

リアルタイム性の高いネットワークの開発とロボット制御への応用

Development of a Network with High Real-time Performance and Its Application to Robot Control

内村 裕 大西 公平¹⁾

要 約

ネットワークを介してロボットや機械システムなどを制御する場合、データ伝送に伴う遅延が制御系全体の安定性を損なう恐れがある。ネットワーク処理の遅延は、伝送路上での遅延と OS を含むコントローラ内部処理による遅延に分類されるが、本研究では後者について注目し、遅延時間を最小化する仕組みを考案し遅延を低減した。具体的には、OS 内部のネットワーク処理部分のアーキテクチャーを改良し、従来は優先度の低いタスクとして扱われていたネットワーク処理をリアルタイムタスクとして扱い、制御系とネットワーク系のいずれの優先度も同等に処理することで、システム全体のリアルタイム性を向上した。また、開発したネットワークのネットワーク機構を、力覚をオペレータに伝えるバイラテラルロボットに実装し、遠隔操作の操作性の改善を実現した。

目 次

- はじめに
- バイラテラルロボットの構成
- リアルタイムネットワークの開発
- バイラテラルロボットによる実験結果
- おわりに

はじめに

コンピュータネットワークの急激な普及により、様々なアプリケーションがインターネットをはじめとするネットワークインフラ上で実現されている。しかしながら、現在ネットワーク上で提供されるほとんどのサービスは情報のみによるサービスである。つまり、利用者は、スクリーン上のテキストあるいはイメージを閲覧することによってサービスを提供されている。たしかに、これらの情報サービスは大変有用なものであるが、現実の世界におけるサービスとは遊離したものであることは否めない。

現実の世界では、人間を含む多くのシステムが、物理的な接触、変形及び運動といった動的な作用を相互に及ぼしている。例えば、何かに触れば、物は変形、移動し、さわった人は、重量や摩擦、弾性力のような触覚の印象を受け取る。こうした（純粋な情報を越えた）物理的なサービスを提供することは、次世代のネットワークにおいて、取り組むべき課題の一つであると認識されている。

一方、近年のモーション・コントロール技術の進展によって、実際の世界における相互作用を提供するサービス実現の可能性が見いだされつつある。こうした可能性の一つがネットワークを介した遠隔操作システムである。ネットワークを介した遠隔操作は、実際の世界の本質的な相互作用を再現するアプローチであり、遠く離れた

対象物に触れ、動かし、操作することを可能にするものである。

遠隔操作における操作性を向上するには、触覚のフィードバックが有効であるが、ネットワークの通信遅延が大きいと制御系が不安定になる問題があり、対策が必要であった。

特にネットワークを介した遠隔操作の場合、五感に係わる情報のリアルタイムな提供は非常に重要であり、通信の時間遅れを伴う（リアルタイム性に欠ける）情報提供は、操作者を混乱させるものとなる。例えば、オペレータが手元の操作レバーを右に倒したのに、伝送に遅れがあって遠隔地の装置が右を向くのが 10 秒後だったとすると、操作性は極めて悪いものとなることは容易に想像がつく。仮に、遠隔地の装置が何かにぶつかった場合、その信号の伝送も遅れるため、対象物を破損する恐れすらある。（そして、オペレータは実際に物が壊れた 10 秒後にその事実を知る。）

一方、制御系の設計問題においても、サンプリングタイムを越える遅延は、これを陽に扱わなければ系の安定性が保証されないため、遅延をむだ時間要素として系にループに取り込む必要がある。この際、個々の情報（パケット通信の場合個々のパケット）の遅延時間が予測不能な場合は、最悪遅延時間を考慮した設計を行うことになるが、これは同時に極めて保守的な設計となり系の性能劣化が免れない。このように、情報伝達における遅れは極力避ける必要があるが、複数の機器を接続したネットワーク構成においては、何らかの遅延が発生するため、これを最小化することが課題となる。

そこで本研究では、ロボット制御系のような時間制約の厳しいシステムへの適用を目指し、ネットワーク処理のリアルタイム性を向上したプロトコルスタックを開発した。また、開発したネットワークをバイラテラルロボットシステムに適用し、触覚の転送実験を行ったので結果を報告する。

1) 慶應義塾大学理工学部 教授

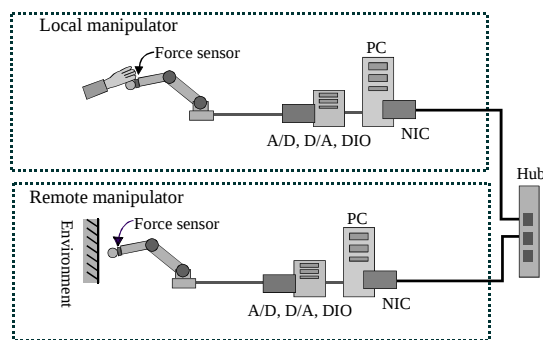


Fig.1 ネットワークベースのバイラテラルロボット
(Network based Bilateral Robot System)

バイラテラルロボットの構成

1. 概要

Fig.1 にネットワークベースのバイラテラルロボットの構成図を示す。バイラテラルロボットでは、遠隔側のマニピュレータが対象物に触れた場合の力を、操作者が感じることを可能とするものである。本システムでは、遠隔側（リモート）と操作者側（ローカル）の2台のロボットマニピュレータのコントローラ（PC）をネットワーク経由で接続し、操作者指令及び対象物との干渉力（カセンサで取得）、各関節の角度情報（エンコーダで取得）などを互いに送受信するように構成している。

2. ネットワークを含む制御系の構成

Fig.2 に制御系のブロック線図を示す。

図中 G_r, G_l はマニピュレータのモデル、 Z_r, Z_l は操作者と対象物のインピーダンスモデル、 K_p, K_{fr}, K_{fl} は制御器である。このうち、制御器 K_{fr}, K_{fl} は、(1) に示すように遠隔側と操作者側の反力の和を最小化し、作用、反作用の関係を実現するように設計した。

$$\text{minimize } |F_l + F_r| \quad (1)$$

ブロック線図から明らかなように、フィードバックループは、遠隔側と操作者側の間のネットワークを含むため、通信遅延によるむだ時間を含む系となる。このような系においてフィードバック系の安定性を確保するためには、系の位相余裕がむだ時間要素による位相遅れを上回る必要がある。

むだ時間を含む系に対する制御系の設計には幾つかの方法が存在するが、系の安定性を保証するためには、通信遅延時間の最大値の推定が不可欠である。また、制御系の性能の劣化を防ぐためには、遅延時間はできるだけ小さいことが望ましい。このため、以下で述べるようなネットワークプロトコルの開発が必要であった。

リアルタイムネットワークの開発

1. リアルタイム性の高いプロトコルスタック

ネットワークにおける通信遅延は、様々な要因が複合して発生するが、時間制約の厳しい制御系においては、ノード内部の通信処理に起因する遅延も無視できなくなる。この遅延はプロトコルスタックの実装方法に依存するが、例えば Linux カーネルにおけるパケット受信においては、Fig.3 に示すようにパケットが割込処理（トップハーフ）によって一旦タスクキューに格納された後、スケジューラが低速割込（ボトムハーフ）を起動するまで処理が延期されるこ

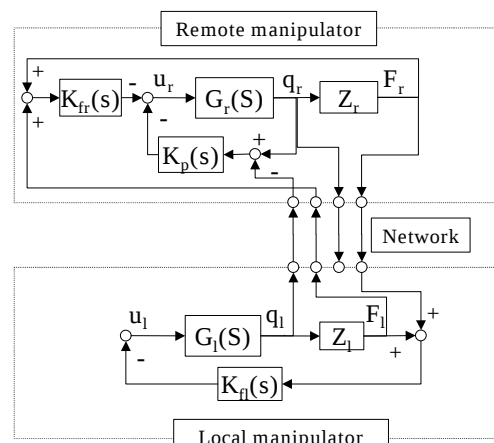


Fig.2 制御系のブロック線図

(Block Diagram of the Main Control Loop)

とに大きな遅延要因がある。また、送信時にも出力パケットを一旦キューにバッファリングして、ある程度の容量になるまでキュー内部に保持することが、遅延時間増大の原因である。これは、帯域の効率を向上するためには有効であるが、リアルタイム性が犠牲になっている。

そこで筆者らは、パケットの受信においては従来ボトムハーフ経由でネットワーク層より上位で行っていた処理をデータリンク層内で完結させ、送信においては、出力キューのサイズを必要最低限に抑えることでリアルタイム性向上を図った。Fig.4 に、開発したプロトコルスタック（RTNP）の構成を示す。

パケット受信においては、ハードウェア（NIC）からの割込を RT-Linux の割込ハンドラがフックし、パケットヘッダのプロトコルタイプ（ID）からリアルタイム処理パケットか否かを判断する。リアルタイム処理パケットであれば、リアルタイムスレッドにパケットを渡すが、それ以外のパケットは、Linux の割込（非リアルタイム）ハンドラによって処理を行う。

一方、送信においては、送信スレッドが作成したパケットをデータリンク層の関数（hard_start_xmit）に直接渡すとともに、送信用のリングバッファの数を小さくすることで、バッファ内での遅延を縮減した。

本プロトコルスタックの実装は、Linux をベースにリアルタイム拡張した OS である RT-Linux 上で行った。RT-Linux の強力なスケジューリングによって他のタスクの実行を中断して送受信を行うことが可能となり、極めてリアルタイム性の高い通信が実現された。また、TCP/IP など既存のプロトコルと共存できるため、従来のネットワーク環境の利便性を損なうことなく使用することが可能である。

2. ベンチマークテスト結果

リアルタイム性能を検証するためにベンチマークテストを行った。本ベンチマークテストでは、Fig.5 に示すように2台のPC間のパケットの往復時間（RTT）の計測によって性能を評価した。

計測条件として、無負荷状態（Case 1）とFTPによる負荷のある状態（Case 2）の二つのケースについて実施した。また、比較のためにUDPを使って同じベンチマークテストを行った。

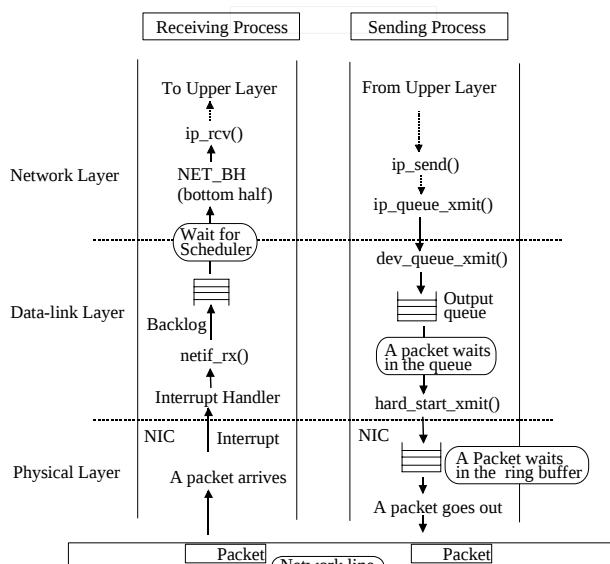


Fig.3 Linux カーネルにおけるパケット処理
(Sequence of Message Process in Linux)

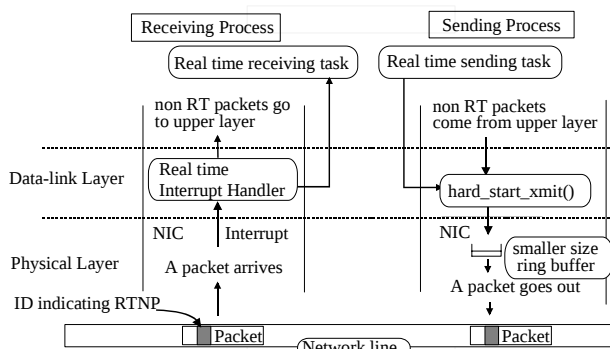


Fig.4 RTNPにおけるパケット処理
(Sequence of Message Process in RTNP)

Table 1 及び Fig.6 にベンチマーク結果を示す。UDP による転送では、無負荷時 (Case 1) は安定した通信を行っているが、FTP のようなバースト転送負荷の下 (Case 2) では、1 msec を超える RTT が頻発したうえ、ジッタ (揺らぎ) が大きい結果となった。

一方、開発したプロトコル (RTNP) においては、高負荷の下でも遅延・ジッタとも少ない安定した送受信が行われており、同プロトコルの有効性が確認された。

・バイラテラルロボットによる力覚転送実験

1. 概要

開発したプロトコルスタックをバイラテラルロボットに実装し、硬さの違う 2 種類の対象物を使用して実験を行った。Photo 1 に実験装置の概要を示す。

2. 静止摩擦を排除した駆動機構

バイラテラルロボットの操作性向上には、マニピュレータの静止摩擦を極力小さくすることが望ましい。そこで、本マニピュレータの各軸の駆動には、対向して配置した 2 台のモータの間にディファレンシャル機構を設ける「ツインドライブ機構」を採用した。Fig.7 にツインドライブシステムの機構図を示す。図に示すように、ツイ

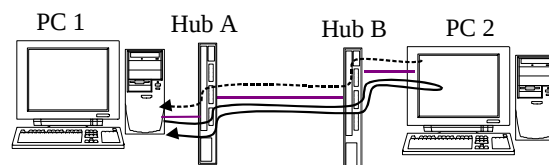


Fig.5 ベンチマークテストの設定
(Benchmark Test Configuration)

Table 1 ベンチマークテスト結果
(Benchmark Test Results)

	UDP (μs)	RTNP (μs)
Case 1 (平均)	197	157
Case 1 (標準偏差)	216	1
Case 2 (平均)	1384	561
Case 2 (標準偏差)	1155	48

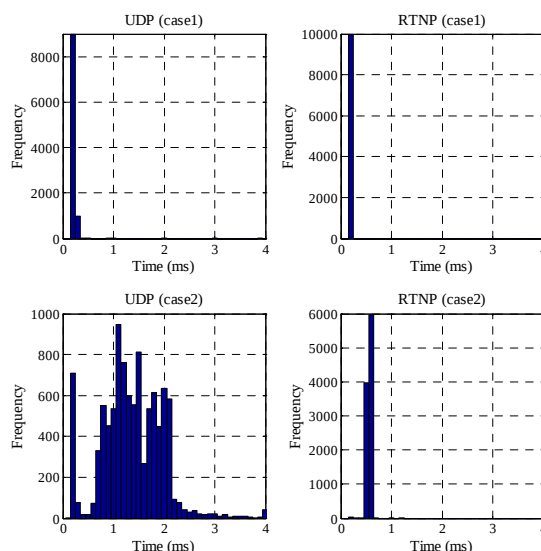


Fig.6 ベンチマーク結果のヒストグラム
(Histogram of the Benchmark Tests)

ンドライブ機構は対向して配置したモータ 1 とモータ 2 の出力軸に直結した 2 枚のホイールで、出力軸に直結したローラーをはさみ込む構造となっている。

モータ 2 がモータ 1 と同一方向に同一速度で回転している場合は (図中で $Rot2 = Rot1$)、出力軸は静止する。また、モータ 2 がモータ 1 と逆方向に同一速度で回転している場合 (図中で $Rot2 = -Rot1$) は、出力軸はモータ 1 と同じ速度で回転する。つまり、ツインドライブの出力速度は次式で表される。

$$\dot{\theta}_e = \frac{1}{2}(\dot{\theta}_1 - \dot{\theta}_2) \quad (2)$$

ここで、 $\dot{\theta}_e$ は出力軸の回転速度、 $\dot{\theta}_1$ はモータ 1 の回転速度、 $\dot{\theta}_2$ はモータ 2 の回転速度である。

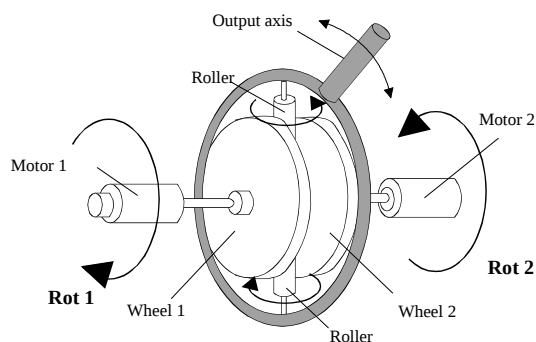


Fig.7 ツインドライブシステム機構図
(Mechanical Overview of the Twin Drive)

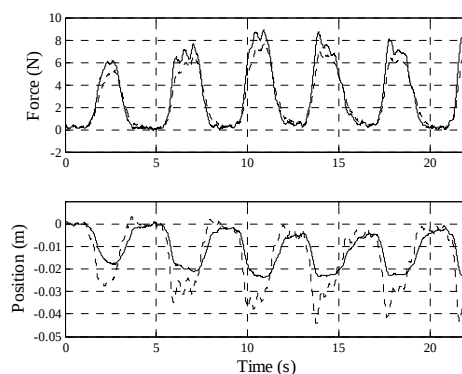


Fig.8 スポンジに対する力応答及び位置応答
(Reaction Force and Position against Sponge Block)

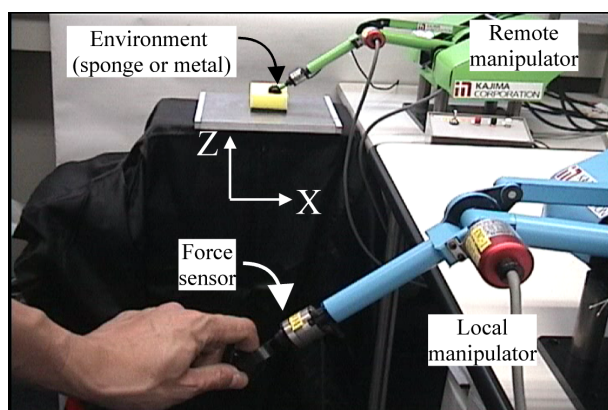


Photo 1 実験の概要
(Overview of the Experiment)

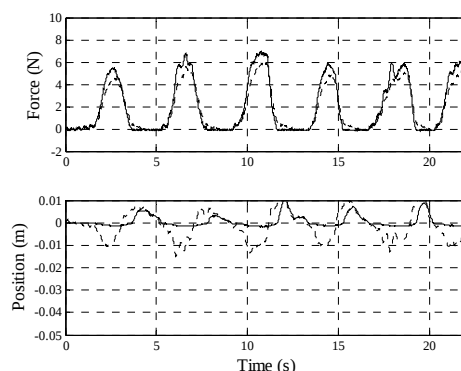


Fig.9 鉄に対する力応答及び位置応答
(Reaction Force and Position against Metal Plate)

ツインドライブ機構では、出力軸が静止した状態から始動する際において、各モータ及び伝達機構が常に回転しているため、原理上静止摩擦が発生しない構造となっている。

3. 実験結果

Fig.8 にリモートマニピュレータがスポンジに接触した場合を、Fig.9 に鉄製の板に接触した場合の結果を示す。なお、実線がリモートのマニピュレータで点線がローカルマニピュレータの応答である。両図とも、上のグラフが鉛直方向の力の応答を示しており、下のグラフはマニピュレータ先端の位置の軌跡を示している。なお、ローカルとリモートの力応答の比較を容易とするため、ローカルの力応答の値は符号を反転して示している。

いずれの場合も、力応答は6～8N程度であるが、スポンジの場合のマニピュレータ先端位置は、鉄の場合に比べ大きく変位している。これは、ローカル側にリモート側の対象物による仮想的な環境が再現されたことを示しており、この結果からネットワークを介した触覚の転送の実現が実証された。

．おわりに

ロボット制御など時間制約の厳しい制御系アプリケーションに適用可能なリアルタイム性の高いネットワークプロトコルを開発した。本プロトコルは、優先度スケジューリングが可能なリアルタイムOS上で実装したもので、既存のプロトコルと共存しながら、負荷の大きなタスクの下でも、高いリアルタイム性を維持出来る。

また、開発したプロトコルをバイラテラルロボットシステムに実装することによって、遠隔地の硬さの異なる対象物の触感を操作者に伝えることを可能とした。

謝 辞

本研究は、文部科学省の平成13年度科学技術振興調整費による「人間支援のための分散リアルタイムネットワーク基盤技術の研究」の一環として行われました。

参考文献

- 1) Y.Uchimura, K.Ohnishi, T.Yakoh; Bilateral robot system on the real time network structure, IEEE. Int. Conf. on Advanced Motion Control, (2002), pp.63 - 68.
- 2) T.Murakami, N.Hayashida, T.Yakoh, K.Ohnishi; A Friction Compensation in Twin Drive System. IEEE. International Conference on Advanced Motion Control, (2000), pp.187 - 192.
- 3) O.J. Smith; A controller to overcome dead time. ISAJ., Vol.6, no.2, (1959.2), pp.28 - 33.
- 4) V. Yodaiken and M. Barabanov; A Real - Time Linux, Proceeding of the Linux Applications Development and Deployment Conference (USELINUX), (1997).
- 5) H. Sato, T. Yakoh; A Real - Time Communication Mechanism for RTLinux., IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, (2000), pp.2437 - 2442.