

メカトロニクス系テストツール "Virtual Device" の開発

6K-1

小池 太* 児玉隆一郎** 永松 祐嗣***

* (株)日立製作所システム開発研究所
 ** (株)日立製作所計測器事業部
 *** (株)日立製作所情報システム事業部

1. はじめに

大部分のメカトロニクス系装置にはマイコンが組込まれている。従来から、メカトロニクス系装置の開発においては、

- ・実機テスト主体による開発コストの増大
- ・制御ソフト設計における並行処理動作確認困難
- ・バッチ式開発による不具合集中発生
- ・仕様の頻繁な変更

などの問題が指摘されていた。

近年、ますますメカトロニクス系装置の多様化、高機能化が進み、装置を制御するソフトウェア(以下、制御ソフト)開発環境に対する高機能化が望まれている。

そこで我々は、対象装置の各機構をオブジェクトモデルとして捉え、これらを用いて制御ソフトのためのテストを行うツール "Virtual Device" を開発した。この Virtual Device は、

- ・制御言語による統合的な開発環境
- ・非実機/実機共通のデバッグ/トレース環境
- ・装置動作のビジュアル検証

をめざしている。また、我々はテストのみに留まらず、制御ソフト開発の高信頼化/高効率化をめざした開発ツールとしての位置付けもしている。

2. 装置モデリング

メカトロニクス系装置の機構制御は、主に

- ・モータ駆動による機構動作(自由度)
- ・機構間(自由度間)の制約
- ・モータ制御(自由度への操作)

によって決定される。これらの要因を、ハード/ソフトで分離し、ハードを静的モデル(機構の自由度と機構間の制約)、ソフトを動的モデル(自由度への操作)としたオブジェクトモデルとして捉えた。このモデリング方式

"Virtual Device" - A Test Tool for Mechatronics Instruments.

Futoshi KOIKE

HITACHI, Ltd. Systems Development Laboratory

Ryuichiro KODAMA

HITACHI, Ltd. Instrument Division

Yuji NAGAMATSU

HITACHI, Ltd. Information Systems Division

によると、

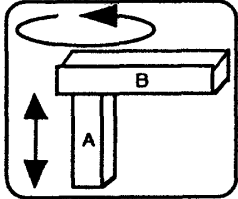
- ・メカニカルに関連して動作する機構を静的モデルで表現
- ・モータ種類による制御の違いを動的モデルで吸収

することが可能となり、ハード/ソフトが協調した制御が可能となる。

2.1 静的モデル記述

静的モデル記述では、対象の機構(名称)に対する3次元の形状とそのモータ駆動による動作(自由度)、および、自由度間の制約条件を定義する。

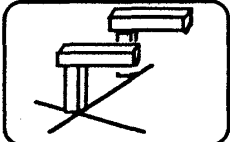
<pre>sub (sarm) boxe(-2, -2, 0, 4, 4, \$sarm_z); trans(0, 0, \$sarm_z); rotz(\$sarm_th); boxe(-5, -5, 0, 70, 10, 5); </pre>	<p>機構sarm定義 直方体A(Z軸方向に可変) 座標系の移動(Z軸方向) 座標系の回転(Z軸) 直方体B</p>
--	--



記述例1 静的モデル記述その1

複数の機構からなる装置に対しては、静的モデルのインスタンス化と組み合わせ記述により、全体を構築する。

<pre>unit (main) task(s1) {\$sarm.arm_th, \$sarm.arm_z}; unit (sarm); trans(0, 70, 0); task(s2) {\$sarm.arm_th, \$sarm.arm_z}; unit (sarm); </pre>	<p>実行部 インスタンス生成 機構描画 座標系の移動 インスタンス生成 機構描画</p>
---	--



記述例2 静的モデル記述その2

組み合わせ記述により、各機構、動作を共用することが可能である。この共用で、

- ・システム構成の変更に柔軟に対処
- ・コード量の縮小と実績あるコードの利用

が可能となり、開発の高効率化/高信頼化を望める。

2.2 動的モデル記述

動的モデル記述では、機構に対する操作を記述する。動的モデル記述は、並行処理記述が可能なBASIC言語を用いる。このBASIC言語による制御の基本記述は、

```
機構名称.自由度 /メッセージ 引数1,引数2,...
```

の形式で行われる。各種モータコントローラ違いをメッセージにより吸収することにより、様々な制御装置への対応が可能である。

```
fork *s1:fork *s2:end           インスタンスの並行起動
*s1
*s2
sarm.arm_th /rate 90           回転方向速度設定
sarm.arm_z /rate 20*2         上下方向速度設定
while (1)
  sarm.arm_th /move -45:delay 600 45度回転
  sarm.arm_z /move -20:delay 500 20ポイント下降
  sarm.arm_z /move 20:delay 500 20ポイント上昇
  sarm.arm_th /move 90:delay 1300 90度回転
wend

記述例3 動的モデル記述
```

動的モデル記述では、同じ機構を1つのプログラムで記述できるため、同じ動作を保証し、制御ソフト開発の信頼性を向上させる。

3. ビジュアル検証機構

制御ソフトの非実機テストを効率的に行うため、多角的なビジュアル検証機構を実現した。前述の静的モデル/動的モデルが、ここで検証される。Virtual Deviceの画面構成を図1に示す。ビジュアル検証機構は、OSF/Motif¹¹⁾, X-Windowシステム¹²⁾を利用している。

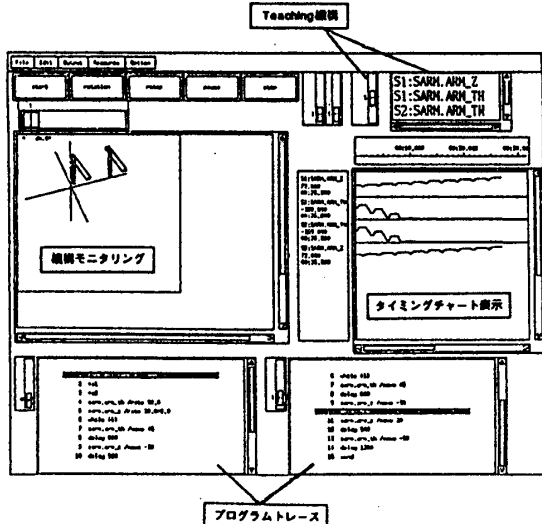


図1 Virtual Deviceの画面構成

¹¹⁾ OSF/Motifは、OSFの商標である。

¹²⁾ X-Windowシステムは、米国マサチューセッツ工科大学の登録商標である。

3.1 機構モニタリング

静的モデルにおける形状記述をもとに、機構を3次元グラフィック表現し、制御ソフトの実行に合わせてアニメーション動作する。実機を使ったテストと同様な確認が可能となる。

3.2 並行動作の確認機構

実機では確認困難な並行処理部分の確認手段として、以下の2つの機構を実現した。

- ・ プログラムトレース
並行に動作するプログラムの実行位置の確認
- ・ タイミングチャート表示
各機構で発生する動作状況をリアルタイムにタイミングチャート表示し、同期関係の詳細を確認

3.3 Teaching機構

ビジュアル検証機構を制御ソフト設計に活かすため、制御ソフトの記述とは別に、ビジュアル画面から直接機構を操作することを可能とした。この機能は、

- ・ 設計段階では予測しづらい機構動作の確認
- ・ 機構制御のプログラミングレス化への足掛かり

として有用である。

4. 考察

機構のオブジェクトモデルとビジュアル検証機構を利用することによって、以下の効果が期待できる。

- (1) 機構のオブジェクトモデルを実現することで、非実機上でハード/ソフト協調制御を確認することができる。従って、制御ソフトの開発効率向上と高信頼化が期待できる。
- (2) ビジュアル検証機構により、実機と同様な検証方法が実現できた。このことは、非実機/実機間で共通なテスト環境を提供し、効率的な開発を可能とする。
- (3) BASIC言語と、機構のオブジェクトモデルから、機構部品の作成と検証の繰り返しが可能となった。これにより、プロトタイプからのインクリメンタルな開発が可能となる。不具合発生の影響が段階的に抑えられ、信頼性が向上し、かつ、効率的な開発が可能となる。

5. おわりに

Virtual Device は、機構をオブジェクトモデルとして採用した。機構のハードの関連を静的モデルとして、制御ソフトを動的モデルとして記述している。そして、これらモデルにもとづいて検証ツールを提供することで、ハード/ソフト協調制御を高信頼、高効率に実現できる。我々は、Virtual Device を医用自動分析装置へ適用することを検討している。

6. 参考文献

J.ランボー/M.ブラハ他著、羽生田栄一監訳、「オブジェクト指向方法論OMT-モデル化と設計」、トッパン、1992