

ロボットの遠隔操作のための環境構築*

長 井 達 一 郎**
 楠 瀬 正 士***
 荒 牧 重 登**
 鶴 岡 知 昭**

Development of Environment for Remote Control of Robot Networked

Tatsuichiro NAGAI**, Masashi KUSUNOSE***, Shigeto ARAMAKI** and Tomoaki TSURUOKA**

Recently, it came to be able to connect various equipment with the network. However, only a computer connected directly with the cable can control the industrial-use robot of the old generation. Therefore, the range where the person was able to use the robot was being limited by cable of several meters. In this paper, we developed the system by which the user enabled the use of the robot through the network. This system is using the server-client method. This system enables two or more users to use the robot on an educational site etc. Moreover, this system can tell the work of one robot to another robot. As a result, two or more robots can do the same work at the same time.

Key Words : Remote Control, Industrial-use Robot, Server-Client Method, Blowfish Encryption Algorithm

1. はじめに

近年、ロボットの進化はめざましく、エンターテインメント性を取り入れた家庭用ロボットなどは超高機能でありながら安価に普及してきている。また、ロボットに限らずいろいろなコンピュータや機器はネットワークに接続され、さまざまな用途に活用されている。しかし、一般に、数世代前の産業用ロボットは直接ケーブルで繋がれたパソコンからしか制御できないため、ロボット制御の利用範囲がロボットから数m圏内に限られていた。

本研究では、ロボットのネットワークを介した利用を可能とし、ロボットの利用範囲を拡張することにある。通常の産業用に使用されるロボットは、機械部品の取り付けなど定型作業をするためのもので、基本的には単体

で動くものである。また、ロボットにはシリアルケーブルなどで直接接続されたロボット操作用コンピュータが必要になる。この状態では、ロボットのすぐそばにロボット操作用コンピュータが必要になるだけでなく、複数のロボットを操作する際に複数のロボット操作用コンピュータが必要になる。これでは、ロボットを利用できる物理的な範囲が限られてしまい、それにより利用用途にも制限が生じる。

そこで本研究では、ロボットに操作命令を送るコンピュータにサーバを構築し、サーバにロボットの操作命令を送るコンピュータに対してクライアントを構築することで、ネットワークを介したロボット操作を試みた。このようにネットワークを利用することにより、従来のロボットと従来のロボット操作用コンピュータの1対1の通信から、ロボット1台に対して複数のコンピュータでの操作、またその逆の複数のロボットを1台のコンピュータで操作することを可能とした (Fig. 1).

* 平成 21 年 6 月 25 日受付

** 電子情報工学科

*** 電子情報工学科：現在 株式会社富士通エフサス

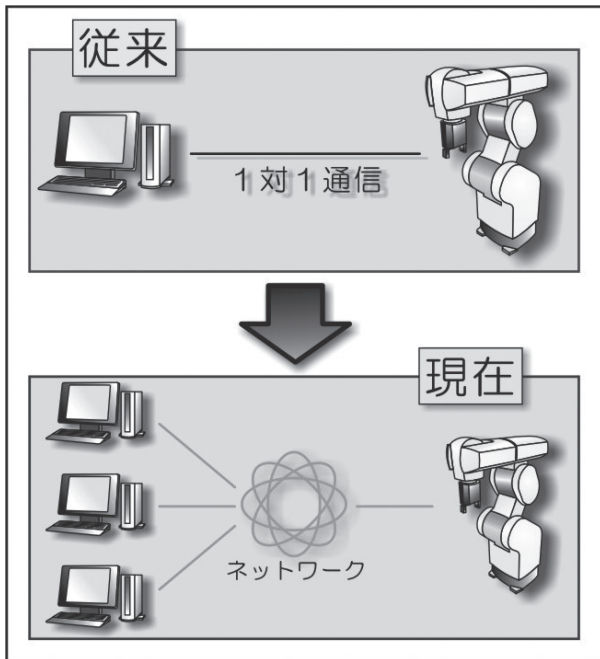


Fig. 1 システム概要

2. 産業用ロボットの概要

本研究では、三菱電機製の産業用ロボット (MoveMaster)RV-E2 と RV-1A を使用した。それぞれの特徴を以下に示す。

▽ RV-E2 の特徴

- 1) パソコンとの接続には、RS-232C ケーブルのみでしか接続できない。
- 2) 何もしない処理でも応答時間が長い。(最短でも 0.5 秒)。
- 3) 動作命令は、文字列で送る。
- 4) 基本的には、パソコンから緊急停止ができない。

▽ RV-1A の特徴

- 1) パソコンとの接続は、LAN ケーブル。
- 2) 特定の IP アドレスからの命令パケットしか受け付けない。
- 3) いつでもロボットの座標が取得可能。
- 4) ハンド部に力センサーがついている。
- 5) 動作命令は、RV-1A の仕様により MXTCMD 構造体パケット*で送る。
- 6) 基本的には、パソコン側から緊急停止はできない。

上記のロボット RV-E2 は RS-232C ケーブルでしか接続できず、また RV-1A についてはネットワークへの接続はできるものの特定の IP アドレスからしか動作命令パケットを受け付けない。このような産業用ロボットでは、利用範囲が制限されてしまっている。また、上記の仕様はハードウェア的な仕様が多く変更は難しいのが現状である。

そこで、本研究ではロボットの制御を行うパソコンに、サーバ/クライアント技術を用いて、各種サーバやクライアントを構築し、ロボットをネットワークに対応させるというソフトウェア的なアプローチを試みた。

3. ネットワークアプリケーションに関する概要

3.1 通信プロトコル

本研究で構築したネットワークアプリケーションに用いている通信プロトコルには主に UDP、一部 TCP (ロボットの仕様によるもの) を利用している。主に UDP で実装していた理由については、TCP に比べ信頼性は欠けるが、通信速度が速く、ネットワークアプリケーションへの実装が比較的容易であるためである。また、UDP の通信の信頼性に関しては、サーバ側に通信パケットのチェックを行う機構を実装することでカバーしている。

3.2 暗号化アルゴリズムの使用

一部のクライアント/サーバ間の通信において、通信パケットを暗号化して通信を行っている。その際、暗号化アルゴリズムとして Bruce Schneier 氏による Blowfish 暗号化アルゴリズムを使用している。Blowfish アルゴリズムの特徴を以下に示す。

- 1) 同じブロック暗号化アルゴリズムである DES より高速である。
- 2) フリーライセンスで、ソースコードが公開されている。
- 3) C++ による実装があり、暗号部分をオブジェクトとして簡単に利用できる。

3.2.1 Blowfish アルゴリズムの利用

この Blowfish アルゴリズムを利用して、通信パケットを暗号化してもそのキーが攻撃者に突き止められてしまうと、通信パケットの盗聴による通信パケット偽造や攻撃などが行われてしまう。そこで、ただ単に Blowfish アルゴリズムを使用した Blowfish オブジェクトを生成し、それを利用するのではなく以下のような利用時の工夫をしている。

- 1) クライアントの初回通信開始時にサーバ及びクライアントそれぞれで、Blowfish オブジェクトを7つ自動生成。
- 2) その7つの Blowfish オブジェクトを通信回数を元にローテーションさせて使用。
- 3) Blowfish オブジェクトを生成する際に必要なキーはクライアントからの初回通信時に時間(分)を元にして、ほぼ通信開始ごとに自動生成している。

以上のような3つの方法を組み合わせることで、暗号化のためのキーの推測を困難にすると共に、暗号化した通信パケットをそのまま送りつけるなどの攻撃による危険性の軽減などを行っている。

4. 構築したシステム

本研究では、従来の産業用ロボットにおける課題に対してソフトウェア的アプローチを行った。ロボットの特性やさまざまな利用用途を想定し、複数のサーバやクライアントを構築した。

4.1 遠隔操作クライアント

4.1.1 遠隔操作クライアント概要

遠隔操作クライアントは、上下、左右、前後などのボタンをクリックすると、中継サーバに操作命令パケットが送信され、遠隔にあるロボットの操作を可能とするクライアントアプリケーションである。また、ロボットの近くに設置された2台のWebカメラの映像を表示する。

4.1.2 機能

- 1) 操作命令パケットの生成、サーバへの送信 (スピード、移動距離)。
- 2) サーバに送る操作命令パケットの Blowfish アルゴリズムを利用した暗号化。
- 3) サーバへの送信ステータス及びサーバからの応答ステータスの表示。
- 4) ロボットの近くに設置した Web カメラからの画像の表示。
- 5) 緊急停止命令の送信機能。

4.1.3 動作画面

遠隔操作クライアントの動作画面を Fig. 2 に示す。

4.2 中継サーバ

4.2.1 中継サーバ概要

中継サーバは、クライアントからの操作命令パケットを受け取り、それをロボット操作命令に変換して、ロボット (RV-E2) を操作するためのサーバである。1 動作ご

とにロボット (RV-E2) の現座標を取得し、クライアントに返信する。

4.2.2 機能

- 1) クライアントから送られてくる Blowfish アルゴリズムで暗号化された操作命令パケットを復号化。
- 2) 操作命令パケットをロボット操作命令に変換し、ロボット (RV-E2) に送信。
- 3) ロボット (RV-E2) の現座標を取得し、クライアントに返信。
- 4) ロボット (RV-E2) の動作範囲制限。
- 5) 各種エラーをクライアントに送信。
- 6) 緊急停止命令の処理。
- 7) クライアント制限 (IP アドレスベース)。
- 8) ロボットの動作範囲の制限。

4.2.3 ロボット (RV-E2) の動作範囲制限

中継サーバでは、クライアントからの操作命令パケット内の情報をチェックしており、ロボット (RV-E2) の現座標と照らしあわせ、ロボット (RV-E2) がある動作範囲内でしか動作しないように制限をかけている (Fig. 3)。

- ・ X 座標...260 ~ 450
- ・ Y 座標...-380 ~ 350
- ・ Z 座標...20 ~ 540 (垂直方向)

4.2.4 緊急停止のしくみ

ロボット (RV-E2) の仕様により、操作用コンピュータから直接緊急停止ができない。そこで、中継サーバではソフトウェア的に緊急停止するための仕組みを実装した (Fig. 4)。

通常のクライアントからのロボット操作命令は、7777 番ポートで受け付けており、緊急停止命令は 9999 番ポートで受け付けるようにした。クライアント

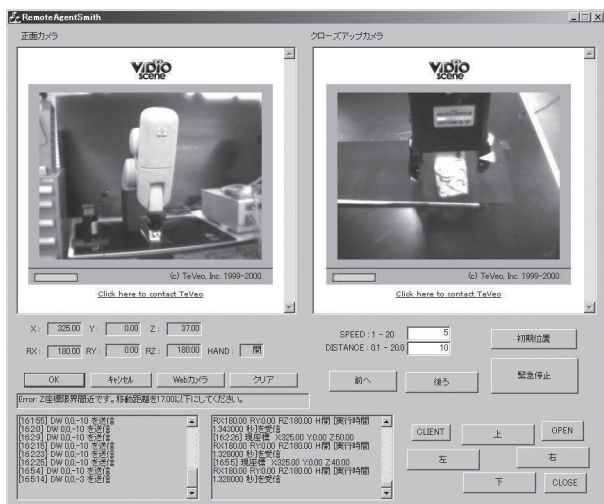


Fig. 2 遠隔操作クライアント動作画面

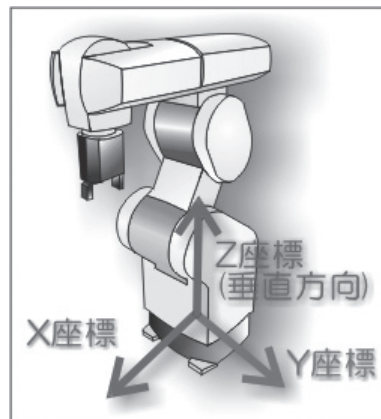


Fig. 3 ロボットの X,Y,Z 座標系

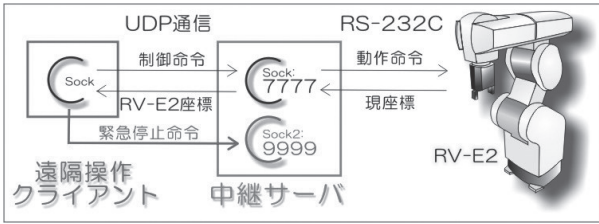


Fig. 4 緊急停止のしくみ

から 9999 番ポートに緊急停止命令 HALT を受信すると、7777 番ポートのソケットオブジェクトを Delete する。これにより、7777 番ポートにたまっている操作命令パケットはソケットとともに破棄され、ロボット (RV-E2) にロボット操作命令を送信しなくなる。これにより、操作命令パケットが破棄されることで実質的に実行前にたまっているロボット操作命令がキャンセルされる。

しかし、この緊急停止のしくみは完全なものではない。7777 番ポートのソケットオブジェクトと 9999 番ポートのソケットオブジェクトのどちらのソケットオブジェクトが、操作命令パケットが先に検出するイベントを発生するかによって、緊急停止命令をクライアントが送信しても、場合によって 1 命令は実行されてしまう。また、ロボット (RV-E2) の仕様により現在実行されている命令はキャンセルすることが出来ない。

4.2.5 動作画面

中継サーバの動作画面を Fig. 5 に示す。

4.3 ブリッジサーバ

4.3.1 ブリッジサーバ概要

ロボット (RV-1A) は、LAN ケーブルで接続可能であり、ネットワークを介したロボット操作に対応している。しかし、その操作命令パケットを特定の IP アドレスか

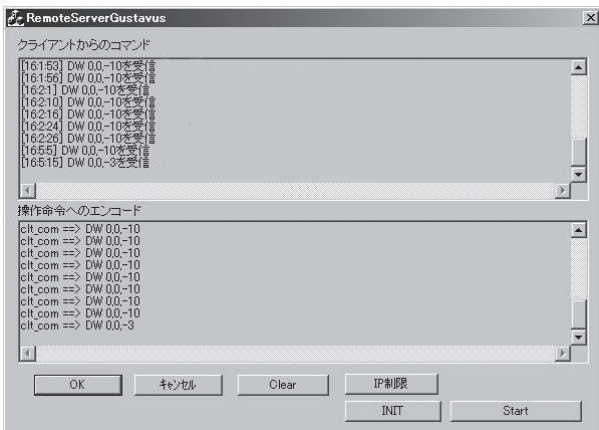


Fig. 5 中継サーバ動作画面例

らしか受け取らない仕様になっている。そのため、任意クライアントマシンからの動作命令パケットをロボット (RV-1A) で受け入れようとする、その任意クライアントが変わるたびにロボット (RV-1A) の設定を変更しなくてはならない。そこで、ロボット (RV-1A) の設定変更をしなくても、任意クライアントからロボット (RV-1A) を制御できるようにブリッジサーバを構築した。

4.3.2 機能

- 1) 任意クライアントからの動作命令パケットを受信し、ロボット (RV-1A) に送信する。
- 2) ロボット (RV-1A) からの応答パケットをクライアントに送信する。
- 3) ロボットに外部から取り付けられた力センサーの値を利用し、Z 軸下方向からの反力を 200 以下に維持。
- 4) ロボットの現座標を 5 秒おきに取得・表示。

4.3.3 動作命令パケットをブリッジするための通信オブジェクトの動作

ブリッジサーバは、2つのネットワークインターフェイスを備えているホストコンピュータを使用する。任意クライアントとブリッジサーバが通信するネットワークインターフェイスをインターフェイス1とし、ロボット (RV-1A) とブリッジサーバが通信するネットワークインターフェイスをインターフェイス2とする。

また、それぞれのネットワークインターフェイスで通信するソケットオブジェクトを以下の名前と呼ぶことにする (Fig. 6)。

- 1) インターフェイス1 → Sock1 オブジェクト
- 2) インターフェイス2 → Sock2 オブジェクト

まず、Sock1 オブジェクトで任意クライアントからの MXTCMD 構造体パケットを受け取る。次に、その MXTCMD 構造体パケットを Sock2 オブジェクトからロボット (RV-1A) に送信する。ロボット (RV-1A) の動作が完了すると、応答パケットが返ってくるので、その応答パケットをクライアントに返信する。基本的には、この動作の繰り返しとなっている。なお、ロボットに対して同一 IP アドレスから送信されたように見せかけるために、MXTCMD 構造体パケットの中身は基本的にはリレーしているだけである。

4.3.4 力センサーによる衝突制御

ロボット (RV-1A) には、外付けで力センサーが取り付けられている。この力センサーの値を利用し、ブリッジサーバは物体との衝突時の制御機能を実装した。1 通信ごとに力センサーの Z 軸上 (垂直) 方向値を読み取り、その値が 200 以上になると、任意クライアントからの操作命令に関わらず少し上に上げるように操作命令を発行して 200 以下になるように圧力調整を行っている。

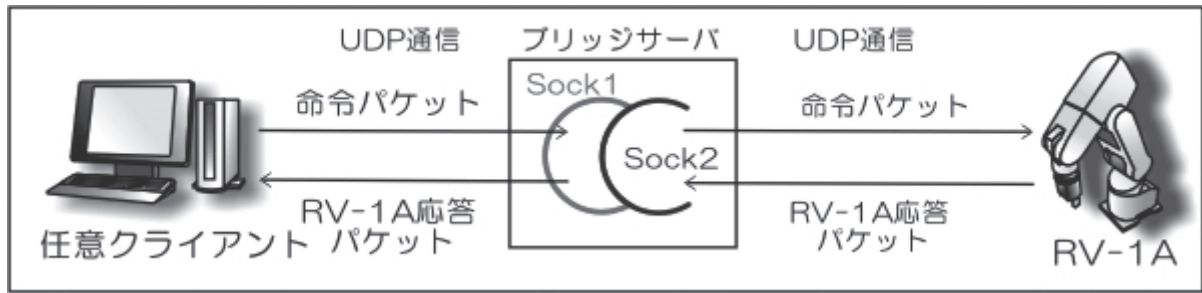


Fig. 6 ブリッジの仕組み

実際にロボットを下向きに操作し続けるように操作命令を送り続けた状態で、厚さ約 2cm ほどのスポンジを置いてそれに衝突させると、スポンジをつぶすことなくスポンジに接触した状態でアイドル状態となる。

4.3.5 動作画面

ブリッジサーバの動作画面を Fig. 7 に示す。

4.4 モニタリングエージェント

4.4.1 モニタリングエージェント概要

一定間隔で、ロボット (RV-1A) の現座標を取得し、前時間に取得したロボット座標との差分より生成した操作命令パケットを中継サーバに送信するモニタリングエージェントである。

4.4.2 機能

- 1) 5 秒おきにロボット (RV-1A) の現座標取得。
- 2) 現座標と前時間のロボット座標との差から生成したロボット制御命令の中継サーバへの送信。
- 3) ロボット (RV-1A) との通信は、TCP を使用。
- 4) 中継サーバとの通信には、UDP を使用。
- 5) 緊急停止機能。

※モニタリングエージェントでは、通信パケットの暗号化は行っていない。

4.4.3 ロボット (RV-1A) の現座標取得

ロボット (RV-1A) のコントローラは、一種のサーバのようになっていて、2つのポートが開いている。

- 1) 通常の動作命令パケットを受け付ける 10000 番ポート (UDP)
- 2) モニタリング用パケットを受け付ける 10001 番ポート (TCP)

ロボット (RV-1A) のコントローラをサーバと見ると、ロボット動作とモニタリングのためのパケットは干渉しない事が分かる。これにより、TCP10001 番ポートを使用することで、いつでもロボット (RV-1A) の現座標などは取得可能である。ただし、ロボット (RV-1A) 応答には 0.5 秒かかる。

4.4.4 動作画面

モニタリングエージェントの動作画面を Fig. 8 に示す。

5. 実験

5.1 ロボットの遠隔操作実験

5.1.1 概要

遠隔操作クライアント及び中継サーバ、ロボット (RV-E2)、Web カメラ 2 台を用いて、遠隔操作の実証実験を行った (Fig. 9)。実験内容は、遠隔操作クライアントから中継サーバを経由して動作命令を送り、単純な動

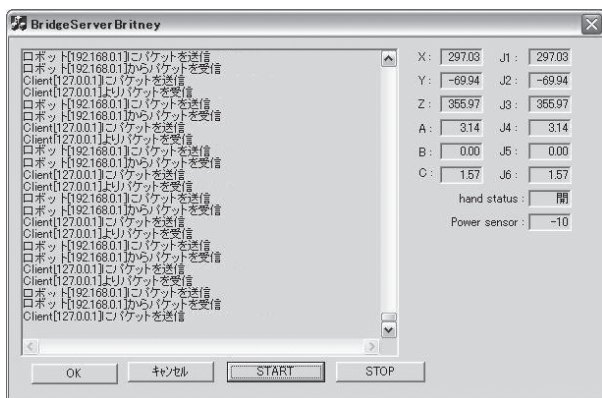


Fig. 7 ブリッジサーバ動作画面

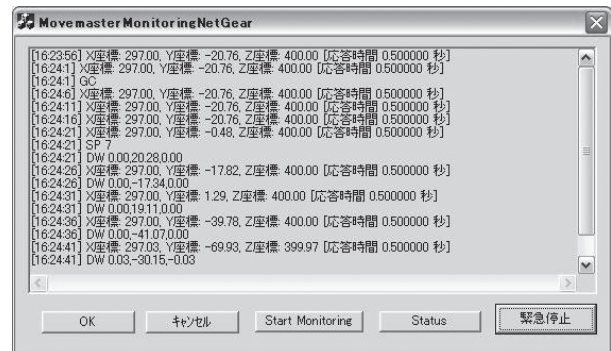


Fig. 8 モニタリングエージェント動作画面



Fig. 9 ロボットの遠隔操作実証実験

作で小さい箱をロボット (RV-E2) のハンドでつかむものである。

5.1.2 結果

結果としては、動作自体には問題はなく、遠隔操作クライアントからの操作だけで箱をつかむことに成功した。

5.1.3 考察及び問題点

遠隔操作クライアントからロボット (RV-E2) を制御し、箱をつかめただけでなく、床付近になったときの衝突防止機能が意図した通り正常に動作した。

しかし、第一の問題点としては、現在の構成では Web カメラ 2 台 (正面から全体を映すカメラと対象物付近を移すクローズアップカメラ) を用いて実証実験を行ったわけだが、結論から言うと Web カメラの数が圧倒的に足りない。Web カメラが 2 台だけでは、ハンド部分などに死角が出来てしまい、今回のような障害物のない実験では成功したが、実際は予期せぬ対象物体との衝突も考えられる。解決方法としては以下のようなことが考えられる。

- 1) Web カメラの台数を増やす。
- 2) Web カメラではなくもっと高密度で鮮明な高性能カメラからの画像を遠隔操作クライアントで表示できるようにする。
- 3) 物体追跡機能のついたカメラを用いて、ロボット (RV-E2) のハンドの先を追跡させる。

次に、動作遅延の問題。人がボタンをマウスでクリックする速度は、大体 1 秒あたり 4 クリックしたと考えても、1 クリックあたり 0.25 秒。ロボット (RV-E2) の動作には、1 動作命令あたり最低でも 0.5 秒はかかってしまうため、圧倒的に人が制御命令を中継サーバに送る速度の方が早い。動作完了を待たずにボタンを押しすぎたりすると、中継サーバに操作命令パケットが確実にたまってしまいます。また、現状では次のことが遠隔操作クライアントからは分からない。

- 1) どのくらい制御命令を送っているか
- 2) またどのくらい制御命令が中継サーバにたまっている

- か
- 3) 現在実行されている動作命令も含めすべての動作命令を実行するとロボットはどのような状態になるのか
- 上記の 3 つことが分からないため、誤って多くボタンをクリックしてしまったりして、緊急停止が遅れると対象物に衝突してしまう危険性がある。

5.2 ブリッジ構成実験

5.2.1 概要

ロボット (RV-1A) は LAN ケーブルでの接続が可能であるが、特定の IP アドレスからしか命令動作パケットを受け取らない。このハードウェア的な制限をソフトウェア的に解決するには、動作命令パケットをブリッジするブリッジサーバのようなものが必要となる。ブリッジ構成実証実験では、既存の任意クライアントプログラム、ブリッジサーバ、ロボット (RV-1A) を用いての実証実験を行った (Fig. 10)。

5.2.2 結果

結果としては、任意のクライアントプログラムからロボット (RV-1A) に動作パケット送り、意図したとおりの動作をさせることができた。

また、ある動作をロボット (RV-1A) にさせるのに、直接ロボット (RV-1A) に動作パケットを送信して動作させた場合とブリッジサーバ経由での動作時間を測定し、結果は以下ようになった。

- ・直接制御の場合 27.35 秒
 - ・ブリッジサーバ経由 28.85 秒
- 差：0.5 秒程度

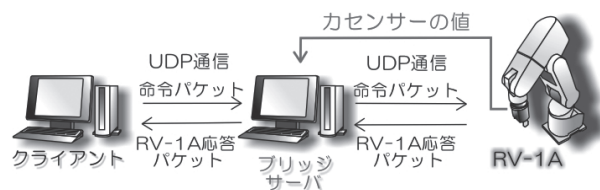


Fig. 10 ブリッジ構成実証実験

5.2.3 考察及び問題点

本実証実験により、ロボット (RV-1A) をブリッジサーバを介して利用することで、任意クライアントからの利用が可能になり、ハードウェア的制限をソフトウェアで拡張することができた。

問題点として、ログの表示の問題がある。クライアントプログラムからの動作命令パケットの送信間隔があまりにも狭い上、クライアントから1動作命令パケットを受信すると、ブリッジサーバでは以下の4つのログを出力する。

- 1) ブリッジサーバがクライアントから動作命令パケットを受信したこと。
- 2) ブリッジサーバからロボット (RV-1A) に動作命令パケットを送信したこと。
- 3) ロボット (RV-1A) からブリッジサーバが応答パケットを受信したこと。
- 4) ブリッジサーバから3)の応答パケットを送信したこと。

本実証実験においては、このログ表示に関する動作に及ぼす影響についての実験も行っている。以下に実験結果を示しておく。

- ・ログ表示桁数無限 50.04 秒
- ・ログ表示桁数 100 桁 34.04 秒

この結果からブリッジサーバは、GUIでの実装に向いていないのかもしれない。

5.3 ロボット動作のマスター・スレーブ制御実験

5.3.1 概要

ロボット動作のマスター・スレーブ制御実証実験では、モニタリングエージェントと中継サーバ、ロボット2台 (RV-1A → マスタ, RV-E2 → スレーブ) を用いて実験をした (Fig. 11)。また、本実証実験では、特に以下の点について実験を行った。項目番号は、そのまま実験1～4を意味する。

- 1) ロボット (RV-1A) の現座標を取得するまでにかかる時間
- 2) ロボット (RV-1A) を制御しているパソコンからロボット (RV-E2) を制御しているパソコンまでのネットワ

ーク応答時間

- 3) ロボット (RV-E2) が動作を完了に要する最大時間
- 4) マスタ・スレーブ制御動作の確認

実験1～3は、マスター・スレーブ制御で最大の問題になるモニタリングエージェントがロボット (RV-1A) の現座標を確認する時間の間隔の最適値を求めるための実験である。また、実験2においては、エコーバックサーバ及び遠隔操作クライアントを用いて測定した。

前提条件を以下に示す。

- 1) ロボット (RV-1A) の動作速度を20とする。
- 2) ロボット (RV-1A) はティーチングボックスでランダムな動作を行う。
- 3) ロボット (RV-E2) の動作速度を5～10とする。
- 4) 実験3に関しては、安全を考え実験時の仮の値として5秒間隔とした。
- 5) 実験4は、実験1から3の結果を用いて、再度モニタリングエージェントの最適なロボット (RV-1A) の現座標確認時間間隔を決定し行った。

なお、時間の測定には、動作前後の時間をC言語標準のclock()関数を用いて、ms単位で行った。

5.3.2 結果

まず、実験1のロボット (RV-1A) の現座標取得にかかる時間は、0.5秒であることが分かった。

実験2においては、遠隔操作クライアントより制御命令を0.5秒間隔で2000回トライした。結果は、ほとんどが0.000秒 (= 0.001秒以下)であり、時折0.015秒、0.031秒となった。

実験3では、ロボット (RV-E2) の最低応答時間は約0.5秒。複数回実験を行い、ロボット (RV-1A) の動作速度20で、モニタリングエージェントのロボット (RV-1A) 現座標確認時間間隔を5秒としたときのロボット (RV-E2) の動作速度の最適値は、7であると考えられる。ロボット (RV-E2) の動作速度が7より遅いと、ロボット (RV-1A) を連続して動作させると、中継サーバに動作命令がたまってしまい、動作速度が7以上だとロボット (RV-E2) の動作速度が早すぎて、移動距離が短すぎると停止する際にガクガクと振動してしまうことが分かっ

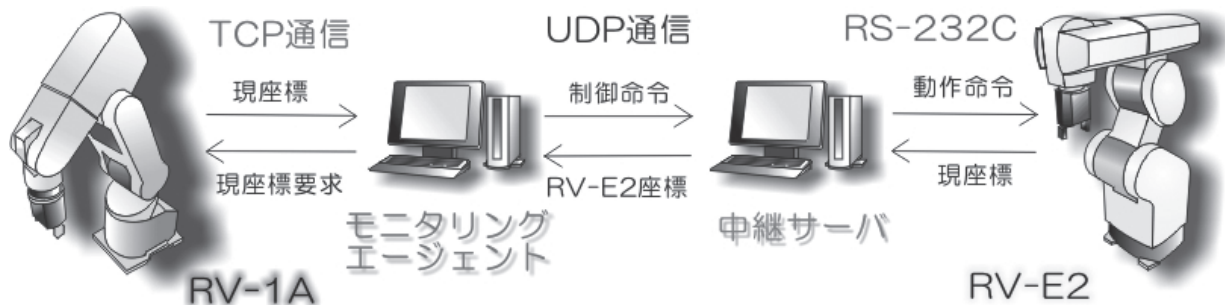


Fig. 11 ロボットのマスター・スレーブ制御実証実験

た。また、ロボット (RV-1A) の動作速度 20 において連続した移動を行った際、ロボット (RV-E2) の動作速度 7 での最大実行時間は 4.7 秒ほどであることが分かった。

ここまでの実験 1～3 を整理すると以下のようになる。

- 1) モニタリングエージェントがロボット (RV-1A) の現座標確認に要する時間は、約 0.5 秒。
- 2) ロボット (RV-1A) を制御するパソコンからロボット (RV-E2) を制御するパソコンまでのネットワーク応答時間は、約 0.001 秒～0.031 秒。
- 3) ロボット (RV-1A) の動作速度が 20 で連続移動したとき、ロボット (RV-E2) の動作速度 7 での最大実行時間は、約 4.7 秒。

以上の実験 1～3 の結果より、1 サイクルにかかる時間は約 1～5.2 秒ほどであることが分かった (Fig. 12)。よって、実験 4 ではマスタ・スレーブ制御エージェントのロボット (RV-1A) 現座標確認時間を 5 秒とした。

最後に、実験 4 で通してのロボット動作のマスタ・スレーブ制御実証実験を行ったわけだが、結果はほぼ滞りなく同じ動作をさせることに成功した。

5.3.3 考察及び問題点

本実証実験 1～3 では、ロボット (RV-1A) の動作速度に対するモニタリングエージェントのロボット (RV-1A) の現座標確認時間、ロボット (RV-E2) の動作速度の最適値を得ることができた。また、実験 4 での実際の実証実験で意図した通りに動作し、これによりネットワークを介してのロボット動作のマスタ・スレーブ制御が可能となった。しかし、現状のモニタリングエージェントのロボット (RV-1A) 動作を監視する上で問題がある。それは、ある一定時間間隔で現座標の確認しかしておらず、ロボット (RV-1A) の動作をトラッキング (追尾) しているわけではない。よって、ある時間にロボット (RV-1A) がある座標にロボット (RV-E2) を動かすことしかできない。すると、次のような問題が出てくる。

- 1) 動きが斜めにショットカットされてしまう (Fig. 13)。
- 2) 最大で約 10 秒 (現座標確認時間間隔+ロボット (RV-

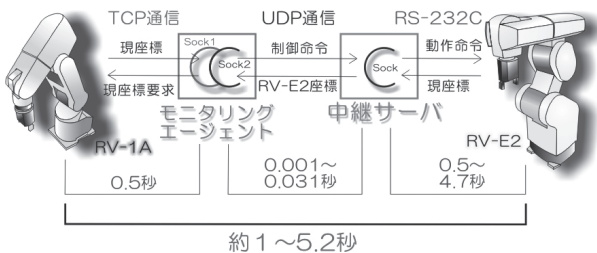


Fig. 12 マスタ・スレーブ制御実証実験 (速度)

E2) の動作時間) の動作遅延が生じてしまう。

1) に関しては、Fig. 13 に示す例では 1 現座標確認時間間隔内にロボット (RV-1A) が、A → B → C → D と移動したとき、モニタリングエージェントは D の座標しか取得しないため、ロボット (RV-E2) の動きは、A → D という動きになってしまう。このようなことが起きると、結果的には D という同じ位置に移動するわけだが、A と D の直線上に何か物体があつたりすると大変危険な状態になる。

2) の問題は、ロボット (RV-1A) に対する現座標確認間隔 (5 秒) とロボット (RV-1A) が最大移動動作距離をロボット (RV-E2) で動作にかかる最大時間 (約 5 秒) に合わせて、最大で 10 秒の動作遅延が生じる問題である。更なる実証実験を重ね、ロボット (RV-1A) の動作速度に対するロボット (RV-E2) の動作速度、モニタリングエージェントのロボット (RV-1A) の現座標確認時間間隔をチューニングすることで、動作遅延はある程度小さくすることができるがリアルタイム制御を考えなければ特に問題ではないと考えている。

6. 利用構成例

本研究では、4 章で示した複数サーバやクライアントを構築した。それによりさまざまな利用が可能となった。その想定される使用用途に応じた構成例を以下に示す。

6.1 ロボットの遠隔操作構成例

6.1.1 ロボットの遠隔操作構成例概要

以下に示すロボットの遠隔操作構成例は、クライアントマシンからネットワークを介して、ネットワークの向こう側にあるロボットを操作するものである。

これにより、操作クライアントはネットワークを介してロボットを操作できるようになった。

6.1.2 構成図

ロボットの遠隔操作構成例の構成図を Fig. 14 に示す。

6.1.3 説明

- 1) 遠隔操作クライアントで上下、左右、前後などのボタ

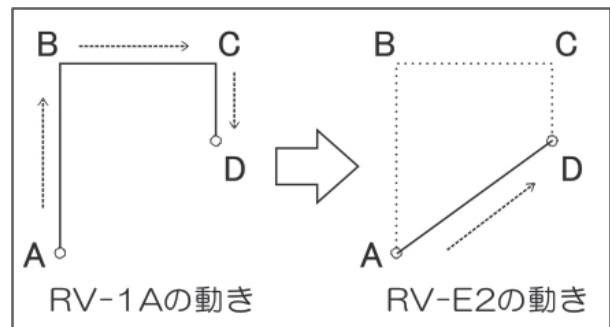


Fig. 13 動作のショートカット

ンが押されると、スピードと移動距離を設定し、操作命令パケットが生成される。

- 2) クライアントからサーバに送られる操作命令パケットは Blowfish アルゴリズムによって暗号化され中継サーバへと送信される。
 - 3) 中継サーバでは、遠隔操作クライアントから受信した操作命令パケットを復号化する。
 - 4) 前回の通信時に取得したロボット (RV-E2) の現座標とロボット操作命令のチェックを行う。
 - 5) 4) で指定された操作が安全であると判断されると、ロボット操作命令 (MP 命令) に変換されロボット (RV-E2) へ送られる。
 - 6) ロボット (RV-E2) は動作が終わると、中継サーバはロボット (RV-E2) の現座標を取得する。
 - 7) 取得された現座標は、遠隔操作クライアントに返される。
 - 8) 遠隔操作クライアントでは、中継サーバから受信したロボット (RV-E2) の現座標をモニタリング表示する。
- 基本的には、上記のようなプロセスが繰り返される。また、中継サーバによる緊急停止なども可能である。

6.2 ブリッジ構成例

6.2.1 ブリッジ構成例概要

ロボット (RV-1A) は、LAN ケーブルで接続可能であり、ネットワークを介した操作に対応しているが、特定の IP アドレスからしか動作命令パケットを受け付けない。ブリッジ構成では、ブリッジサーバを介することで任意のクライアントからロボット (RV-1A) を操作できるようにするものである。動作としては、ルータに似た動作となる。

ブリッジサーバは、任意クライアントからのロボット (RV-1A) の操作用の MXTCMD 構造体パケットをそのまま中継している。よって、任意クライアントの使用に IP アドレス以下の変更を加えることなく、任意のクライアントマシンからロボット (RV-1A) を操作が可能となった。

6.2.2 構成図

ブリッジ構成例の構成図を Fig. 15 に示す。

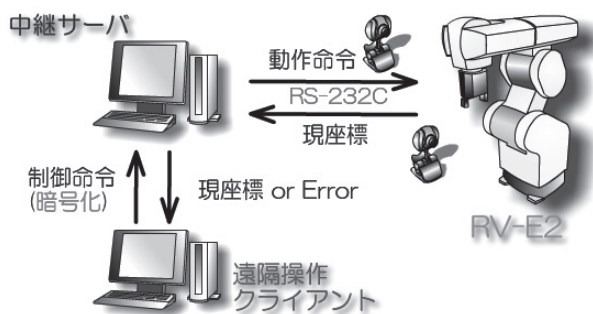


Fig. 14 ロボットの遠隔操作構成例

6.2.3 説明

ブリッジサーバには、2つのネットワークインターフェイスがあり、それぞれのネットワークインターフェイスは、Sock1 オブジェクト (172.16.1.17) 及び Sock2 オブジェクト (192.168.0.2) で利用する。

- 1) まず、任意クライアントから命令パケットを Sock1 オブジェクトに送信する。
- 2) ブリッジサーバは、その動作命令パケットをブリッジして Sock2 オブジェクトからロボット (RV-1A) に動作命令パケットを送信する。
- 3) ロボット (RV-1A) の動作が完了すると、ロボット (RV-1A) から応答パケットが返ってくる。
- 4) このときブリッジサーバでは、力センサーの値を取得する。
- 5) ロボット (RV-1A) からの応答パケットは任意クライアントに返される。

もし、4) で下方において何かに衝突しその反力が一定値 (上向きに 200 以上) あった場合、以降の任意クライアントからの動作命令パケット内の Z 座標などを調整し、反力が一定値以下になるように移動距離 (圧力) 調整が行われる。

6.3 ロボット動作のマスター・スレーブ制御構成例

6.3.1 ロボット動作のマスター・スレーブ制御構成例概要

モニタリングエージェントと中継サーバを組み合わせることで、ロボット (RV-1A) をマスター、ロボット (RV-E2) をスレーブとみて、ロボット (RV-1A) の動作を、ネットワークを介して向こう側にあるロボット (RV-E2) で再現する動作が可能となった。

6.3.2 構成図

マスター・スレーブ制御構成例の構成図を Fig. 16 に示す。

6.3.3 説明

モニタリングエージェントでは、2つのソケットオブジェクトを生成して利用している。Sock1 オブジェクトは TCP 通信オブジェクト、Sock2 オブジェクトは UDP

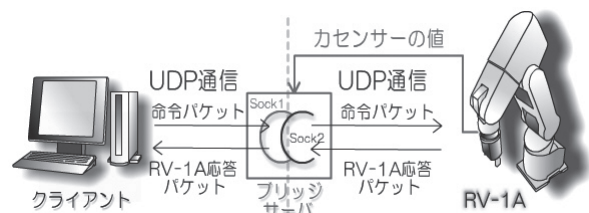


Fig. 15 ブリッジ構成例

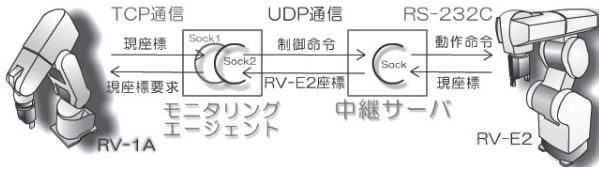


Fig. 16 マスタ・スレーブ制御構成例

通信オブジェクトである。

- 1) 5秒おきにモニタリングエージェントは、ロボット (RV-1A) の現座標を取得する。
- 2) 前時間に取得したロボット (RV-1A) の現座標との差を取り、差分制御命令 (DW 命令) を生成し、中継サーバへ送信する。
- 3) 中継サーバでは、モニタリングエージェントからの制御命令を動作命令に変換して、ロボット (RV-E2) へ送信する。
- 4) ロボット (RV-E2) の動作が完了すると、ロボット (RV-E2) の現座標をモニタリングエージェントに返信する。

以上、1)～4) までの動作を繰り返すことで、ロボット (RV-1A) の動作をネットワークを介して向こう側にあるロボット (RV-E2) に動作の再現をさせることができる。

また、モニタリングエージェントが5秒おきに現座標を取得するのは、制御用のポートとは異なるので、いつでも取得可能である。そのため、ロボット (RV-1A) の動作はネットワーク経由でもティーチングボックスでも可能である。

6.4 応用例

6.4.1 応用例概要

前述のマスタ・スレーブ制御とブリッジサーバを組み合わせることで、ネットワークを介した広域でのロボット操作が可能になった。また、この構成により、複数の人が場所を移動することなくロボットを操作できるだけでなく、遠隔から操作するロボットの動作をネットワークの向こう側にある複数のロボットに動作を再現させることが可能となった。

6.4.2 構成図

応用例の構成図を Fig. 17 に示す。

6.4.3 説明

- 1) まず、任意クライアントからブリッジサーバを介して、ロボット (RV-1A) を操作する。
- 2) ロボット (RV-1A) の動作はモニタリングエージェントに監視されているので、前時間とのロボット (RV-1A) の座標と現座標から制御命令を生成し、中継サーバへ送信する。
- 3) 中継サーバはモニタリングエージェントからの制御命令から動作命令に変換し、ロボット (RV-E2) を動作させる。

構成図中にある3つの四角のエリアにあるものは、それぞれネットワークに繋がれていれば、世界中のどこにあってもかまわない。

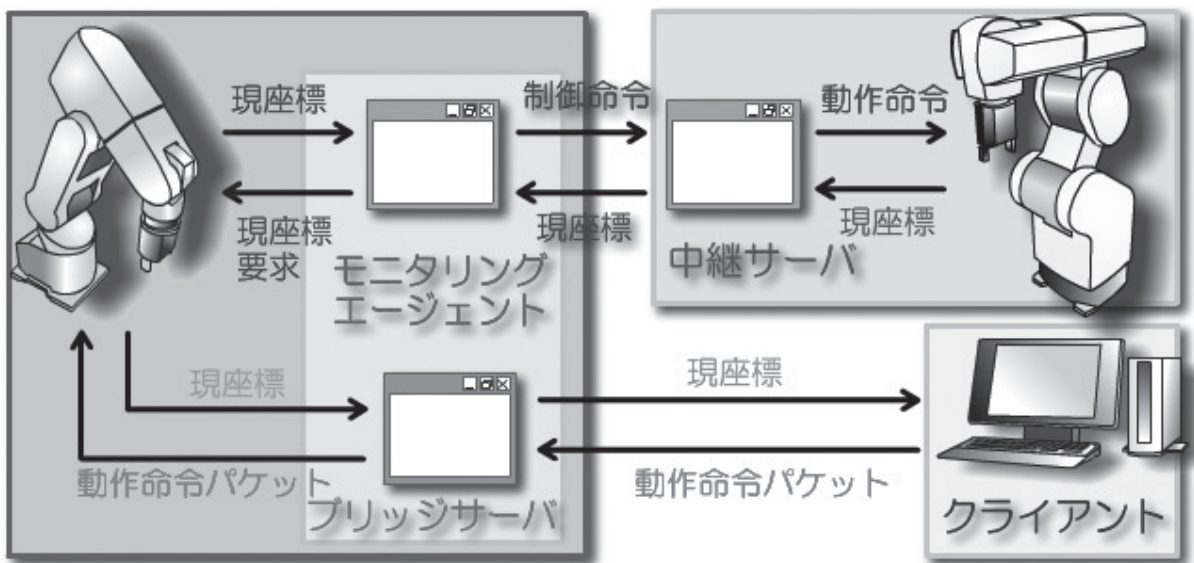


Fig. 17 応用例

7. おわりに

本研究では、ネットワークを介したロボットの利用をあまり考慮されていない産業用ロボットのハードウェア的な制限を、ロボットに直接繋がっているロボット制御用パソコン上に、各種サーバやクライアントを構築するというソフトウェア的なアプローチによりネットワークを介した利用を可能とした。また、これまでロボットに直接接続されたロボット制御用コンピュータからしかロボットを制御することが出来なかったが、あらゆる利用構成を想定し、その用途に応じた各種サーバやクライアントを構築したことで、ロボットの操作はネットワークに対応させた。

高速なインターネット回線(インフラ)が整備され、その利用料も安価となった現代では、パソコンのみならず家電までもがネットワークに接続され、様々なものが電子化され、ネットワークを介して集中管理されるようになってきている。産業用ロボットもネットワークに繋がることで、それ自体がネットワークの一要素とすることができた。しかし、今回使用した産業用ロボットだけでなく、センサーネットワークへの接続などを行うことでさらに利用範囲を拡大できると考えられる。しかし、本研究で使っている操作命令パケットは産業用ロボット(MoveMaster 専用)の仕様になっているので、センサーネットワークなどの拡張を視野に入れると新たに汎用的な通信プロトコルを策定し、ロボット操作パケットを汎用化する必要となってくるだろう。

謝 辞

本研究を行うにあたり、ロボットの操作の際に利用するライブラリを作成してくださった本学卒業生(現:東芝情報システム株式会社)の松尾英氏、そして、高速で素晴らしい Blowfish 暗号アルゴリズムをフリーライセンスで提供していただいている Bruce Schneier 氏に感謝します。

参 考 文 献

- [1] Napper, Lewis: 「Winsock2 プログラミング - Windows Sockets API によるネットワークプログラミングのすべて (WinSock 2.0)」, ソフトバンクパブリッシング, 2005.01
- [2] ブルース・シュナイアー / 著 山形浩生 / 監訳 安達真弓 / [ほか] 訳: 「暗号技術大全」, ソフトバンククリエイティブ, 2003.6
- [3] 「MoveMaster(RV-E2 及び RV-1A) の仕様書・マニュアル一式」, 三菱電機(株)
- [4] 「Intelligent Force Sensor(レシーバボード) の仕様書・マニュアル一式」, ニッタ株式会社
- [5] <http://www.schneier.com/blowfish.html> (The Blowfish Encryption Algorithm)