

## ヒューマノイドロボットの制御システムの開発

### Development of Control System for a Humanoid Robot

○中村友則\*, 那須康雄\*, 山野光裕\*, 武田謙郎\*,  
Genci Capi\*, 西田幸弘\*, 水戸部和久\*

○Tomonori Nakamura\*, Yasuo Nasu\*, Mitsuhiro Yamano\*,  
Kenro Takeda\*, Genci Capi\*, Yukihiro Nishida\*, Kazuhisa Mitobe\*

\*山形大学工学部

\*Yamagata University

キーワード: 人間型ロボット(Humanoid Robot), 制御システム(Control System), CORBA  
ロボットコントロール・アーキテクチャー(Robot Control Architecture)

連絡先: 〒992-8510 米沢市城南4-3-16

山形大学大学院理工学研究科 機械システム工学専攻 那須研究室 中村友則

Tel: (0238)26-3237, Fax: (0238)26-3237, E-mail: nakamura@mnasu2.yz.yamagata-u.ac.jp

## 1. はじめに

将来, ヒューマノイドロボットは災害救助や福祉などの分野での活躍が期待され, 現在, 実用化に向けた研究, 開発が盛んに行われている.

ヒューマノイドロボットは, 歩行, 物体のハンドリング, 視覚など実装すべき機能が多く, 研究分野は非常に幅広いものとなっている. ヒューマノイドロボッ

トの研究者が, これらの研究を全て一人で行うことは効率が悪く完成度も低くなる. 複数の研究者がそれぞれの機能について研究, 開発を行い, その研究成果を統合してヒューマノイドロボットのシステムを構成する方が効率がよく, 完成度も高くなるため, 各分野の研究成果の連携を容易にするシステムが必要である.

これらのことから本研究では, 各分野の研究成果を容易に連携するために,

Common Object Request Broker Architecture (CORBA)<sup>1)2)</sup>を用いたクライアント/サーバ・ロボット・コントロール・アーキテクチャ(CSRCA)を提案する。CORBAは、プログラミング言語、OSに独立であり、CORBAを用いることで、異なるプログラミング言語、OSで開発されたプログラム間の通信が可能となる。すなわち、ロボットの各機能をCORBAを用いて実装することで、各機能の連携、追加、削除、更新を容易に行うことが出来る。また、今回提案するCSRCAの制御対象であるヒューマノイドロボットの開発について述べる。

## 2. ヒューマノイドロボット

### 2.1 自由度配置

人間の動作は一般的なロボットと比較すると非常に汎用性があるが、人体の骨格系をそのままロボットへ導入することは、極めて複雑な動的システムが必要となり<sup>3)</sup>、現在の技術力ではハード、ソフトの両面において困難である。そこで、ヒューマノイドロボットの自由度配置を決定する際に、極力、少ない自由度で目的とする動きを実現できるように図1に示すように自由度を配置した。脚部は3次元空間内で2足歩行が出来るように、足首関節にロール、ピッチ軸の2自由度、膝関節にピッチ軸の1自由度、大腿部にヨー軸の1自由度、股関節にロール、ピッチ軸の2自由度の片脚6自由度を配置した。腕部は歩行時のバランスの維持、4足歩行(這う動作)が出来るように、肩にロール、ピッチ軸の2自由度、肘にピッチ軸の1

自由度、手首にピッチ軸の1自由度の片腕4自由度を配置した。頸部には、頭部の姿勢をロボットの姿勢と独立して与えることが出来るように、ロール、ピッチ、ヨー軸の3自由度を配置した。

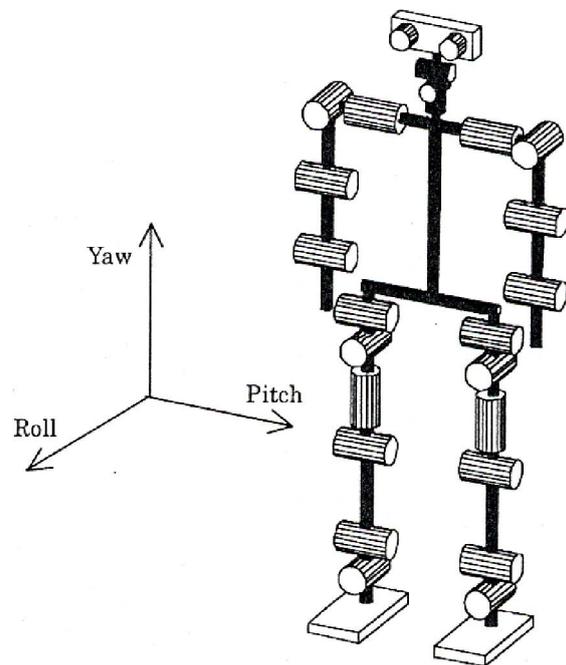


図1 自由度配置

### 2.2 ハードウェア構成

開発中のヒューマノイドロボットは、全長 1049 [mm]、全幅 522 [mm]、全重量 32.6 [kg]である。上半身の重量は 12.6 [kg]、下半身の重量が 20.0 [kg]である。ヒューマノイドロボットの各部の寸法を図2に、全身の写真を図3に示す。構造部は全てアルミ合金を使用し、軽量化のために細部に至るまで中抜きを行った。

頸部以外の各関節にはハーモニックドライブを配置し、動力はDCサーボモータからプーリとベルトを介して伝達される。頸部の各関節は遊星歯車が一体となったDCサーボモータを用い、さらに平歯車で減速している。各関節のモータの定格出力と減速比を表2に示す。また、各関節

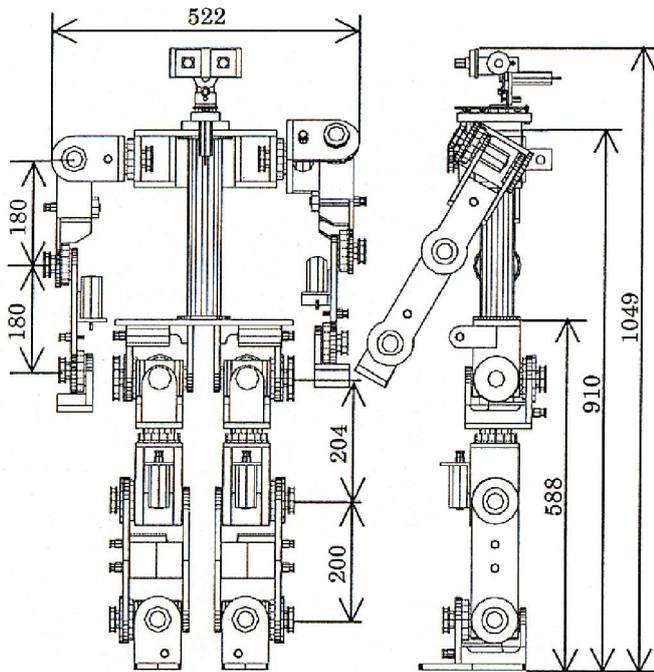


図2 ヒューマノイドロボットの寸法

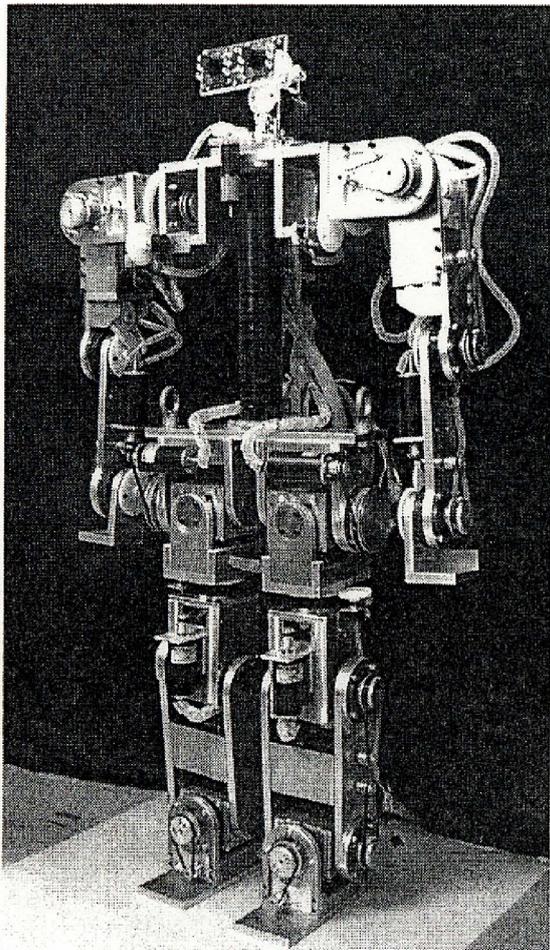


図3 ヒューマノイドロボットの写真

の回転角度はポテンショメータを用いて測定している。各関節の可動範囲を、表3, 4, 5に示す。システム構成を図4に示す。PC1からPC3はネットワークに接続されており、PC1がクライアントである。負荷分散のために、モータ、ポテンショメータ、ジャイロはPC2、CCDカメラは、PC3に接続している。

表2 各関節のモータ出力と減速比

	モータ出力	減速比
脚部, 腕部	90 W	333
首関節	3.5 W	128
首関節	3.2 W	157

表3 脚部(左脚)の可動範囲

	可動範囲(deg)
足首関節1	- 18.32 ~ +20.03
足首関節2	- 63.12 ~ +30.3
膝関節	- 80.0 ~ +27.51
腿関節	- 80.0 ~ +80.0
股関節1	- 18.32 ~ +20.03
股関節2	- 45.67 ~ +80.0

表4 腕部(左腕)の可動範囲

	可動範囲(deg)
肩関節1	- 80.0 ~ +80.0
肩関節2	- 25.0 ~ +158.4
肘関節	- 103.8 ~ +103.8
手首関節	- 106.7 ~ +82.0

表5 頸部の可動範囲

	可動範囲(deg)
首関節1	- 90.0 ~ +90.0
首関節2	- 90.0 ~ +90.0
首関節3	- 135.0 ~ +45.0

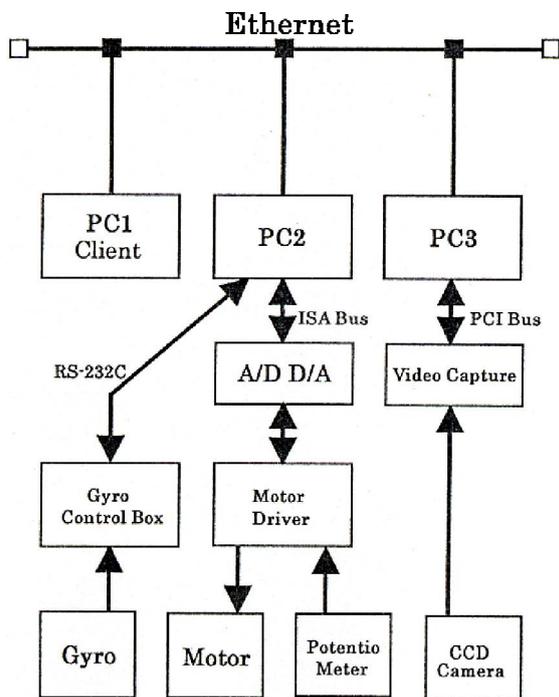


図4 システム構成図

### 3. 制御システム

#### 3.1 CORBA

CORBAはOpen Management Group(OMG)<sup>4)</sup>で規定を進めている、オブジェクト間のメッセージ交換を標準化した仕様である。CORBAの仕様は、プログラミング言語、プラットフォームに非依存である。すなわち、異なるプログラミング言語で開発されたプログラム間や、異なるOSで開発されたプログラム間の連携を、CORBAを用いて互いに通信させることで可能となる。

全てのプログラムはORB(Object Request Broker)を通じて通信が行われる。プログラミング言語、OSの違いは、このORBで吸収される。ORBとは、ネットワーク上に分散しているサーバ・オブジェクト(プログラム)に、クライアント・アプリケーションからの要求を配信

するソフトウェアである。図5にORBの概念図を示す。

CORBAは基本的にはクライアント/サーバモデルであり、機能の追加や削除をする場合、その機能をネットワークに接続または切断することによって、他のサーバプログラムの変更なしに、クライアントプログラムだけが追加された機能呼び出すように変更するだけでよい。

以上より、CORBAをヒューマノイドロボットの制御システムに組み込むことは、ロボットの各機能の連携、処理の分散という点で有効である。

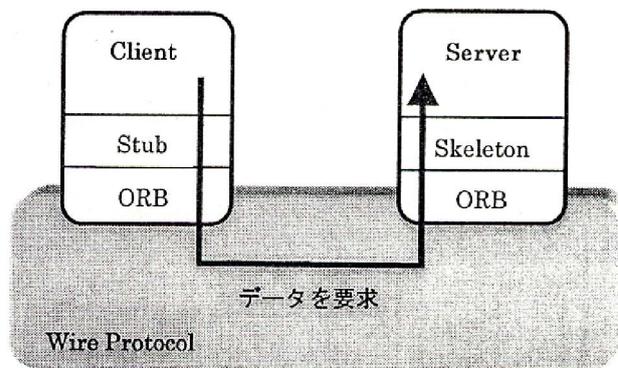


図5 ORBの概念図

#### 3.2 クライアント/サーバ・ロボットコントロール・アーキテクチャ

ヒューマノイドロボットの各機能の連携、処理の分散に有効な制御システムとして、CORBAを基礎にしたクライアント/サーバ・ロボットコントロール・アーキテクチャ(CSRCA)を提案する<sup>5)</sup>。

提案するCSRCAは1つのユーザインタフェースクライアントと、ヒューマノイドロボットの各機能ごとに八つのサーバが設けられている。提案するクライアントと各サーバを図6に、実装したものを図7示す。クライアントと各サーバ

の機能を以下に示す。

・ **The Data Transmission Control Server (DTCS)**

DTCS は全てのサーバとのインターフェースを持ち、各サーバが必要とする、または提供するデータの流れを制御する。

・ **Motion Description Server (MDS)**

MDS は、ロボットの各関節の目標軌道が保存されたデータベースから、モータの指令値(回転角度)を検索し DTCS へ提供する。

・ **Motor Movement Server(MMS)**

MMS は、MDS から DTCS を経由した指令値(回転角度)により、ロボットの各関節を駆動させるモータを制御する。

・ **Gyro Sensor Server(GSS)**

GSS は、ジャイロからの角度、角速度、角加速度データを DTCS へ提供する。

・ **Joint Angle Server(JAS)**

JAS は、ロボットの各関節の角度データを DTCS へ提供する。

・ **Stability Control Server(SCS)**

SCS は、各関節の指令値を MDS、ジャイロからのデータを GSS から DTCS 経由で受取り、指令値を現在のロボットの姿勢の安定に最適な値に補正する。

・ **Image Capture Server(ICS)**

ICS は、CCD カメラからの画像データを DTCS へ提供する。

・ **Image Recognition Server(IRS)**

IRS は、ICS から DTCS を経由した画像データを扱いやすいデータに処理し、そのデータを DTCS へ提供する。

・ **Client**

ユーザの要求するロボットの動作を DTCS へ出力し、センサから出力されるロボットの状態を DTCS からユーザに提供する。

### 3. 3 データの流れ

例として、現在実装されているサーバを用いたデータの流れを以下に示す。

1. Client が「walk」, 「turn」, 「stop」といったユーザの要求を DTCS へ送る。
2. DTCS がユーザからの要求に見合ったデータを MDS へ要求する。
3. MDS が要求されたデータを DTCS へ送る。
4. DTCS がジャイロデータを GSS へ要求する。
5. GSS が要求されたデータを DTCS へ送る。
6. DTCS が、受取った MDS と GSS のデータを SCS へ転送する。
7. SCS が受取ったデータを元に、最適なデータを求め DTCS へ転送する。
8. DTCS がそのデータを MMS へ転送する。
9. MMS が受取ったデータを元に、モータを駆動させる。

以上の手順で、ユーザの要求が実行される。

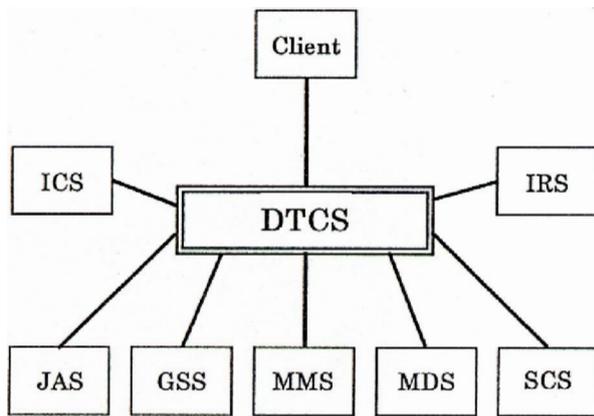


図6 CSRCA モデル

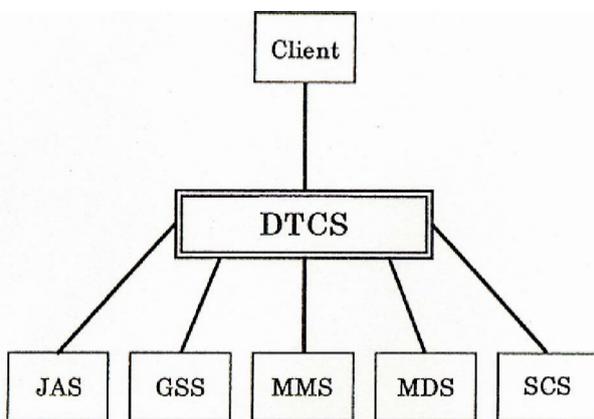


図7 実装した CSRCA モデル

## 4. 性能評価実験

### 4.1 実験環境

提案した CSRCA の性能を評価する。実験には 10Base-T LAN に接続した PC1(Celeron 433MHz, Linux 2.0.36), PC2(Celeron 266MHz, FreeBSD2.2.8R) の 2 台を使用する。PC1 に Client, DTCS を、PC2 に MMS, GSS, MDS, SCS, JAS を実装する。図 8 に実験環境を示す。

### 4.2 実験 1

ネットワークに接続された複数のコンピュータを使用して、ロボットを制御したときの時間遅れを評価する。

実験は、MMS, GSS と関節の回転角度

との遅れを評価する。ジャイロの傾斜角度とロボットの足首関節が同期して傾くようにプログラムして、ジャイロが傾いたときから、実際に関節が回転するまでの時間を計測した。実験結果を図 9 に示す。実験結果より、MMS と GSS の間の時間の遅れはほとんどない。これは、データサイズが 16 [Byte] しかないためだと考えられる。もし、画像データのようにデータサイズが大きく、PC 間の距離が長くなると時間遅れが大きくなることが予想されるため、今後改善しなくてはならない。

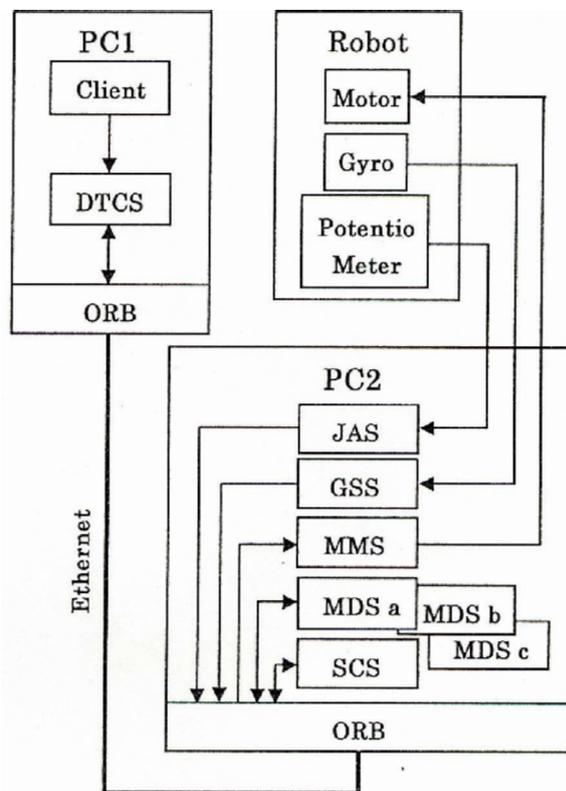


図8 実験環境(remote)

### 4. 3 実験 2

提案した CSRCA では、ヒューマノイドロボットを操作する時、「walk」, 「turn」, 「stop」のように動作によって MDS を切替える。この時に、サーバの切替えに要する時間が動作に影響を及ぼすことが考えられるので、サーバ切替えに要する時間を評価する必要がある。

実験 2 では、「walk」, 「turn」, 「stop」替わりに、直立姿勢から静歩行までの動作を、

- a: 直立姿勢から一歩目の最後
- b: 一歩目の最後から二歩目の最後
- c: 二歩目の最後から三歩目の最後

の動作に分割し、それぞれ MDS a, MDS b, MDS c とし、MDS a → MDS b → MDS c と切替えることにより、サーバ切替えにかかる時間を計測した。

ネットワークの接続による影響も調べるため、サーバの実行は図 10 のように 1 台の PC で CSRCA を構築した場合 (Local), 図 8 のように 2 台の PC で構築した場合 (Remote) の両方で行った。

軌道計算はオフラインで行い、各関節の制御はモータドライバの位置制御モードで行う。

図 11, 12, 13, 14 に実験結果を示す。実験結果より、Local のときのサーバ切替時間は約 3.0 [sec], Remote 時のサーバ切替時間は 3.5 [sec] となり、いずれも今回のような静歩行の場合には問題にならないが、動歩行のように時間に厳しい制御が要求される動作には問題がある。今後、受け渡すデータの量を少なくし、不足した情報を補間するなどの改善策が必要である。

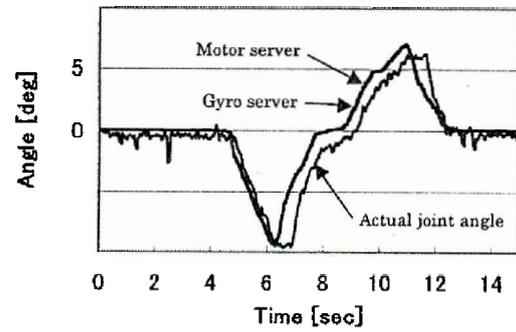


図 9 実験結果 1

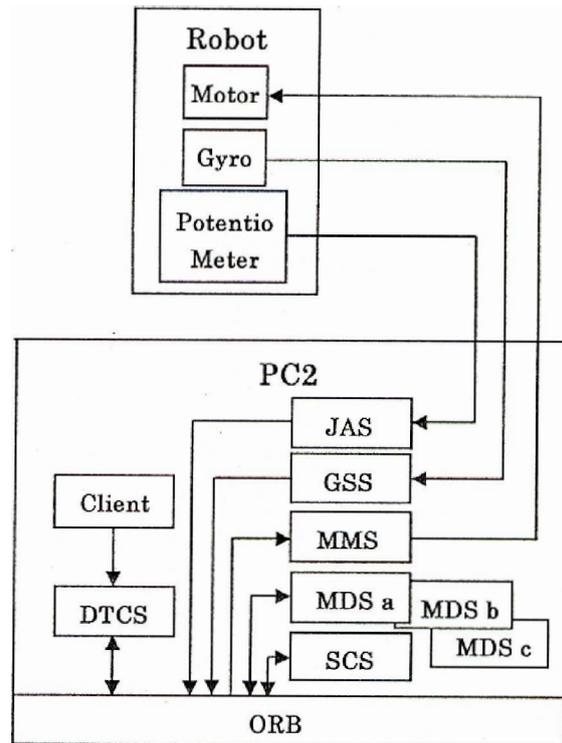


図 10 実験環境(local)

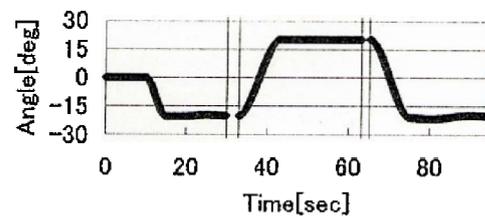


図 11 MMS data (local)

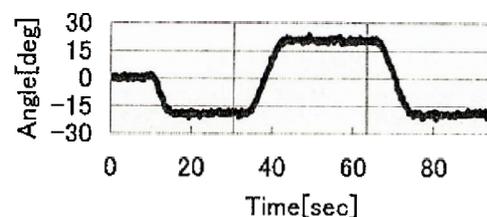


図 12 JAS data (local)

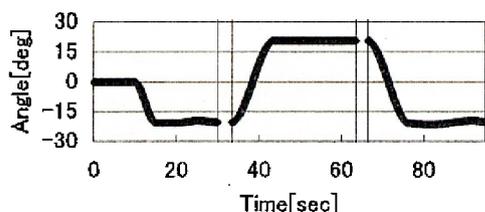


図 13 MMS data (remote)

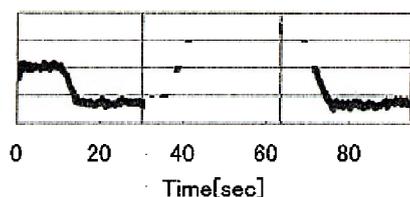


図 14 JAS data (remote)

## 5. おわりに

本研究では、CORBAに基づくCSRCAの提案、性能評価を行なった。

現在のところ、実行できる動作は静歩行のみであるが、今後、様々な動作が出来るように拡張していく。また、歩行の際に必要なZMP計測用の圧力センサの設置など、ハードウェアの拡張も行う予定である。

## 参考文献

- 1) Open Management Group,OMG's site for CORBA Success Stories, <http://www.corba.org/>.
- 2) T.J.Mowbrey and W.A.Ruh,Inside CORBA:Distributed Object Standardsand Applications,Addison Wesley,1997.
- 3) Miomir Vukobratovic(著), 加藤一郎 山下忠(訳):歩行ロボットと人工の足,

日刊工業新聞社, p9, 1975

- 4) Open Management Group, Object Management Group Home Page, <http://www.omg.org/>.
- 5) 武田謙郎, 那須康雄, バロリ レオナルド, チャピ ゲンツィ, 水戸部和久, CORBA を用いたエージェント指向ロボット制御プラットフォームに関する研究, IPSJ Symposium DICOMO2000, 2000, 7, pp37-42,2000