3次元CADデータを用いた制御シミュレーションシステムの開発

Development of control simulation system in which 3-D CAD data is used

新井　健睦　*Kenboku Arai*
岩﨑　孝夫　*Takao Iwasaki*
後藤　　務　*Tsutomu Goto*
佐藤　裕一　*Yuichi Sato*
橋間　正芳　*Masayoshi Hashima*
千田　陽介　*Yosuke Senta*

要　　旨

当社では、1997年からCDチェンジャなどの車載用デッキメカニズムの開発に3次元CAD(Computer Aided Design)を導入している。以来、3次元による形状確認やアセンブリ状態での部品相互の静的な干渉チェックをはじめ、CAE(Computer Aided Engineering)による応力解析などへの3次元データ活用により、部品レベルでの最適化設計に効果を上げている。しかしながら、メカの動的な機能に対する検証やレビューには、3次元データの活用が十分ではなかった。

今回、3次元CADデータを用いてパソコン上に実物そっくりの仮想モデルを構築し、このモデルを実際のメカと同じように制御できるシミュレーションシステムを開発した。

本システムを用いることで、"メカ制御" という観点でのメカ構造の動的検証を設計初期段階から効果的に進めることができ、さらに制御ソフトウェアを実機レスで評価できる。

本稿では、開発の背景、本シミュレーションシステムの機能と活用効果について紹介する。

Abstract

Since 1997, we have adopted the usage of 3-D CAD (Computer Aided Design) for the development of vehicle-mounted deck mechanisms such as CD changers. Thereafter we have achieved considerable effects in optimizing the design especially for component parts, by taking advantage of 3-D data for the stress analysis based on CAE (Computer Aided Engineering), the verification of shapes, and the check of static interference between assembled parts. However, 3-D CAD data has not been fully utilized for the verification and review of the dynamic functions of the deck mechanisms.

We thus developed a simulation system in which a virtual model created on a PC by using 3-D CAD data, which can be controlled in the same way as the real deck mechanism.

Making use of this system has made it possible to carry out dynamic verification for the structure of the mechanism from the standpoint of "control of mechanism" effectively even at the early step of designing and evaluating the control software without the real mechanism.

This paper outlines the background of the development, functions of the simulation system, and effectiveness in its utilization.

１．はじめに

システムの多機能化と小型・軽量化に対応するため、構成部品の高精度化と高密度化が求められている。当社が開発しているCDチェンジャをはじめとする車載用デッキメカニズムについても同様で、限られた製品スペースでの機能実現のため、メカ部品の高密度化が必要となっており、その構成も複雑化している。

このような設計対象の複雑化に対応するとともに、開発期間の短縮、コスト削減と設計品質の向上をねらい、当社では1997年からデッキメカニズムの開発に3次元CAD「Pro/ENGINEER」*1を導入している。

以来、3次元モデルによる形状確認やアセンブリ状態での部品相互の静的な干渉チェックをはじめ、CAEによる応力解析、振動解析、樹脂流動解析など、3次元データの利用拡大を図り、主に構成要素部品レベルでの最適化設計に効果を上げている。

しかしながら、メカの動的な機能に対する検証やレビューには、3次元データの活用が十分ではなかった。

3次元データを用いたメカの動的なレビューや検証課題は、大きく次の二つに分けることができる。

１）機構の運動特性と力学的構造を厳密に解析するもの。

２）構造的な動き、操作性、組立性、幾何的な可動性と可制御性など力学的要素を厳密に扱わないもの。

本稿で紹介するシミュレーションシステムは、特に後者の検証課題に対して機能的に優れた富士通製3次元機構シミュレータをベースに開発したものである。

3次元データを基にパソコン上に実物そっくりの仮想モデルを構築し、このモデルを実際のメカと同じように動かし、制御できるシミュレーションシステムである。

これにより、前記後者の課題に対するレビューや検証、特に今回紹介する "メカ制御" という観点でのメカの可制御性と制御仕様の検討が設計初期段階から効果的に進められ、さらに設計された制御ソフトウェアの検証も実機メカの試作に頼ることなく進めることができる。

本システムは株式会社富士通研究所殿に研究開発を委託したもので、当社の要望が随所に織り込まれている。

先ず、開発の背景となった、従来の制御ソフトウェア開発とメカ開発における問題点について述べる。

２．制御ソフトウェア開発における問題

メカ構造の複雑化に伴って、これを最適に制御する組み込み型ソフトウェア(ファームウェア)の制御ロジックも複雑化している。 また、システムの高度化によってソフトウェア全体の規模も年々増大しており、如何に効率的に開発を進めていくかが課題となっている。

ここで、ソフトウェア開発の基本的な流れを図-1に示す。

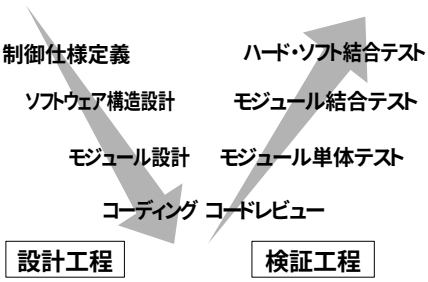

図-1　ソフトウェアの開発工程
Fig.1 Software Development Process

開発工程は、制御仕様を定義して段階的に設計仕様を展開しながらソフトウェアを作成する設計工程と、各段階の設計仕様に照らしてソフトウェアの品質を確認していく検証工程の2つからなる。

この一連の開発工程において、従来、以下のような問題があった。

２．１ 設計工程における問題

設計工程においては、制御仕様の細部がなかなか決まらず最終仕様が確定しない、という問題があった。

メカをどのような部品で構成し、各部をどのように動作させて製品機能を実現するかは、機構設計者の設計意図によって決まる。従って、どのようなタイミングでモータを駆動し、どのようにセンサ信号を検知しながら制御を行うかは、先ずは機構設計者によって定義され、1次的な制御仕様書にまとめられることになる。

ソフトウェア設計者は、機構設計者から提示される仕様を基にソフトウェア設計作業に着手することになる。

ところが、このような一次的な仕様書には(機構制御に限ったことではないが)、仕様としての曖昧さ、矛盾など細部にモレや問題があることが多い。

従って、ソフトウェア設計者が行うべき第一の作業は、"制御"という観点から与えられた仕様と制御対象とを分析し、矛盾などを解決し、曖昧さを排除しながら細部仕様を再定義していくことになる。 場合によってはメカ構造についての変更を提案しなければならない。

これが設計工程での制御仕様と制御対象の可制御性の検討である。この作業は、ソフトウェア設計者と機構設

＊1: Pro/ENGINEERは、米国Parametric Technology社の商品名です。

計者との協力によって進められる。

ところが、従来の開発環境ではこの作業を進めるのが非常に困難であった。ソフトウェア設計者にとっては、設計段階にあるメカ(制御対象)の構造を平面的なCAD図面やタイミングチャートだけから理解するのが困難であるからである。メカ構造の複雑化に伴って、この傾向は強くなってきている。

設計段階での十分な解析が現実的には不可能という場合があり、基本的な制御仕様だけを定義しておいて、細部は実際にメカを試作して動かしながら決める、というケースも多くなっていた。

２．２ 検証工程における問題

設計された制御ソフトウェアは図-1に示したように段階的にテストされ、ソフトウェア単体での評価から制御対象(メカ)を実際に結合した状態での検証へと進む。

この検証工程においては、次のような問題があった。

１）実機メカが完成するまで本格的な結合テストが進められない。

メカの試作には部品製作手配のリードタイムと組み立て調整作業が必要になり、一般に制御ソフトウェアの作成よりメカの試作には時間がかかる。そのため、ソフトウェアは既に評価開始できる状態にあるのに試作メカが未だ完成していなかったり、あるいは試作メカの完成時期に合わせてメカ制御部分のソフトウェア設計計画を意図的に遅らせる(後回しにする)という場合がある。いずれにしても、実機メカとの結合テストによる問題の摘出と対策が遅れ、全体的な開発遅延を招く。

1個のモータと数個のセンサでレバー機構を位置決め制御するだけの単純なメカ(例えば、CDプレーヤのディスク挿排機構や、ナビゲーションシステムなどの液晶表示パネルの開閉機構)の制御であれば、実機を用いなくても図-2に示す例のように適当な評価治具を用意して入出力テストを行うだけで、ソフトウェアが制御仕様を満たしているかどうかの検証は可能である。しかし、多くのモータとセンサで構成されるCDチェンジャやMDチェンジャのような複雑なメカでは、その評価は容易ではなく、入出力テスト用の専用評価治具をその都度製作するか、実機に頼らなければ評価ができなかった。また、評価治具を用いた入出力テストだけでは、制御仕様そのものが妥当かどうかの検証までは出来なかった。

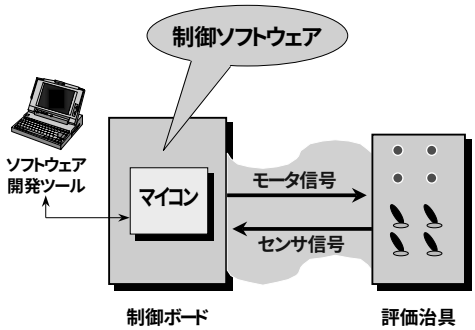

図-2　評価治具によるソフトウェア評価
Fig.2 Software Evaluation by Evaluation Equipment

２）試作メカが破損または故障して検証作業が中断してしまう。

開発の初期段階では、メカと制御ソフトウェアの完成度がともに低い為、ソフトウェアバグによる制御の暴走やメカの設計不良などにより、試作メカを破損させたり故障させてしまうことが度々ある。このような事態が発生した場合は、原因を調査して修理を行うか、代替品を準備する必要があり、時間のロスが発生する。この間制御ソフトウェアの検証作業が中断してしまう。

３）異常状態などに対する制御ソフトウェアの検証が容易でない。

正常時のメカ動作や、正常な操作を想定した基本機能の検証は、実機を用いれば比較的容易に作業を進めることができる。しかし、メカが異常な状態に陥った場合を想定した検証や、異常な製品操作を想定した検証は容易ではない。異常なメカ状態とは、モータの故障、機構部品の引っ掛かり、センサの故障や誤作動、ディスクのスタックなどである。制御ソフトウェアには、このような異常を想定したリカバリ処理、フェールセーフ処理が多く含まれる。その規模は、ソフトウェア全体の7〜8割を占める。

このような制御ソフトウェアを検証する為には、異常状態を効率よく、繰り返し再現させる必要がある。

実機メカを用いた評価では、力ずくで強引にメカ動作を停止させたり、メカ内部のスイッチをイジワル操作してみたり、ギアを指で回してみたり、分解して細工するなどして所望のメカ状態をつくり出すしかなかった。

４）不具合に対するメカ要因と制御ソフトウェア要因の切り分けが容易でない。

開発の初期段階では、試作メカの動作が不安定である場合があり、発生した不具合がメカに起因するものなのか、ソフトウェアに起因するものなのかの切り分けが困難な場合がある。例えば、センサが誤作動したのか、ソフトウェア側の検知処理が誤動作したのか、というケー

スである。現象の再現頻度が低い場合、ソフトウェア側の要因解析に時間を費やしたけれども実はメカ側の要因だった、ということが度々発生する。

３．メカ開発における問題

メカ開発においては、先に述べたように"制御"という観点からのソフトウェア設計者との共同によるメカ構造の解析や他の担当者を交えたレビューが困難であったため、先ずは実機メカを試作してから検討する、という開発スタイルをとることが多かった。

その結果、試作後の検討過程で問題が判明することとなり、メカ設計の変更と再試作という手戻りが生じていた。

以上述べたように、制御ソフトウェアとメカの開発が実機メカの試作に依存していたことにより、開発上多大なロスを生じていた。

メカ構造とその制御に関連して摘出される、主な不具合事例を表-1に示す。

表-1　主な不具合事例と対策
Table 1 Examples of Major Problems and Solutions

不具合事例	対策
特定のメカ状態に陥ると、正常状態への復帰や、メカ状態の初期化ができなくなる。	・メカ構造の見直し ・制御仕様変更 （フェールセーフ制御追加）
センサ構成の不備から、複数の異なるメカ状態を識別できず、制御できない。	・センサ構成の見直し ・制御仕様変更

3次元CADデータの利用拡大を進める中、前述した"メカ制御"に絡む問題、制御ソフトウェアの設計工程と検証工程の問題に対しても3次元データの利用が有効であると考え、今回、以下に述べるシミュレーションシステムを企画、開発した。

４．シミュレーションシステムとその活用

ここに紹介するシステムは、富士通製3次元機構シミュレータ「FJVPS (Fujitsu Virtual Product Simulator」をベースに開発したものである。

本システムで実現される仮想メカモデルを用いた開発の手順に沿って、システムの機能と活用効果を紹介する。

４．１ 3次元データに基づく仮想メカモデルの構築

先ず、3次元CADデータを用いて仮想メカモデルを構築していく。

１）3次元CADで設計された部品形状データとアセンブリ情報を変換ツールを用いてポリゴンデータなどに変換し、システムに取り込む。

２）図-3に示すような回転部品・スライド部品の動きと、ギア・カム・溝機構・クラッチ機構などに対する部品間のリレーションを設定する。

図-3　各部品の動きの設定
Fig.3 Movement Settings of Individual Parts

上記設定は、ベースとなる機構シミュレータの基本機能により実現される。

今回、上記基本機能に対して機構定義の拡張と制御シミュレーションのための機能を新たに組み込んだ。

３）ギアのバックラッシュや溝機構のガタなどを定義する。

これは、前記リレーションにヒステリシス関係を定義することで実現している。(図-4)

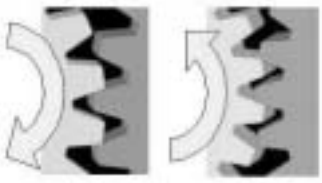

図-4　ギアのバックラッシュ
Fig.4 Gear Backlash

４）被搬送モデルの動きを定義する。

CDチェンジャメカニズムでは、複数のディスクの装填、排出、そしてディスク交換が主な動作となる。このような動きをモデル化するためには、機構構成部品としてのメカの動きとともに、搬送されるディスクの動きをモデル化する必要がある。本システムでは、図-5に示すよう

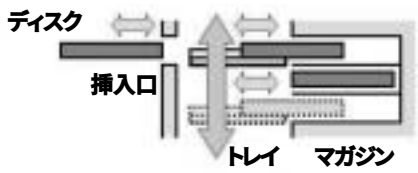

図-5　被搬送モデルの動き
Fig.5 Movements of the Model Being Transferred

な被搬送モデルの動きも定義できるようにした。

５）制御ソフトウェアとのインタフェース部分であるモータモデルとセンサモデルの定義設定を行う。

仮想メカモデルを "制御" するためには、制御ソフトウェアによって出力される制御信号に基づいてメカを駆動するモータモデルと、制御ソフトウェアへの入力となるセンサモデルが必要になる。センサモデルは、メカ動作に応じてセンサ信号を出力する。本システムでは、モータモデルとしてDCモータとステッピングモータの2種類を用意した。また、センサモデルとして、ON/OFFスイッチ、ポテンショメータ、エンコーダの各モデルを用意した。

以上の設定で仮想メカモデルの構築は完了する。これらの設定は、設定項目毎に順次表示されるダイアログ(図-6)に対して必要パラメータを入力していくだけの簡単な操作で行える。図-7 は、当社デッキメカニズムのモデル規模と実際にモデル化に費やした工数である。比較的短

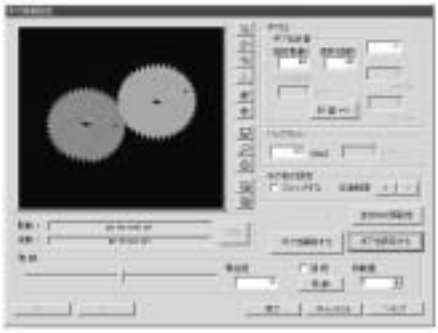

図-6　設定ダイアログの例
Fig.6 Example of Setting Dialog

a）CDチェンジャメカニズム

所要工数：約20時間

モデル規模：

部品点数	1,067点
ポリゴン数	557,486面
関節数（回転、スライド）	103点
モータ数	5個
センサ数	13個

b）カセットプレーヤメカニズム

所要工数：約8時間

モデル規模：

部品点数	350点
ポリゴン数	192,830面
関節数（回転、スライド）	46点
モータ数	2個
センサ数	6個

図-7　モデル規模と設定工数の事例
Fig.7 Example of Model Dimension and Time for Settings

時間でモデルを構築することができる。

尚、本作業が機構設計者自身によって行われるのは言うまでもない。

４．２ 仮想メカモデルによる機構のレビュー

パソコン上に構築されたモデルは、実際のメカを操作するように自由に動かすことができる。例えば、任意のギアやレバー機構をマウスによるドラッグ操作でリアルタイムに動かすことができる。また、モデルに対する視点の変更、断面表示、部品の分解や交換などが自在にできる。

高速描画技術の導入により大規模なモデルでも、違和感なく操作でき、高速干渉チェック機能により、メカを動作させながらの動的な干渉とクリアランスのチェックが行える。

機構設計者以外でも容易に操作方法を習得できるので、製造/営業/サービス担当者などによるレビューも効果的に進めることができる。

４．３ ソフトウェア設計者による制御対象の分析

ソフトウェア設計者にとっても操作は容易である。機構設計者から提示された一次的な制御仕様に基づいて、マウス操作などで各部を "制御" しながら制御仕様の不備、妥当性を解析していくことができる。

メカ構造を断面表示したまま、あるいは外部の部品を透明/半透明表示にした状態で、内部構造の動きを見ながら解析できるなど、実機メカでは不可能な作業も実現できる。

このような作業により、"制御"という観点からのメカ構造と制御仕様の徹底的な分析が行える。

ここで摘出された問題は、制御仕様やメカ設計にフィードバックされることになるが、ここでは未だ設計段階にあり変更は容易である。確認と変更のサイクルを仮想モデルを用いて繰り返しながら、メカ構造と制御仕様の細部について、設計の完成度を高めていくことができる。

４．４ 制御シミュレーションのための準備

ソフトウェア設計者は、前記作業で確定した制御仕様に基づいてソフトウェアの設計とコーディングを行うことになる。設計された制御ソフトウェアは、本システムが提供する制御シミュレーション機能によって検証できる。

制御シミュレーションを行うためには、前述したモータモデルとセンサモデルの定義設定に加えて、以下の設定と準備が必要となる。

1）シミュレータと同期をとる為の準備

仮想モデルをマウスで操作しながらレビューを行う場合には、シミュレータのレスポンスは問題にならなかった。しかし、リアルタイムで動作するソフトウェアとの接続では、パソコンの処理速度が遅いことが問題となった。

この問題を解決するため、本システムでは制御ソフトウェアとシミュレータとで同期をとる方法を採用した。制御ソフトウェアがシミュレータの処理速度に合わせて制御処理を進める。同期信号をお互いに送受信することで交互に処理を進め、例えばシミュレータが10msec分の計算を行ったら、次に制御ソフトウェアが10msec分の処理を進めるとということを繰り返す。(図-8) これによりスローモーション的ではあるが、厳密なタイムスケールでのシミュレーションが可能になる。実際にシミュレーションした事例では、10msec毎の同期で、実物の約4倍の時間でのシミュレーションを実現している。

完全なリアルタイムでのシミュレーションではないが、ソフトウェアによる制御とモデル挙動を細部までじっくり確認するのには好都合である。

制御ソフトウェアには、シミュレータと同期をとるためのロジックを組み込む必要があるが、その規模は僅かである。尚、シミュレータ側では、同期条件などを設定する。

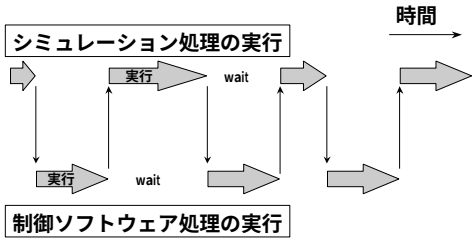

図-8　シミュレータと制御ソフトウェアの同期
Fig.8 Synchronization of Simulator with Control Software

2）シミュレータと制御ソフトウェアの接続

パソコン上に構築された仮想メカモデルと制御ソフトウェアとの接続は、パソコンに装着された汎用I/Oボードを介する。入出力される信号は、モータ制御信号、センサ信号と前述した同期信号である。同期信号以外は、実際の製品での制御ソフトウェアとメカの接続と同じである。

一方、制御ソフトウェアは制御ボードに実装されたマイコン、あるいはボードに接続されたソフトウェア開発ツール上で動作させることになる。

４．５ 制御シミュレーション

以上の設定を行うことで、仮想メカモデルを用いた制御シミュレーションが可能になる。実機を用いた従来の結合テスト(図-9)に対し、本システム上に構築された仮想メカモデルを制御することで、実機と同様に検証を行うことができる。(図-10) このような、仮想モデルと実際の制御回路とを接続した状態での制御シミュレーションをHIL(Hardware In the Loop)シミュレーションと呼んでいる。

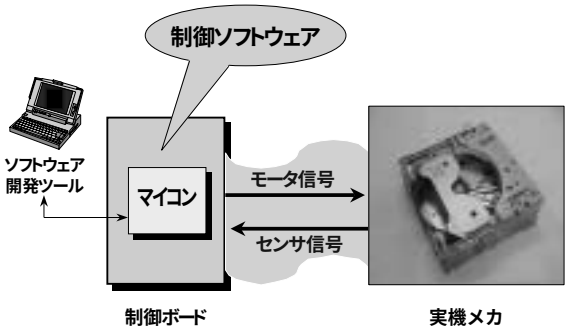

図-9　実機メカを用いた結合テスト
Fig.9 Combination Test Using Completed Mechanism

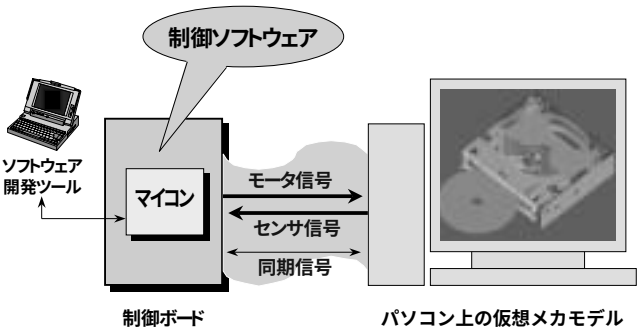

図-10　仮想メカモデルを用いた結合テスト
Fig.10 Combination Test Using Virtual Mechanism Model

４．６ 制御タイミングの検証

制御シミュレーションにおいて、制御タイミングは重要な評価項目の一つである。前述したように、本システムでの制御シミュレーションは完全なリアルタイムではなく、またタイムスケールもリニアでないため"現実世界"の測定器は使えない。

そこで、本システムには、シミュレーション中に入出力される信号を正しいタイムスケールで捉えるためのロジックアナライザ機能を組み込んだ。これによって、時間的に正確な入出力信号タイミングを測定することができる。(図-11)

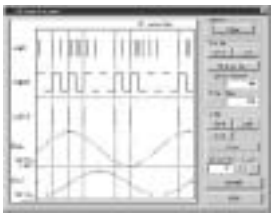

図-11　ロジックアナライザ機能
Fig.11 Logic Analyzer Function

４．７ 異常状態などの再現

本システムでは、実機では困難なメカの異常状態の再現も容易である。モータ故障やセンサ故障(非動作など)は、モータモデルやセンサモデルの設定ダイアログで容易に設定できる。また、センサモデルでは、ON/OFFスイッチで問題となるチャタリングを任意の条件で生成できる。(図-12)

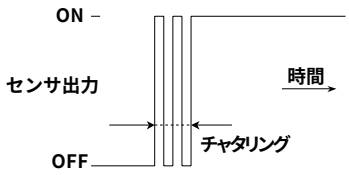

図-12　チャタリング生成
Fig.12 Chattering Generation

機構の引っ掛かりや跳ね返り、イジワル操作なども簡単に再現できる。 図-13は、ディスクを強引に引き止めたときの制御ソフトウェアによるリトライ処理とメカの挙動を確認した事例である。

異常状態の再現により、制御ソフトウェアのリカバリ制御の検証が容易にできる。

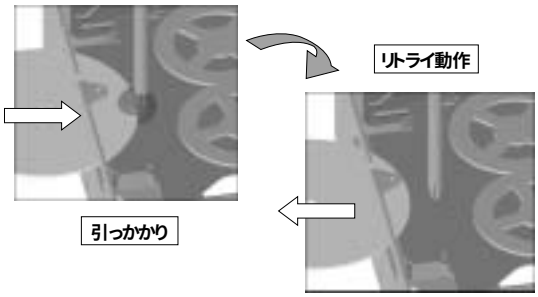

図-13　制御ソフトウェアによるリトライ
Fig.13 Retry by Control Software

本システムでは、部品を操作して任意のメカ状態を作り出せる。また、それらの状態の登録と再呼出しができるので、メカ状態の再現が容易である。

以上、本システムの機能により、制御ソフトウェアを実機メカの試作に頼らず検証することができる。

４．８ 実機による調整と最終確認

本システムでは、力学的な要素を厳密に扱わない。従って、仮想メカモデルと実機メカでは動作タイミングが異なる。

メカ制御では、モータのブレーキ時間や駆動時間などをメカの特性に合わせて決定することがあるが、ソフトウェアのこれらの制御パラメータについては、実機メカを用いて最終調整を行い、動作確認を行う必要がある。

５．状態遷移解析/設計ツールとの連携

本システムでは、さらに次に述べる機能に対応している。

制御ソフトウェアの品質をより早い段階で高め、開発上のロスを最小限に抑えるためには、より上流工程からのシミュレーションが有効である。

大規模、複雑化するソフトウェアの開発ではソフトウェアの振る舞いを状態遷移図/表を用いて分析定義しながら設計を進めることが有効である。本システムでは、このようなソフトウェア設計作業を支援する状態遷移解析/設計ツールMATLAB/Stateflow ＊2、あるいはZIPC ＊3に連携するインタフェース機能を持たせている。 (図-14)

この機能を使えば、仮想メカモデルとの結合テストをソフトウェア設計段階から進めることができ、より早い段階での品質の作り込みが可能となる。

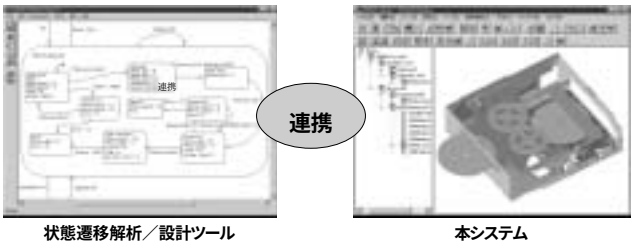

図-14　状態遷移解析／設計ツールとの連携
Fig.14 Linkage to State Transition Analysis/Design Tools

６．本システムによる開発の効果

従来の実機メカの試作に依存した開発プロセス(図-15)に対し、本システムを用いた開発プロセス(図-16)の特徴と効果は以下のとおりである。

１）設計初期段階から、ソフトウェア設計者と機構設計者が協調して各設計作業を進めることができる。

＊2: MATLAB/Stateflowは、米国MathWorks社の商品名です。
＊3: ZIPCは、キャッツ株式会社の商品名です。

2）メカ構造が複雑であっても、ソフトウェア設計者による制御仕様とメカ(制御対象)の詳細な動きの分析が可能となり、制御上の問題点を早期に摘出して設計に反映できる。
3）実機メカの試作を待たずに制御ソフトウェアの検証ができ、メカと制御ソフトウェアのコンカレント開発が可能となる。
4）メカの破損、故障によるソフトウェア検証の中断が発生しない。
5）実機では困難な異常状態などに対する動作検証が、再現性よく効率的に進められる。
6）再現性が良いので、不具合要因の切り分けが容易である。
7）メカ開発においては、出図前に制御系との連携動作を確認できるので、不具合対策のための手戻りと再試作のためのロスコストを抑えることができる。

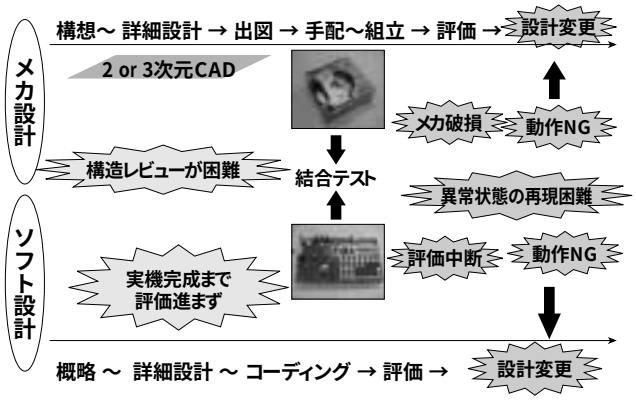

図-15　従来の開発プロセス
Fig.15 Conventional Development Process

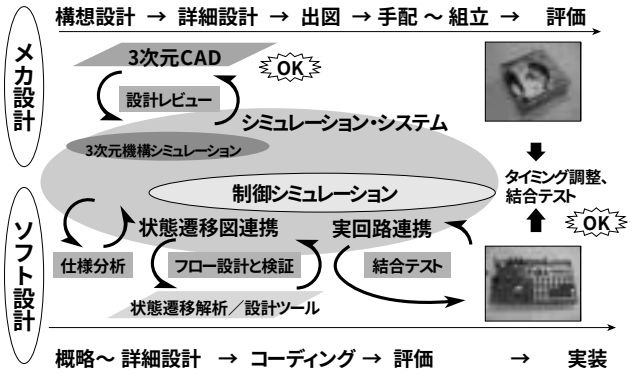

図-16　本システムを用いた開発プロセス
Fig.16 Development Process Based on This System

今回開発したシステムは、機構設計と制御ソフトウェアの協調設計、コンカレント開発を可能にし、準リアルタイムで仮想メカモデルと制御ソフトウェアの結合テストができるところに特徴がある。

本システムを用いることで、従来工数の3～4割削減が可能となる見込みである。

7．おわりに

今回このシミュレーションシステムを開発途上のデッキ開発において部分的に試行する事で、今後のデッキ開発が大幅に効率化できる見通しを得た。今後は、他のシミュレーション（応力解析，振動解析など）や本システムを積極的に活用したデジタルエンジニアリングを推進し、さらなる効率化と高品質化に取り組みたい。

尚、本稿に紹介したシステムは、富士通株式会社殿で商品化予定である。

謝辞

本シミュレーションシステムの開発にあたり多大なご支援とご協力を頂いた富士通株式会社並びに、株式会社富士通研究所の関係者の方々に深く感謝の意を表します。

参考文献

1）富士通株式会社編:組込み用ソフトウェア開発ソリューションFJVPS/HIL、ＦＩＮＤ、Vol.19,Nov.1,2001
2）橋間 他:組込み用ソフトウェア開発支援システム(VPS/HIL)の構築、第10回日本機械学会設計工学・システム部門講演会講演論文集,pp148-149,2001

筆者紹介

新井　健睦　（あらい　けんぼく）

1981年入社。以来カセット、CDプレーヤなどマイコン応用機器のソフトウェア開発に従事。
現在AVC本部コンポーネント事業部デッキ技術部評価実験課課長。

後藤　務　（ごとう　つとむ）

1986年入社。以来カセットプレーヤの開発に従事。
現在AVC本部コンポーネント事業部デッキ技術部評価実験課在籍。

橋間　正芳　（はしま　まさよし）

1991年（株）富士通研究所入社。以来、ロボット用視覚計測技術、シミュレータ技術の研究開発に従事。日本ロボット学会会員。現在、自律システム研究部在籍。

岩﨑　孝夫　（いわさき たかお）

1986年入社。以来カーオーディオのファームウェア開発に従事。
現在AVC本部コンポーネント事業部デッキ技術部ＤＦプロジェクト在籍。

佐藤　裕一　（さとう　ゆういち）

1986年（株）富士通研究所入社。以来、制御、シミュレーション技術の研究開発に従事。
日本ロボット学会、情報処理学会各会員。現在、自律システム研究部在籍。

千田　陽介　（せんた　ようすけ）

1997年（株）富士通研究所入社。以来、VPS 用・磁気ディスク用HILシミュレータの開発に従事。
日本機械学会、計測自動制御学会各会員。現在自律システム研究部在籍。