

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

尾崎 敦夫[†] 古市 昌一[†]
西 乃 武夫^{††} 黒田 悦司^{††}

我々は道路交通シミュレーションにおいて実世界の観測値を活用し、模擬精度を高めるための研究を行っている。模擬精度を高める方式としては、感知器を通して得られる観測値とシミュレーション結果が等しくなるようにシミュレーションで使用するパラメータを調節する方式を考案した。そして、分散計算機システム上に本方式を実装した。また、実装では、分散シミュレーション接続仕様 HLA (High Level Architecture) を利用したシステム開発を行った。本論文では、問題規模が大きくなる場合に必要となる並列処理の性能、および並列処理と模擬精度との関係について述べる。結果としては、問題規模が大きくなるほど負荷がバランスし、これが実行性能の向上だけでなく、模擬精度の劣化の度合を抑えることにもつながることが分かった。また、実行性能向上のための手法、およびその適用効果に関して説明する。

Design and Implementation of High Accurated Car Traffic Simulation on Parallel and Distributed System and Its Evaluation

ATSUO OZAKI,[†] MASAKAZU FURUICHI,[†] NOBUO NISHI^{††}
and ETSUJI KURODA^{††}

We have been researching for a fusion algorithm in Car Traffic Simulation (CTS), which is to improve the simulation's accuracy by using the real world's observed data. The basic concept of the fusion algorithm is to control the parameters, which are used in the CTS, in order to adjust the simulation to the road monitoring sensor data. HLA (High Level Architecture) interface has been employed to develop the CTS on the workstation cluster system. In this paper we discuss the design of HLA based CTS system and the preliminary evaluation results of this system. The results show that the system performance can be enhanced without degrading the accuracy of CTS, specially when the problem size is enlarged. The technique for enhancing the system performance and its effect have been also presented.

1. はじめに

近年、全世界的に技術協業がさかんに行われ、日本国内においても官・学・民協力のもと推めている高度道路交通システム (ITS: Intelligent Transport Systems) は、安全性、輸送効率、快適性の向上と環境への配慮を目的としたプロジェクトである。ITS の大きなテーマの 1 つは、交通渋滞の解消である。この解決策として、的確な信号制御、ドライバへの旅行時間情報の提供¹⁾ 等が考えられるが、どのような解決策においても対象とする道路の交通状況を正確かつ詳細に把握する必要がある。

このため、我々は実世界から得られた観測値を活用

して、対象とする道路の交通状況を精度良く模擬するシミュレーション技術の実現が重要であると考えた。そして、この技術を確立するべく、観測値が得られるたびに、シミュレーションが観測値に整合するように、シミュレータで使用するパラメータを調節する手法に関して研究を行ってきた^{2),3)}。ここでの研究課題は、どのパラメータ要素が、系全体にどのような影響を与えるかを解析すると同時に、どのようなパラメータ要素群を調節の対象に選べば、模擬する車群の挙動を観測値に効率良く近付けることができるかを解明することである。なお、道路交通における観測値とは、たとえば感知器を通して得られる感知器設置地点での統計データ (“車両通過台数” や “平均速度” 等) のことであり、本データは実際には 5 分間の周期で得ることができる。5 分間隔という短い時間刻みで感知器設置地点における観測値との整合を行い、高い模擬精度を実現するシミュレータは、観測値が得られない道路箇所

[†] 三菱電機株式会社情報技術総合研究所
Information R&D Center, Mitsubishi Electric Corporation

^{††} 三菱電機株式会社鎌倉製作所

Kamakura Works, Mitsubishi Electric Corporation

においても高い精度でその道路状況を模擬できていると考える。このため、本シミュレータを活用することにより、対象とする道路全体を考慮した的確な実時間信号制御や信頼性の高い旅行時間情報の提供等が可能になる。

また、本シミュレーションシステムでの模擬処理は、5分間という実時間性が要求されるため、高速に最適なパラメータ群を選出する必要がある。特に模擬対象となる車の台数が多くなるような場合には、複数の計算機を用いた並列・分散処理が不可欠となる。また、様々な応用への適用を目指す本システムは、高い拡張性や再利用性が要求されるため、開発する際には機能ごとにモジュール化すると同時に、各モジュール間を標準仕様のインタフェースにより実現しておくことが重要である。このため、ワークステーション (WS) をネットワーク接続した分散計算機システム上で、西暦2000年にIEEE標準となる分散シミュレーション接続仕様HLA (High Level Architecture)⁴⁾を利用したシステム開発を行った。

なお、並列交通シミュレータはすでにいくつかのものが開発されており^{5)~7)}、都市開発を計画する際の利用や環境への影響を解析することを目的としたものとしてはPARAMICS⁵⁾が存在するが、実世界の観測値を活用して模擬精度を高めることが可能なシミュレータは存在しない。また、HLAは異機種シミュレータ間の相互接続性の確保を本来の目的としたものであり、すでに多くの開発事例⁸⁾があるが、並列・分散シミュレーションのための実行基盤としてHLAを利用した事例はない。

本論文では、シミュレーションパラメータ調節方式の紹介と、分散計算機システム上への本方式の実装および本システムの性能評価について述べるとともに、並列処理と模擬精度との関係について考察する。以下、2章ではシミュレーションパラメータ調節方式について説明し、3章ではHLAの概要について述べる。そして4章で、分散計算機システム上でHLA仕様を基本に開発した並列分散型交通シミュレーションシステムの実現方式について説明する。5章では本システムにおける並列処理方式の基本性能と、並列処理と模擬精度との関係を報告し、6章ではこの並列処理方式における実行性能向上のための手法について考察する。

2. シミュレーションパラメータ調節方式

我々が目指す交通シミュレーションシステムは、渋滞解消を目的とする信号制御や、旅行時間情報の提供¹⁾を主な応用としている。渋滞のできはじめや渋滞

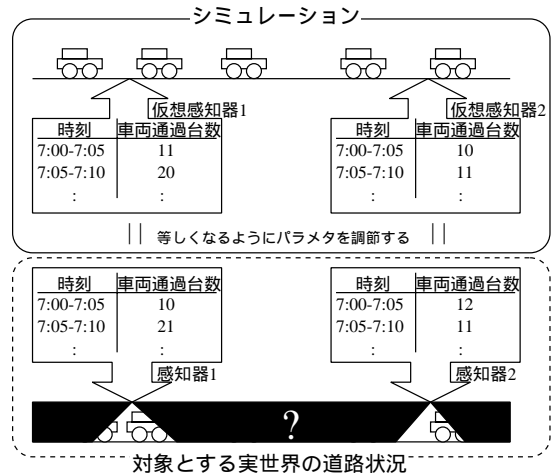


図1 シミュレーションパラメータ調節方式

Fig. 1 Concept of the fusion algorithm.

解消時の道路状況のようなモデル化が困難である問題にはマイクロモデル^{7),9)}を基本とする実現方式が考えられる。このため、本システムではマイクロモデルを基本とするシミュレーションを採用した。模擬精度を高める手法としては、感知器を通して得られる観測値とシミュレーションによって模擬される車の挙動が等しくなるように、このマイクロモデルシミュレータで使用するパラメータを調節する方式を考案した^{2),3)}。具体的には、シミュレータ内においても実際と同様の位置に仮想感知器を設定し、5分間ごとに算出した統計データが、実際の観測値 (統計データ) と等しくなるように、シミュレータで使用するパラメータを調節するものである (図1参照)。

本方式において選んだパラメータは、以下の車両追従方程式、ドライバの視野範囲、そして車両流入分布に関するものである。

車両追従方程式 (c, m): 式 (1) に示す車両追従方程式を採用し、この式の中に含まれる c と m を、調節対象のパラメータとした。式 (1) を基本とする車両追従モデルは、米国連邦道路庁 (FHWA) によって開発されたマイクロシミュレーションシステム「TRAF-NETSIM」に使用されたものであり、本モデルを基本とする交通シミュレータは渋滞現象等を高精度で模擬できる¹⁰⁾。

$$\dot{X}_s(t + \Delta t) = c \cdot \frac{\dot{X}_f(t) - \dot{X}_s(t)}{\{X_f(t) - X_s(t)\}^m} \quad (1)$$

ここで、 t は時刻、 Δt はシミュレーション時刻の刻み幅、 X は車両の位置座標、 s は模擬対象車、 f は s の前方車である。

ドライバの視野範囲 (Scope): ドライバが前方を

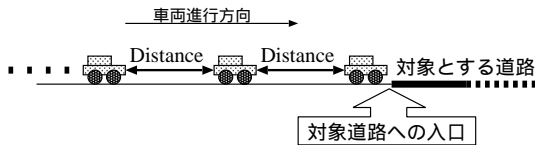


図2 Distanceによる車両流入分布の調節

Fig.2 The Distance between cars when they enter the road, which controls a distribution of cars on the road.

見る範囲を規定するものであり、基本的に走行速度に比例する。設定した限界スピードまで加速を続ける加速処理と追従処理(式(1)の適用)の切り分けに使用する。

車両流入分布 (Distance): 対象とする道路へ車両が流入するときの車間距離を調節するものであり、このパラメータを調節することにより車両流入分布を変えることができる(図2参照)。

各パラメータがシミュレーションに及ぼす影響はすでに解析済みであり³⁾、各々を適宜調節することによってシミュレーションを観測値に近付けることをはかる。我々が用いたパラメータ選定方式²⁾は、対象とする道路のラッシュアワーやランチャイム等の時間帯ごとに、c, m, Scopeの値は各々異なるが、この時間帯内ではc, m, Scopeの値は各々固定値になるという仮定に基づいている。このため、本方式では対象とする時間帯の道路状況に合ったc, m, Scopeの値を事前に選出しておき、5分ごとにDistanceのみを変えることによって車両流入分布を調節し、観測値との整合をはかることを基本としている。

朝のラッシュアワーを対象に、実際のフィールドデータを用いて実施したシミュレーションパラメータ調節方式の評価実験では、整合対象となる感知器設置地点以外の地点においても、最大約1.5%の誤差で実際の観測値にシミュレーションが整合できていることが検証できた²⁾。

3. HLA: High Level Architecture

HLAは分散シミュレーションシステムにおけるシミュレータ間のデータ交換や論理時刻の同期方式を共通化するための接続仕様である。HLAでは、分散シミュレーション全体をフェデレーションと呼び、各々独立して動作可能なシミュレータをフェデレートと呼ぶ。各フェデレートはそれぞれHLAインタフェース(HLA I/F)を備え、RTI(Run Time Infrastructure)と呼ばれる中核のソフトウェアとメッセージの交換を行う。各フェデレートは複数のオブジェクトクラス(以下、HLAオブジェクトと呼ぶ)を持つこと

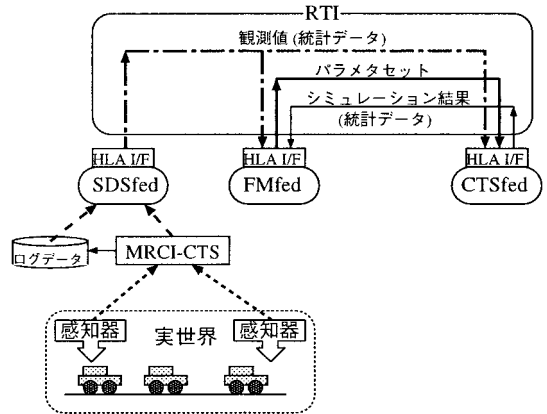


図3 モジュールの基本構成

Fig.3 Basic module structure in CTS system.

ができる。HLAオブジェクトの属性情報は、各フェデレートがRTIに対して公開および購読を宣言することにより、RTIの介入によって公開側から購読側へ定期的送信される。また、非定期的フェデレート間で情報交換を行う場合は、インタラクションクラスと称する通信手段を用いる方法がある⁴⁾。

4. HLAを基本とする並列分散型シミュレーションシステム

我々は、交通シミュレーションシステムの開発において、様々な応用への適用性や、並列・分散処理の実現の容易性を高めるだけでなく、たとえば将来考えられる気象シミュレータ等の他分野のシミュレータとの接続性までも高めておくことが重要と考え、HLA仕様に基づくシステム開発を行った。本章では、分散計算機システム上でシミュレーションパラメータ調節方式を実現するためのHLAを基本とするシステム構成と、並列・分散処理の実現方式について説明する。ここで、本システムは複数のWSから構成される分散計算機システム上で問題をうまく分割し、各WSに割り付けて並列処理させることから並列分散型と称することとした。

4.1 システムの基本構成

4.1.1 モジュール構成

機能ごとにモジュール化することを目的に、構築するシミュレーションシステムは次の3つのフェデレートから構成するものとした(図3参照)。また、各フェデレートは1つのHLAオブジェクトからなり、フェデレート間通信は公開と購読の宣言による定期的なデータ送信により行うものとした。

Sensor Data Surrogate federate (SDSfed):

5分ごとに感知器を通して得られた観測値(統計データ)を, FMfedと CTSfedに送信する役割を担う. また, 図3に示すように取得済みの統計データ(ログデータ)を使用して, アルゴリズムおよびシステムの動作検証等に使用することも可能である. ここで, MRCI-CTS(Modular Reconfigurable Communication Interface for CTS)は, 実システムである感知器からのデータをシミュレーションシステムへ転送するためのインタフェース機能を備えた装置である.

Car Traffic Simulation federate (CTSfed):

シミュレーションを担当するモジュール. SDSfedからの統計データを基に, FMfed から送られてきたパラメータセットを用いて規定された時間分(5分間分)のシミュレーションを行う. そして, この5分間分のシミュレーションを通して, 仮想感知器設置地点での統計データを算出し, FMfedへ送る.

Federation Manager federate (FMfed): 各モジュールを統括する役割を担う. 具体的には, CTSfed が計算した統計データの結果と, SDSfed 経由で受信した観測値(統計データ)を比較して, 両者が等しくなるように, より適したパラメータセットを解析し, そのパラメータセットを再び CTSfed へ送る処理を行う. この処理は時間の許す限り(感知器から次の観測値が届くまでの間), 何度でも繰り返すことが可能である.

このように各機能を標準仕様(HLA)の独立したモジュールとして開発することにより, 異なるタイプの CTSfed との交換, FMfed の交換による様々なタイプのパラメータ抽出アルゴリズムの試行, そして他の HLA 仕様のシミュレータとの接続等が比較的容易に実現できる.

4.1.2 シミュレーションパラメータ調節アルゴリズム

図4は各モジュールの処理の流れを示したものであり, モジュール間の矢印は RTI を介したデータの流れを示している. 以下のステップは, 感知器から観測値を得てから次の観測値が届くまでの間(これを1サイクルと呼ぶことにする. 図4の“One cycle”に相当)に, 我々が用いたパラメータ選定方式に基づいて, システムが行う基本的な処理の流れを示したものである.

step 0 (Initialization): SDSfed は, 感知器を通して得た観測値(統計データ)を FMfed および CTSfed へ送る.

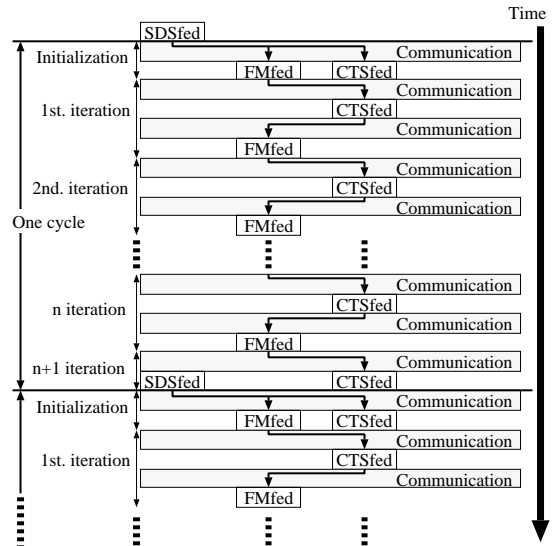


図4 シミュレーションパラメータ調節方式の処理の流れ
Fig. 4 Process and data flow in the fusion algorithm.

step 1 (1st. iteration):

- (1) FMfed より, 事前に選定してあるパラメータ c , m , $Scope$ と, $Distance$ がとりうる最小値を CTSfed へ送る.
- (2) CTSfed では step0 で得たデータと, FMfed から送られてきたパラメータセットを使用して対象とする道路の5分間分の模擬を行う. そして, この模擬を通して, 仮想感知器設置地点での統計データを計算し, この結果を FMfed へ送る.
- (3) CTSfed より模擬結果(統計データ)を受けとった FMfed は, その結果と step0 で得た観測値を比較する. そして, 観測値との整合性がより高くなる模擬結果を導出できると思われる, 前回(1)の値より大きい $Distance$ を再選出する.

step 2 (2nd. iteration):

- (1) FMfed より前ステップ(この場合 step 1(3))で選出したパラメータセットを, CTSfed へ送る.
- (2) step 1(2)と同様の処理を行う.
- (3) step 1(3)と同様の処理を行う. 基本的に, 今まで選出した $Distance$ から導出された模擬結果と観測値を比較して, 整合性がより高くなる模擬結果を導出できると思われる新たな $Distance$ を再選出する.

step 3 ~ step n (3rd. iteration ~ n iteration):

step2の処理を繰り返す.

step n+1 (n+1 iteration):

- (1) FMfed において, step 1 ~ step n で試行したパラメータセットの中で, 最も step 0 で得たデー

次に近い結果を導出してきた最適なパラメータセットを CTSfed へ送る。

- (2) CTSfed では、この最適なパラメータセットを用いて 5 分間の模擬を行う。本処理は、次のサイクルのために対象とする道路を初期化しておくためのものである。

以上の処理 (step 0 ~ step $n+1$) は、新たな観測値が得られるたびに実行される。

4.2 パラメータ空間の並行探索方式

より現実に近い道路状況を模擬しようとする場合は、限られた 5 分間の間にできるだけ多くのパラメータセットを試すことにより、対象とするパラメータ空間を細かく調べ、最適なパラメータセットを選出する必要がある。このため、要求される精度および試したいパラメータセットの数に応じて、CTSfed を増やし、対象とするパラメータ空間を並行探索することが考えられる。図 5 は CTSfed を実行する計算機を複数台使用する場合のシステム構成例を示したものである。

また、この場合の処理の流れを示したものが図 6 である。基本的には、CTSfed の数を増やしただけで 4.1.2 項で示したアルゴリズムと同等のものであるが、

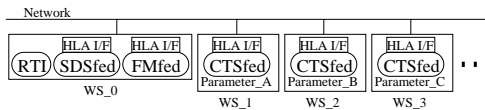


図 5 CTSfed 実行用計算機を複数台使用する場合のシステム構成例

Fig. 5 An example of system structure when multiple WS (CTSfed) is used.

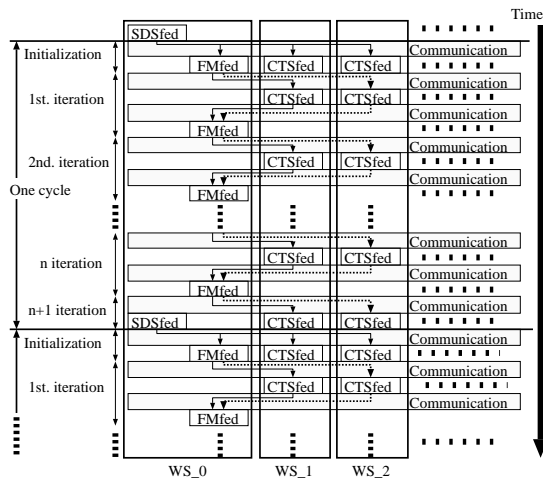


図 6 複数の CTSfed を用いてパラメータ空間を並行探索する場合の処理の流れ

Fig. 6 Parallel retrieve on a parameter space by using multiple CTSfed.

イタレーション (図 6 : iteration) ごとに、FMfed は各々の CTSfed に異なるパラメータセットを送信し、また各々から異なる模擬結果 (統計データ) を受信することとなる。我々が用いたパラメータ選定方式において、図 6 のように $N (> 1)$ 台の CTSfed を用いる場合には、Distance のとりうる範囲を N 等分し、各々を初期値として並行検索することとなる。そして、各 CTSfed からの結果に基づき、最も観測値に整合できたパラメータ値の近辺を再度 N 等分し、同様にこのイタレーションを繰り返すことにより、高い整合性を実現できるパラメータを求めていくこととなる。しかし、ラッシュアワーやランチタイム等の時間帯の初めや終り (道路状況の特徴が変化する時刻) 付近では Distance のみならず、 c 、 m 、Scope のいずれかも同様の方式で状況に応じた調節³⁾が必要となるため、このような並行探索による高速化は特に必須となる。

3 km の道路においてつねに車が 200 台存在するという渋滞した状況 (朝のラッシュアワー) を対象に実施した性能評価では、5 分間に 100 組のパラメータセットを実行する場合は 2 台、500 組の場合では 8 台の CTSfed 実行用 WS が必要となるという結果が得られている²⁾。

4.3 領域分割による並列処理方式

対象とするシミュレーション領域が大きいために模擬する車の台数が非常に多くなる場合や、高い模擬精度が要求されるため、各車を詳細に模擬しなければならない場合等では、計算量が非常に多くなる。本システムでは、この処理を実時間処理する必要があるため、分散計算機システム上で複数の WS により並列処理し、高速化をはかる方式を採用した。

基本的に対象とするシミュレーション領域は固定であり、車はその対象領域に入ったり出たりするために、車オブジェクトの生成/消滅処理は頻繁に行われることになる。また、車オブジェクトはどの道路を走行し

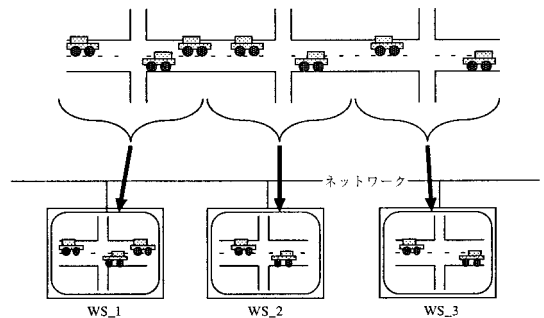


図 7 領域分割による並列処理の実現

Fig. 7 Parallel processing by dividing a simulation space.

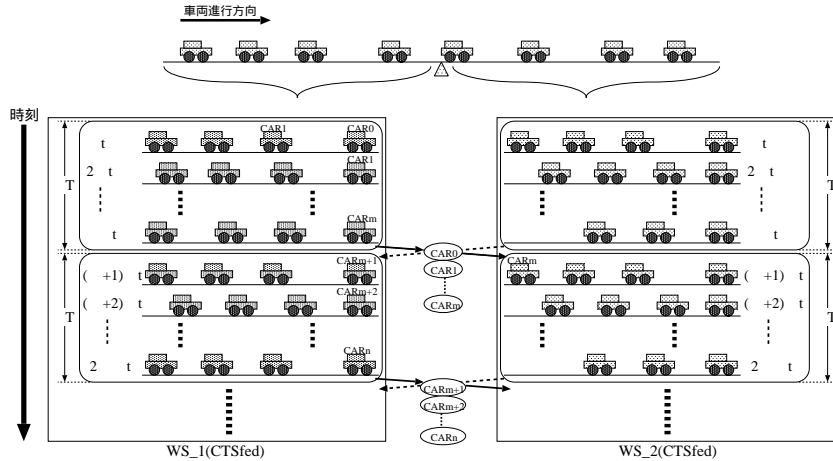


図 8 並列処理方式

Fig. 8 Parallel processing.

ていようとつねにその車の近傍の状況を必要とする。このため、並列処理を実現するにあたり、オブジェクト並列で実現するよりは、領域分割による並列処理⁷⁾が適していると考えられる(図 7 参照)。

領域分割による並列処理を実現する際に課題となるのが、車が部分領域間を通過する場合や、前方車が自車と異なる部分領域に存在する場合に必要な WS 間での通信処理である。シミュレーションの時刻進行の単位を Δt とすると、部分領域間での情報交換や、車が部分領域間を通過する際の処理もこの Δt の頻度で処理することが好ましい。なぜなら、本来 Δt の頻度で各車は情報交換を行い動作するようにモデル化されており、この頻度を粗くすると、本来意図する車の動作、および系全体の挙動に誤差が生じ、“模擬精度の劣化”につながるからである。しかし、多くの応用問題には満たさなければならない実行性能が課せられている。このため、要求される模擬精度を満たす許容範囲内であれば、この通信頻度を適度に粗くして、実行性能を高めることが考えられる。本システムでは、通信頻度 λ (正の整数) を導入して、部分領域間での情報交換や、車が部分領域間を通過する際の処理の間隔を $\Delta T (= \lambda \times \Delta t)$ で行うように実装した(図 8 参照)。このため、 λ が大きくなるほど模擬精度は劣化することとなる。

たとえば、片側車線の道路(“単路”と称する)において、前方車の情報を必要とし、その車が自車と異なる部分領域(WS)へ移動してしまっている場合、すなわち図 8 に示す時刻 $(\lambda + 1)\Delta t$ において CAR_{m+1} が、前方車である CAR_m の状態値を必要とする場合は、 CAR_{m+1} は時刻 $\lambda\Delta t$ のときの CAR_m の値を使

用することとなる。この場合、 CAR_m は移動先の WS (図 8 の WS.2) においても同じ状態等で等速運動を行っているものと見なして処理することとなる。また、任意の部分領域を通過した車でも、 ΔT の間は、その領域を担当する WS が処理することとなる。すなわち、図 8 の例では、WS.1 はシミュレーション時刻 Δt から $\lambda\Delta t$ の間は、 $CAR_0 \sim CAR_m$ の模擬処理も行うことになる。

各部分領域を担当するシミュレータは、基本的には CTSfed にこの部分領域間の通信処理機能を付加しただけのものであるため、並列処理を行う各々のシミュレータも CTSfed として HLA 仕様で実現した。そして、この通信処理は HLA のインタラクションクラスにより行うものとした。また、並列処理を行う各々の CTSfed は、すべて同一のパラメータセットを使用する必要があるため、各々の CTSfed は同一のパラメータセットを FMfed から受信するものとした。このようにして実現した並列処理環境を図 5 の中の 1 つの CTSfed として組み込んだシステム構成を示したものが図 9 である。

5. 性能評価

本章では、国内の国道 3 km 区間の単路を対象に、並列処理方式(4.3 節参照)における実行性能向上のための最適な λ を求めることと、本方式の台数効果を調べることを目的に実施した、システムの性能評価について述べる。なお、実験ではシミュレーションの時刻進行単位 Δt を 0.1 秒とした。

5.1 実験環境

図 10 に示す構成で、並列処理方式の性能評価を実

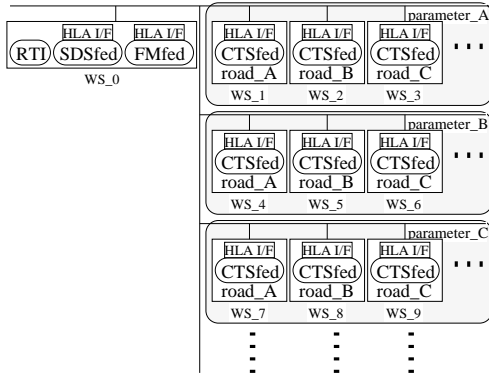


図 9 並列分散型シミュレーションシステムの構成例

Fig. 9 A structure of parallel and distributed simulation system.

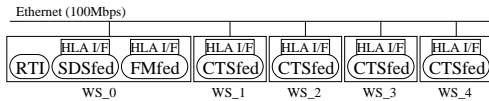


図 10 実験環境

Fig. 10 System structure for performance evaluation.

表 1 性能評価に使用した計算機の性能一覧
Table 1 Performance of the machines.

計算機名	機種	CPU (MHz)	主記憶 (MB)
WS_0	Ultra2-2200	UltraSparc (200) × 2	256
WS_1 } WS_4	Ultra2-1300	UltraSparc (300)	128

施した。図 10 の各 WS の性能を示したものが表 1 であり、ネットワークは 100 Mbps の Ethernet を使用した。

HLA-RTI として現在利用できるのは DMSO (Defense Modeling and Simulation Office) の RTI⁴⁾ や我々が開発した eRTI¹¹⁾ 等があり、本実験では台数拡張性に優れた eRTI を使用した。

5.2 評価結果

並列処理方式の性能評価では、対象とする単路は使用する WS (CTSfed) の数で均等に領域分割するものとした。評価実験では、次の 3 通りの場合に関して性能評価を実施した。

- (a) : 単路上に 35 台の車が存在する場合
- (b) : 単路上に 200 台の車が存在する場合
- (c) : 単路上に 300 台の車が存在する場合

(a) は単路上を車がスムーズに走行している場合、(b) は平均的な渋滞が発生している場合、そして (c) は非常に激しい渋滞が発生している場合の代表例である。以下の図 11, 12, 13 は、(a), (b), (c) 各々の場合において、 λ を変化させたときのスピードアップ

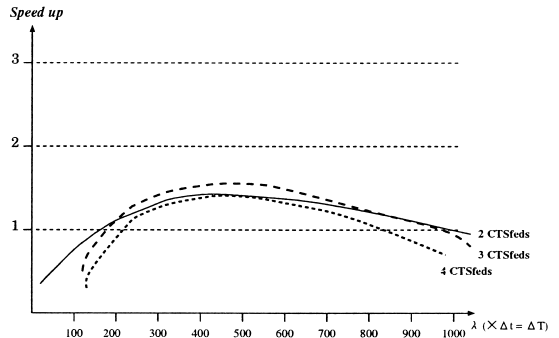


図 11 (a) 単路上に 35 台の車が存在する場合
Fig. 11 (a) Case of 35 cars on the road.

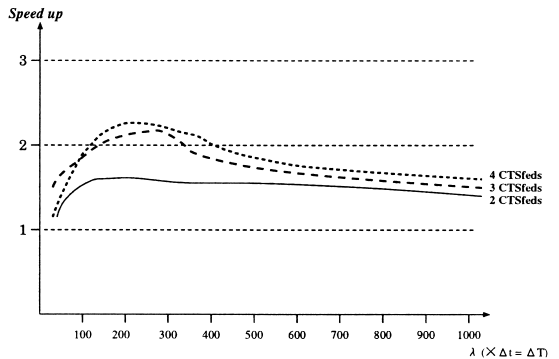


図 12 (b) 単路上に 200 台の車が存在する場合
Fig. 12 (b) Case of 200 cars on the road.

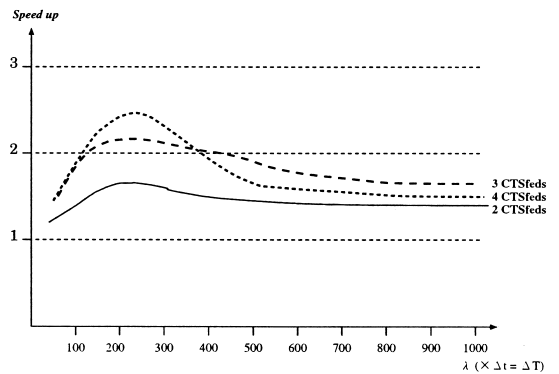


図 13 (c) 単路上に 300 台の車が存在する場合
Fig. 13 (c) Case of 300 cars on the road.

プ [WS (CTSfed) を 1 台使用した場合に対して、 n 台使用したときの実行時間の向上率] を示したものである。

この結果から単路上の車の台数が少なくなるほど、最適な λ (λ_{opt} : スピードアップ値を最も高くすることができる λ) は大きくなる傾向を示すことが分かる。これは WS 間で負荷 (車の台数) がバランスしていないためである。問題規模 (車台数) が小さいケースでの、2 台の WS を使用した場合における双方の部分領

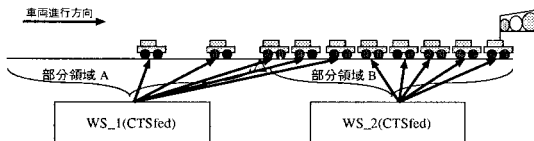


図 14 λ の値を適度に大きくすることによる負荷バランスの実現
Fig. 14 Load balancing by letting $\lambda = \lambda_{opt}$.

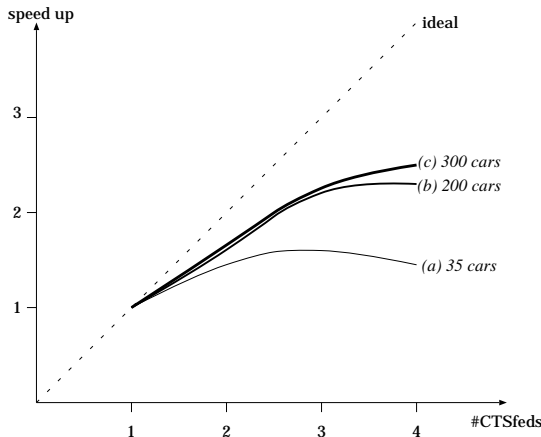


図 15 λ の最適値 (λ_{opt}) を用いた場合の台数効果
Fig. 15 Speed-up when λ_{opt} is applied.

域に存在する車の台数の比率を調べたところ、20%対80%であった。この原因は、対象とした単路の両端に信号機が設置されていることから、車両流出付近の信号が赤になると、この付近で車が渋滞してしまうことになるためである。このため、 λ の値を大きくすることによって WS_1 側でなるべく多くの車を模擬させた方が負荷のバランスにつながるようになる(図 14 参照)。

逆に、問題規模が大きくなると λ_{opt} は小さくなり、またスピードアップは高くなる。これは、問題規模が大きくなるほど、WS 間で負荷はバランスする傾向を示すため、通信間隔 (λ) を粗くして、負荷バランスを行う必要がなくなるからである。このことから、問題規模が大きくなるほど、スピードアップが高まり、これが同時に模擬精度の劣化の度合を抑えることにつながる事が分かる。

また、図 15 は図 11, 12, 13 より CTSfed 数ごとの λ_{opt} を求め、この λ_{opt} を用いたときのスピードアップを示したものである。この結果から、車の台数が多くなるほど、スピードアップ値が高くなり、並列処理の効果が得られることが分かる。

6. 考 察

本章では、並列処理方式において問題規模(車の台

表 2 フェデレート数ごとのパラメータ値
Table 2 Parameters for each federate.

フェデレート数	2	8	16	32
a	0.0039	0.0164	0.0335	0.0669
b	12.4	28.5	61.0	111.3

数) と使用する WS の台数が与えられたときの λ_{opt} の値について考察する。まず、並列処理の単位である粒度を 1 つの WS における $\Delta T (= \lambda \times \Delta t)$ 分の処理とする(図 8 参照)。問題規模を N 、使用する WS (CTSfed) の数を P とし、車両台数 (N) は各 WS につねに均等に配分されているものとする、粒度 1 つの実行時間 T_{unit} は、式 (2) で表される。

$$T_{unit} = \sum_{i=1}^{\lambda} (\#Car_{in} \cdot (i-1) + \frac{N}{P}) \cdot T_{car} \quad (2)$$

ここで、括弧内は Δt の間に処理する車の台数であり、 $\#Car_{in}$ はこの Δt の間に他の部分領域から流入してくる車の台数を示す。また、 T_{car} は 1 台の車の模擬処理に要する実行時間を示す。

次に、部分領域を通過した車群を移動先の WS (CTSfed) に転送するのに要する通信処理時間 T_{comm} を式 (3) に示す。ここで、式 (3) は実験結果¹¹⁾ に基づいた近似式であり、括弧内は通信するデータサイズ、 $Size_{car}$ は 1 台の車のデータサイズを示す。なお、フェデレート数が 2, 8, 16, 32 であるときの T_{comm} は、表 2 のパラメータ値 (a, b) を用いることにより導出することができる。

$$T_{comm} = a \cdot (\lambda \cdot Size_{car} \cdot \#Car_{in}) + b \quad [\text{ミリ秒}] \quad (3)$$

以上より、5 分間の模擬、すなわち 1 イタレーション分の処理を行うのに要する実行時間 T_{total} は式 (4) により表される。ここで、 Δt を 0.1 秒としているので、5 分間は $3,000 \times \Delta t$ (秒) に相当する。

$$T_{total} = (T_{unit} + T_{comm}) \cdot \frac{3000}{\lambda} \quad (4)$$

そして、 λ_{opt} の値を見積るべく、適当と思われる値および実験を通して得られた値を式 (2), (3) に含まれる各要素の値として以下のように設定した。

- $\#Car_{in} = 0.05$
一般道路において、任意地点を車が通過してから次の車がこの地点を通過するまでの時間間隔は 1.8 秒 ~ 2 秒であることが知られている¹²⁾。本実験ではこの値を 2 秒とした。1 台/2 秒 = 0.05 ($\#Car_{in}) \times 10$ (Δt)
- $Size_{car} = 24$ byte
現状では、各車は、“速度”、“加速度”、“走行中のレーン”、“車長”、“可能最高速度”、そして“ポジション”の属性値を持つようにモデル化してある。

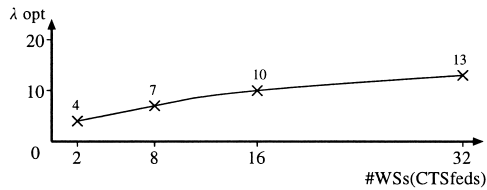


図 16 WS (CTSfed) 数に対する λ_{opt} の (最小) 値
Fig. 16 λ_{opt} vs. Number of WSs (CTSfeds).

- $T_{car} = 26.6$ ミリ秒

実験に使用した WS (表 1 参照) で, 車 1 台を模擬するのに要する実行時間²⁾ .

これらの値を式 (4) に代入することにより, T_{total} の値を最小にする $\lambda(\lambda_{opt})$ を導出した結果, 問題規模 (N) に関係なく WS 数に対する λ_{opt} の値は一意に決まり, また, WS 数が増えるとともに λ_{opt} の値も大きくなるのが分かった (図 16 参照). 問題規模に関係なく, WS 数に対する λ_{opt} の値が一意に決まる理由は, WS (部分領域) 間で負荷がバランスしていると仮定しているからである. また, WS 数が増えるとともに λ_{opt} の値も大きくなる理由は, WS 数が増えると通信コストも大きくなり, このコストを抑えるべく, ΔT を大きくするように λ_{opt} が作用するためである.

次に, 本章で得られた λ_{opt} の結果 (図 16 参照) と, 実験で得た λ の最適値 (5.2 節: 図 11, 12, 13 参照) を比較すると, 実験で得た λ の最適値が非常に大きい値であることが分かる. これは WS 間での負荷のバランス度合いが悪い傾向にあるためである. このことから, 問題規模が小さくなり, 負荷のバランス度合いが悪くなるほど, λ_{opt} の値は本章で得られた λ_{opt} (負荷がバランスしている場合の値) より大きくなる傾向を示すことが分かる. すなわち, 図 16 の λ_{opt} は, WS 数ごとの最小値を示すものであり, 負荷のバランス度合いが高くなるほど, この値に近付くこととなる.

また, 図 17 は $\lambda = 1$ ($T_{total}(\lambda = 1)$) に対する, λ_{opt} を用いたとき ($T_{total}(\lambda = \lambda_{opt})$) の性能向上率 ρ を示したものである (式 (5) 参照).

$$\rho = \frac{T_{total}(\lambda=1) - T_{total}(\lambda=\lambda_{opt})}{T_{total}(\lambda=1)} \cdot 100 (\%) \quad (5)$$

この結果から, WS 数が多くなるほど λ_{opt} の適用効果は高くなることが確認できる. これは, WS 数が多いほど λ_{opt} の値が ($\lambda=1$) よりも大きくなるために (図 16 参照), $T_{total}(\lambda = 1)$ と $T_{total}(\lambda = \lambda_{opt})$ の差がいつそう開くためである. また, WS 数が一定の場合には, 問題規模 (N) が小さくなるほど, λ_{opt} の適用効果は高くなることも分かる. これは, 問題規模が

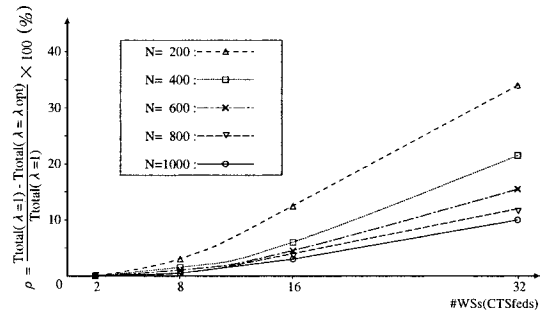


図 17 λ_{opt} を用いた場合の性能向上率
Fig. 17 Ratio of performance improvement when λ_{opt} is applied.

小さいほど, 通信時間に対する WS 内の処理時間の比率が小さくなり, $\lambda = 1$ の場合はこの傾向が顕著になるために, $T_{total}(\lambda = \lambda_{opt})$ に比べて $T_{total}(\lambda=1)$ が特に大きくなるからである. 他方, 問題規模が大きい場合には, この比率は大きくなり, 並列処理の台数効果が高くなるため, 問題規模が大きい分には問題ない. ここで, λ_{opt} の適用効果が高いとは, $\lambda=1$ とした場合の実行性能が悪いことを意味し, 逆に適用効果が低いとは, $\lambda=1$ とした場合でも比較的実行性能が落ちないことを意味する. なお, 図 17 の結果は, λ_{opt} の最小値 (図 16 参照) を用いて導出したものである. このため, WS 間で負荷がバランスしていない場合は, 図 17 で示した結果より性能向上率は高くなることとなる. すなわち, λ_{opt} を適用することにより, 最低でも図 17 で示した性能向上率が得られることとなる.

以上のことから, 通信頻度を適度に高めて実行性能の向上をはかる手法を適用する場合は, 適用効果と, この適用により劣化する模擬精度が対象とする応用の許容範囲内であるかを見定める必要がある. この際, 図 17 に示す効果と, 図 16 に示す λ_{opt} の最小値は指標として有効に活用することができる. なお, ここで得た理論的な結果を実験により検証することは今後の課題である. また, 実行性能の向上と模擬精度の劣化度合いを抑えることの双方を目的に, 問題規模に応じて各 WS が担当する領域の大きさを調節する負荷バランス方式や, すでにその効果が知られている多重マッピング方式⁷⁾ の適用と評価も今後の課題である.

7. おわりに

実世界の観測値を活用して, シミュレーションの模擬精度を高める方法について述べた. リアルタイムでこの観測値を活用し, 模擬精度を高める本シミュレーションシステムでは大きな計算パワーを必要とするた

め、並列・分散処理が不可欠である。本論文では、問題規模が大きい場合に必要となる並列処理の性能に関して述べるとともに、並列処理と模擬精度との関係を明らかにした。また、実行性能向上のための手法、およびその適用効果についても説明した。

今後の課題は、単路のみならずネットワーク状の道路網への展開を行いつつ、負荷のバランス度合いを変化させた様々なケースを対象に、考察した内容を実験により検証すること、この問題に適した負荷バランス方式を適用し、評価することである。そして、得られた結果および知見を活用し、信号制御や旅行時間情報の提供等の応用へ、本システムを適用することが最終的な目標である。

参 考 文 献

- 1) Nishi, N., et al.: Evaluation for Effectiveness of CDRGS (Centrally Determined Route Guidance System), *4th World Congress on Intelligent Transport Systems* (1997).
- 2) Ozaki, A., et al.: Exploiting HLA in Car Traffic Simulations, *Fall 1998 Simulation Interoperability Workshop*, Vol.2, pp.546-554 (1998).
- 3) 尾崎敦夫ほか：ミクロモデル交通シミュレーションにおける感知器情報融合方式の検討，第56回情報処理学会全国大会論文集，Vol.4, pp.432-433, 5V-4 (1997).
- 4) Dahmann, J.S. and Fujimoto, R. and Weatherly, M.: The Department of Defense High Level Architecture, *Winter Simulation Conference*, pp.142-149 (1997).
- 5) Cameron, B.G. and Wylie, D.M.: PARAMICS - Moving Vehicles on the Connection Machine, *Super Computing'94*, pp.291-300 (1994).
- 6) 滑川光裕ほか：道路交通ネットワークを対象とする並列シミュレーションのための同期アルゴリズム，第16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.185-188 (1997).
- 7) 尾崎敦夫ほか：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).
- 8) <http://www.sisostds.org/doclib/>
- 9) 後藤幸夫ほか：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 116.5 (1996).
- 10) 吉川康雄ほか：渋滞改善のための交通流マイ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).
- 11) Furuichi, M., et al.: Performance Evaluation Model of HLA-RTI and Evaluation Result of eRTI, *Fall 1997 Simulation Interoperability Workshop*, Vol.2, pp.1099-1109 (1997).
- 12) 交通工学研究会：交通工学ハンドブック，技報堂出版 (1998).

(平成 11 年 8 月 27 日受付)

(平成 12 年 2 月 4 日採録)



尾崎 敦夫 (正会員)

1964 年生。1988 年九州工業大学工学部情報工学科卒業。1990 年同大学院電気工学部計算機コース博士課程前期修了。同年 4 月より三菱電機(株)情報電子研究所に勤務。1992 年から 1996 年までリアルワールド・コンピューティング(RWC)プロジェクトに参画。以来、同社情報技術総合研究所において並列分散シミュレーションシステムの研究開発に従事。1996 年 European Simulation Symposium Best Paper Award 受賞。IEEE 会員。



古市 昌一 (正会員)

1958 年生。1982 年広島大学総合科学部総合科学科卒業。同年 4 月より三菱電機(株)情報電子研究所に勤務。1994 年イリノイ大学アーバナシャンペン校コンピュータサイエンス学科大学院修士課程修了。現在三菱電機(株)情報技術総合研究所において、並列分散シミュレーションシステムの研究開発に従事。ACM, IEEE-CS, SCS 各会員。



西 乃武夫

1973 年生。1996 年慶應義塾大学理工学部電気工学科卒業。同年 4 月より三菱電機(株)鎌倉製作所に勤務。交通管制システムおよび画像処理道路監視システムの関連機器開発・設計業務に従事。



黒田 悦司

1963 年生。1985 年大阪大学工学部機械工学科卒業。1987 年同大学院工学研究科産業機械工学専攻修士課程修了。同年 4 月より三菱電機(株)鎌倉製作所に勤務。交通管制システム関連機器の開発・設計業務に従事。