

ヒューマノイドロボット総合ソフトウェア開発環境：OpenHRP

産業技術総合研究所知能システム研究部門 比留川博久

1. はじめに

経済産業省は、1998年から5年間・総額約50億円の予算で、人間協調・共存型ロボットシステム研究開発プロジェクト(Humanoid Robotics Project, HRP)を実施中である。本講演では、HRPの開発の一環として産業技術総合研究所などが開発中のヒューマノイドロボット総合ソフトウェア開発環境 OpenHRP(Open Architecture Humanoid Robotics Platform)のを紹介する (<http://www.is.aist.go.jp/humanoid/openhrp/>よりユーザ登録後、ダウンロードが可能)。

OpenHRPはヒューマノイドのような多自由度を持ち、環境との頻繁な衝突/接触を起こす対象の動力学シミュレーションを効率的に実行する環境を提供する。OpenHRP上で開発したソフトウェアは、バイナリ互換で実ロボットを制御する実時間OS上で稼動する。OpenHRPを用いることにより、ヒューマノイドロボットなどの多自由度ロボットを制御するソフトウェアを効率良く開発することが可能になる。以下、OpenHRPの概要、シミュレータと実機間のバイナリ互換性の実現方法、運用例について述べる。

2. OpenHRP概要

OpenHRPはCORBAを用いた分散オブジェクトシステムとして実装されており、ネットワーク上での分散計算やオブジェクト単位での並行開発、機能拡張が可能な構成となっている。CORBAの実装としては、ORBacus 4.1.0を利用している。OpenHRP

を構成するCORBAサーバ群と、これらを利用するための統合ユーザインターフェースの構成を図1に示す。

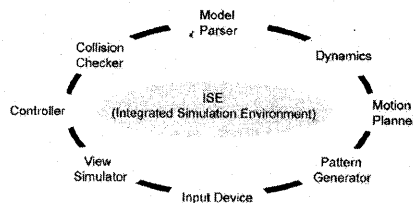


図1 OpenHRPの構成

OpenHRPは、Windows及びLinux上で稼動し、ロボットのモデルはVRML97により記述される。以上述べた様に、OpenHRPは可能な限り業界標準の仕様に従って開発されており、オープンアーキテクチャなシステムを目指している。各サーバの機能は次の通りである。

- ・ Model Parser—ロボット及び動作環境のモデル読込を行う。
- ・ Collision Checker—ロボットと環境の干渉チェックを行う。
- ・ Dynamics—ロボット等の稼動物体の動力学シミュレーションを行う。
- ・ View Simulator—ロボットの視野画像をシミュレートする。
- ・ Motion Planner—ロボットの障害物回避動作の計画を行う。
- ・ Pattern Generator—ヒューマノイドロボットの歩行パターンを生成する。
- ・ Controller—ロボットの姿勢安定制御

を行う。

・ **Input Device**—ジョイスティック等の対話的入力装置からの入力を処理する。これらの各サーバの起動は、ORBacus の **IMR (Implementation Repository)**機能を用いて必要に応じて自動的に行われる。また、各サーバ内部は **CORBA** インタフェース部分と演算ロジック部分に分かれており、演算ロジックのネイティブインタフェースを実時間制御ソフトウェアから利用することができる。

シミュレーションは、**Collision Checker**、**Model Parser**、**Dynamics**、**View Simulator** の4つの **CORBA** サーバを図2に示す様な GUI を備えた **ISE (Integrated Simulation Environment)** と呼ばれる **CORBA** クライアントから利用することによって行われる。

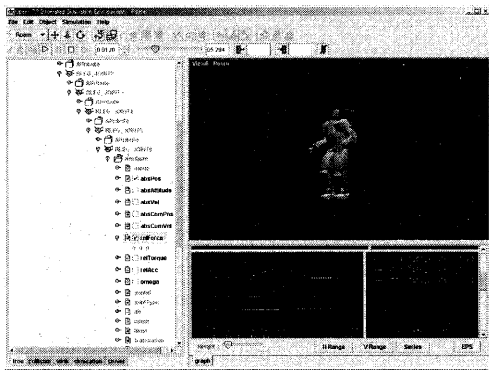


図2 Integrated Simulation Environment

ISE はシミュレーションにおける各種初期設定、シミュレーションデータの解析を行う機能を提供する。シミュレーション準備段階ではユーザはこの GUI を用いて、シミュレーション世界を構成するモデルを読み込み、初期位置/姿勢、関節角を 3D 画面内でインタラクティブに設定することができる。積分方式、積分刻み、各モデルへのコントローラの割り当ても GUI から設定ことができ、それらの情報はプロジェクトファイルに保存して再利用することがで

きる。シミュレーション実行時には途中経過が 3D 画面に表示され、必要に応じて途中中断し設定を変更して再実行することが可能である。またシミュレーション終了後には、シミュレーション結果をグラフ化したり、動画化したりすることが可能である。

ユーザはあらかじめ用意されているコントローラサーバの雛型に実時間制御のロジックを組み込み、**CORBA** サーバ化してそれをシミュレーション世界のロボットに割り当てることができる。また、歩行パターンジェネレータや障害物回避動作を生成する動作計画機能も適宜利用することができる。

ロボット及び環境のモデルは、**VRML97** をベースに **h-anim WG** で策定されたヒューマノイドの記述フォーマットを一部拡張して、一つのファイルに形状、機構、動力学パラメータといった情報を埋め込んでいる。**h-anim** フォーマットでは、**Humanoid**、**Joint**、**Segment** の各ノードが定義されており、各々一体のヒューマノイド全体、関節、リンクを表す。以下にヒューマノイドのモデル定義の一部を示す。

```
DEF HRP1 Humanoid {
```

```
  humanoidBody [
```

```
    DEF WAIST Joint {
```

```
      jointType "free"
```

```
      translation 0 0 0
```

```
      rotation 0 0 1 0
```

```
      children [
```

```
        DEF BODY Segment {
```

```
          mass 0.5
```

```
          momentsOfInertia [1 0 0 0 1 0 0 0 1]
```

```
          children [
```

```
            Inline {url "shape.wrl"}
```

```
          ]
```

```
        }
```

```
      DEF LEG_JOINT0 Joint {
```

```
        jointType "rotate"
```

3. 実時間システムとのコード共有の実現

応用の立場から見た場合シミュレータと実時間システム上で稼動するソフトウェアの共有が必要な理由は、次の通りである。

- ・ ソフトウェアの移植に時間を割きたくない。
- ・ シミュレーションで検証を行ったコードをそのまま実機でも使いたい。
- ・ ロボットを制御するためではなく、そのシミュレーションをするために開発したコードを、出来るだけ制御コードに転用することにより、信頼性の高い制御系を構築したい。

以下、OpenHRP におけるコード共有の実現方法について述べる。前述した様に、OpenHRP の各サーバの内部は、図3の様な構成になっている。

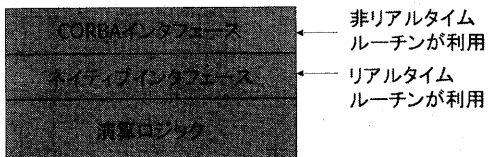


図3 OpenHRP のサーバの内部構造

すなわち、演算ロジックの上にまずネイティブインターフェイスが載り、その上にCORBA サーバ用インターフェイスが載っている。ネイティブインターフェイスはリアルタイムルーチンが利用し、CORBA インターフェイスはシミュレーション時に利用する。

ここで、コントローラ側から見た場合、シミュレーション時と実験時にAPIが異なるとコントローラのバイナリ互換性は成立しない。そこで、アダプタと呼ぶ中間的なAPIを用い、これらの差を吸収している。シミュレーション時と実時間実行時のAPIの構成を図4に示す。実時間システムにおいても、コントローラは機構データ等の各

種パラメータを読み込む必要があり、これは初期化時に Model Parser サーバからORB 経由で読み込んでいる。

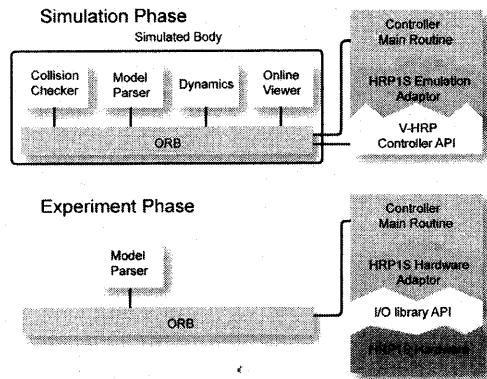


図4 アダプタ

互換性実現のもう一つのポイントは、時間の同期である。一般に、ヒューマノイドロボットの様な多自由度系の動力学シミュレーションを実時間で実行することは困難で、何らかの同期方法が必要となる。OpenHRP では、シミュレーションの順動力学の単位積分ステップと、実時間システムの OS として利用している ART-Linux の art_wait() コールを対応させることにより、この同期

4. 運用例

OpenHRP を、HRP において本田技研がハードウェアを開発したヒューマノイド HRP-1S に適用した例を紹介する。HRP-1S の外観を図5に示す。

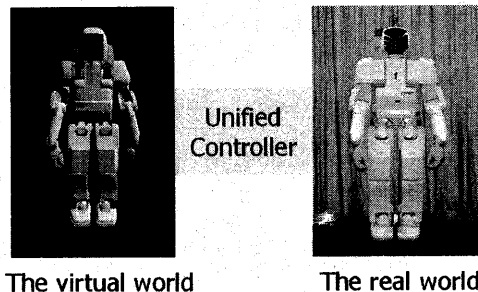


図5 ヒューマノイドロボット HRP-1S

図6は、HRP-1S を用いて歩行シミュレーションと実験を行った際のボディのロール及びピッチ角の振動と、床から足部への半力を比較したものである。十分な一致結果が得られていることが分かる。

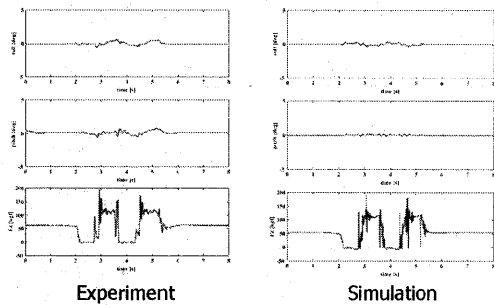


図6 シミュレーション及び実験結果

これらシミュレーション及び実験を行ったときのスナップを図7に示す。

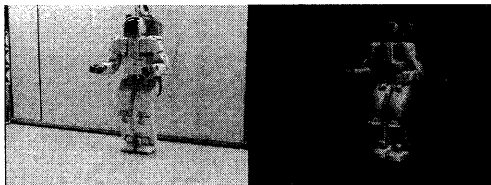


図7 歩行実験及びシミュレーション

この他、図8に示す様な不整地歩行の実験も行っている。

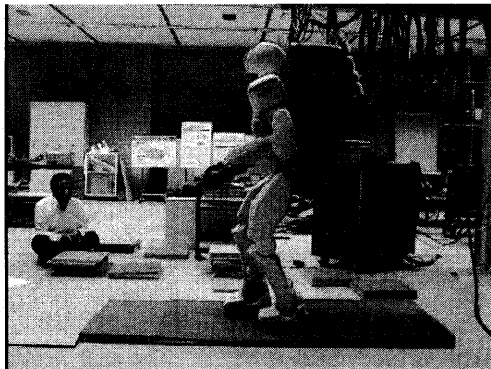


図8 不整地歩行実験

また、異なるロボットへの適用例としては、図10に示す様なHRP-2脚モジュールをジョイスティックによって対話的に操作する実験等も行っている。

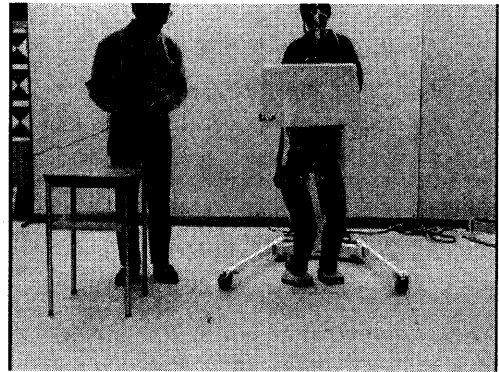


図10 HRP-2Lを用いた実験

この際、HRP-1S 用に開発したコントローラのHRP-2L への転用を実現している。

5. おわりに

本稿では、ヒューマノイドロボット等の実時間制御ソフトウェアを安全かつ効率的に行うために開発したプラットフォームであるOpenHRPを紹介した。OpenHRPが、ヒューマノイドロボットに関するソフトウェア開発の有力なツールになることを期待している。

謝辞

OpenHRPの開発を共同で行っている、金広文男を始めとするヒューマノイド研究グループ各位、東京大学中村仁彦研究室の諸氏に感謝します。