動的タイムステップ制御方式に基づく道路交通シミュレーションの実現と評価

尾崎敦夫[†] 松下和隆[††] 白石將[†]
渡部修介[†] 古市昌一[†] 佐藤裕幸[†]

我々は計算機クラスタ環境上において，航空機，艦船，車両等を主対象とする移動体シミュレーションの高速化手法である，動的タイムステップ制御方式（DTSS: event aware Dynamic Time Step Synchronization method）をすでに考案している．しかし，問題が大規模になった場合には，1 つの計算機上でも多数の移動体を高速に模擬することが求められる．本論文では，移動体が道路網上を移動する道路交通シミュレーションを取り上げ，模擬結果を変えずに移動体間の同期および模擬コストを低減させて実行性能向上を図る，道路交通向け動的タイムステップ制御方式（DTSS-RT: DTSS for Road Traffic simulation）を提案する．本方式を災害シミュレータの代表例であるロボカップレスキューの道路交通サブシミュレータへ適用した結果，ひどい渋滞状態がなければ従来方式の約 2〜3 倍の性能向上が実現できることが確認できた．

Design and Implementation of Road Traffic Simulation Based on Dynamic Time Step Synchronization Method and Its Evaluation

Atsuo Ozaki,[†] Kazutaka Matsushita,[††] Masashi Shiraishi,[†]
Shusuke Watanabe,[†] Masakazu Furuichi[†] and Hiroyuki Sato[†]

We have already proposed DTSS (event aware Dynamic Time Step Synchronization method) for speeding up moving objects (MOs) simulation, e.g. aircraft, ships, vehicles and so on, in a computer cluster environment. However, speeding up the simulation for a number of MOs on a single processor is required as well, when the problem size is enlarged. In this paper, we propose DTSS-RT (DTSS for Road Traffic simulation), which can speed up the simulation by reducing the simulation cost of each MO and synchronization cost between MOs without changing the simulation result. For evaluating DTSS-RT, we employed a road traffic sub-simulator of RoboCupRescue simulation. The results show that the performance of the simulation based on DTSS-RT is improved approximately 2〜3 times over that of the conventional method except when the situation includes a heavy traffic jam.

1. はじめに

計算機（PC，ワークステーション等）のコストパフォーマンスが向上するにつれ，防災・防衛等の分野では，航空機，船舶，車両，人等の多数の移動体（MO: Moving Object）をより詳細に，かつより高速に模擬することが可能な訓練・演習用または解析用のシミュレーションシステムが求められる．また，コスト低減を目的としたシミュレータの再利用性向上や，複数の異なる組織による大規模システムの開発効率向上といった観点から，シミュレータの標準化が求められている．

† 三菱電機株式会社
Mitsubishi Electric Corporation
†† 三菱電機インフォメーションシステムズ株式会社
Mitsubishi Electric Information Systems Corporation

我々は，シミュレーションの高速化に関しては，複数の計算機をプラットフォームとする並列・分散シミュレーション技術を，またシミュレータの標準化に関しては，HLA（High Level Architecture，IEEE1516）[1)〜3)] の適用を検討および実施してきた[4)〜9)]．しかし移動体数が膨大となった場合は計算機台数を増やすことによる高速化だけでなく，計算機単体上での実行性能も向上させる必要がある．

本論文では，我々が並列・分散シミュレーション高速化手法として考案した動的タイムステップ制御法（DTSS: event aware Dynamic Time Step Synchronization method）[8),9)] の概念に基づいて，移動体が道路網上を移動する応用問題を対象に，計算機単体上で実行性能向上を図る方式（DTSS-RT: DTSS for Road Traffic simulation）を提案する．そして，DTSS-RT

の実装法と道路交通シミュレーションへの適用効果を説明する．なお，本技術課題の検討は大規模シミュレーションを効率的に開発するためのフレームワーク“MAЯS☆”構築の一環として実施しているものである[10]．

以下2章では，まず本研究の位置づけを示す．3章では動的タイムステップ制御法の道路交通シミュレーションへの適用方法を議論し，提案方式であるDTSS-RTを説明する．そして4章で，災害シミュレータの代表例としてロボカップレスキュー（RCR）シミュレータ[11]を取り上げ，本シミュレータの道路交通サブシミュレータへDTSS-RTを適用した場合の性能評価結果を示す．また5章で，この結果をさらに粒度および問題規模の観点から考察する．そして6章では，道路交通シミュレータに関する関連研究を紹介する．

2. 本研究の位置づけ

移動体を対象とする計算機シミュレーションでの，シミュレーション時刻進行方法には大きくイベントベース法とタイムステップ法の2種類が存在する[12),13),☆☆]．イベントベース法は，移動体間で発生するイベントをシミュレーション時刻順に並べて，時刻の小さいものから模擬していく方法である．他方，タイムステップ法は，対象とするアプリケーションに応じて，タイムステップを決め，このタイムステップに基づいてシミュレーション時刻を進める方法である．イベントベース法では，不要な模擬を抑えることができるが，多数の移動体が登場し，お互いが複雑に影響を及ぼし合う場合は，イベント発生時刻の算出が非常に困難となる．他方，タイムステップ法は，シミュレーション時刻の精度がタイムステップの大きさに左右されてしまうが，イベント発生時刻を計算する必要がない．また，不要な模擬が多発するが，移動体の数だけ並列度が存在するため計算機クラスタ環境等を利用した並列・分散シミュレーション技術により高速化を図ることが可能となる．

すでに我々が考案した動的タイムステップ制御法（DTSS）は，計算機クラスタ環境を対象に移動体の単位で負荷分散させた場合の高速化手法に関するものである．一般的に，計算機クラスタ環境上で移動体シミュレーションの高速化を図る有効な方策が，移動体

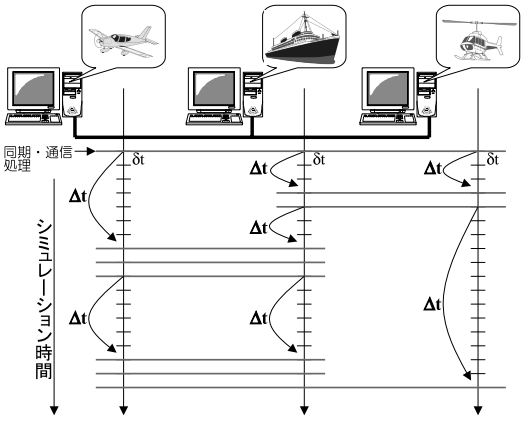

図1 動的タイムステップ制御法（DTSS）による実行例
Fig. 1 Example execution by applying DTSS.

間のデータ交換にともなうプロセッサ間の通信および同期処理を極力減らすことである．DTSSはタイムステップ法に基づく移動体シミュレーションを対象に，移動体間で相互作用が生じるイベントが発生する可能性がある場合のみ，通信および同期処理を行うものであり，このタイミングを基準に各移動体のタイムステップの大きさを動的に設定するものである．DTSSでは，イベント発生時刻を正確に算出するのではなく，イベントが発生する可能性がある時刻を簡単なメカニズムにより小さい演算量で算出する．なお，**図1**はDTSSによる実行例を示したものであり，従来は一般的に固定のタイムステップ δt で全移動体の模擬および移動体間の通信・同期処理を行っていたが，DTSSでは，Δt（$= \delta t \times$ n：nは正の整数）の間隔で通信・同期処理を行う．したがって，Δt を大きく設定できる場合では，通信・同期コストを抑えることができるため高速化が図れることになる．しかし，移動体数が大規模になる場合では，1つの計算機上でも複数の移動体を高速に模擬するための仕組みが求められる．

本論文で提案するDTSS-RTはDTSSをベースとしたものであり，特に移動体が道路網上を移動する道路交通シミュレーションを対象に，1つのプロセッサ上で複数の移動体を高速に模擬することを目的とする．したがって，DTSS-RTは，**図2**に示すように移動体間での同期コストを低減させるだけでなく，個々の移動体の模擬コスト（図2中の太線部分）をもDTSSの概念に基づいて低減させるものである．さらにDTSS-RTはDTSS同様，細かい固定の δt によるタイムステップにより模擬する従来方式と同一の模擬結果を実現するものである．

☆ Mitsubishi Architecture Framework Required for Modeling and Simulation Systems.

☆☆ 並列・分散シミュレーションでのオプティミスティック法もあるが本論文での主テーマは計算機単体上でのシミュレーション実行であるため，議論から除外する．

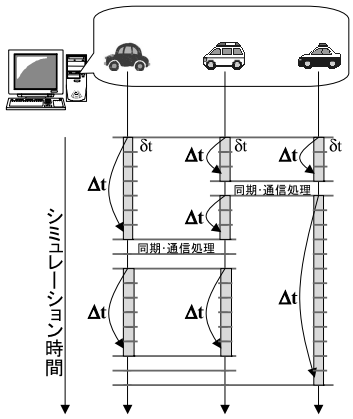

図 2 道路交通向け動的タイムステップ制御法（DTSS-RT）による実行例
Fig. 2 Example execution by applying DTSS-RT.

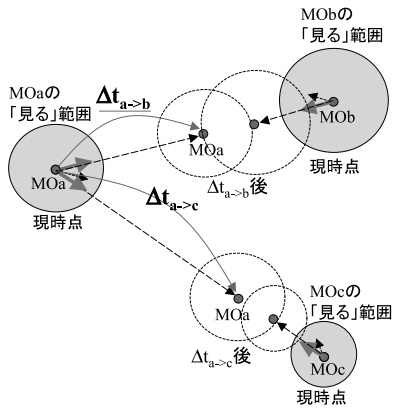

図 3 DTSS の基本概念：移動体 MOa の Δt を決定する場合の例
Fig. 3 The concept of calculating Δt for MOa based on DTSS.

3. 動的タイムステップ制御法の道路交通シミュレーションへの適用検討

本章では，DTSS を道路交通シミュレーションへ適用する際の課題を示し，本課題を解決し実行性能向上を図る DTSS-RT を説明する．

3.1 動的タイムステップ制御法とその課題

DTSS では，各移動体は，次の模擬時刻を他の移動体と相互作用を及ぼし合う可能性のある最小の時刻に設定するものであり，他の移動体と直線距離上を互いに最高速度で向かい合うと仮定して算出する．この相互作用を及ぼし合う状態とは移動体間で「見る」または「見られる」状態（以下，「会合状態（SoD: Scene of Detection）」と呼ぶ）のことを示す．これは最悪ケースを想定して次の Δt および模擬時刻を設定するためであり，かつその Δt 算出コストを小さくするためである．**図 3** 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．ただし，会合状態では因果関係に矛盾が発生しないよう最小の Δt，すなわち δt のタイムステップで模擬する必要がある．

しかし，移動体が道路網上を移動する場合，他の移動体との距離を直線距離で定義して次の模擬時刻を算出する方法では Δt が小さくなり非効率である．また，道路が存在しない地点で会合することを仮定して Δt を設定する場合もおおいにありうる．このため，道路網に沿った距離情報に基づいて Δt を算出することが好ましい．この距離情報が実際に走行する経路とかけ離れていればやはり性能劣化につながるが，実際に近づけようとするとそのための計算コストが増してしまう．しかし，我々は各移動体が持つ計画路情報を利用することによって，計算コストを増やさずに実行性能向上を図る “道路交通向け動的タイムステップ制御方式 DTSS-RT” を考案した．次節では DTSS-RT について説明する．

3.2 道路交通向け動的タイムステップ制御方式

3.2.1 基 本 概 念

DTSS-RT では，各移動体は自身の計画路上に他の移動体および静止体が存在する（この地点を「会合地点（CP of SoD: Core Point of SoD）」と呼ぶ），または他の移動体の計画路と重複する地点（この地点を「会合可能性地点（CP of pSoD: Core Point of possible SoD）」と呼ぶ）が存在する場合に，その地点を基準に次の Δt および模擬時刻を算出する．ここで，静止体とは信号機や目的地等の自律的に移動しない物体または地点のことである．ただし，信号の挙動が既知で，当該移動体通過時に「青」であることが明らかな場合は，この信号を会合地点（静止体）としなくてよい．**図 4** はこの計画路の例を示したものであり，以下はこの例における各移動体の計画路情報である．

- 移動体 A の計画路：[R-8, N-6, R-6, N-2, R-3, N-3, D-1]
- 移動体 B の計画路：[R-10, N-7, R-9, N-5, R-5, N-1, R-2, N-2, R-3, N-3, R-4, N-4, D-2]
- 移動体 C の計画路：[R-11, N-8, D-3]

この場合，N-2 地点が移動体 A と B の会合可能性地点となる．ここで R は道路，N は交差点，そして D は目的地を意味する．また，本方式は図 4 の移動体 A にとっての移動体 C のように，会合地点および

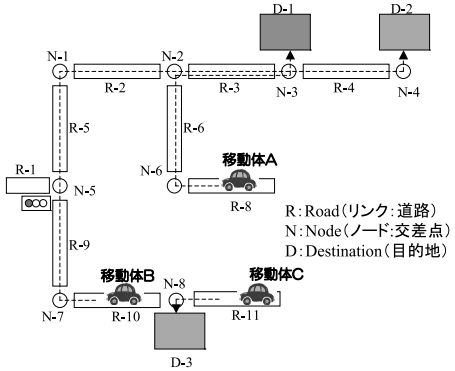

図 4 計画路情報の例
Fig. 4 Example of the planning route data.

会合可能性地点が存在しない他の移動体はいっさい考慮せずに Δt を算出できるという特長がある．

基本的に，各移動体での 1 タイムステップ Δt 分の処理は，周囲を「見て」，その状況から次の動作を「判断」し，「動く」ことである．このタイムステップを大きくすることによる効果は，1 度「見て」，「判断」する処理を行えば，このタイムステップの間は，周囲の状況を考慮せずに自身の状態だけで 1 タイムステップ分の模擬が実行できることであり，この「見て」，「判断」する処理の回数を削減できることにある．また，本方式では「動く」処理も計画路情報と後述する走行モデルの概念を利用することにより模擬コストの低減を図る．これらの部分が DTSS-RT によりシミュレーションを高速化できる主要部分となる．DTSS-RT の基本概念は DTSS と同じだが，DTSS-RT では各移動体の計画路情報を利用してまずは会合地点および会合可能性地点を求め，それを基準に次の Δt および模擬時刻を設定する．

なお，我々が想定している道路交通シミュレーションのモデルでは，各移動体は自身の走行車線上の走行方向のみの情報を利用して走行するものとし，反対車線等の状況は考慮しないものとしている．したがって，自身の走行に影響を与えるものは，自身の計画路上の最寄の前方物（移動体または静止体）または他の移動体との会合可能性地点であり，前方物が移動体であるときでも自身の方向に近づくことはない．このため，各移動体の最寄りの会合地点または会合可能性地点が次模擬時刻設定の基準となる．この地点が決まると，次に移動体の走行モデルを作成し Δt を計算する．走行モデルは基本的に，他の移動体の状態から影響を受けない等加速走行および等速走行と，影響を受ける追従走行より構成される（**図 5** (a) 参照）．この追従走行の状態は会合状態と同義であるため，追従走行開始時

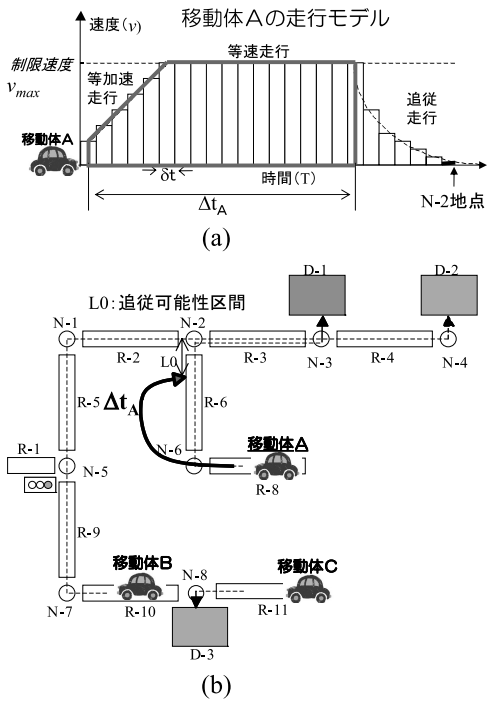

図 5 移動体 A への自主的 Δt 設定法の適用例
Fig. 5 Example of calculating Δt (Δt_a) for MO_A when Self-motivating Δt set scheme is applied.

点までを Δt に設定し，次模擬時刻（現在時刻 + Δt）を決定する．また，会合状態または会合する可能性がある状態では，δt（Δt の最小値）により次模擬時刻を設定する．会合可能性地点ではどちらの移動体が先に当該地点を通過するかは未定であるため，δt によるタイムステップにより模擬することとなる．

したがって，DTSS-RT は他の移動体から影響を受けない等加速および等速走行区間はタイムステップ Δt ($= \delta t \times$ n：n は正の整数）に，また影響を受ける追従走行区間およびその可能性がある区間は δt に基づきシミュレーションを実行するため，実行性能の向上を図り，かつ固定のタイムステップ δt に基づく従来方式と同じ模擬結果が実現できることになる．

3.2.2 実装検討

上記したように DTSS-RT では，各移動体は自身の計画路情報に基づいて，まず他の移動体もしくは静止体との会合地点および会合可能性地点を求める．次にこの地点で速度が 0 になる走行モデルを作成する．この理由は，会合する可能性のある他の移動体がこの地点をいつ，どのように走行したとしてもその動きに矛盾なく対応できるよう，最悪ケースを想定して δt により模擬する追従走行区間を十分確保するためである．また，会合対象が静止体である場合，この地点で停止

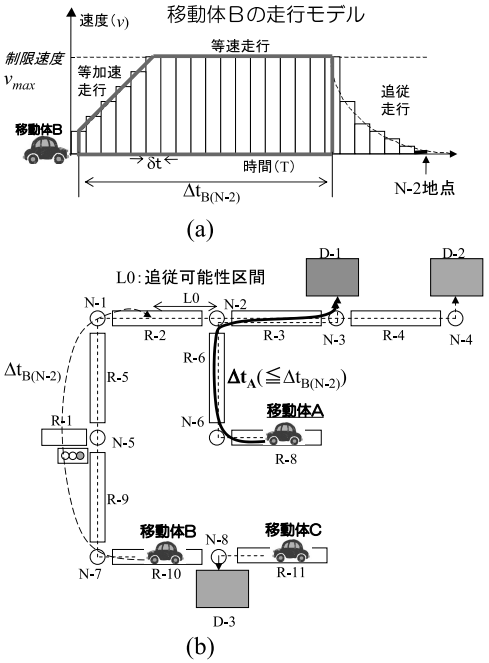

図 6 移動体 A への他主的 Δt 設定法の適用例
Fig. 6 Example of calculating Δt (Δt_a) for MO_A when Other-motivating Δt set scheme is applied.

する必要があるためである．

ここで，最寄地点が会合可能性地点である場合，次模擬時刻決定のための Δt の設定方法には主に以下の 2 つの方法が考えられる．

自主的 Δt 設定法： 当該会合可能性地点までの自身の走行モデルを作成して次の Δt および模擬時刻を決定する．

他主的 Δt 設定法： 自身の計画路上で会合可能性地点を持つ他の移動体に関して当該地点までの走行モデルを作成し，この走行モデルにより導出した Δt を用いて自身の次模擬時刻を決定する．

図 5 および**図 6** は移動体 A の次の Δt（$= \Delta t_A$）を決定する場合の，自主的 Δt 設定法および他主的 Δt 設定法の適用例である．自主的 Δt 設定法を用いた場合，堅実に Δt を決定するが他の移動体と会合する可能性がないときでも会合可能性地点付近で δt によるタイムステップで模擬する状態が多発するため非効率である．他方，他主的 Δt 設定法を用いた場合では，非効率なときはあるが，自主的 Δt 設定法のように δt によるタイムステップで模擬するような状態が多発することはない．この非効率なときとは，たとえば，図 6 (b) の状態で移動体 B の次の Δt を決定するときのことで，この場合，N-2 地点を基準とする移動体 A の走行モデルによる Δt が採用されてしまう．しかし，他主的 Δt 設定法では図 6 の例のように移動体 A の目的地 D-1 までの移動時間が，$\Delta t_{B(N-2)}$（移動体 B において N-2 を基準に追従走行になるまでの時間）よりも小さければ，移動体 A は 1 タイムステップで目的地まで移動できる可能性がある．この場合，移動体 A の走行モデルは，移動体 B の模擬を追従走行開始時点までとしているので，移動体 B との会合をいっさい考慮しなくてよいため，図 5 (a) において N-2 地点を D-1 地点に置き換えたものでよい．移動体 A が交差点で減速走行する場合でも，図 5 (a) で N-2 を D-1 に置き換えた走行モデルにより Δt を算出しておけば，この値（Δt）は減速走行を反映した走行モデルにより算出した Δt よりも小さくなり，D-1 地点に至るまでのタイムステップ数は増えてしまうが模擬結果が変わることはない．なお，会合対象が移動体の場合でも，減速走行を反映した走行モデルにより算出した Δt よりも小さい当該 Δt を用いていれば，実行性能は劣化するが，移動地点が追従走行区間に入ってしまうという問題は生じない．そのほか，他主的 Δt 設定法では移動体ごとに算出した Δt は互いに利用し合えるので，両設定法の Δt 算出コストは同程度である．これらの理由で，我々は他主的 Δt 設定法に基く実装方法を採用することとした．

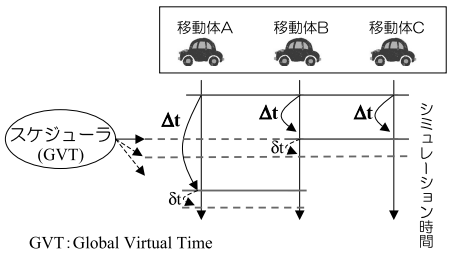

図 7 シミュレーション時刻進行機構
Fig. 7 Mechanism for advancing the simulation time.

なお，**図 7** は DTSS-RT での各移動体のシミュレーション時刻進行方法を示したものである．図 7 に示すように道路交通シミュレータには，シミュレータ全体を統括するスケジューラが存在し，当該スケジューラは本シミュレータで唯一の GVT（Global Virtual Time）を管理する．そして，この GVT を基準に各移動体が算出したタイムステップに基づく模擬時刻の小さいものから順番に，対応する移動体は模擬されることとなる．

3.2.3 アルゴリズム

ここでは，DTSS-RT での走行モデルの作成および タイムステップ Δt の設定方法を説明する．

図 8 に示すように，各移動体のタイムステップ Δt

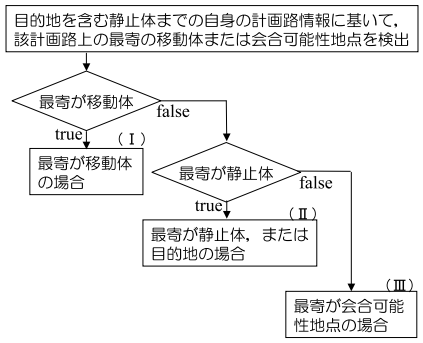

図 8 各移動体のタイムステップ Δt 設定の枠組み
Fig. 8 The framework for setting Δt of each MO.

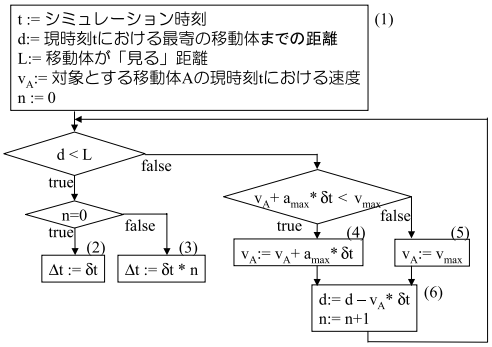

図 9 (I)：最近接物体が移動体の場合の走行モデル作成および Δt 設定のためのフローチャート
Fig. 9 (I): Flowchart for developing travel motion model and for setting Δt in the case where the nearest object is MO.

を設定するには，(I)〜(III) の場合がある．そして，(I) の最近接物体が移動体の場合についての，移動体 A の走行モデルの作成および Δt の設定方法に関するフローチャートを示したものが**図 9** である．この図 9 中の (4) は等加速走行，そして (5) は等速走行の速度を示し（図 5 (a) および図 6 (a) 参照），(3) で導出される Δt が当該移動体が会合状態でない場合のタイムステップとなる．また (1) は図 11，12 とも初期設定を示し，v_{max} は制限速度，a_{max} は最高加速度を示す．そして，**図 10** (a) は対象とする移動体 A の「見る」距離 L と，前方最寄の会合地点（移動体）までの距離 d，そして同図 (b) は前方最寄の会合可能性地点までの距離 d を説明したものである．他方，図 9 (2) で導出される Δt が会合状態である場合のタイムステップ δt となる．会合状態である場合は，たとえば，式 (1) および (2) に示す追従方程式を用いる．ここで，式 (1) は模擬対象の移動体と前方移動体との距離に対する両移動体の速度差から導出される δt 後の加速度，そして式 (2) は同移動体の δt 後の速度を示す．

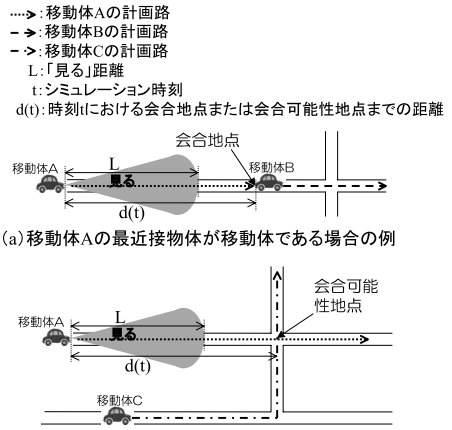

図 10 移動体の「見る」距離（L）と最寄の会合地点および会合可能性地点までの距離に関する説明図
Fig. 10 The distance MO looks ahead (L), and the distance to CP of SoD/pSoD.

$$a_A(t+\delta t) = c \times \frac{v_B(t) - v_A(t)}{\{x_B(t) - x_A(t)\}^m} \qquad (1)$$

$$v_A(t+\delta t) = a_A(t+\delta t) \times \delta t + v_A(t) \qquad (2)$$

A：模擬対象の移動体，B：A の前方の移動体または静止体，t：シミュレーション時刻，δt：タイムステップ，a：加速度，v：速度，x：移動体または静止体の位置，c，m：定数

この場合でも制限速度 v_{max} 以下になるように，式 (2) により導出した速度 $v_A(t+\delta t)$ が v_{max} を超えた場合は v_{max} に設定する．なお，式 (1) および (2) は米国連邦道路庁（FHWA）によって開発されたミクロシミュレーションシステム「TRAF-NETSIM」に使用された追従方程式であり，多くの道路交通シミュレータがこの式に基づいた追従モデルを採用している[14),15)]．

また，**図 11** は図 8 (II) の最近接物体が静止体である場合の，移動体 A の走行モデルの作成および Δt の設定方法を示したフローチャートである．この静止体には計画路の最終地点である目的地も含まれる．この場合，図 11 中の (2)〜(4) に至るまでのステップに示すように，等加速走行および等速走行だけでなく，静止体直前までの追従走行までも含めた区間を 1 タイムステップ Δt とすることができる．ここで，(2) での a_A はたとえば，式 (1) に示す追従方程式により導出された加速度を示す．

最後に，**図 12** は図 8 (III) の最近接物体が会合可能性地点である場合の Δt の設定方法を示したフローチャートである．この場合において，自身の計画路上に会合可能性地点が複数存在するときは，図 12 中の (3)〜(6) に示すように，各々に対応する他の移動体に

関して走行モデルの作成および追従走行開始時刻までの時間 Δt を計算し，その中から最小の Δt を選定することになる．なお，計画路上の遠くの位置で会合可能性地点を持つ他の移動体が存在し，この移動体により算出した Δt が小さい値であった場合，非常に非効率な実行となる．このような Δt 算出コストの増加や非効率な実行を避けるため，図 8 で示したように，自身の計画路上の最寄の移動体か静止体までの間の会合可能性地点だけに限定している．これは，計画路上に移動体か静止体が存在すればそれを基準に当該移動体は必然的に起動され模擬することになり，その時点でそれから先の会合可能性地点を評価すればよいからである．また，上記 (I)〜(III) での当該移動体 A の走行距離は，基本的に，v_A を Δt の間，積分することにより求まる．

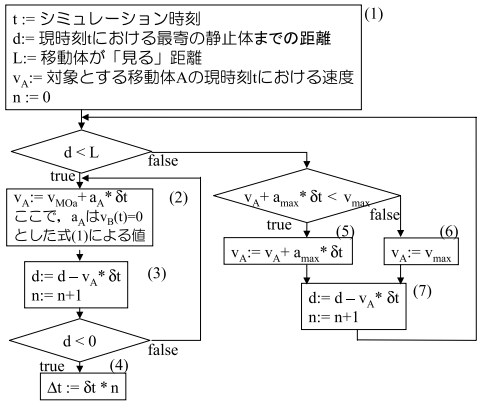

図 11 (II)：最近接物体が静止体の場合の走行モデル作成および Δt 設定のためのフローチャート

Fig. 11 (II): Flowchart for developing travel motion model and for setting Δt in the case where the nearest object is stationary object.

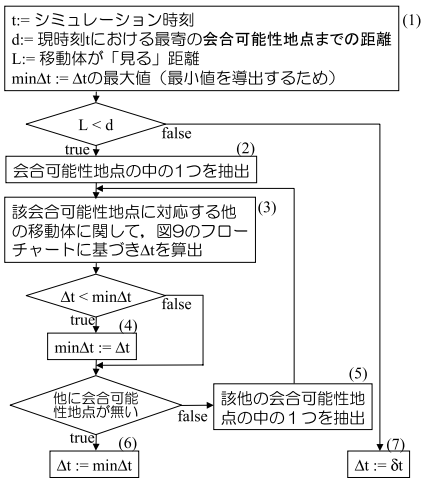

図 12 (III)：最近接物体が会合可能性地点の場合の走行モデル作成および Δt 設定のためのフローチャート

Fig. 12 (III): Flowchart for developing travel motion model and for setting Δt in the case where the nearest object is CP of pSoD.

4. 性能評価

我々は，DTSS-RT の応用例の 1 つとして様々なタイプの移動体が登場する災害シミュレーションを視野に入れている．以下では災害シミュレータの代表例としてロボカップレスキュー（RCR）シミュレータを取り上げ，その道路交通サブシミュレータ上での DTSS-RT および従来方式による実行性能の比較評価を行う．

4.1 RCR シミュレータの概要

図 13 は RCR シミュレータの基本部分の構成とビューア画面のサンプルを示したものである．本シミュレータの目的は，毎年開催されるロボカップ大会[16] において，参加チーム間で互いのレスキュー方法を競い合うツールとして利用することである．具体的には，参加チームがレスキューエージェント（救急隊，警察隊，消防隊）のレスキュー方法を実装し，どれだけの市民（被災民）を救助できたかを競うものである．各エージェントは，情報伝達を仲介するシミュレーション統括部（カーネル）を介して得られた周囲情報を利用して，次にとるべき行動・計画（計画路等）を作成する．作成した行動・計画情報はカーネルを介して，主に道路交通サブシミュレータに転送され，本サブシミュレータはこの情報に基づいて 1 分間（60 タイムステップ）分の模擬を行う．この繰返しによって，シミュレーションが進行する．対象場面は大震災直後の状況であり，初期段階で倒壊および道路閉鎖サブシミュレータにより建屋の倒壊による道路上への瓦礫散乱状態を模擬する．そして，その後の時間のほとんどを道路交通および火災延焼サブシミュレータにより災害現場の状況を模擬する．

また，当該道路交通サブシミュレータでの道路網は，図 4 に示すように，道路をリンク，交差点をノードと

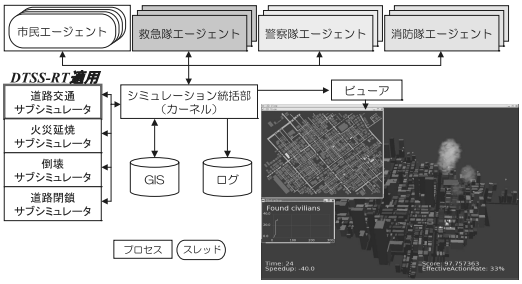

図 13 RCR シミュレータの基本構成

Fig. 13 Basic system configuration of RoboCup rescue simulator.

表 1 計算機環境
Table 1 Machine environment.

CPU	OS	Memory
Intel Pentium D (3 Ghz × 2)	Linux (RedHat9)/VMware on Windows-XP	2 GB

表 2 神戸市中央区に関する統計情報
Table 2 Statistical information for Kobe Chuou-ku.

総面積	交番・駐在所数	救急車数
27.84 km	20	3

する構造を採用している．道路上の車は車線単位で管理されており，ビル等の建屋は図 4 に示すようにノードを介在して道路網とつながっている．災害時ということで，信号は機能せず，すべてのレスキューエージェントは制限速度 20 km/h に規制されており，右左折時は減速せずに走行するモデルを採用している．このため，RCR の道路交通サブシミュレータの場合では，レスキューエージェントの移動距離は図 5，6 で説明した v-t グラフに基づく走行モデルでの面積と等しくなる．

4.2 計算機環境および設定条件

表 1 は性能評価で用いた計算機環境である．RCR シミュレータは Linux 環境上で動作するが，本評価の主目的は DTSS-RT と従来方式の性能比較であるため，作業効率を重視し，Windows 上で VMware により Linux 環境を構築した．

また，道路交通サブシミュレータ上での DTSS-RT の実行性能の特性解析も目的の 1 つである．火災延焼サブシミュレータを並行して実行させるとこの特性解析に悪影響を与えることになるため，火災延焼サブシミュレータおよび消防車は排除して評価することとした．以下は，本評価での道路交通サブシミュレータにおける設定条件である．

対象領域： 神戸中心街（中央区・三宮）の 500 m 四方

レスキューエージェント（移動体）数： 15
内訳はパトカー（警察）10 台と救急車 5 台であり，実際の競技で用いる台数を採用．神戸市中央区の統計情報[17)~19)]（**表 2**）から面積比で換算すると多めの数であるが，その他の車も考慮すると震災直後の状況としては妥当な車両数と考える．

シナリオ： 大震災直後の瓦礫撤去および救助活動
パトカー（警察）は道路上の瓦礫を順次除去．また，車両の走行に無関係な被災民は登場させないが，救急車は通行可能な道路を通り，被災民が閉じ込められている可能性がある建屋を順次捜索．

シミュレーション時間： 110 分間

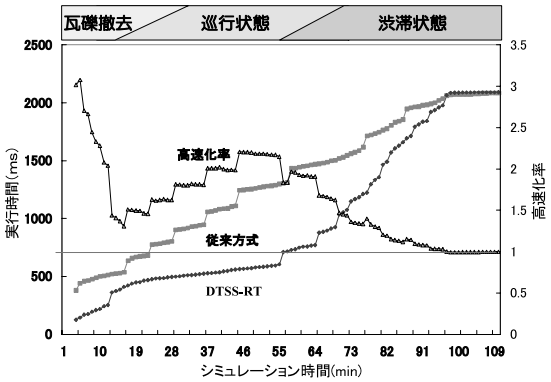

図 14 DTSS-RT と従来方式の性能比較結果
Fig. 14 Comparison result of execution times based on DTSS-RT and conventional method.

シナリオ（瓦礫撤去および被災民捜索）が完了するまでの時間．本シミュレータでは 1 タイムステップ (δt) = 1 秒間であるので，110 × 60 タイムステップ．

4.3 評価結果

図 14 は，DTSS-RT と $\delta t = 1$ 秒間とした従来方式の道路交通サブシミュレータ上での実行時間および，その従来方式に対する DTSS-RT の高速化率を示したものである．初期状態では瓦礫の撤去作業が主となり，会合対象が静止体であることから，追従走行区間までも含めて Δt を設定できるので高速化率が約 3 倍となる．そして瓦礫が取り除かれるにつれ，会合対象が移動体へと変わるので実行性能は劣化するが，スムーズに移動できる巡行状態に移行するにつれ約 2 倍程度の高速化が達成できる．しかしその後，移動体が連なってセンターに帰還するため，つねに渋滞状態となり性能が徐々に劣化していくことが確認できた．このことから，ひどい渋滞状態でなければ DTSS-RT は実行性能向上のための有効な方式であることが検証できた．また，同じ設定条件で，DTSS-RT による模擬結果と従来方式による模擬結果が同一であることも確認できた．すなわち，1 分間ごとの各移動体の状態およびシナリオ終了時点での各移動体の状態が両方式で同一であった．

5. 考　　察

本章では，前章の DTSS-RT による実行性能評価結果を粒度および問題規模の観点から考察する．ここでの粒度とは各移動体の模擬コストの大きさを示し，問題規模とは移動体の数を示すものとする．なお，RCR の道路交通サブシミュレータは Java により実装されているため，前章の結果はガーベッジコレクショ

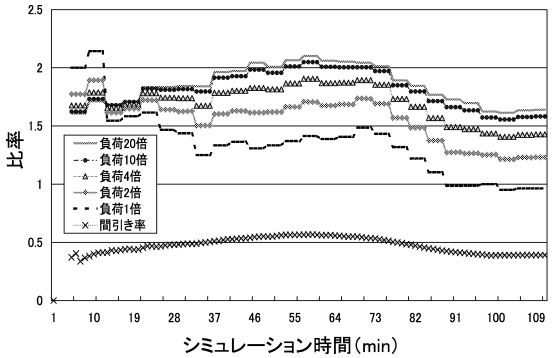

図 15 粒度（移動体の模擬コスト）を変えた場合の高速化率と間引き率の結果

Fig. 15 The results of speed up when the simulation cost of the movement is enlarged up to 20 times, and reduction ratio of simulation time steps.

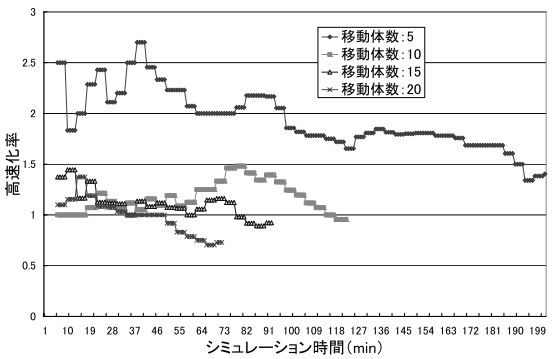

図 16 問題規模（移動体数）を変えた場合の結果

Fig. 16 Speed up ratio of the execution time for DTSS-RT to that for conventional method, when the number of MOs (police units) is changed from 5 to 20.

ン（GC）による擾乱が含まれたものであるが，ここでは実行時間への影響を詳細に解析するために，GC を別スレッドにして，GC の処理は実行時間から省いた．

5.1 粒度の影響解析

DTSS-RT では，各移動体は模擬時刻になると 1 タイムステップ分の行動模擬（「見て」，「判断」し「動く」）と，次の模擬時刻を決定するための Δt 算出を行う．DTSS-RT による実行性能が従来方式に勝るためには，この行動模擬コストが Δt 算出コストに比べて十分大きい状態であるべきだが，現状の RCR のレスキューエージェントは簡易モデルであるため，両コストは同程度の状態である．**図 15** は，この行動模擬コストを現状の 20 倍まで増やした場合の高速化率と間引き率を示したグラフである．この間引き率とは，従来方式での総タイムステップ数に対する，DTSS-RT により模擬を省けたタイムステップ数の比率の平均である．この図から，行動模擬コストを増やしていくと Δt 算出コストの影響が小さくなり，間引き率に応じた実行性能に近づいていくことが確認できる．これらの結果は，実行性能向上のための行動模擬コストと Δt 算出コストのトレードオフを行う指標となるため，今後有効活用できると考える．

また，行動模擬コストが比較的大きい応用では，多少 Δt 算出コストが増えても間引き率を上げる（Δt を大きくする）ような工夫を施すことが有効であり，たとえば以下のような方法が考えられる．

- 自主的 Δt 設定法と他主的 Δt 設定法の双方ともで Δt を計算し，大きい方の値を採用する．
- 交差点での移動体の減速走行等，実際の移動体の走行状態を反映した走行モデルを作成し，Δt を決定する．

5.2 問題規模の影響解析

ここでは，移動体数を変えてみた場合の実行性能への影響を考察する．なお，パトカーと救急車の台数配分が変わると，シナリオおよび実行時間に与える影響が変わることと，救急車だけでは瓦礫により走行および活動できない状態が多発することから，単独でも影響なく活動および走行できるパトカーのみに限定して本評価を実施した．

図 16 は移動体（パトカー）の数を 5〜20 に変えた場合の従来方式に対する DTSS-RT の高速化率を示したものである．この結果から，移動体数にほぼ反比例して高速化率が減少することが確認できた（シナリオ終了時の高速化率に着目した場合）．前節で述べたように粒度が大きくなれば，図 16 で示す高速化率も向上することになるが，計算機単体で高速化を図れる問題規模には限界があり，これらの結果に基づく，問題規模が与えられた場合での最適な計算機台数および該計算機への移動体のマッピング方法の検討は今後の課題となる．

そのほか，シナリオが終了するシミュレーション時間も移動体数にほぼ反比例して短くなる結果となった．これは，移動体（パトカー）が増えると瓦礫撤去作業が早く完了するためである．このような傾向は，問題規模を変えた場合の実行時間やシミュレーション時間の予測に役立つものである．

6. 関 連 研 究

道路交通シミュレータは計算機単体で動作するものがほとんどであるが，PARAMICS を代表とする並列・分散計算機環境上で動作するものもあり，すでに多くのものが研究開発されている[14),15),20)〜23)]．PARAMICS を含めほとんどのシミュレータは道路交通に関す

る応用研究が主であるため実装が容易な固定のタイムステップによるタイムステップ法を採用している．また，HLA 等の普及により異機種のシミュレータを統合して新たな付加価値を創出する動きもある．ドイツの Dresden 工科大および Otto-Guericke 大のチームは道路交通シミュレータ，人間が操作するドライビングシミュレータ，そして表示系を HLA で連接する統合シミュレーションシステム[24] を研究開発した．国内でも同様の研究がある[25),26)]．さらに同チームは計算機単体で動作する当該道路交通シミュレータ上でイベントベース法とタイムステップ法との比較研究をも行い，実行性能の優劣はイベントベース法におけるイベント発生時刻の算出コストの大きさと，不要な模擬を省ける効果とのトレードオフに依存するとしている[27)]．

本論文で提案した方式はタイムステップ法をベースとするが，タイムステップの大きさをイベントベース法の概念に基づいて設定するものである．ただし，イベント発生時刻を正確に算出するのではなく，最悪ケースを想定してイベント発生の可能性が生じる時刻を小さな演算量で求め，その可能性の生じる時間帯はタイムステップ法により実行することでイベント発生時刻の問題を解決するというハイブリッド方式である．さらに本方式は計画路情報を利用することにより，タイムステップをより大きくし，かつ効率的にタイムステップを設定することを可能とするものである．

7. おわりに

移動体が自由空間を移動する航空機等を対象とした応用問題では，各移動体は巡航速度により定速で移動する状況が多いため，動的タイムステップ制御法（DTSS）の適用による実行性能向上は容易に実現できる[9)]．しかし，移動体が道路網上を移動する場合では，移動体間の距離が比較的短いため，加減速走行や追従走行の状態が多く，DTSS の適用による実行性能向上は難しい．本論文では，DTSS を道路交通シミュレーションへ適用するための効率的な実行方式 DTSS-RT を提案した．DTSS-RT の主な特長は以下の 3 点である．

- 移動体間の同期コストを低減させるだけでなく，移動体自身の模擬コストも低減させることが可能．
- 各移動体の計画路情報に基づいた移動体間の会合可能性地点および会合地点を利用することにより，次の模擬時刻を効率的に設定可能．
- 細かい固定の δt によるタイムステップにより模擬する従来方式と同一の模擬結果を実現．

ただし，DTSS-RT では，計画路情報に変更が生じたときは一連のプロセスを再計算し直すこととなり，この頻度が多い場合は実行性能劣化の主要因となる．しかし，本論文で応用例として取り上げた RCR シミュレータの場合，1 分間に 1 回のタイミングで，エージェントは計画路を変更することがあるが，道路交通サブシミュレータで 1 分間分の模擬を実行している間は計画路が変更されないので，このようなタイプのシミュレータでは問題ない．DTSS-RT を RCR の道路交通サブシミュレータへ適用した結果，瓦礫撤去作業時で約 3 倍，巡行走行時で約 2 倍の性能向上が達成できた．しかし，渋滞時ではタイムステップ Δt が大きくとれず，Δt 算出コストがオーバヘッドとなり実行性能が劣化するため，このような状況での性能改善が今後の課題となる．このような状況が多発するようなシナリオでは，従来方式に適宜切り換える折衷方式等を検討している．また，行動模擬コストが大きい応用を対象に Δt をさらに大きくするような方式検討および評価と，計算機クラスタ環境への DTSS-RT の効率的な実装方法の検討も今後の課題である．

なお，防災分野では救助計画，防衛分野では作戦計画等に基づき各移動体は行動するものであり，日常生活でも基本的に目的地へ向かうまでの計画を立てて行動するものである．したがって，計画路情報を活用する DTSS-RT は多くの応用問題へ適用できる可能性が高い．たとえば，多数の航空機が登場し，各航空機の航空路線が決まっている航空管制シミュレータ等へも効果的に適用できる可能性が高く，道路交通以外の応用問題への DTSS-RT の適用検討も今後の課題である．

参考文献

1) DMSO (Defense Modeling and Simulation Office). https://www.dmso.mil/public/transition/hla/
2) Dahmann, J.S., Fujimoto, R.M. and Weatherly, M.: The department of defense high level architecture, *1997 Winter Simulation Conference*, pp.142–149 (Dec. 1997).
3) IEEE Std 1516-2000 IEEE Standard for Modeling & Simulation (M&S) High Level Architecture (HLA) — Framework and Rules.
4) 尾崎敦夫，古市昌一，西乃武夫，黒田悦司：並列分散型高精度交通シミュレーションシステムの実現と評価，情報処理学会論文誌，Vol.41, No.5, pp.1587–1596 (2000).
5) Ozaki, A., Furuichi, M., Nishi, N. and Kuroda, E.: The Use of High Level Architecture in Car

Traffic Simulations, *IEICE Trans.*, Vol.E83-D, No.10, pp.1851–1859 (2000).
6) Ozaki, A., Furuichi, M., Takahashi, K. and Matsukawa, H.: Design and Implementation of Parallel and Distributed Wargame Simulation System and Its Evaluation, Vol.E84-D, No.10, pp.1376–1384 (2001).
7) 古市昌一，尾崎敦夫，松川 仁：HLA をベースとした並列分散シミュレーション開発実行支援環境の実現と評価，電子情報通信学会論文誌，Vol.J84-D-I, No.12, pp.1610–1622 (2001).
8) 尾崎敦夫，渡部修介，宮沢 稔，古市昌一，佐藤裕幸：移動物体を対象とした分散シミュレーション時刻同期方式，コンカレント工学研究会 (2006).
9) Ozaki, A., Shiraishi, M., Watanabe, S., Miyazawa, M., Furuichi, M. and Sato, H.: Event-Aware Dynamic Time Step Synchronization Method for Distributed Moving Object Simulation, *IEICE Trans.*, Vol.E89-A, No.11, pp.3175–3184 (2006).
10) Furuichi, M., et al.: MARS: A M&S Framework for Large Scale Simulation Based on the HLA, *Fall 2005 Simulation Interoperability Workshop*, 05F-SIW-033 (Sep. 2005).
11) RoboCup-Rescue Official. http://www.rescuesystem.org/robocuprescue/
12) Fujimoto, R.M.: Parallel discrete event simulation, *Comm. ACM*, Vol.33, No.10, pp.30–53 (1990).
13) Fujimoto, R.M.: Parallel and distributed discrete event simulation: Algorithms and applications, *1993 Winter Simulation Conference*, pp.106–114 (Dec. 1993).
14) 吉川康雄，森田育宏，香月伸一，羽藤英二，貴志泰久：渋滞改善のための交通流ミクロシミュレータの開発，第 14 回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.189–192 (1995).
15) 後藤幸夫，駒谷喜代俊，福田豊生：自律型走行モデルによる道路交通シミュレータの開発，電気学会論文誌 D，Vol.116, No.5, pp.569–577 (1996).
16) RoboCup. http://www.robocup.org/
17) 神戸市企画調整局：神戸の統計. http://www.city.kobe.jp/cityoffice/06/013/toukei/
18) 兵庫県警察. http://www.police.pref.hyogo.jp/index.htm
19) 神戸市消防局. http://www.city.kobe.jp/cityoffice/48/
20) 石川 亮，本多中二，風間 洋，猪飼國夫：微視的道路交通シミュレータ MITRAM による広域交通解析，情報処理学会研究報告：数理モデル化と問題解決，Vol.2004, No.130, pp.29–32 (2004).
21) 尾崎敦夫，古市昌一，阿部一裕，中島克人，田中秀俊：大規模並列交通シミュレータの実現と負荷分散方式の評価，情報処理学会論文誌，Vol.40, No.6, pp.2810–2818 (1999).
22) Cameron, G., Wylie, B.J.N. and McArthur, D.: PARAMICS —moving vehicles on the connection machine, *Super Computing'94*, pp.291–300 (1994).
23) Liu, H.X., Wenteng, M., Jayakrishnan, R. and Recker, W.: Distributed large-scale network modeling with paramics implementation, *IEEE conference on Intelligent Transportation Systems*, pp.232–238 (Sep. 2005).
24) Klein, U., Schulze, T. and Strassburger, S.: Distributed Traffic Simulation based on the High Level Architecture, *Fall 1998 Simulation Interoperability Workshop*, 98F-SIW-016 (1998).
25) 志磨 健，高橋和範，中島憲宏：ITS シミュレータの開発，第 19 回シミュレーション・テクノロジー・コンファレンス発表論文集，pp.23–26 (2000).
26) 伊川雅彦，後藤幸夫，熊澤宏之，古澤春樹：異種分散シミュレーションによる ITS 開発環境の構築，情報処理学会論文誌，Vol.45, No.12, pp.2805–2814 (2004).
27) Schulze, T. and Fliess, T.: Urban Traffic Simulation with Psycho-physical Vehicle-following models, *1997 Winter Simulation Conference*, pp.1222–1229 (1997).

(平成 19 年 1 月 22 日受付)
(平成 19 年 5 月 7 日採録)

尾崎 敦夫（正会員）

昭和 63 年九州工業大学工学部情報工学科卒業．平成 2 年同大学院電気工学研究科計算機コース博士課程前期修了．同年より現在まで主に三菱電機株式会社情報電子研究所（現，同社情報技術総合研究所）にて並列・分散処理技術およびモデリング&シミュレーション技術に関する研究開発に従事．この間，平成 4 年〜8 年 RWC（Real World Computing）プロジェクトに参画し，並列シミュレーションの研究開発に従事．平成 8 年 ESS96（8th European Simulation Symposium）Best Paper Award，平成 18 年電子情報通信学会コンカレント工学研究会優秀論文賞各受賞．博士（情報工学）．電子情報通信学会，IEEE 各会員．

松下　和隆（正会員）

昭和 56 年金沢大学理学部物理学科卒業．同年より現在まで三菱電機インフォメーションシステムズ株式会社にて基本ソフトウェアの開発に従事．主に言語処理系，推論支援ツール，および並列分散シミュレータの開発に従事．日本ソフトウェア科学会準会員．

白石　　將（正会員）

平成 6 年東京大学大学院工学研究科電子工学専攻修士課程修了．同年三菱電機（株）情報システム研究所（現，同社情報技術総合研究所）に入社，入社後は最適化，データ解析，並列分散処理に関する研究開発に従事．

渡部　修介（正会員）

平成 2 年図書館情報大学（現筑波大学図書館情報専門学群）卒業．同年三菱電機（株）情報電子研究所（現，同社情報技術総合研究所）に入社，入社後は同社研究所および製作所にて主に分散処理に関する研究・製品開発に従事し，平成 13 年より同社鎌倉製作所にて並列分散シミュレーションによるモデリング&シミュレーションシステムの事業化に従事．

古市　昌一（正会員）

昭和 57 年広島大学総合科学部卒業．同年三菱電機（株）情報電子研究所（現，同社情報技術総合研究所）に入社，平成 4 年〜6 年にイリノイ大学へ留学し修士課程修了，平成 13 年〜16 年に慶應義塾大学大学院博士後期課程に在籍し博士（工学）を取得．入社以来並列推論計算機の研究を経て並列分散シミュレーションの基盤と応用技術の研究に従事し，平成 14 年からは同社鎌倉製作所にてモデリング&シミュレーションシステムの事業化に従事．IEEE，ACM，電子情報通信学会等各会員．

佐藤　裕幸（正会員）

昭和 34 年生．昭和 57 年筑波大学第三学群情報学類卒業．同年三菱電機（株）入社．昭和 60 年 6 月〜平成元年 4 月（財）新世代コンピュータ技術開発機構に出向し，逐次型および並列型推論マシンのシステムソフトウェアの研究開発に従事．現在，三菱電機（株）情報技術総合研究所にて，並列処理ソフトウェア，最適化システムの研究開発に従事．博士（工学）．