led to a trend of shorter product lifecycles. To keep up with such market trends, it is necessary to shorten launching time for production facilities. We introduced virtualization technology in our FA integrated development environment, Sysmac Studio, which enabled debugging without physical machines using a 3D simulation.

A production facility consists of various elements such as robots and peripheral devices, and if it contains even a single device that cannot be reproduced in the virtual environment, a 3D simulation of the whole facility is not possible. Although a 3D simulation is realized by the combination of virtualization models of each mechanism, our conventional virtualization technology only supported the general mechanisms and virtualization models were not available for the unsupported mechanisms. To solve this challenge, we have developed virtualization models for custom mechanics that allow users to easily define movements of movable parts and joint points. This enabled a 3D simulation of a facility containing the mechanisms that had not been supported and further contributed to reduction in launching time.

1. まえがき

近年は消費者の嗜好変化が激しく、製品のタイムリーな市場投入が必要である。そのため、生産設備の短期立ち上げが求められている。これまでにオムロンはFA統合開発環境 Sysmac Studio において仮想化技術を導入し、3Dシミュレーションによる設備の事前検証を可能にした¹⁾。

3Dシミュレーションでは設備を構成する装置を仮想化する必要がある。Sysmac Studio では装置を仮想化するために13種類の汎用メカニカル機構の仮想化モデルを用意している。ユーザは仮想化したい装置に含まれる各機構に対して機構種別を選択し、可動部品のCADデータをインポートして決められた設定を行う。この機能により、XYテーブルや直交ロボットなど、FA業界で使用されている多くの機構を含む装置の仮想化が可能となった。しかし、

Contact : HASEGAWA Naoto naoto.hasegawa@omron.com

定義済みの13種類の機構種別に当てはまらない、メーカー独自の動作機構や、クランプ機構のような物理的なジョイントを含む装置の仮想化は実現できない。そのため、これらのサポート外の機構を一つでも含む装置がある場合は、3Dシミュレーションによる設備の事前検証ができなかった。

この問題を解決するために、可動部品の動作や可動部品間の結合方法をユーザーが任意に設定できる、カスタムメカニカル機構の仮想化モデルを開発した。

本稿では、2章で従来の仮想化技術の効果とその技術に対する課題を述べる。3章ではカスタムメカニカル機構を実現するための技術の詳細を説明し、4章でカスタムメカニカル機構の有効性の検証結果について述べる。そして、5章にて本稿での総括及び今後の展望と課題を述べる。

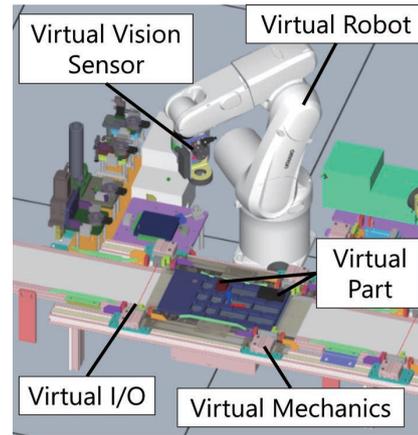


図2 仮想化した生産設備の例

2. 従来技術と課題

2.1 オムロンが提供する従来の仮想化技術とその効果

仮想化技術が普及する以前の生産設備の開発は、一般的に実機のみで行われていた。オムロンでは生産設備の立ち上げ工数の削減を目指し、FA統合開発環境であるSysmac Studioに生産設備の仮想化技術を導入した。生産設備全体の構成例を図1に示す。構成要素にはセンサ制御に使用するI/O、画像センサ (Vision Sensor)、周辺機器の制御に用いるサーボモータ (Servo Motor)、及びロボット (Robot) が含まれる。そして、それらの構成要素はワーク (Part) を作業対象として検出・搬送等を行う。これらの要素の仮想化を実現するため、5つの仮想モジュール「仮想I/O (Virtual I/O)」、「仮想画像センサ (Virtual Vision Sensor)」、「仮想周辺機器 (Virtual Mechanics)」、「仮想ロボット (Virtual Robot)」、「仮想ワーク (Virtual Part)」を実現した。仮想化を実現した生産設備の例を図2に示す。

従来技術を活用すると実機のみの場合に比べ、立ち上げ工数を56%削減できることが分かった¹⁾。実機のみによる生産設備立ち上げの開発プロセスと仮想化技術を用いた時の開発プロセスの違いを図3に示す。

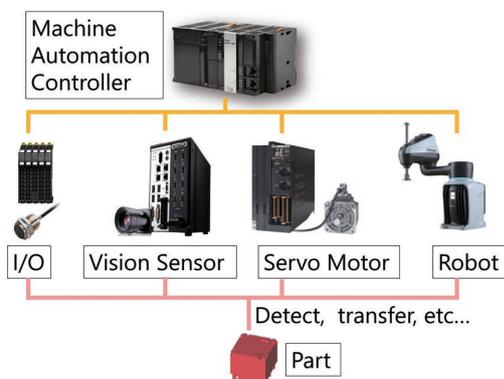


図1 生産設備の制御システム構成例

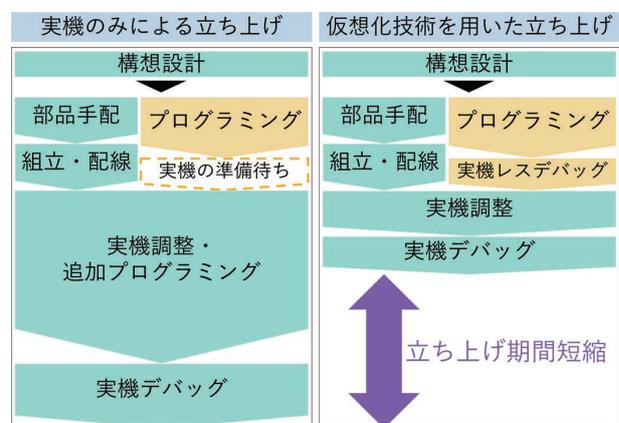


図3 生産設備の立ち上げの開発プロセスの違い

2.2 従来の仮想化技術の課題

生産設備を構成する要素のうち、特に周辺機器の種類は多岐に渡る。従来の仮想周辺機器の要素であるメカニカル機構でサポートする種別は「直動機構」、「回転機構」、「エアシリンダ」など汎用的な13種類であり、これらを作成可能な機構として定義していた。しかし、周辺機器はユーザーの生産設備に対応するようにカスタマイズして作られることがある。カスタマイズして作られた周辺機器は、汎用的な13種類でサポートできない機構を含むことがある。具体的には設定内容が製品メーカーによって異なる電動シリンダ及び電動チャック、物理現象のシミュレートが必要なクランプ機構などである。このような機構を含む周辺機器は、従来の仮想化技術では仮想環境上で再現できない。生産設備の中に一つでも再現不可の周辺機器があると、仮想環境上での3Dシミュレーションができなくなる。また再現不可の周辺機器は実機を使ってデバッグや調整をせざるを得なくなり、その分生産性が低下する。仮想環境で設備全体を対象とした3Dシミュレーションを行うには、これらの周辺機器を仮想環境上で再現する必要がある。

3. カスタムメカニカル機構の実現

2章で述べた従来の技術に対する課題を解決する手段として、ユーザが任意の機構の仮想化モデルを作成できる機能の実現に取り組んだ。ユーザがメーカーごとの設定項目、設定値を任意に変更、あるいは物理シミュレーションを含めた機構を独自に定義し、それを仮想化モデルとして実現することができれば、仮想環境での3Dシミュレーションが可能となり、上記課題を解決することができる。汎用的な13種類の機構をサポートするメカニカル機構に対し、上記の機能を持つ機構をカスタムメカニカル機構と定義した。

カスタムメカニカル機構を実現するにあたり、重要な要素となるのが、可動部品同士を結合するジョイントである。ジョイントの動作を実現するためには物理現象をシミュレートする物理シミュレーション技術を用いる必要がある。その実現にあたっての課題解決策を3.1節で述べる。3.2節では、カスタムメカニカル機構と既存の仮想モジュールとの同期を実現し、仮想環境上で設備全体を対象とした3Dシミュレーションを実現する技術について述べる。また、複雑なカスタムメカニカル機構を含むシステムをユーザが簡単に設定できるようにするための技術を3.3節で述べる。

3.1 可動部品同士を結合するジョイントの実現

可動部品間が物理現象に従った結合を含む機構を再現するためには、物理シミュレーションを用いて可動部品間を接続し、その間の動作の制約条件を設定する必要がある。Sysmac Studioにおける物理シミュレーションは、NVIDIAが提供している、リアルタイムの物理演算エンジンPhysXを用いて実現している。PhysXにはx, y, z軸とそれぞれの軸の回転を独立して制御できるD6ジョイントが定義されている²⁾。しかし、D6ジョイントは機構のジョイントのメカニズムを直接表現したものではないため、そのままのインタフェースでは簡略な設定を目指すカスタムメカニカル機構のジョイントとして使用できない。そこでユーザに対して直感的に理解しやすいカスタムメカニカル機構のジョイントとして使用できるようにするために、機構設計の定石集であるからくり設計のメカニズムの分類を参考にジョイントを定義した³⁾。この定義に基づいてD6ジョイントのインタフェースをカスタマイズして表1に示す独自のジョイント群を開発した。

表1 カスタムメカニカル機構のジョイント群

ジョイント	説明
固定ジョイント	2つの可動部品を固定する接続方法。
ヒンジジョイント	2つの可動部品をヒンジジョイントで接続する方法。
ボールジョイント	2つの可動部品をボールジョイントで接続する方法。
スライダージョイント	2つの可動部品を直線方向に接続する方法。
回転ジョイント	2つの可動部品を一つの軸を基準に回転し接続する方法。
親子ジョイント	2つの可動部品の親子関係による接続方法。

独自ジョイントの一例として、スライダージョイントの動作を図4に示す。本ジョイントはD6ジョイントのx軸以外のすべての並進・回転動作をロックすることで実現している。親子ジョイント以外のジョイントは、同様にD6ジョイントの一部動作方向をロックする方法により実現した。

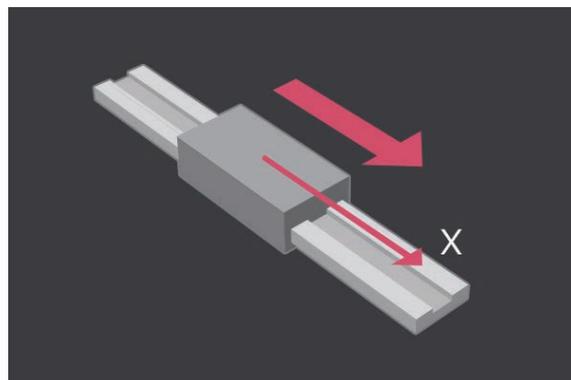


図4 スライダージョイント

PLC (Programmable Logic Controller) のモーション軸を用いた位置制御と、ジョイント機構を組み合わせた電動チャック機構などを再現するために親子ジョイントを実現した。この機構においては可動部品の一部がPLCのモーション制御で動作すると、その可動部品の動作に合わせてジョイントを持つ別の可動部品が追従するという部品間の座標更新に親子関係が存在する。単純にそれぞれの機構部品を動作させた場合、追従すべき部品の位置の更新が次の周期で行われるため、機構全体として見ると本来期待される位置に対して遅れが生じてしまう。これを解決するためPLCのモーション軸の指令値に合わせて可動部品の位置を更新し、その直後に更新された位置を用いてジョイントの物理シミュレーションを行い、その結果から機構全体の

3D ワールド座標を求めるという一連動作を機構ごとに独立させた。これによって親子ジョイントを持つ機構が他の機構などの動作から遅れることなくシミュレーションできるようになった。これらの技術で実現したカスタムメカニカル機構により、仮想化可能な装置の大幅な拡大を実現した。

3.2 設備全体を対象とした 3D シミュレーションを実現するための要素間同期の実現

Sysmac Studio のシステム全体構成図を図 5 に示す。

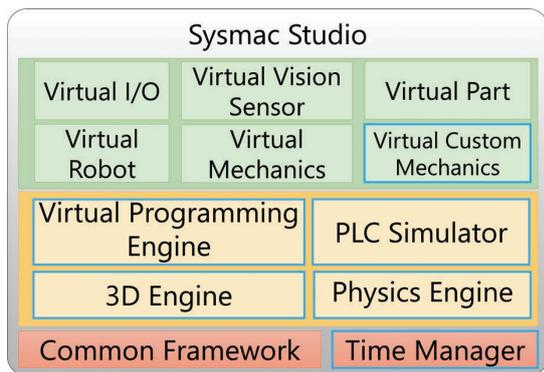


図 5 システム全体構成図
(緑: 仮想化対象、黄: 仮想化エンジン、橙: ソフト基盤)

仮想化した全てのモジュールを一つのソフトウェアで実現するための基盤となるのが、「Common Framework」である。各 Engine の実行タイミングを管理する仕組みが「Time Manager」であり、詳細については後述する。次に、生産設備の制御プログラムやシミュレーション用のプログラムを実行するのが、「仮想プログラミング環境 (Virtual Programming Engine)」である。実機 PLC の動作シミュレーションを行うのが「PLC Simulator」である。生産設備全体の 3D モデルの形状データを管理し、3D 表示機能やモデル同士の干渉チェックを実現するのが「3D Engine」である。そして、重力や物理シミュレーションの演算を行うのが「Physics Engine」である。3D Engine 上で、生産設備の要素であるロボット、周辺機器、I/O 機器、画像センサ及びワークを仮想化したモジュールがそれぞれ、「仮想ロボット (Virtual Robot)」、「仮想周辺機器 (Virtual Mechanics)」、「仮想 I/O (Virtual I/O)」、「仮想画像センサ (Virtual Vision Sensor)」、「仮想ワーク (Virtual Part)」である。今回、新たに「仮想カスタムメカニカル機構 (Virtual Custom Mechanics)」モジュールを追加した。なお、それに伴う I/F 追加などの変更を行った要素を青枠で示している。

図 5 で示されるカスタムメカニカル機構を含む仮想環境を実現するためにはシステムに含まれる全ての要素が同期して動作する必要がある。システムとして各機能が同期し

ていない場合、シミュレーション実行ごとに結果が変わる。そのため、コンベア上を流れるワークに対してカスタムメカニカル機構を用いてピックして別のコンベア上にブレースする場合など、正しくピックできないなどの問題が発生してしまう。また要素間の同期が高速に行われない場合、3D 表示画面が滑らかに表示されないという問題が発生する。

これらの問題を解決するためにはカスタムメカニカル機構とシステムを構成する既存モジュール間の高速な同期が必要となる。図 5 のシステムからカスタムメカニカル機構を含むシステムを抽出し、各モジュールの実行単位であるプロセスの関係を表したものを図 6 に示す。各プロセス間で高速な通信を実現するために Windows OS が提供している EventObject を用いて、プロセス間同期を実現している。これによりプロセス間を約 0.02ms で通信することが可能であり、カスタムメカニカル機構を含む場合でも、3D 表示画面上で滑らかに動作の様子を確認することができるようになった。

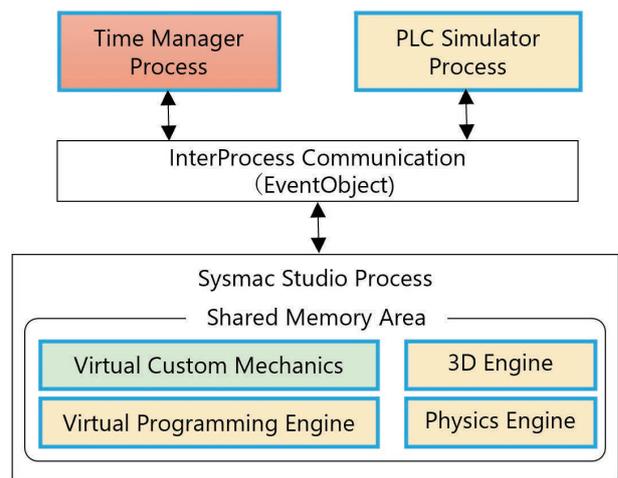


図 6 Sysmac Studio のプロセス構成

さらに、カスタムメカニカル機構のジョイントと、ワークの重力や、衝突の 3D シミュレーションをリアルタイム動作に近い形で行うためには Sysmac Studio プロセス内のメモリ空間における異なる要素間のデータ交換速度も課題となる。Sysmac Studio は C# を用いた仮想化用の汎用プログラム言語である Shape Script を用いて仮想空間上のワークを動作させている。Virtual Programming Engine にある Shape Script は PLC シミュレータの変数に対し、値の読み書きを行うため、Sysmac Studio プロセス内の他の要素に対し、高速通信する必要がある。Virtual Programming Engine と他の要素間は、OS が提供するメモリ空間の異なる要素間の高速通信の仕組みである .NetRemoting を活用して高速通信を実現している^{1,4)}。カスタムメカニカル機構においては Shape Script に機構を同期動作させるための専

用インターフェースを用意することで、これら通信の仕組みを利用し、リアルタイム動作に近い 3D シミュレーションを実現した。

今回、カスタムメカニカル機構を含む要素間の同期をシステム全体で管理するため、それぞれの要素のタイミングを管理する Time Manager¹⁾ の拡張を行った。システム全体のシミュレーション時間同期タイムチャートを図 7 に示す。

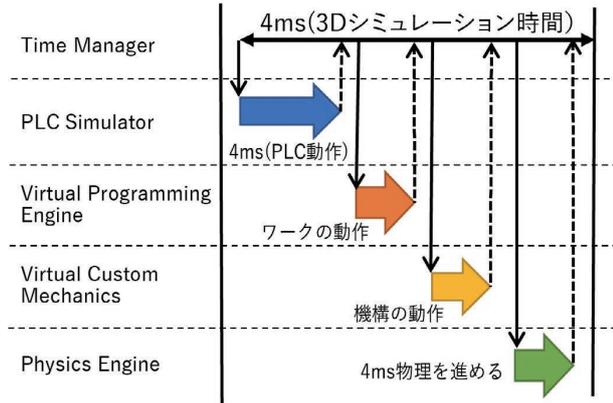


図 7 Time Manager を用いたシミュレーション時間同期タイムチャート

Virtual Custom Mechanics の「機構の動作」を Time Manager で管理することにより、カスタムメカニカル機構の動作と既存の仮想モジュールとの同期を実現した。

3.3 カスタムメカニカル機構を設定する上でのユーザビリティ課題の解決

本節ではカスタムメカニカル機構をユーザが簡単に設定するための課題とその解決手段について述べる。

3.3.1 ジョイントの位置を簡単に設定するためのスナップ機能の実現

先に述べている通り、カスタムメカニカル機構のジョイントの動作を実現するために、物理シミュレーションを用いる。物理シミュレーションを正確に行うためには、可動部品同士のジョイントの位置を正確に決める必要がある。ジョイントの位置は図 8 に示すようにマウス操作によって設定することが可能である。しかし、目標とする位置が画面上に表示されないことや、点の移動がユーザのマウス操作に依存するため、正確にジョイントの位置を設定することは困難である。

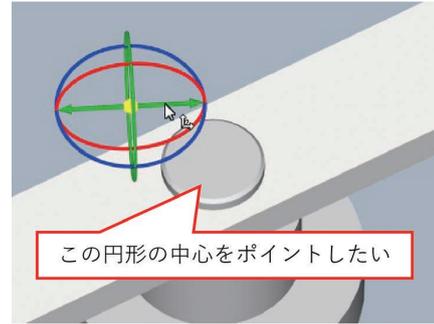
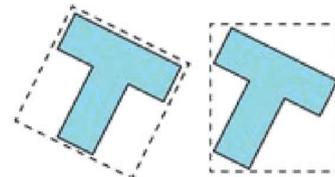


図 8 マウス操作によるジョイントの位置の設定

この問題を解決するため、スナップ機能を開発した。スナップ機能とは、特定の 3D オブジェクトを選択した後、マウスで別の 3D オブジェクト上にマウスカーソルを移動させると、マウスカーソル近くの辺の両端あるいは面の重心などの点を表示し、ユーザがその点をクリックすることで、対象をその点に移動させることができる機能である。スナップ機能における、移動先の候補点の表示を実現するためには、マウスカーソル上に最も近い面を検索する必要がある。しかし、このスナップ先の点の検索には大量の計算が必要であるため、ユーザのマウス操作に影響が出るなどの性能悪化につながる。そこで計算量を減らすために、3D オブジェクトを直方体で簡素化した。簡素化の方法には図 9 に示す OBB と AABB という手法があるが、今回は操作性に関わる箇所であるため、応答速度で有利な AABB を採用した。



OBB AABB

OBB: Oriented Bounding Box
AABB: Axis-Aligned Bounding Box

図 9 OBB と AABB の比較

AABB を適用して簡素化する手法を用い、マウスカーソルが示す箇所にあるオブジェクトの内、最前にあるオブジェクトを特定する。最前にあるオブジェクトの特定は、カメラとマウスカーソルを結ぶ光線を作成し、光線と交差している 3D オブジェクトのうちカメラに一番近いオブジェクトを取得することで実現した。さらに、特定した 3D オブジェクトを構成している内部の形状データについてもそれぞれを直方体で囲い、検索することで最前の面を特定する。特定した面から辺の両端あるいは面の重心の情報を取得し、画面上にスナップ先の点として表示する。

性能改善の更なる手段として、計算処理をバックグラウ

ンド処理で行う工夫をした。計算はマウスカーソルの存在している位置に対して常に実行されるが、マウスカーソルが移動すると、新たに計算をし直す。バックグラウンドで処理が実行されるため、早く動かすとスナップ先の点は表示されないが、マウスカーソルがスナップ先に近付き、ゆっくり動かされることで、スナップ先の点が表示されることになる。これはユーザのマウス操作方法に着目して設計を行った。具体的には目標座標付近に来るまでは早くマウスを動かし、目標位置付近ではゆっくり動かすという人間の操作方法に基づいて設計した。これらの技術により実現したスナップ機能を用いてジョイントの位置を設定する様子を図 10 に示す。ジョイントを選択してスナップモードに入った後にマウスカーソルを目標位置付近に移動させると、水色の点が表示される。ここでクリックすると、ジョイントが水色の位置に移動する。

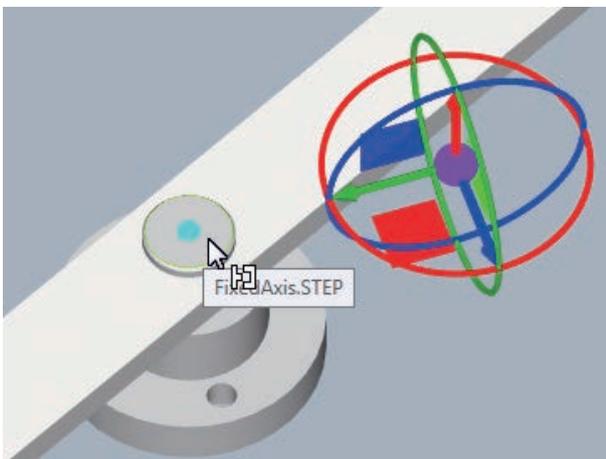


図 10 スナップ機能によるジョイントの位置の設定

このように、ユーザ操作を妨げることなく自然なスナップ機能を実現し、物理シミュレーションに必要なジョイント位置の正確な位置決めを実現した。

3.3.2 複数可動部品の動作定義を簡単にする動作設定機能の実現

カスタムメカニカル機構で実現したい重要な機構として電動チャックが挙げられる。電動チャックは複数の爪で物をつかむことを想定しているため複数の可動部品をセットにして動作させる必要がある。これを実現するために一般的な電動チャックには動作番号機能⁵⁾という特徴的な機能が存在する。この機能はあらかじめ複数可動部品の速度や位置などの動作をグルーピングし、そのグループ設定を1つの番号として設定することができる。例えば SMC 社のステップモーターコントローラは電動チャックを制御できるが、複数の可動部品の動作指示をまとめたステップデータ指示により運転制御している。カスタムメカニカル機構で複数の可動部品をステップデータ指示により動作させるためには、Shape Script で可動部品毎に動作を定義することが必要である。しかし、一般にカスタムメカニカル機構のステップデータは十数個定義され、その十数個もあるステップデータの管理をスクリプトで行うことは非常に複雑であり、また動作番号を指定する PLC 変数との紐づけのために、制御プログラムの変更も必要であることが、ユーザにとって障壁となる。

そこで、電動チャックの動作番号機能と同等である、複数の可動部品の動作設定をグルーピングして管理できる機能を実現した。この機能により動作設定に基づいた速度および位置の計算と、動作番号を指定する PLC 変数との紐づけを内部的に自動で行う。また、直感的にこれらの動作設定を実施できるように、図 11 に示すような表形式ベースの UI を用意した。



図 11 カスタムメカニカル機構の動作設定画面

動作設定においては、カスタムメカニカル機構の各可動部品に対して、直動・回転・直動回転のいずれかの動作方向と、それぞれの動作方向に対応する動作を予め最大 64 個まで設定することができる。表 2 にそれぞれの動作方向に対して設定する内容を示す。

表 2 動作方向と動作設定

動作方向	動作設定内容
直動	直動速度 [mm/s] と目標位置 [mm] を設定する。可動部品は目標位置 [mm] に到達するまで指定の直動速度 [mm/s] で直動することを表す。
回転	回転速度 [degree/s] と目標角度 [degree] を設定する。可動部品は目標角度 [degree] に到達するまで指定の回転速度 [degree/s] で回転することを表す。
直動回転	直動速度 [mm/s] と目標位置 [mm] と回転速度 [degree/s] と目標角度 [degree] を設定する。可動部品は目標位置 [mm] に到達するまで指定の直動速度 [mm/s] で直動した後、目標角度 [degree] に到達するまで指定の回転速度 [degree/s] で回転することを表す。

各可動部品には動作開始 I/O 変数、位置決め完了 I/O 変数、動作番号指定変数、現在値取得変数の割り当て設定が存在する。表 3 に各変数についての説明を示す。動作開始 I/O 変数が ON (True) である間、各可動部品は動作番号指定変数で指定された動作番号の速度で、動作番号の目標位置へ向かって移動する。その後、目標位置にて停止し、位置決め完了 I/O 変数が ON になる。可動部品の現在値は常に現在値取得変数に出力される。それぞれの変数には、コントローラ変数もしくはロボットの I/O シグナルを割り当てることができる。3.2 節で説明した通り、これらの変数の取得、書込はシステム全体で同期して動作するように管理されている。

表 3 可動部品に割り当てる変数

変数	説明
動作開始 I/O 変数	ON である間、指定された動作番号の設定に従い、可動部品を動作させる変数。
位置決め完了 I/O 変数	可動部品が目標位置に到達した際に ON になる変数。
動作番号指定変数	最大 64 個まで設定される動作の内、どの設定で可動部品が動作するかの番号を指定する変数。
現在値取得変数	可動部品の現在位置が常に書き込まれる変数。

図 12 に可動部品である爪が 2 つと、各可動部品に対する動作設定が 2 つずつ存在する場合の電動チャックの動作例を示す。動作番号 0 によりチャックを閉じ、動作番号 1 によりチャックを開く動作を定義している。ユーザは動作設定画面で必要な項目を入力するだけで、これらの動作を定義できる。

(A) はチャックが閉じた初期状態を表す。(B) に示す通り、動作番号指定変数の値を 1 に設定し、動作開始 I/O 変数の値を True に変更すると、可動部品が動作番号 1 の動作設定に従って直動速度 10mm/s で目標位置 12mm に向かって動作する。動作番号 1 の目標位置に可動部品が到達すると、(C) の通り、動作開始 I/O 変数の値が false となり、位置決め完了 I/O 変数の値が True となる。これによりチャックが開いた状態となる。次に (D) の通り、動作番号指定変数の値を 0 に設定し、動作開始 I/O 変数の値を True に変更すると、可動部品が動作番号 0 の設定の通り、直動速度 10mm/s で目標位置 0mm に向かって動作する。動作番号 0 の目標位置に可動部品が到達すると、(E) の通り、位置決め完了の値が True となり、チャックが閉じた状態になる。

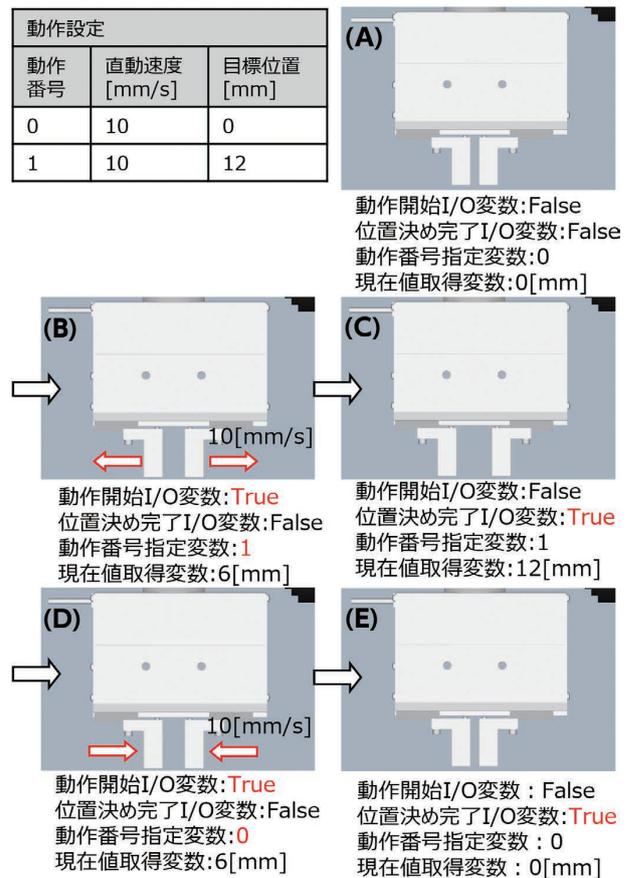


図 12 動作設定による動作例

本項で述べた通り、複数の可動部品の動作をグルーピングして動作させる動作設定機能を実現した。これにより複雑なカスタムメカニカル機構の動作定義をユーザが簡単に設定できるようになった。

4. カスタムメカニカル機構の有効性の検証

カスタムメカニカル機構により実現される仮想化技術の有効性検証を目的に、本稿で記載した技術要素をすべて含む装置としてロボットとカスタムメカニカル機構によるドアスイッチ取り付けシステムの構築を行った。

ドアスイッチ取り付けシステムでは、垂直多関節ロボット Viper650 がドアスイッチのボディ、ヘッドを順に運搬することで、ドアスイッチのボディにヘッドを連結させる。運搬先では、ボディはヘッドの取り付けを可能にするために、クランプ機構により押さえ付けられ、定位置に固定される。システムの内、Viper650 のエンドエフェクタに取り付けられるロータリテーブルと電動チャック、ボディを押さえるクランプ機構がカスタムメカニカル機構により実現される。図 13 に検証に用いたシステム構成、表 4 にそれぞれの機構についての説明を示す。

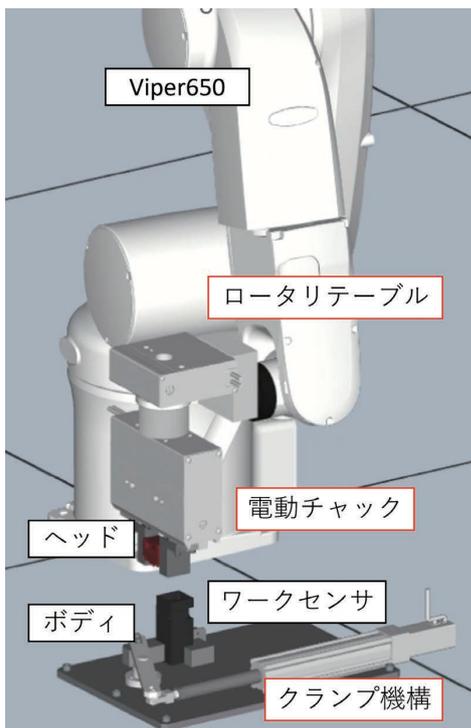


図 13 ドアスイッチ取り付けシステムの構成 (赤枠：カスタムメカニカル機構)

表 4 ドアスイッチ取り付けシステムの機構

項目	説明	
Viper650	PLC により制御される垂直多関節ロボット。	
ボディ (ドアスイッチ)	運搬されるワーク。上部にヘッドが取り付けられる。	
ヘッド (ドアスイッチ)	運搬されるワーク。ボディの上部に取り付けられる。	
ワークセンサ	ワークが固定位置に押さえつけられていることを検知するセンサ。	
カスタムメカニカル機構	ロータリテーブル	回転して手先の位置調整を行うカスタムメカニカル機構。
	電動チャック	ワークを把持するカスタムメカニカル機構。
	クランプ機構	ドアスイッチのヘッドを取り付けるために、ボディを押さえつけて固定するカスタムメカニカル機構。

検証はドアスイッチ取り付けシステムにおける、メカ設計から、実機制作、制御設計、全体検証を含む制御デバッグまでを行い、これらの各工程に要した工数を記録し、仮想化技術未使用時と使用時において工数を比較した。比較結果を図 14 に示す。

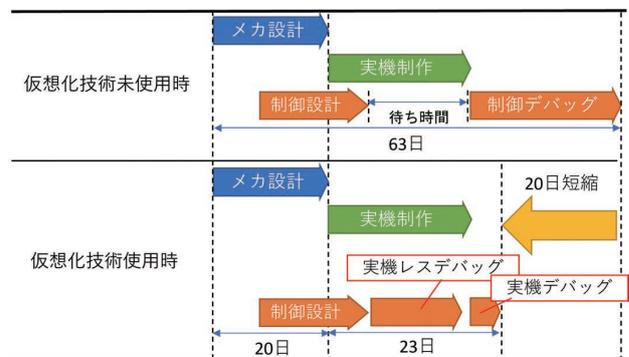


図 14 仮想化技術未使用時と使用時の工数比較結果

図 14 に示した通り、仮想化技術を使用しなかった場合、立ち上げには 63 日を要した。それに対し、仮想化技術を使用した場合は 43 日となり、仮想化技術を使用することで立ち上げ工数を約 32% 削減することができた。

仮想化技術を使用しない場合、メカ設計を行ってから実機制作が完了し、手元に実機が揃うまで制御デバッグを始められないため待ち時間が生じていた。それに対し、仮想化技術を使用した場合は、実機制作の完了を待たずして並行して制御デバッグを進めることができるため、システム全体の製作期間を短縮することができた。

また、制御デバッグにおけるティーチング工程では、実機を使用する場合、干渉による機器の破損を避けるためロボットを低速で動かす必要があり工数を要する。それに対し、仮想化技術を使用すると仮想環境上で仮想ロボットを高速に動かすことができるため、ティーチング時間を短縮できた。さらに、最終的な実機での調整はほぼ確認と微調整のみで済むため、実機での作業時間が削減され、結果として制御デバッグ全体の工数も削減できた。

また、仮想化技術を使用して作成・実機レスデバッグが行われたプログラムは、実機にそのまま使用することができるため、一般的な仮想化シミュレーションのように実機立ち上げを行う際の追加プログラミングは必要なく、実機での確認作業開始までの期間を短縮できる。

5. むすび

本稿では、従来技術でサポート外の機構を一つでも含む装置がある場合に3Dシミュレーションによる設備の事前検証ができないという問題を解決するため、物理シミュレーション技術を導入したカスタムメカニカル機構の実現手段や活用について述べた。

可動部品同士を結合する各種ジョイントや、設備全体を対象とした3Dシミュレーションのためのカスタムメカニカル機構動作と既存の仮想モジュールとの同期を実現した。また、ジョイント位置のスナップ機能や動作設定機能など、カスタムメカニカル機構を使用するためのユーザビリティ課題を解決した。

ドアスイッチシステムの製作に関して本技術を活用すると工数が32%削減されることが確認でき、本技術の有効性を検証することができた。これにより、生産設備の短期立ち上げが可能となり、製品のタイムリーな市場投入に寄与することができる。

今後は、仮想化技術をより進化させ、カスタムメカニカル機構で実現できていないギアやカムなど、別の固有のパラメータ設定が必要である機構も仮想化できるようにする。あるいは柔軟物の仮想化を行うために、Sysmac Studioで未導入である有限要素法の技術を用いるなど貪欲に技術を取り入れ、より幅広い設備を仮想化できるようにしていく。

参考文献

- 1) 島川はる奈, 岩村慎太郎. FA統合開発環境における生産設備の仮想化技術. OMRON TECHNICS. 2021, Vol.53, No.1, p.8-16.
- 2) PhysX Joints. "NVIDIA PhysX SDK 3.4.0 Documentation". <https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Joints.html>, (参照 2023-01-13).
- 3) 熊谷英樹. 必携「からくり設計」メカニズム定石集一ゼロからはじめる簡易自動化一. 日刊工業新聞社, 2017.

- 4) Microsoft. "NET Remoting Framework の概要". [https://learn.microsoft.com/ja-jp/previous-versions/msdn/architecture-center/cc440094\(v=vs.71\)](https://learn.microsoft.com/ja-jp/previous-versions/msdn/architecture-center/cc440094(v=vs.71)), (参照 2023-01-13).
- 5) SMC. "コントローラ (ステップデータ入力タイプ) JXC51/61 Series". <https://ca01.smcworld.com/catalog/Electric/mpv/s100-136-JXC51-61/data/s100-136-JXC51-61.pdf>, (参照 2023-02-06).

執筆者紹介



長谷川 直人 HASEGAWA Naoto
 インダストリアルオートメーションビジネス
 カンパニー
 商品事業本部 コントローラ事業部
 ソフトウェア開発部
 専門：ソフトウェア工学



岩村 慎太郎 IWAMURA Shintaro
 インダストリアルオートメーションビジネス
 カンパニー
 商品事業本部 コントローラ事業部
 ソフトウェア開発部
 専門：ソフトウェア工学
 所属学会：日本ロボット学会



島川 はる奈 SHIMAKAWA Haruna
 インダストリアルオートメーションビジネス
 カンパニー
 商品事業本部 コントローラ事業部
 ソフトウェア開発部
 専門：ソフトウェア工学



白田 聖人 SHIRATA Seito
 インダストリアルオートメーションビジネス
 カンパニー
 商品事業本部 コントローラ事業部
 ソフトウェア開発部
 専門：ソフトウェア工学

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。