

# QoS を考慮したラジコンカーのネットワーク制御

末次 新 菊池 浩明 中西 祥八郎  
東海大学工学部

{marumi,kikn}@ep.u-tokai.ac.jp

概要：インターネットでの遅延時間や帯域を考慮して、安全性、即時性を満たす遠隔制御方式を議論する。我々は、RCカーに制御シーケンス送るために使う最適なパラメータ同定実験について報告する。

## Remote control algorithms of the radio-controlled car based on a QoS of Network

Arata Suetsugu Hiroaki Kikuchi Shohachiro Nakanishi  
Tokai University Faculty of Engineering

Abstract : This paper studies a safe and real-time remote control protocol which considers the transmission delay and the bandwidth of a given network. We report about an experiment in order to identify optimal parameters used to send control sequence to the radio-controlled car.

### 1 はじめに

インターネットを通信路に用いた遠隔制御システムに関する研究が盛んに行われている。コミュニケーション手段の拡大<sup>1)</sup>、移動ロボットを用いた美術鑑賞<sup>2)</sup>、遠隔医療、情報家電など応用分野は幅広い。通信論の帯域は、ADSLやATM、衛星インターネットなどの導入により、高品質な通信路が保証されようとしている。QoSもRSVPやATMの技術の進歩により保証されるはずである。しかし、どのアプリケーションに、どれだけQoSが必

要であるかはアプリケーションの要求条件によって大きく異なるため、最低限保証されるQoSを同定する必要性が生じてきている<sup>3)</sup>。

そこで我々は、Radio Control (RC) カーの遠隔制御システムを開発し、操作性が伝送メディアにどれほど影響するかを報告してきた。RCカーに、装備された、カメラからの映像のみを用いて、操作を行った場合、通信路には、PHSクラス(32kbps)の帯域が必要であり<sup>4,5)</sup>、送信する映像の符号化形式には、十分な帯域(64kbps)の基で、Motion-jpegが適している事を示してきた<sup>6)</sup>。我々のシステムは、CGIをRCカーの制御に利用していたが、

CGI は、元々即応性を必要としない処理用に設計されており、例えば処理した確認のデータをクライアント側に返信しないと、次の命令を実行不可能であるという問題点があった。そのため単一操作の応答時間が長く、少しずつ RC カーを走行させる事ができなかった。そこで、より応答性を期待できるように、文献 3) で原氏が述べている Java Applet を用いた制御方式を用いて実装し、評価実験を行った。その実験結果を報告する。

## 2 システム設計

### 2.1 システム構成

本システムは、RC カー(田宮模型, Clod Buster)を基にした走行部と無線 LAN (Xircom, WaveLAN, 2Mbps)を介して接続された操作部から成る。走行部には、CCD カメラ (Canon, Powershot, 27 万画) とマイクとスピーカを備え、ノート PC(東芝, Libretto100, Pentium166MHz, Windows95)が制御を司る(図 1,2)。PC とモータ制御回路間のインターフェイスには、パラレルポートを用い、8 ビットの信号を前後のモータに 4 ビット、左右に 3 ビット割り当てている。前後は、PWM 制御により 2 個の DC モータを駆動して速度を制御する。左右の制御には RC サーボの駆動回路を採用した。これは、50Hz の正パルスの幅によって操舵角を変化させるもので、左が 1.4 ms, 右が 2.3 ms に割り当てられていた。そこで、クロック信号を分周して D0, D1, D2 の 3 ビットで指定される N 進カウンタへ入力し意図するパルス幅を作りだした。

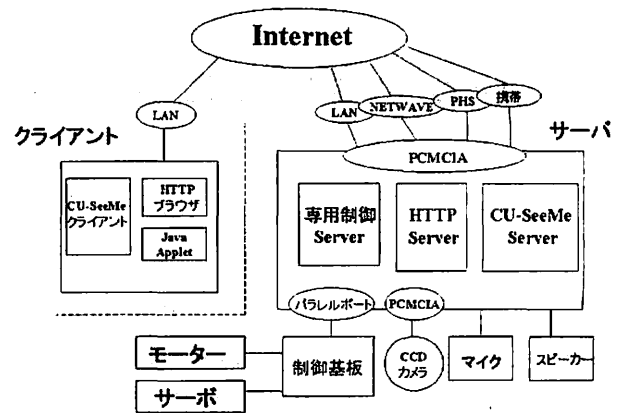


図 1：システム構成図

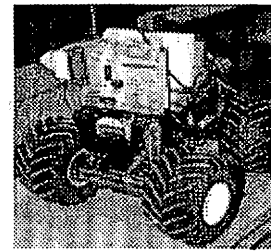


図 2：RC カー

### 2.2 通信プロトコル

画像と音声の送受には、電子会議システム Cu-SeeMe Ver3.1.1 を用いた。UDP ポート 7648 と 7649 により通信し、Cornel Cu-SeeMe Gray, White Pine Motion Jpeg, White Pine H.263 の 3 種類の動画像符号 codec をサポートしている。

## 3 提案操作方式とその評価

### 3.1 方式定義

#### 3.1.1 従来方式

人が操作する側をクライアント、RC カー側をサーバと呼ぶ。クライアント側は、前進や左折などの操作信号を操作命令と動作時間の組をサーバに送信する。サーバは信号を受信したら対応する命令を指定時間動作させる。本方式には、単位時間当たりのアクション回数が約 0.1 回しかないという即応性の問題点が挙げられる。

### 3.1.2 A方式

本方式では、クライアント側はボタンをクリックするたびに、操作命令をサーバに送信する。サーバ側は受取った信号の命令を実行し、次の命令を受取るまで制御ポートの状態を保持する。つまり、走行し始めると停止信号を受取るまで走り続ける。必要最低限の信号を送信するために、通信効率が良い。信号の推移の状態を図3(a)に示す。ここで、前進の信号を F、停止の信号を S、横軸に経過時間  $t$  をとり、C はクライアント側、S をサーバ側の時間軸、停止信号の送信開始時刻  $t_1$ 、実際の停止時間を  $t_2$  とする。次の問題点が挙げられる。

#### ・ 安全性の問題

通信路が故障すると RC カーが走りっぱなしになる。例えば、無線 LAN の有効圏内を超えると止める術がなくなる。

#### ・ 操作性の問題

静止させるには、停止ボタンをクリックするという、新たなアクションを要する。インターフェイスの問題だけでなく、新たなパケットによる伝送遅延が生じるという問題もある。

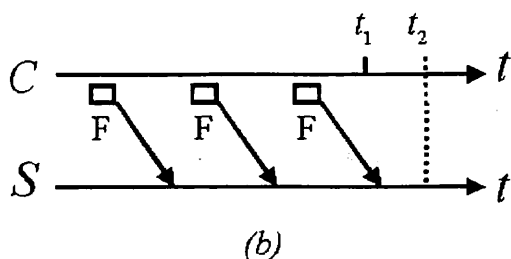
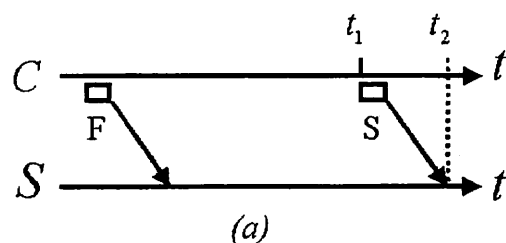


図3：方式AとBの違い

### 3.1.3 B方式

本方式では、クライアント側が、マウスボタンをクリックしている間中、一定の間隔で信号を送信する。ボタンを離すと信号の送信も停止する。サーバ側は、信号を受取るとある一定時間、制御命令を実行し、信号を受取らない時は停止する。本方式による信号推移の関係を図3(b)に示す。マウスを離すと停止するインターフェイスは操作者の負担が軽く、レスポンスの速い制御が期待される。信号の損失が起きても RC カーは、停止するフェールセーフな構造になっている。次の問題点が挙げられる。

・ 信号の送信間隔、命令実行時間の問題  
信号の送信間隔は、短ければ短いほど、操作者の意図がサーバに早く伝わる。しかし、実際には、短すぎるとサーバにキューがたまり、停止の実行が遅れてしまう。

#### ・ 通信効率の問題

常に一定間隔でデータを送信しているので、余分な通信が生じ、通信効率は、良くない。

## 1.1 試験実装

### 3.2.1 従来方式

クライアント側のインターフェイスには、Web ブラウザを用いた。走行側の PC に HTTP サーバ (Omni Server Ver2.0b1) を置き、CGI/Post メソッドによりブラウザから制御信号を送信する。8 ビットの制御信号を動的に変化させる CGI コマンドと ms 単位の出力時間の列を与えるコマンドを用意しており、これにより、8 の字走行などの複雑な走行プログラミングを実現している

### 3.2.2 A、B方式

クライアント側には Web ブラウザと Java Applet を用いた。これは、クライアント側を様々なプラットフォームの上から、操作可能

にする為である。サーバ側には HTTP サーバと、RC 車制御専用サーバを、インストールした。ここで、HTTP サーバは Java Applet を送信するために使用し、RC 車の制御には使用しない。専用制御サーバは C 言語で実装した。これは、制御に関する処理を高速化する為である。信号の送信プロトコルには、UDP を使用した。これは、本システムのような制御においては、パケットの送信が致命的でなく再送も必要な為と、TCP コネクションの確立にかかるオーバーヘッドを省略して、より即応性の高い制御を実現する為である。

### 3.3 操作遅延の評価実験

#### 3.3.1 実験 1 (CGI と Java の応答時間)

CGI を用いて制御する従来方式と、Java Applet と専用サーバを用いて制御する A 方式との遅延の違いを測定するために実験を行った。

無線 LAN (Netwave, 1Mbps) でつながれているクライアント (Win98, P III 550MHz) とサーバ (Win95, P II 166MHz) の環境で、操作ボタンをクリックしてから、RC 車が動作を始めるまでの時間を測定した。測定にはストップウォッチを用い、各々 10 回ずつ繰り返した。結果を表 1 に示す。

表 1: 従来方式と A 方式の応答時間

	従来方式	A 方式
サーバ/クライアント	HTTP, CGI/ ブラウザ	専用サーバ /Java applet
平均応答時間 [s]	1.29	0.47
標準偏差	0.066	0.084

#### 3.3.2 実験 2 (B 方式の最適間隔測定)

B 方式では、クライアント (Win98, P III 550MHz) とサーバ (Win95, P II 166MHz) の環境で実験を行った。クライアント側の信号送信間隔と、サーバ側の命令実行間隔の時間

によって、平均応答時間が決まる。そこで、最も操作遅延の少ない最適な信号送信間隔時間を調べるために、次の実験を行った。

無線 LAN で接続されたクライアントとサーバの環境において、クライアント側のマウスを 5 秒間押し続ける。次にマウスのボタンを離してから、RC 車が停止するまでの時間  $E_B$  を測定した。ここで、クライアントの信号間隔を  $T_c$ 、サーバの命令実行時間を  $T_s$ 、UDP パケット長を  $L$ 、帯域を  $B$ 、停止させようとマウスのボタンを離した時刻を  $t_1$ 、実際に RC 車が止まった時刻を  $t_2$ 、応答時間を  $E_B = t_2 - t_1$ 、通信遅延時間を  $T_1$  とする。クライアント (C) とサーバ (S) における信号の推移を図 4 に示す。 $t_1$  は、区間  $T_c$  の中に一様に分布するので、平均  $T_c/2$  の待ち時間がかかる。また、ネットワーク上でのパケット競合などにより、生じる遅延時間のばらつきを  $M/M/1$  で近似すると、

$$\text{平均到着率: } \lambda = \frac{1}{T_c + L/B}$$

$$\text{平均サービス率: } \mu = \frac{1}{T_s + L/B}$$

$$\text{平均遅延時間: } T_2 = \frac{1}{\mu - \lambda}$$

となり、サーバ側の待ち行列に平均  $T_2$  だけの待ち時間が生じる。更に、ネットワークの伝送遅延の平均を  $T_1$  とする。以上より、応答時間  $E_B$  は

$$E_B = T_1 + \frac{T_c}{2} + T_2$$

となる。

一方、方式 A の応答時間  $E_A$  は、

$$E_A = T_1 + \frac{L}{B}$$

であたえられる。

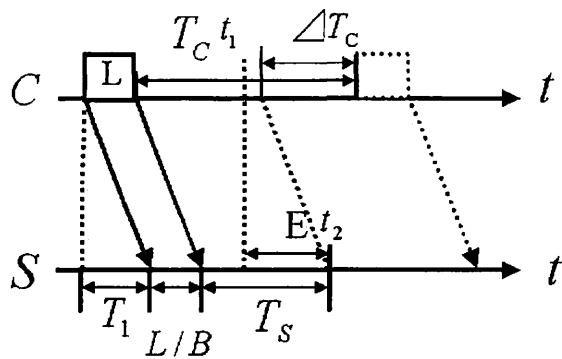


図4：B方式の信号推移詳細

実験では、 $T_c$ を $\pm 50$ [ms]、 $10$ [ms]刻みで変化させ、それぞれ10回ずつ $E_B$ を測定した。さらに、 $T_s$ を $50, 100, 200, \dots, 1000$ [ms]と変化させ、それぞれ10回ずつ測定した。 $\Delta T_c = T_c - T_s$ とおき、結果を図5,6に示す。

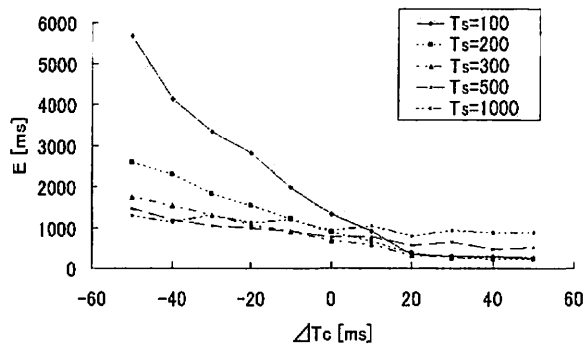


図5： $\Delta T_c$ に対する応答時間E

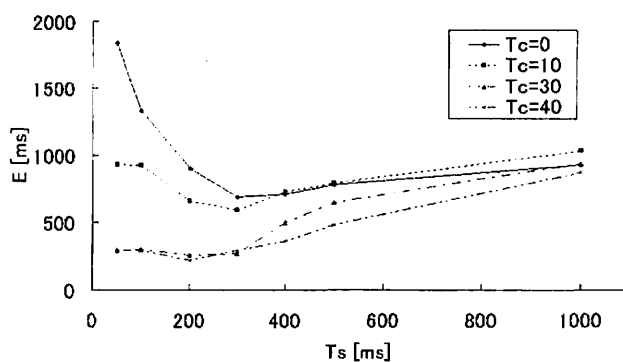


図6： $T_s$ に対する応答時間E

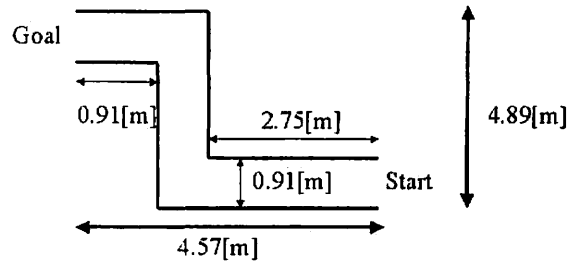


図7：走行コース

### 3.3.3 実験3（走行実験）

クライアント (Win95, P 133MHz) とサーバ (Win95, P II 166MHz) の環境で、実際にRCカーを操作し、図7のようなクランク型のコース、全長  $4.57$ [m]、全幅  $0.91$ [m]、コースの幅  $0.91$ [m] を走行した時間を測定した。測定にはストップウォッチを用い、3人の被験者にそれぞれ3回ずつ走行させ、測定を行った。

結果を表2に示す。

表2：従来方式とA方式の応答時間

	従来方式	A方式	B方式
C/S	ブラウザ /HTTP,CGI	Java Applet /専用サーバ	
平均走行 時間[s]	39.93	15.58	13.92
標準偏差	3.12	2.84	2.12

### 3.4 考察

実験1より、従来方式と比較して、提案方式の応答時間の方が短い。このことから、Java Applet と専用サーバを使ったアプリケーションの有効性が実証された。

実験3より、実際に複雑な動作をさせた場合でも、従来方式と比較して、提案方式の方が、操作にかかる時間を約3倍近く短縮していることから、操作性が向上している事が実証された。

実験2から、 $T_c$ と $T_s$ が応答時間に影響していることが示された。

図5では、 $\Delta T_c$ が増加するとEが減少して

いる。特に、 $\Delta T_c$ が20 [ms]以上になると、 $E$ がほぼ一定に収束する。この時、サーバにキューがたまらなくなっている。これは、待ち行列による解析結果に従っている。待ち行列による遅延は、 $T_s$ が小さい程顕著である。 $T_s$ が短くなることより、帯域における利用率があがり、キューがたまる為である。しかし、 $T_s$ を大きくすれば、ボタンを離してから、反映されるまでの時間が大きくなってしまふ。このトレードオフを表しているのが、 $T_s$ についての $E_B$ の式である。図6では、最も応答時間が短くなっているのは $T_s=200$ [ms]、 $\Delta T_c=40$ [ms]の時であり、応答時間は0.22[ms]であった。 $T_s$ が300[ms]以上の時は、 $T_s$ に線形に、緩やかに応答時間 $E$ が増加する。 $T_s=200$ [ms]以下の時応答時間 $E$ が急激に増加している点に、注目すべきである。以上の結果から、 $T_1<0.47, B=1$ [Mbps]のQoSにおける、B方式の制御方式に最適な間隔は $T_c=200$ [ms]、 $T_s=240$ [ms]となる。この値は、本実験環境における最適値であり、 $T_1$ と $B$ に依存して決まる。より競合性の激しいネットワークや、損失の大きいネットワークなどにおいては、 $T_s$ の最適値は更に大きくなることが予想される。また、A方式とB方式とを操作性に関して比較すると、B方式の方がアクセラに対応するマウスを離すだけなので、より安全性と安定感があり、より人の感覚に近いインターフェイスを得られた。

## 4 おわりに

IPネットワーク経由でRCカーを制御する際に生じるQoSの違いが操作方式に与える影響を実験的に考察した。提案するB方式を実装した本システムでは、従来方式より、応答性があり、安全性が高い事がわかった。また、

操作性においても良好であった。B方式の信号伝送時間の実験から、与えられたQoSにおける本システムの最適な伝送間隔を固定した。本システムは、クライアント側をJava Appletで実装した事から、Java Appletの使用できる端末からならどこからでも操作することができる汎用性がある。将来的には携帯電話などからでも操作が可能になるだろう。

## 参考文献

- 1) 國井他：「握手マシンの開発およびインターネットを介した遠隔握手」, 電学論C, Vol.117, No.5, pp.500-505, 1997.
- 2) 前山他：「移動ロボットの遠隔操作による美術鑑賞」, 日本ロボット学会誌, Vol.17, No.4, pp.486-489, 1999.
- 3) 原：「インターネットを利用した遠隔操作システム」, 日本ロボット学会誌, Vol.17, No.4, pp.477-480, 1999.
- 4) 菊池, 安部, 中西：「通信品質による遠隔操作性の評価」, 情報研報, Vol.99, No.30, 99-DPS-93, pp.49-54, 1999.
- 5) 佐々木他：「インターネットカー：ネットワークを介した遠隔操作システムの開発と評価」, 情処全大, Vol.58, No.3, pp.557-558, 1999.
- 6) 末次他：「インターネットを介したラジコンカーの遠隔制御システムの実装とサービス品質についての遠隔操作性評価」, 日本ロボット学会, Vol.17, No.2, pp.669-670, 1999.
- 7) 藤本, 佐野, 岡本：「インターネットを介した力帰還型バイラテラル遠隔操作」, 日本ロボット学会誌, Vol.18, No.5, pp.713-720, 2000.