

シミュレーション技術を応用したシステム開発手法

Technique for Factory Automation System Development Using Simulation Technology

長谷川 文 夫 技術開発本部総合開発センター制御技術開発部 課長
 志 田 倫 教 技術開発本部総合開発センター制御技術開発部
 袖子田 志 保 技術開発本部総合開発センター制御技術開発部
 高 濱 雄 一 原子力セクター原子力プラント技術部

産業機械や物流システムの制御システムの開発では、開発のコスト低減に伴い、開発期間の短縮化が進んできている。このような状況のなか、短期間で高い信頼性をもったソフトウェアを開発するためには、制御対象の装置がまだできていない状態である設計開発段階でのソフトウェアの検証が重要となる。ソフトウェア検証は、従来はスイッチやランプなどを使用して、簡易的に実施されてきたが、この方法では、実際の装置の動作タイミングに応じたソフトウェアの検証が難しい。そこで、シミュレーション技術を応用して、ソフトウェア検証を実施する手法を構築したので、内容を報告する。

Reduction of development costs has led to a reduction in development time for control systems for industrial machinery and logistical systems. In such a situation, it is important to have ready-made techniques for verifying software developed in the design phase in order to develop highly reliable software within a short period. We have carried out software verification using simple methods such as lamp testing and bottom up testing. However, using these simple methods it is difficult to verify developed software based on actual operation of the machine. In view of this, we devised a technique that uses simulation technology for developing Factory Automation System, which we would like to present here.

1. 緒 言

産業機械や物流システムの制御システムの開発では、開発コスト低減のために、開発期間の短縮が求められている。このような状況のなか、短期間で高い信頼性をもったソフトウェアを開発するためには、装置がまだできていない設計開発段階でのソフトウェアの検証が重要となる。

設計開発段階でのソフトウェア検証は、従来はスイッチやランプ等を使用して、簡易的に実施されてきたが、この方法では、実際の装置の動作タイミングに応じたソフトウェアの検証が難しい。そこで、シミュレーション技術を応用して、ソフトウェア検証を実施する手法を構築したので、内容を説明する。

2. 制御ソフトウェア検証技術

2.1 装置をモデル化したソフトウェア開発手法

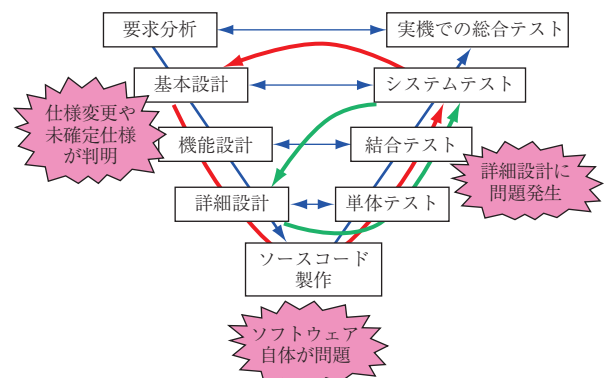
ソフトウェアの開発は要求分析の結果を基に、基本、詳細設計段階で制御仕様や制御アルゴリズムを検討し、基本設計書、詳細設計書にまとめていく。これらを基にプログラム設計を実施して、ソースコードを製作していく。

開発段階で発生する問題はさまざまであるが、装置実機

での検証テスト段階からの大幅な後戻りが必要になる要因は主に以下の三つである。第1図に概念を示す。

- (1) 仕様変更や未確定の仕様が判明した場合
- (2) 制御仕様や制御アルゴリズムなど詳細設計に問題があった場合
- (3) ソフトウェア自体に問題があった場合

ソフトウェア開発の最終段階である実機での検証段階で上記の問題が発生した場合は、大幅な後戻りが必要になり、開発プロジェクトにおいて大きな問題となることがしばしばある。また、ソフトウェア自体に問題があった場



第1図 ソフトウェア開発段階での後戻り要因
 Fig.1 Relapse factors in software development phase

合、最悪の場合は装置の破損などが発生し、装置全体の開発が大幅に遅れることになる。

このような大幅な後戻りをなくすためには、実機試験の前に、開発の上流段階である設計フェーズで制御仕様や制御アルゴリズムの検証、装置の運用を踏まえた形での仕様の確認およびソフトウェア自体の検証をする手法が必要である。

現在のソフトウェア開発においては、工場の生産設備や搬送装置などの制御対象をモデル化して、ソフトウェアの検証をする手法が用いられるようになってきている。この手法では、開発ステップごとに検証結果に基づいてフィードバックを掛けるため、問題を後工程にもち越さず、実機試験での後戻りが大幅に低減でき、品質の高い制御ソフトウェアが短時間で開発できるようになってきている。検証手法は大きく HILS (Hardware In the Loop Simulation) と SILS (Software In the Loop Simulation) に分類され、制御対象である実機および実機を制御する制御装置をモデル化して、実機環境を想定したシミュレーションを行い、ソフトウェアの検証を行う。

2.2 HILS と SILS

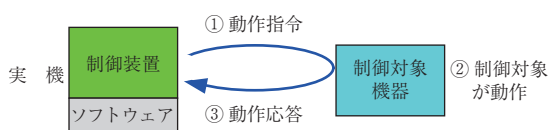
制御対象をモデル化してシミュレーションするとは、以下の一連のフィードバックループをモデル化し、シミュレーションすることである。

- (1) 制御対象の装置のアクチュエータが、制御装置からの動作指令を制御信号として受け取る。
- (2) 制御対象の装置が動作指令に従い、動作する。
- (3) 制御対象の装置からのセンサ信号を制御装置が受け取って、次の制御信号を出力する。

第 2 図に装置動作の一連の流れの概念を示す。

HILS は、この一連のフィードバックループのなかに実機ハードウェアを含めるシミュレーションである。装置をモデル化する場合に、一部、ハードウェアを使用することによって、応答性を高め、装置の動特性の検証など応答にリアルタイム性が求められる場合には、HILS での検証が必要になる。

一方、SILS はフィードバックループ全体をシミュレーションで実行する形式である。ソフトウェアで装置をモデル



第 2 図 装置動作の一連の流れの概念
Fig. 2 Conceptual diagram of machinery control flow

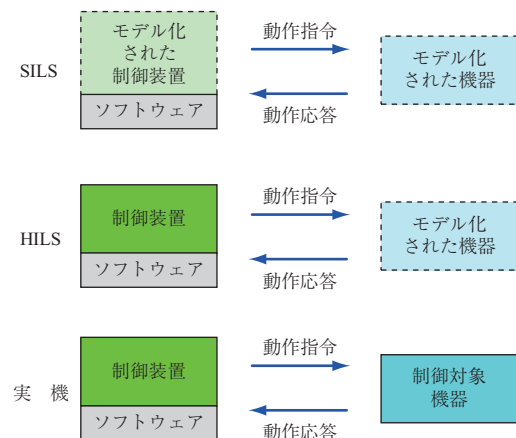
化するため、HILS に比較してモデルからの応答にリアルタイム性はないが、モデル化をソフトウェアで行うため、短時間で簡単に装置のモデル化が可能である。このため、設計の上流段階である制御仕様や制御アルゴリズムの検討段階でも適用が可能である。第 3 図に HILS と SILS でのモデル化範囲を示す。破線線部がモデル化された部分である。

2.3 FA システムのソフトウェア検証

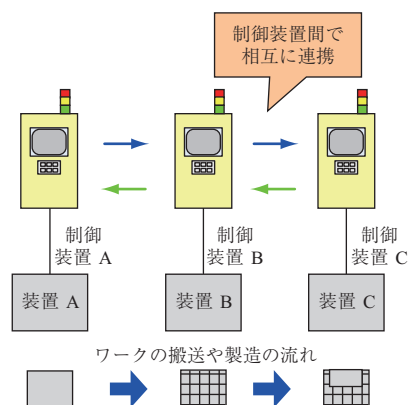
Factory Automation (以下、FA と呼ぶ) システムでは、多くの場合、複数の装置が連携してワークの搬送や部品の組み付けなどを実行する。

複数の装置間の連携は、① 個々の制御装置間が互いに連携して行う場合 ② 上位の Programmable Logic Controller (以下、PLC と呼ぶ) や Personal Computer (以下、PC と呼ぶ) が統括してライン全体の動きを制御する場合、の大きく二つの場合に分類される。第 4 図および第 5 図に概念を示す。

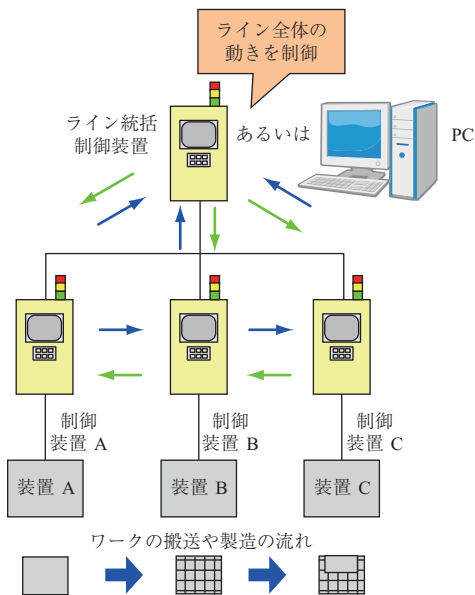
これら制御システムの形態に応じたソフトウェア検証手法の開発状況を説明する。



第 3 図 HILS と SILS
Fig. 3 HILS and SILS



第 4 図 制御装置間で相互に連携する場合
Fig. 4 Controller B working with controller A and controller C



第 5 図 ライン全体の動きを制御する場合
Fig. 5 PC or PLC controlling all lines

ソフトウェアの検証手法として、複数の制御装置が互いに連携して行う場合はシーケンスシミュレータを用いた手法について、上位の PC や PLC がライン全体を統括して制御する場合は物流シミュレータを用いた手法について説明する。

3. シーケンスシミュレータの開発

3.1 シーケンスシミュレータを用いた手法

近年、3D-CAD を用いて製品の外見、内部構成などを比較、検討するためのアプリケーションが普及し始めている。また、これらのアプリケーションで製作されたデジタルモックアップに製品モデルや人体モデルの挙動をシミュレートして確認する機能も備わっており、これらの機能を応用してソフトウェアの検証を実施するアプリケーションが各社から提供されてきている。

各社から提供されるアプリケーションは、機能面は充実しているが高価であり、使い方を習得するのに時間が掛かるという問題がある。また、装置の開発段階で使用する場合は、既存のアプリケーションでは対応ができない場合があるため、開発時に新規に開発する装置の機能追加が容易に行える環境が必要である。各社から提供されるアプリケーションのカスタマイズは各提供ベンダにお願いする必要があり、費用面で問題がある。

そこで、ソフトウェア開発技術者が短期間で簡単に装置のモデル化を実現でき、ソフトウェア検証を容易に実施できるシーケンスシミュレータを自社開発した。

シーケンスシミュレータは PLC で制御するシーケンスロジックの検証を対象とする。ソフトウェア検証対象とモデル化の範囲を第 6 図に示す。

3.2 対象規模と開発コンセプト

3.2.1 対象規模

シーケンスシミュレータが対象とするシステム規模は入力点数 256 点、出力点数 256 点、サーボ軸が 3 軸、アクチュエータ数が 50 個程度の中規模なシステムである。

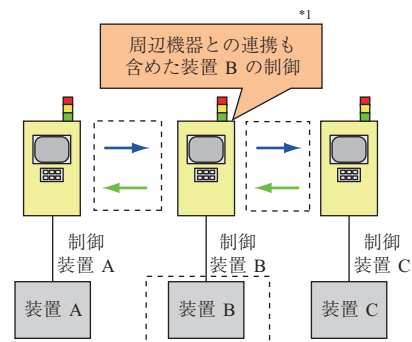
3.2.2 開発コンセプト

シーケンスシミュレータの開発コンセプトは以下である。

- (1) 容易にモデル作成・検証テストができること
1 週間程度でモデル作成、およびテストが可能であること。
- (2) 汎用的であること
インデント製品、そのほかの社内製品の制御ソフトの開発・製作段階で使用可能であり、さまざまなメーカーの PLC に対応可能であること。
- (3) 過渡状態・異常状態の模擬が可能であること
センサのチャタリングやアクチュエータ動作の過渡状態の模擬および異常状態の模擬が可能であること。

3.3 装置のモデル化

シーケンスシミュレータでの装置のモデル化は基本的には、設計段階で作成されるドキュメントを参照して実施する。装置の動作フローを記述した「フローチャート」、装置間の信号のやりとりを記述した「タイミングチャート」、装置と制御装置間の信号のやりとりを示した「入出力リスト」、装置の各ユニットの動作の干渉を記述する「インタロックリスト」を基に、Excel で設定ファイルを作成する。これらの設定ファイルをシーケンスシミュレー



(注) *1 :ソフトウェア検証対象
[] :シーケンスシミュレータでモデル化された部分

第 6 図 モデル化、ソフトウェア検証範囲
Fig. 6 Scope of modeling and software verification

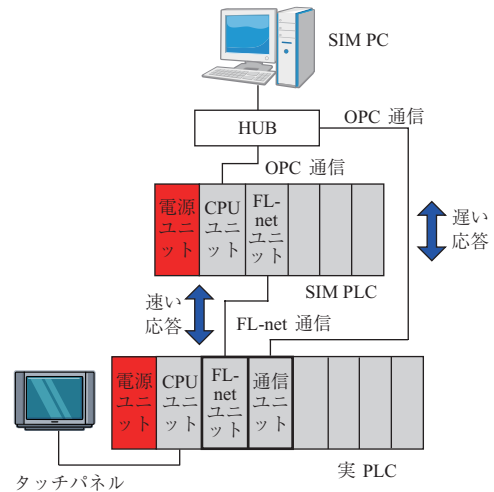
タが読み込み、装置をモデル化する。装置のモデル化の流れを第 7 図、設定ファイルの一覧を第 1 表に示す。

3.4 システム構成

シーケンスシミュレータは Simulation PC（以下、SIM PC と呼ぶ）、Simulation PLC（以下、SIM PLC と呼ぶ）、実機 PLC（以下、実 PLC と呼ぶ）から構成される。システム構成を第 8 図に示す。

3.4.1 SIM PC

SIM PC では、シミュレータの設定や、自動運転模擬などの比較的遅い信号の応答を模擬する。SIM PLC、実 PLC とは OPC サーバで接続する。OPC とは OLE for Process Control の略で、OPC サーバを使用することで、さまざまな型式、メーカーの PLC と同じプロトコルで通信が可能である。



第 8 図 システム構成

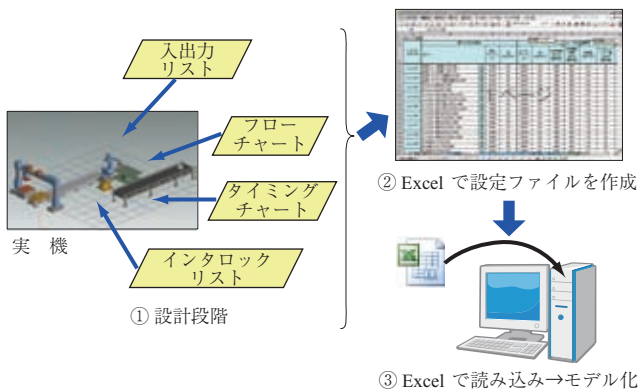
Fig. 8 System architecture diagram of sequence simulator

3.4.2 SIM PLC

SIM PLC では、主に単体アクチュエータの動作信号などの比較的速い信号の応答を模擬する。正常な応答だけでなく、センサのチャタリングや、過渡応答など異常な応答をすることも可能である。実 PLC と FL-net 通信で接続する。FL-net は ① FA 機器 ② PLC ③ パソコン、などを相互接続するオープンなネットワークの規格であり、この通信プロトコルを採用することでさまざまなメーカーの PLC と同じプロトコルで通信が可能である。

3.4.3 実 PLC

実 PLC は I/O ユニット、位置決めユニットなどが実装されているが、シミュレーションを実施する場合は、基本



第 7 図 装置のモデル化の流れ

Fig. 7 Modeling flow

第 1 表 設定ファイル一覧

Table 1 Settings file list

大分類	中分類	シート名称	内 容
基本設定	初期設定	初期値	単体アクチュエータ以外のデバイス値の初期設定を行う。
		シリンダ	シリンダのデバイス割付、動作設定を行う。
		サーボ	サーボモータのデバイス割付、動作設定を行う。
		インバータ	インバータのデバイス割付、動作設定を行う。
		アナログ	アナログセンサなど連続的に変化するデータのデバイス割付、動作設定を行う。
	操作盤	操作盤	操作盤画面における、デバイス割付などの設定を行う。
	システム処理オペレータ操作	条件応答	自動運転時以外も、常時応答を返すセンサなどの条件を設定する。
その他	一括登録	一括で動作させたいデバイスをひとまとめに登録する。	
自動運転	自動シナリオ	周辺装置 I/F	周辺装置 I/F の自動運転シナリオを設定する。
		センサ動作	アクチュエータ動作に連動しないセンサの動作条件を設定する。(例：ワーク検出センサなど)
ソフトウェア検証	表示	ワーク表	ワークの在籍条件を設定する。
	動作ロジックチェック	動作順序	動作順序チェックにおける、動作順序を設定する。
		機械干渉	機械干渉チェックの条件を設定する。
ログ	ログ	ログをとるデバイスを設定する(最大 16 個)。	

(注) I/F : Inter Face

的にこれらのユニットは実装せず、SIM PC、SIM PLC との通信ユニットのみの実装とする。ただし、シミュレーションモードがない位置決めユニットを使用する場合は、位置決めユニットのみ実装する。

実 PLC 内のソフトウェアは実際の制御するものと同じものであり、シミュレーション用にコードを変更するなどの必要はない。実 PLC には、実運用時に使用するタッチパネルも接続し、タッチパネルのソフトウェアの検証も実施する。

3.5 ソフトウェア検証までの流れ

ここでは、装置のモデル化からソフトウェアの検証までの流れに沿ってシーケンスシミュレータの機能を説明する。

3.5.1 制御対象の装置、操作ボックスのモデル化

(1) 各アクチュエータの動作

シリンダ、バルブ、モータ、インバータ、アクチュエータ動作に伴うアナログセンサなど、動作指令に対して、動作とそれに伴う応答信号を模擬する。

(2) 操作ボックスやセンサ

装置内にある操作ボックス、シグナルタワー、安全センサなどの ON/OFF を模擬する。

(3) 表示

装置の動作を簡易的に表示する。

3.5.2 初期状態や自動運転シナリオの設定

装置の自動運転時の各機能を模擬することによって、自動運転時のシーケンス制御ロジックの検証を実施する。

(1) 装置の初期状態

設定ファイルによって、自動運転開始時の装置の状態、周辺装置の状態、ワークの状態などの初期状態を設定する。

(2) 自動運転シナリオ

自動運転時のワークの動きや周辺機器とのやりとりを設定ファイルに記述して、自動運転のシナリオを作成する。

3.5.3 ソフトウェアの検証

(1) 自動運転時のシーケンス制御ロジックの検証

自動運転シナリオに従い、実 PLC 内のシーケンス制御ロジックによって、モデル化された装置の制御を実行する。装置の動作状況を SIM PC の画面や実 PLC のタッチパネルで確認し、シーケンス制御ロジックに問題がないか検証を実施していく。検証に当たって、装置の動作状況の表示や検証したい信号のログを参照して、実施していく。

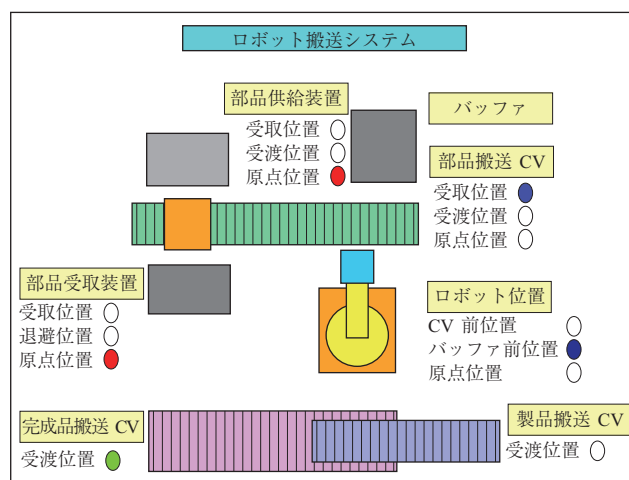
装置の動作状況の表示例を第 9 図に、ログ表示画面例を第 10 図に示す。

(2) 装置の干渉チェック

装置の干渉条件を、あらかじめ、設定ファイルで設定しておき、装置の干渉が発生した場合は異常箇所を表示する。インタロック条件が正しく設定されているかを検証することが可能である。

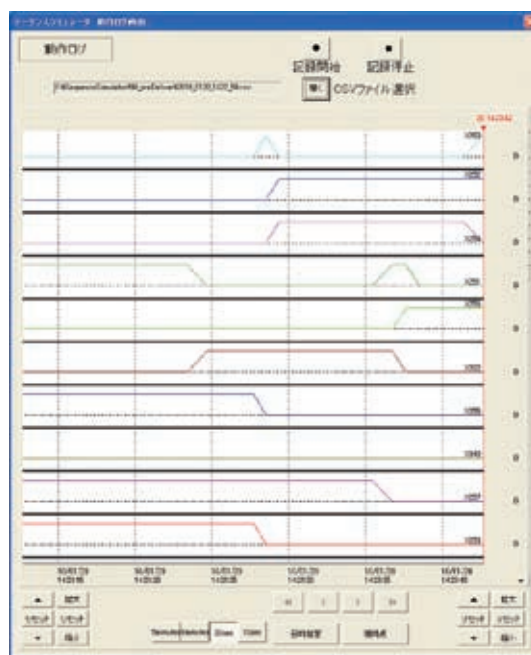
(3) 異常状態でのソフトウェアの検証

自動運転中のワークの異常、安全装置作動異常、各周辺機器とのインタフェースの異常を自動運転のシナリオファイルにあらかじめ設定しておくことによって、自動運転中に発生する異常を模擬し、異常



第 9 図 装置の動作状況の表示例

Fig. 9 Example of display showing machinery movement



第 10 図 信号のログ表示例

Fig. 10 Example of display showing signal log

時の復旧ロジックに問題がないかを検証することが可能である。

3.6 シーケンスシミュレータの仕様

シーケンスシミュレータでモデル化可能な装置の規模やモデルの応答時間、各装置間の通信仕様を第2表に示す。

3.7 シーケンスシミュレータの評価

実際の FA システムで動作中のソフトウェアを使用して、シーケンスシミュレータの機能、性能の評価を実施した。

今回評価に使用した制御システムは、ロボット1台とコンベヤ2台および部品供給装置から構成されており、製品をバッファに搬送する装置であり、第9図で表示例として示したロボット搬送システムである。システムの規模を第3表に示す。

このシステム規模では、SIM PC - 実 PLC 間は数百 ms 程度、SIM PLC - 実 PLC 間は数十 ms 程度で応答が可能であり、実 PLC のソフトウェアの検証をすることができた。また、モデルの製作期間は今回のシステム規模では4日程度であった。

シーケンスシミュレータを開発したことで、実際の装置の動作タイミングに応じたソフトウェアの検証ができる環境が整った。今後は、開発段階での制御ロジックの検証や制御仕様の検討時、あるいはソフトウェアのデバッグ時に使用し、開発期間の短縮とソフトウェアの信頼性の向上を図っていく。

4. 上位システムシミュレータの開発

4.1 物流シミュレータを用いた手法

上位の PC や PLC が全体を統括し、制御する場合は、シーケンスシミュレータのように、装置内の個々のアクチュエータやセンサの応答を模擬する必要はなく、各装置への動作指令に対する各装置の動作状況を応答することが

第3表 システム規模
Table 3 System scale

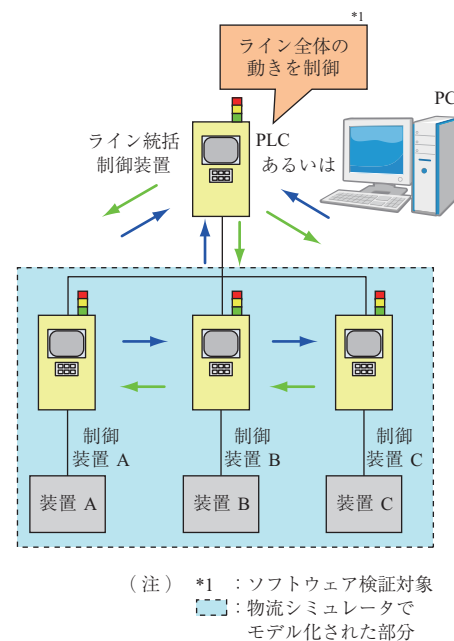
項目	単位	仕様
入力点数	点	430
出力点数	点	250
サーボ軸	軸	2
シリンダ数	本	40
ラダーステップ	ステップ	21 000

できれば、上位 PC や PLC の制御ロジックの検証作業は可能となる。

そこで、これらの複数の装置の動作を物流シミュレータでモデル化し、モデルを仮想装置として、上位 PC や PLC の制御ロジックの検証をする手法を開発した。物流シミュレータを用いた手法の概念を第11図に示す。

4.2 システム構成

ライン全体の統括制御は PLC、または PC で実施され



第11図 物流シミュレータを用いた手法の概念

Fig. 11 Conceptual diagram of method using logistical simulator

第2表 シーケンスシミュレータの主な仕様
Table 2 Specifications of sequence simulator

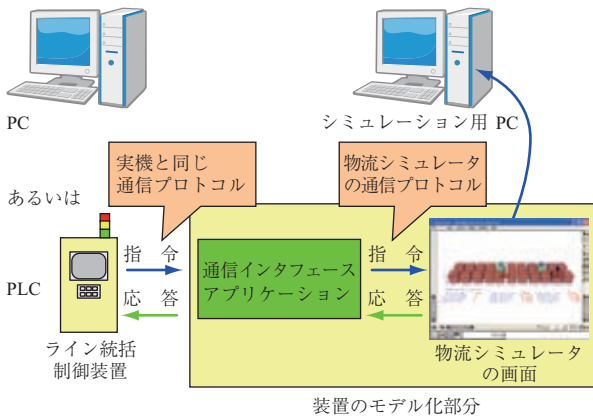
大項目	小項目	単位	仕様	備考
モデル化	シリンダ	本	100	Excel ファイルによって設定
	サーボ軸	軸	10	
	インバータ	個	10	
	アナログ応答	個	20	
	入出力点数		制限なし	
応答時間	SIM PC 部	ms	500	システム規模によって確定
	SIM PLC 部	ms	50	
通信	SIM PC - 実 PLC 間		OPC 通信	Ethernet ケーブルで接続
	SIM PLC - 実 PLC 間		FL-net 通信	Ethernet ケーブルで接続

る。それぞれ、PLC、または、PC から出力される指令を物流シミュレータで読み込める形に変換する必要がある。また、物流シミュレータ内の動作に対する応答信号を PLC または PC で読み込める状態に変換して、応答する必要がある。PC、PLC どちらの場合でも、これらの通信を仲介するインタフェースアプリケーションが必要となる。システム構成図を第 12 図に示す。

4.3 適用例のシステム構成

今回は、同じレーン内に 2 台のクレーンが設置された場合のクレーンの制御ロジックの検証に適用した。検証時のシステム構成を第 13 図に示す。

クレーンの制御システムは 2 台のクレーンの動作を制御する PC (図中の STK-PC を示す) とクレーン地上盤、クレーン機上盤で構成される。



第 12 図 システム構成

Fig. 12 System architecture diagram of method using logistical simulator

クレーン地上盤は STK-PC とクレーン機上盤のインタフェースの役割をもち、クレーン機上盤はクレーン本体の制御を行う。

今回は、このなかのクレーン機上盤とクレーン本体の部分を物流シミュレータでモデル化して、クレーン地上盤と接続することによって、STK-PC 内で動作する 2 台のクレーンの制御ロジックの検証を行った。

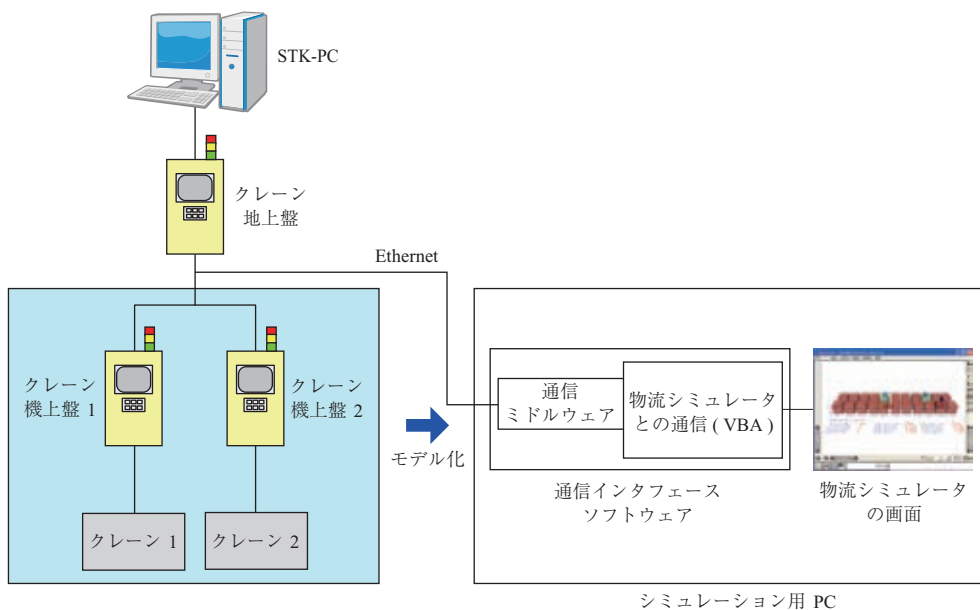
4.4 物流シミュレータを用いた手法の評価

シミュレーションモデルを仮想装置として使用する場合、モデルの応答時間が実際の装置の応答時間と同等程度とすることができるか否かが課題となる。

装置をモデル化する場合、実際の装置をどこまでモデル化するか、つまり、モデル化の詳細度の検討が必要であり、検証する目的に応じて決定していく。モデル化の詳細度を上げるとより実際の装置に近い状態でのソフトウェアの検証が可能となるが、通信する内容も多くなり、結果としてモデルの応答時間が長くなってしまいう問題が生じる。

今回のソフトウェア検証の目的は、「2 台のクレーンを衝突させることなく制御するロジックの検証」が目的であるので、クレーンの走行動作を中心にモデル化し、ワークを棚に搬送するクレーン上の昇降ユニットやフォークユニットの動きは簡易的にモデル化した。

上記の詳細度でシミュレーションモデルを製作することで、シミュレーションモデル内のクレーンがクレーン地上盤からの指令に応じて、ワークを棚から棚へ搬送する動き



第 13 図 適用例のシステム構成

Fig. 13 System architecture diagram of actual application

をシミュレーションすることができた。クレーン地上盤の指令に対するシミュレーションモデル応答時間は数秒であり、応答時間としては、問題ないレベルであった。

今回の開発によって、実際の装置を使用することなく、ソフトウェア開発の段階で装置の動きに応じた制御ロジックの検証をすることができる環境が整った。

今後は、制御ロジックやソフトウェア自体の検証だけでなく、システム開発初期のシステム仕様検討や実運用を想定しての搬送能力の検証などにも応用していく考えである。また、ほかの装置への展開も図っていく。

参 考 文 献

- (1) 夏目明典, 青井文男, 横浜浩二, 筒井健司, 上藤陽一, 中本正彦: 信頼性の高い製品を支える制御ソフトウェア検証技術 三菱重工技報 第43巻 第2号 2006年 pp.20 - 24
- (2) 細川健司, 吉田光男, 松尾智裕, 鷹取 剛, 高橋淳二: 制御アプリケーション仕様開発への SILS の適用 富士通テン技報 第26巻 第2号 2008年12月 pp.16 - 22