B-004

大規模移動体シミュレーションを対象とした動的タイムステップ制御方式 ～通信モデル及び分散化の実現検討～

Event-aware Dynamic Time Step Synchronization Method for Large-scale Moving Object Simulation - Study of communication model and distributed method -

尾崎 敦夫† 白石 將† 渡部 修介† 古市 昌一†
Atsuo Ozaki, Masashi Shiraishi, Shusuke Watanabe, and Masakazu Furuichi

1．まえがき

航空機，船舶，車両，人等の移動体（MO：Moving Object）が数多く登場する計算機シミュレーションでは，各 MO を模擬単位としてタイムステップ方式により実装することにより MO 数に比例した並列性が抽出できる．このため，このような移動体シミュレーションは，複数のプロセッサを用いることによる並列分散シミュレーション技術により実行性能向上を図ることが可能となる[1]．このタイムステップ方式に基づく実行では，各 MO のタイムステップを因果関係に矛盾を発生させることなく，如何に大きくすることができるかが，実行性能向上のための重要な鍵となる．我々は既に，このタイムステップを各移動体のその時々の状況に応じて，因果関係に矛盾を発生させることなく大きく設定することにより実行性能向上を図る動的タイムステップ制御方式[2]を提案済みである．しかし本方式は移動体同士が「見る」または「見られる」状態である場合のみをタイムステップ設定の基準にしている．このため，移動体間で情報伝達のための通信を行う場合には，この要素を取り入れたタイムステップの設定方法とそのための効率的な実装方式が必要となる．

本稿では，動的タイムステップ制御方式に基づく移動体シミュレーションにおいて，上記移動体間の通信モデルを導入する場合の課題を示し，この課題を解決するための実行方式を提案する．更にこの実行方式を HLA（High Level Architecture）[3] に基づく分散計算機環境へ実装するための方法についても説明する．

2．動的タイムステップ制御方式と通信モデル導入の課題

我々は MO を主対象とした計算機シミュレーションにおいて，因果関係の矛盾を発生させずに，各 MO のタイムステップ（Δt）を大きく設定することにより実行性能向上を図る動的タイムステップ制御方式（DTSS：event aware Dynamic Time Step Synchronization method）[2]を考案した．本方式では，MO 間でイベントをやり取りする場面を，お互いに，「見る」または「見られる」の関係にある時のみとしている（この場面を“会合場面”と称する）．なお，図 1 は DTSS に基づいて，移動体 MOa の次模擬時刻設定のためのΔt の導出例を示したものであり，他の移動体との会合可能性時間の中で最も小さい値がΔt に設定される．この場合，$\Delta t_{a\to b} < \Delta t_{a\to c}$ であるので，$\Delta t_{a\to b}$ が MOa の次模擬時刻設定のためのΔt となる．但し，会合状態では因果関係に矛盾が発生しないよう定義した最小のΔt で模擬する必要がある．

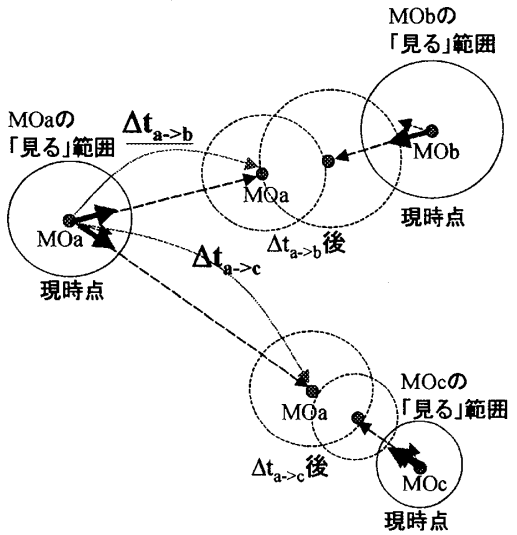

図 1．動的タイムステップ制御方式の基本概念

この DTSS において，例えば，会合とは関係ない遥か遠方の移動体から情報伝達のための通信イベントが送信された場合，そのタイミングで当該通信イベントを受ける移動体は起動し，そのイベントを処理する必要がある．例えば，図 2 に示すように，MOa が DTSS に基づき t_i から t_k に時刻を進めた状態で遥か遠方の MOz より時刻 t_j に通信イベントが MOa に送信された場合，MOa は t_j で当該通信イベントを受信及び処理するために，時刻を巻き戻し（ロールバック）して，時刻 t_i から t_j の模擬を再実行する必要がある．この時，MOa が時刻 t_k に通信イベントを他の移動体に送信していた場合は，当該イベントメッセージを無効にするキャンセルメッセージを発行する必要がある．また，過去の状態に戻すためにログも取得する必要がある．なお，このように模擬の再実行を許す方式を楽観的手法[4]と呼ぶ．他方，このような時刻のロールバックやキャンセルメッセージを用いずに実行するためには，全移動体の中で最小の論理時刻（GVT：Global Virtual Time）を持つ移動体から順番に実行する必要がある（本方式を保守的手法[4]と呼ぶ）．即ち，図 2 の例では，①→②の順番に各移動体を模擬することとなる．但し，保守的手法はこの逐次実行による制約により，複数のプロセッサを用いることによる実行性能向上は困難となる．他方，楽観的手法は，例えば図 2 での①と②を平行して実行することが可能だが，ロールバックが生じた時に，キャンセルメッセージが雪崩式に増加してし

†三菱電機（株）

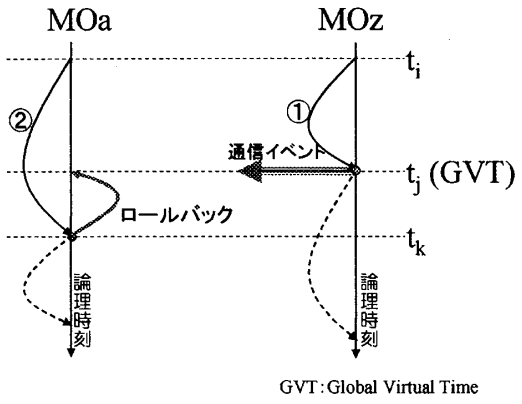

図２．通信イベントを導入した場合の各移動体の動作例

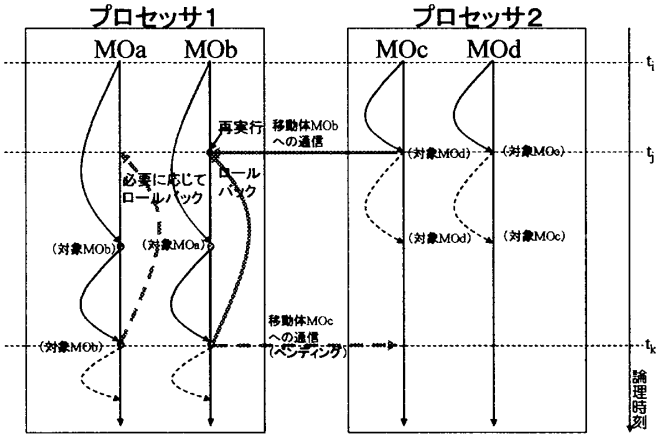

図３．提案方式の実行例

まう可能性があり，当該メッセージがプロセッサ間を跨る場合では，実行性能を大きく劣化させてしまうこととなる．

3. 通信モデルの導入と分散化の実現検討

3.1 基本概念

我々は，多くの移動体を複数のプロセッサを用いることによる並列分散シミュレーション技術により高速化を図ることを考えている．従って，１つのプロセッサは基本的に複数の移動体を扱うこととなる．このため，前章で述べたように，保守的手法では複数のプロセッサを用いる効果が発揮できない．他方，楽観的手法での，キャンセルメッセージがプロセッサ間を飛び交う状況は避けるべきだと考える．但し，プロセッサ内の移動体に関して楽観的手法を用いる場合では，キャンセルメッセージが飛び交うのは当該プロセッサ上の移動体に限定できるため，ロールバック及び模擬の再実行に要するコストは比較的小さく抑えることができる．また，プロセッサ間は保守的手法により逐次実行したとしても，各プロセッサ内は楽観的に実行させておけば，他のプロセッサ上の移動体模擬中に，実行を待たされる状況は少なくなる．

以上の観点から，我々はプロセッサ内の移動体は楽観的に，またプロセッサ間は保守的に実行する方式を検討している．

3.2 分散化方式

図 3 は，前節で述べた基本概念に基づく実行例を示したものであり，２台のプロセッサにより計 4 つの移動体を模擬する場合を説明したものである．なお，DTSS では，図 1 で説明したように最も早く会合する可能性のある他の移動体の情報に基づいて次ぎの模擬時刻を決定する．従って，会合する可能性が高い移動体同士を同一のプロセッサに割り付ければ，当該プロセッサ内で独立して模擬及び時刻を進められる状態を持続できる可能性が高くなる．なお，図 3 中の“（模擬対象 MOx）”は次模擬時刻設定の対象となる移動体を示したものである．しかし，他のプロセッサ上の移動体と会合する可能性が生じた時は，該他の移動体の情報無くしては模擬及び時刻を進めることはできなくなるため，プロセッサ間の通信は必須となる．

提案する実行方式では，プロセッサ間の通信は保守的手法に基づいて時刻の小さいイベントから順番に実行することとなるが，プロセッサ内は楽観的に実行するものであり，上記したようにプロセッサ間の通信が必要となる時刻まで模擬を進めることができる．しかし，プロセッサ間の通信は保守的に実行するため，他のプロセッサ上の移動体への通信イベントは当該時刻が GVT に達するまではペンディングさせる必要がある（図 3 参照）．従って，図 3 に示すように，時刻 t_j で移動体 MOc から MOb へ通信イベントが送信された場合，MOb は時刻 t_j までロールバックして当該イベントを処理することとなり，このロールバックの影響を受ける可能性があるのは同一プロセッサ上の MOa のみとなる．この場合，MOa は必要に応じてロールバック及び再実行することとなる．また，MOa から MOc への通信は，GVT が t_k に達するまでペンディングすることになる．

3.3 HLA 分散計算機環境への実装

HLA は，分散シミュレーション相互接続仕様であり，各シミュレータ間の論理時刻管理には以下の方式を実装するためのインターフェースを備えている．

- Time step method (HLA-TS)
- Event based method (HLA-EB)
- Optimistic method

図 4 は HLA-TS と HLA-EB の動作例を示したものである．なお，図中の FED は“フェデレート”の略で，HLA による各シミュレータを示し，TAR（Time Advance Request）と NMR（Next Message Request）は，それぞれ HLA-TS と HLA-EB における時刻進行及びメッセージ受信要求であり，TAG（Time Advance Grand）は時刻進行許可を示すものである．また，Lookahead は当該 FED が，その区間，メッセージを送信しないことを宣言するものであり，TSO（Time Stamp Ordar）message は時刻付きメッセージを示したものである．DTSS は基本的に保守的に時刻を進める方式であることから，HLA-TS 及び HLA-EB のどちらでも，DTSS に基づく移動体シミュレーションを実装することができる．我々は，既に DTSS に基づく移動体シミュレーションを HLA 分散計算機環境上に実装するための検討を実施し，簡易モデルを用いた評価を実施済みである．この結果から，タイムステップを TAR により動的に設定する HLA-TS が DTSS に基づく移動体シミュレーションの実装に適してい

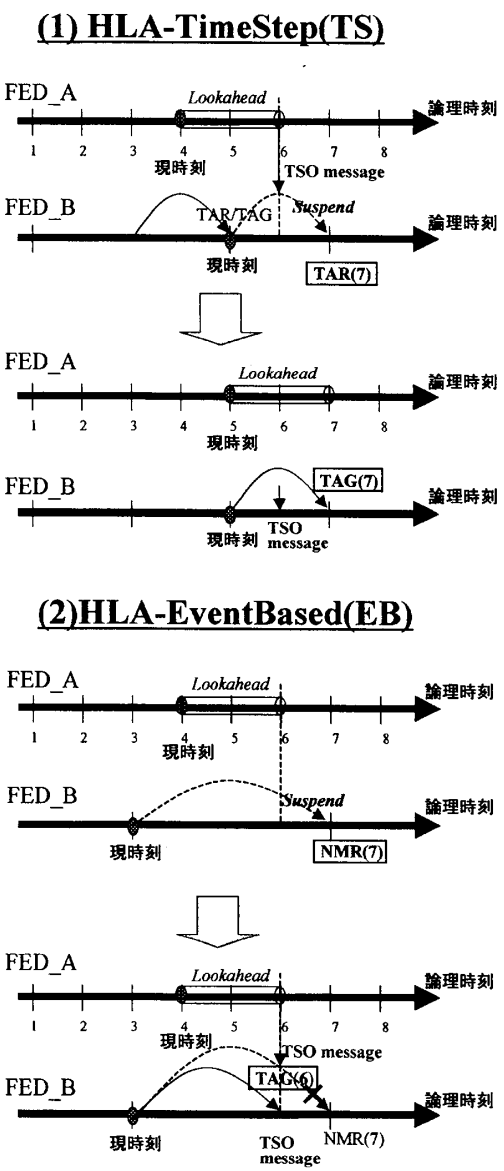

図４．HLA-TS と HLA-EB の動作例

ることが分かっている[2]．これは，図 4 の（1）に示すように HLA-TS では FED_A が時刻 6 に TSO メッセージを FED_B に送信したとしても，FED_B は指定した時刻 7 で起動できるので，不要な時刻に起動されることが無いためである．しかし，通信イベントを扱うこととなった場合，通信イベントを受信する移動体は該通信イベント送信時刻に起動する必要があるため，HLA-EB による実装が適している．これは，HLA-EB では図 4 の（2）に示すように，FED_B が時刻 7 に NMR を発行したとしても，FED_A が TSO メッセージを送信した時刻 6 に FED_B が起動されるためである．

従って，プロセッサ内の移動体群を 1 つの FED に割り当て，FED 内に移動体の模擬とローカルな時刻進行を管理するマネージャを設けて，他の FED 内の移動体と通信しなければならない最も早い時刻を NMR に設定する実行及び実装方式が適していると考える．なお，図 3 の場合では MOa と MOb が同一の FED で管理され，この例では，該 FED は時刻 t_k で NMR を発行することとなる．他方，MOc と MOd を管理する FED は時刻 t_j で NMR を発行することになる．但し，通信イベントが生じない場合は，TAR（HLA-TS）による時刻進行方法が適しており，事前に通信イベント発生の有無が分かっていれば，適宜，NMR と TAR を使い分ける実行方法も考えられる．

4. むすび

本稿では，既に考案した移動体シミュレーションのための高速化手法である DTSS を概説し，本方式に移動体間での情報伝達のための通信モデルを導入した場合の課題を示した．そして，本課題を解決するための実行方式を提案し，本方式に基づく移動体シミュレーションの HLA 分散計算機への実装方法を説明した．本提案方式は，プロセッサ内は楽観的に，またプロセッサ間は保守的に実行する方式であり，具体的なアプリケーションを用いた評価が今後の課題である．また，提案方式において，実行性能向上を図るためには，各移動体の各プロセッサへのマッピング方法が鍵となり，本方式に適したマッピング方法の検討も今後の課題である．

参考文献

[1] A.Ozaki, et al, “Design and Implementation of Parallel and Distributed Wargame Simulation System and Its Evaluation,” IEICE Trans. Vol.E84-D No.10, pp.1376-1384, 2001.

[2] A.Ozaki, et al, “Event-Aware Dynamic Time Step Synchronization Method for Distributed Moving Object Simulation,” IEICE Trans. Vol. E89-A No.11 pp.3175-3184, 2006.

[3] IEEE Std 1516-2000 IEEE Standard for Modeling & Simulation (M&S) High Level Architecture (HLA) - Framework and Rules

[4] R.M.Fujimoto, “Parallel discrete event simulation,” Commun. ACM, vol.33, no.10, pp.30-53, Oct.1990.