

触覚メディア通信における反力の適応制御

藤本 正樹[†](学生員) 石橋 豊[†](正員)

Adaptive Control of Reaction Force in Haptic Media Communications

Masaki FUJIMOTO[†], Student Member and
Yutaka ISHIBASHI[†], Member

[†]名古屋工業大学大学院工学研究科情報工学専攻, 名古屋市
Department of Computer Science and Engineering, Graduate
School of Engineering, Nagoya Institute of Technology,
Nagoya-shi, 466-8555 Japan

あらまし 本論文では, 触覚メディア通信において, 利用者に提示される反力の適応制御方式を提案する. この方式では, ネットワーク遅延に応じて, 反力計算に用いられる弾性係数を動的に変更する. また, 反力だけでなく, オブジェクトに加えられる力も制御する. 本論文では, 実験によって, 触覚メディアの出力品質を評価し, 提案方式の有効性を示す.

キーワード 分散仮想環境, 触覚メディア, 反力, 適応制御, 実験

1. ま え が き

3次元CG(Computer Graphics)によって構築された分散仮想空間において, 触覚メディアを用いて, 手術のシミュレーションや遠隔設計などを行うための研究が活発に行われている[1]. しかし, ネットワークを介して触覚メディアを転送すると, ネットワーク遅延やその揺らぎによって, その出力品質が大きく劣化する可能性がある. これは, メディアの時間構造が乱されるからである. これを防ぐためには, バッファリングによって, ネットワーク遅延揺らぎを吸収するメディア同期制御が必要である[2], [3].

しかし, 触覚メディアは, インタラクティブ性に対する要求条件が厳しい. 文献[4]では, 触覚メディアの許容遅延時間は約30~60msであることが報告されている. したがって, ネットワーク遅延だけでなく, メディア同期のためのバッファリングによっても, 出力品質が大きく劣化するおそれがある.

一方, 文献[5]では, ネットワーク遅延に応じて, 反力を変更し, 触覚メディアの出力品質をできるだけ高く維持する方式が提案されている. しかし, この方式では, ネットワーク遅延の揺らぎや, メディア同期制御によるバッファリング時間などを考慮していない.

そこで, 本論文では, 触覚メディアの発生から出力までの遅延やその揺らぎに応じて, 利用者に提示され

る反力を動的に変更する適応制御方式を提案する. そして, 実験によって, その有効性を示す.

以下では, まず2.で提案方式を説明する. 次に, 3.において実験に用いるシステムや実験方法について述べ, 4.で実験結果の考察を行う.

2. 提案方式

2.1 クライアント・サーバモデル

本論文では, 図1のクライアント・サーバモデルを用いる. クライアントには, 触覚インタフェース装置として, PHANToM DESKTOP [6] (以下, PHANToM と略す) が接続され, 利用者に反力の提示を行う. サーバは, 仮想空間内の物体(オブジェクトと呼ぶ)の位置を更新する. 本論文では, 簡単のため, クライアントの数を1, オブジェクトの数を1とする.

図1には, クライアントとサーバの機能も示されている. クライアントは, PHANToMの1kHzのサーボループ[6]ごとに仮想空間内のPHANToMのカーソル位置情報を取得し, メディアユニット(メディア同期制御のための処理単位. 以下, MU)としてサーバに送信する. サーバは, 受信したMUからオブジェクトの弾性係数を変更する(詳細は次節で説明). 更に, PHANToMのカーソルからオブジェクトへの力の計算を行った後に, 仮想空間内のオブジェクトの位置を更新する. そして, その位置情報をMUとしてクライアントへ送信する.

クライアントで出力される反力の更新も, PHANToMのサーボループごと(すなわち, 1ms周期ごと)に行われる. クライアントでは, サーバからMU(オブジェクトの位置情報を含む)を受信すると, メディ

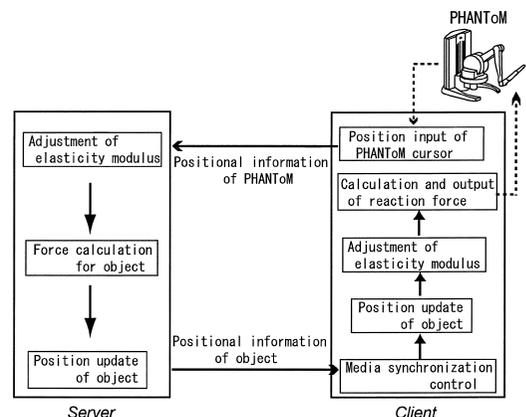


図1 クライアント・サーバモデル

Fig. 1 A client-server model.

ア同期制御を行い、その MU をもとに、オブジェクトの位置情報と弾性係数を更新する。そして、この位置情報と、入力された PHANToM のカーソルの位置情報から、PHANToM に出力される反力の計算を行う（反力と弾性係数の計算方法は次節で示される）。なお、仮想空間の画像の更新レートは 30 Hz である。

2.2 適応制御方式

クライアント側で PHANToM に出力される反力 F は、spring-damper モデル [6] を用いて次のように計算される。まず、クライアントにおいて、仮想空間内の PHANToM のカーソル位置情報とオブジェクトの位置情報を取得する。次に、これらの位置情報から、オブジェクトと PHANToM のカーソルが接触しているか否かを調べる。これらが接触している場合には、反力 F は、次の式から計算される。

$$F = K_S \cdot x + K_d \cdot v \quad (1)$$

ここで、 x はオブジェクトと PHANToM のカーソルとの干渉深度（めり込み）であり、 v はオブジェクトとカーソルとの相対速度の接触面に垂直な成分である。また、 K_S と K_d はそれぞれオブジェクトの弾性係数（elasticity modulus）とダンパ係数である。

ネットワーク遅延やバッファリング時間が大きくなると、オブジェクトの位置が遅れて更新されるので、オブジェクトと PHANToM のカーソルとの干渉深度 x が大きくなる。このとき、クライアントの利用者への反力 F が大きくなり、自然な触感が表現されなくなる。また、サーバでもこれと同じ理由から、オブジェクトを円滑に移動させることが困難になる。

このような問題を防ぐため、反力の適応制御方式では、オブジェクトの弾性係数 K_S の代わりに K'_S を用い、これを変化させることによって、干渉深度 x が反力 F に与える影響を吸収する。以下では、クライアントにおける K'_S の計算方法について説明し、その後、サーバでのそれを示す。

K'_S の計算には、MU がクライアント（サーバ）から送信されてからサーバ（クライアント）で出力されるまでの遅延を用いる。クライアントにおける n 番目 ($n = 1, 2, \dots$) の MU の、この遅延を Δt_n で表し、次式から求める。

$$\Delta t_n = D_n - T_n - \frac{A_n - T_n}{2} \quad (2)$$

ここで、 T_n 、 A_n 、 D_n は、それぞれ n 番目の MU の発生時刻、到着時刻、出力時刻である。なお、上式で

は、クライアントからサーバへの片方向のネットワーク遅延を $(A_n - T_n)/2$ としている（サーバでの MU の処理時間は無視できると仮定）[7]。

Δt_n の値は、ネットワークの状況によっては、大きく変動する。そのため、これをそのまま弾性係数 K'_S の計算に用いると、反力 F の提示に悪影響を与える可能性がある。そこで、以下の式を用いて平滑化を行い、 ΔT_n （MU 遅延と呼ぶ）を得る。

$$\Delta T_n = \alpha \cdot \Delta T_{n-1} + (1 - \alpha) \cdot \Delta t_n \quad (3)$$

ただし、 $\Delta T_0 = 0$ であり、 α ($0 \leq \alpha \leq 1$) は平滑化係数である。

弾性係数 K'_S が ΔT_n に反比例するように、定数 b を ΔT_n で割ることによって K'_S を求めることが考えられる。しかし、 ΔT_n の値が 0 に近いと、 K'_S が大きくなりすぎ、反力が過剰に大きく計算される。これを防ぐため、しきい値 T_{hd} ($T_{hd} > 0$) を導入し、次式から K'_S を決定する。

$$K'_S = \begin{cases} b/\Delta T_n & (\Delta T_n > T_{hd}) \\ b/T_{hd} & (\Delta T_n \leq T_{hd}) \end{cases} \quad (4)$$

本論文では、簡単のため、サーバでも、クライアントで計算された ΔT_n の値を用いる。この値は、PHANToM の位置情報を含む MU に付与され、クライアントからサーバに送信される。

3. 実験方法

実験では、一人の利用者が PHANToM を操作して、仮想空間内のカーソルを動かし、カーソルで立方体のオブジェクト（剛体とする）を持ち上げて球形の目標物体と重なるように移動させる作業を行う [3]。仮想空間の表示例を図 2 に示す。図中の目標物体は軌道に

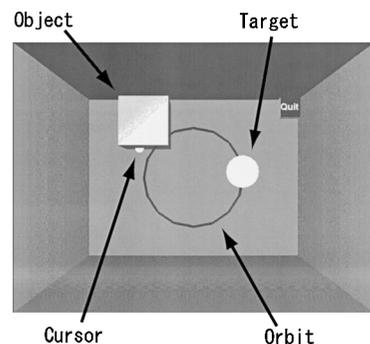


図 2 仮想空間の表示例

Fig. 2 A displayed image of virtual space.

沿って等速円運動をし、軌道上を一周する時間は 10 秒である．なお、目標物体とその軌道は、オブジェクト及び PHANToM のカーソルとは衝突しない．

3.1 実験システム

実験システムでは、クライアント (CPU: PentiumIII 933 MHz, グラフィックカード: 3Dlabs OXXYGEN VX1), サーバ (CPU: Pentium4 1.5 GHz, グラフィックカード: 3Dlabs WildcatII 5000), 及び負荷端末 1, 2 を 10BASE-T の Ethernet ハブに接続する．サーバとクライアント間で送受信される MU は、UDP で転送される．この MU には、カーソルまたはオブジェクトの位置情報、タイムスタンプなどの制御情報、及び反力の適応制御で用いられる MU 遅延の情報が含まれている．MU サイズは両方向とも 41 バイトである．また、MU の送信レートは 1000 MU/s である．

負荷端末 1 から 2 へは、1472 バイトの固定長の負荷メッセージが、指数分布に従う間隔で UDP によって送信される．これにより、サーバ-クライアント間で転送される MU に対して、ネットワーク遅延とその揺らぎを生じさせる^(注1)．

3.2 比較方式

前述のように、反力の適応制御は、メディア同期方式と組み合わせて用いられる．本論文では、メディア同期のために、触覚メディア用に拡張された VTR [3] を用いる^(注2)．VTR は、仮想時間軸や MU の出力時間の伸縮などを行うことにより、MU のバッファリング時間を動的に変更する．

触覚メディアの出力品質の比較には、反力の適応制御を適用する場合 (adaptive control) と適用しない場合 (非適応制御; non-adaptive control) とを扱う．適応制御を用いる場合は、平滑化係数 α を 0, 0.8 [8], または 0.998 [9] と設定する．なお、 $\alpha = 1$ にすると、式 (3), (4) から、弾性係数が変化せず、適応制御を用いない場合と等しくなる．また、サーバとクライアントにおいて、 $K_s = 0.8$, $K_d = 0.2$, $T_{hd} = 15$, $b = 12$ とする [7]． b の値は、式 (4) より、 $K_s \cdot T_{hd}$ に等しくなるように設定されている．

4. 実験結果

平均負荷 (data load) に対するオブジェクトと目標物体との平均距離、反力の平均値、MU 遅延の変動係数をそれぞれ図 3 から図 5 に示す．平均負荷とは、負荷端末 1 が単位時間当りに送信する負荷メッセージのビット数の平均のことである．オブジェクトと目標物体との平均距離は、オブジェクトの中心点と目標物体

の中心点との距離の平均値であり、この値が小さいほど、作業効率が良い．反力の平均値は、PHANToM を介して利用者に出力される反力の平均である．MU 遅延の変動係数とは、式 (3) によって得られる ΔT_n の変化の滑らかさを表す．この値が小さいほど反力の出力が滑らかである．なお、図 5 には、非適応制御の結果が示されていない．これは、非適応制御では前述のように、式 (3) によって MU 遅延を求めないからである．更に、図 3 から図 5 には、95%信頼区間も示している．ただし、この区間が実験結果を示す記号より小さい場合には示されていない．

まず、図 3 において、平均負荷が約 7.0 Mbit/s になると、各方式のオブジェクトと目標物体との平均距

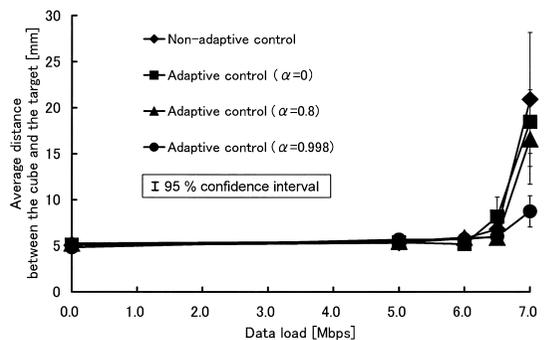


図 3 オブジェクトと目標物体との平均距離

Fig.3 Average distance between the cube and the target.

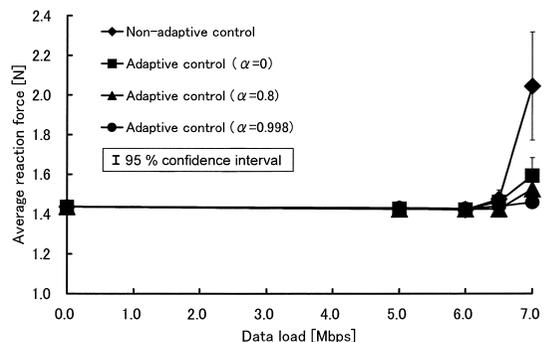


図 4 反力の平均値

Fig.4 Average reaction force.

(注1): 筆者らは、このほか、サーバ-クライアント間で転送される MU に対して、正規分布に従うネットワーク遅延を発生させて、本論文と同様の実験を行った．この実験でも、適応制御方式の有効性が示された．

(注2): 文献 [7] では、バッファ内の MU のうち最も新しいものだけを出し、その他の MU をスキップする Skipping も扱っている．VTR と Skipping に反力の適応制御を適用した結果、高負荷時には VTR の方が出力品質が良かった．

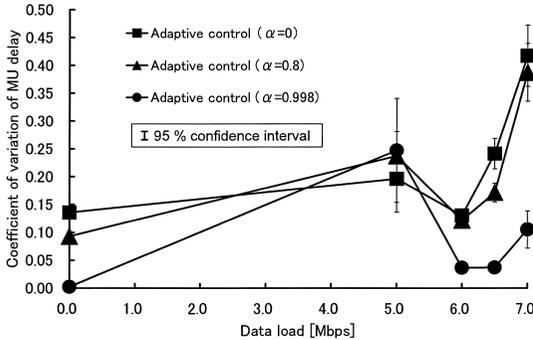


図 5 MU 遅延の変動係数
Fig. 5 Coefficient of variation of MU delay.

離が急に大きくなっている．特に，非適応制御の平均距離が最も大きい．そして，適応制御を用いる 3 方式では， α の値が大きいくほど，平均距離が小さくなっている．このことから，適応制御を用いることにより，触覚メディアの出力品質を高く保つことができ， α を 0.998 にするとよいことが分かる．

次に，図 4 では，平均負荷がおよそ 6.0 Mbit/s 以下の領域において，方式の違いによる反力の平均値の差はほとんどない．しかし，この領域を超えると，反力の平均値が大きくなり始めている．この領域における各方式の反力の平均値の大小関係は，図 3 のそれと同じである．しかし，図 4 では，三つの適応制御の場合の差は大きくないのに対し，図 3 では，適応制御 ($\alpha = 0.998$) と他の方式間の差が大きくなっている．

図 5 から，平均負荷が 6.0 Mbit/s 程度で，MU 遅延の変動係数がいったん小さくなるのが分かる．これは，VTR では，平均負荷が約 6.0 Mbit/s 以上の領域において，MU の発生から出力までの時間が Δ_{al} に抑えられる場合が多くなり，MU 遅延の揺らぎが小さくなるためである．また，図 5 では，平滑化係数 α の値が大きくなるほど，MU 遅延の変動係数は小さくなる傾向にある．更に， α の値が 0.8 と 0.998 のときの変動係数の差は， α の値が 0 と 0.8 のときのものよりも大きくなっている．平均負荷が約 7.0 Mbit/s のとき， α の値による MU 遅延の変動係数の差が，図 3 の平均距離に影響を与えている．

主観的には，適応制御において $\alpha = 0.998$ とする場合が，最もオブジェクトを目標物体に近づけやすいと感じた．また，平均負荷が大きいときでも，提示される反力は滑らかであった．

以上の結果から，適応制御を用いることにより，平均負荷が高いときにも，反力の増大を抑制できることが分かった．また，今回の実験では，平滑化係数 α の値をある程度大きくするのがよいことが判明した．

5. むすび

本論文では，触覚メディアを用いた分散仮想環境における反力の適応制御方式を提案した．そして，実験によって，その有効性を示した．また，平滑化係数 α の値はある程度大きくするのがよいことが分かった．今後，クライアント数を増やした実験を行うとともに，最適な平滑化係数を求める予定である．

謝辞 御討論・御助言頂いた本学田坂修二教授に深謝する．本研究は，平成 15 年度科学研究費補助金基盤研究 (B)(2) (課題番号 13450156) 及び平成 14 年度電気通信普及財団研究調査助成金によって行われた．

文 献

- [1] M.A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computers and Graphics*, vol.21, no.4, pp.393-404, April 1997.
- [2] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda, "The evaluation of delay jitter for haptics collaboration over the Internet," *Conf. Rec. IEEE GLOBECOM '02*, pp.1492-1496, Nov. 2002.
- [3] Y. Ishibashi, S. Tasaka, and T. Hasegawa, "The virtual-time rendering algorithm for haptic media synchronization in networked virtual environments," *Proc. CQR '02*, pp.213-217, May 2002.
- [4] 小森 優, 吉田隆太, 湊小次郎, 菅 幹生, 松田哲也, 高橋 隆, "医療 VR システムにおける反力反応の及ぼす影響の定量測定," 第 39 回日本エム・イー学会大会, 2H1-4, May 2000.
- [5] J.P. Wilson, R.J. Kline-Schoder, M.A. Kenton, and N. Hogan, "Algorithms for network-based force feedback," *Proc. Fourth PHANToM Users Group Workshop*, pp.13-17, Nov. 1999.
- [6] SensAble Technologies, Inc., "GHOST SDK programmer's guide," Version 3.0, 1999.
- [7] 藤本正樹, 石橋 豊, "触覚メディアを用いた分散仮想環境における反力の適応制御," *信学技報*, CQ2003-32, July 2003.
- [8] J. Postel, "Transmission Control Protocol," RFC793, Sept. 1981.
- [9] M. Narbutt and L. Murphy, "Adaptive playout buffering for audio/video transmission over the Internet," *Proc. Seventeenth UK Teletraffic Symposium*, May 2001.

(平成 15 年 9 月 18 日受付)