

車載センサのネットワーク化を実現するハードウェアの提案と評価

河合 智之[†] 本田 晋也[†] 富山 宏之[†] 高田 広章[†] 石原 秀昭^{††}

鈴木 亨市^{††} 手嶋 芳徳^{††} 松岡 俊彦^{††} 山田 健二^{††}

[†] 名古屋大学大学院情報科学研究科 〒464-8601 名古屋市千種区不老町

^{††} 株式会社デンソー IC 技術 1 部 〒448-8661 愛知県刈谷市昭和町 1-1

E-mail: †{t-kawai, honda, tomiyama, hiro}@ertl.jp, ††{HIDEAKI.ISHIHARA, KYOUICHISUZUKI, YOSHINORI.TESHIMA, TOSHIHIKO.MATSUOKA, KENJILY.YAMADA}@denso.co.jp

あらまし 本研究では、車載センサを直接ネットワークに接続した構成において、リアルタイム性の確保を実現するハードウェアを提案した。提案機構は、主として2つの機能を備える。メッセージの衝突を抑制するためのセンサ間の同期と、車両の状況に応じて適切なメッセージを送信できるようにメッセージセットの切替えを実現する。提案機構を SystemC 言語により実装した。実験では、SystemC 言語で記述された車載ネットワークシミュレータを利用して、従来のネットワーク構成と提案機構を用いたネットワーク構成を比較した。

キーワード 車載制御システム、センサ、CAN、SystemC、リアルタイム制約

A Networking Mechanism for Automotive Sensors

Tomoyuki KAWAI[†], Shinya HONDA[†], Hiroyuki TOMIYAMA[†], Hiroaki TAKADA[†],

Hideaki ISHIHARA^{††}, Kyouichi SUZUKI^{††}, Yoshinori TESHIMA^{††}, Toshihiko MATSUOKA^{††},

and Kenji YAMADA^{††}

[†] Graduate School of Information Science, Nagoya University Furou-cho, Chikusa-ku, Nagoya, 464-8601 Japan

^{††} IC Engineering Department 1, Denso Corporation 1-1, Showa-cho, Kariya, Aichi, 448-8661 Japan

E-mail: †{t-kawai, honda, tomiyama, hiro}@ertl.jp, ††{HIDEAKI.ISHIHARA, KYOUICHISUZUKI, YOSHINORI.TESHIMA, TOSHIHIKO.MATSUOKA, KENJILY.YAMADA}@denso.co.jp

Abstract In this paper, we have proposed a hardware mechanism which is useful to meet real-time constraints in the automotive systems where sensors are connected directly to the network. Specifically, this hardware mechanism supports two functions; synchronization between sensors and switching of message sets. We have implemented the hardware mechanism in the SystemC language. In the experiments, we have compared the proposed hardware mechanism with a traditional networking mechanism.

Key words Automotive control systems, Sensors, CAN, SystemC, Real-time constraints

1. はじめに

近年、自動車制御システムは、自動車の基本性能である、「走る、曲がる、止まる」に加え、環境への配慮、安全性の確保、快適性の向上といった要求に応えるため、大規模化、複雑化している。自動車の制御システムには精密さが求められるため、自動車には多くの電子制御ユニット (ECU: Electronic Control Unit) が搭載されている。これら複数の ECU はネットワーク構成をとって制御システムを実現する。

より高度な制御を実現するために、ECU の増加と共にセンサやアクチュエータなどの電子部品の数も増加の一途をたどっている。現在の制御システムでは、センサやアクチュエータはそれらを管理する ECU に直接接続されており、ネットワークを介した情報の伝達は ECU を通して行われる。

車両制御の質を高めるために、1つのセンサの値を、様々な場所に配置される複数の ECU が利用したいケースが増加している。それに伴い、センサを直接ネットワークに接続したいという要求が高まっている。しかし、センサを直接ネットワーク

に接続すると、ネットワーク負荷の上昇やメッセージの滞留時間の増加が問題となる。これらの問題はネットワーク遅延を引き起こし、リアルタイム性の確保を難しくする。本研究の目的は、このような構成においてもリアルタイム性の確保を容易にすることである。本研究では、車載センサのネットワーク化を実現するハードウェアの提案と評価を行った。

本論文の構成は以下の通りである。2章では、本研究で対象とする CAN や本研究と関連の深い TTCAN といった車載ネットワークプロトコルについて紹介する。3章では、車載センサのネットワーク化を実現するハードウェアの提案を行う。提案機構の機能とその特徴について述べ、章の最後で、提案機構の設計と実装について説明する。4章では、提案機構の評価を行う。車載ネットワークシミュレータを利用して、従来のネットワーク構成と提案機構を用いたネットワーク構成を比較した。最後に、5章で、提案機構とそれを用いた車載ネットワーク構成について、結論を述べる。

2. 車載ネットワークプロトコル

現在の車載制御システムは、ネットワーク構成をとって実現されている。車載ネットワークを構成する通信プロトコルには様々なものが存在するが、本節では、本研究で対象とする CAN と本研究と関連の深い TTCAN について紹介する。

2.1 CAN

CAN(Controller Area Network) は、ISO において国際的に標準化されたシリアル通信プロトコルである。CAN は、自動車の高速リアルタイムネットワークを目指し、1980 年代にドイツの自動車電装メーカーであるボッシュ社により開発された。現在、CAN は自動車内の制御系ネットワークにおいてデファクトスタンダードとなっている。

CAN は以下の特徴を持つ。

- ID ベースのアービトレーション
全てのメッセージは、ユニークな識別子 (ID) を持つ。ID はバスにアクセスする際のメッセージの優先度を示し、2 つ以上のノードが同時にメッセージの送信を開始した場合、各メッセージの ID に対してビット単位でアービトレーションを行う。アービトレーションに勝ったノードが、そのまま送信を継続し、負けたノードは送信をやめて受信動作に移る。
- マルチマスタ
制御ノードをバス形式に接続しており、バスが空いている時はどのノードもメッセージの送信を始められる (マルチマスタ)。バスに早くアクセスしたノードが送信権を得ることができ、同時に送信を始めた場合、アービトレーションが行われる。また、ID 付きの送信メッセージはバス上の全てのノードへ配信され (ブロードキャスト)、受信側がメッセージの必要性を判断する。
- 通信速度
仕様では、最大 1Mbps の速度まで通信が可能で、ネットワークの規模に合わせて通信速度を設定できる。た

だし、1 つのネットワーク内では、全てのノードに対して同一の通信速度を設定する必要がある。

- システムの柔軟性
ノードがアドレスのような情報を持たないため、バスにノードを追加する際、他のノードのソフトウェアやハードウェアを変更する必要がない。また CAN バスに複数のノードを接続する場合、接続するノードの数に論理的な制限はない。

2.2 TTCAN

TTCAN(Time Triggered Controller Area Network) は標準 CAN にセッション層を追加し、タイムトリガ型の通信を行えるように拡張したものである [1] [2]。タイムトリガ型の通信では、通信開始のタイミングが衝突することはない。従って、TTCAN は、最大バス負荷時においても、すべての転送デッドラインを満足できるように工夫されている。また、TTCAN は、イベントドリブン型の通信も扱うことができる。詳細については、文献を参照されたい。

TTCAN では、同期用のメッセージとして Reference Message が定義される。どの時間にどのメッセージが送信されるかを Matrix Cycle として定義する。具体的なメッセージ送信は Cycle Time と呼ばれる時間に従って行われる。Cycle Time は Reference Message を受信する度にリセットされる。

3. 車載センサのネットワーク化を実現するハードウェア

3.1 従来の車載ネットワーク構成におけるセンサ接続

従来の車載ネットワーク構成を図 1 に示す。従来の車載ネットワーク構成では、センサやアクチュエータはそれらを管理する ECU に接続される。そして、情報の送受信は、その ECU を介して行われる。つまり、ネットワークには複数の ECU が接続され、センサやアクチュエータは直接ネットワークに接続されていない。

近年、制御の質を高めることを目的として、あるセンサの値を、様々な場所に配置される複数の ECU が利用したいケースが増加している。それに伴い、センサを直接ネットワークに接続したいという要求が高まっている。センサは ECU に比べ、非常に短い周期で情報を送信する。従来の車載ネットワーク構成においてセンサを直接ネットワークに接続した場合、ネットワーク負荷の上昇やメッセージの滞留時間の増加が問題となる。

3.2 センサのネットワーク化による弊害

センサのメッセージ送信は周期送信を基本とし、ネットワークに接続されるセンサの数が増加すれば、おのずとバス負荷は

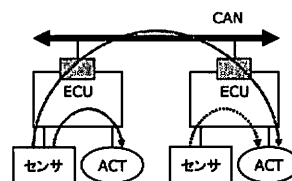


図 1 従来の車載ネットワーク構成

上昇する。バス負荷が上昇すれば、ネットワークに送信されるメッセージの滞留時間は増加する。滞留時間が増加するとデッドラインを守ることができなくなり、リアルタイム性の確保が困難となる。センサは非常に短周期なため、増加傾向にあるセンサをそのままネットワークに接続する構成では、システムが破綻してしまうことが容易に想像できる。そこで、車載制御システムの特性を考慮した解決策が求められる。

3.3 本研究で対象とする車載ネットワーク構成

本研究で対象とする車載ネットワーク構成を図2に示す。これまでECUに接続されていたセンサを直接ネットワークに接続する。厳密には、本研究で提案する機構を介してネットワークに接続される。これ以降、この機構をセンサ接続コンポーネントと呼ぶ。

ネットワークに送信されるメッセージの滞留時間を軽減するには、メッセージ同士の衝突を減らす必要がある。メッセージの衝突が生じると、衝突したメッセージの中で一番優先度の高いメッセージがネットワークへ送信される。それ以外のメッセージは、自分より優先度の高いメッセージの送信が完了するのを待たなければならない。優先度の低いメッセージは、メッセージの衝突が生じると滞留時間が増加する。従ってメッセージの滞留時間を減らすには、メッセージの衝突を抑制する必要がある。

メッセージの衝突は、適切なオフセットをメッセージに付与することで抑制できる。しかし、各ノード内でメッセージが衝突しないようにオフセット付けを行っても、ネットワーク上の他ノードから送信されるメッセージと衝突する可能性がある。ネットワーク全体のメッセージが衝突しないように、全メッセージに適切なオフセット付けを行っても、CANに接続されるノードは独立して動作するため、それぞれのノードのタイマにずれが生じるとメッセージの衝突が生じる。このようなメッセージの衝突が起きると、オフセットの付加による効果が薄れてしまう。そこで、本研究では、各ノードが時刻の同期を行う仕組みを備えたハードウェアを提案する。

また、車載制御システムでは、走行時の状況に応じて、値が必要とされるセンサやその周期は異なると考えられる。例えば、車両が停車している時と走行している時では、必要なセンサの情報やタイミングは異なると考えられる。そこで、本研究では、車両の状況に応じて、送信するメッセージを変更する機能を備えたハードウェアを提案する。これにより、不要なメッセージの送信が防止され、バス負荷の軽減やリアルタイム性の確保が実現できる。システム全体のリアルタイム性を保証するには、メッセージセット毎のリアルタイム性を解析しリアルタイム性を保証できるメッセージセットを作成すればよい。リアルタイム性を解析する手法には、従来の手法 [3] [4] [5] を用いることができる。

3.4 センサ接続コンポーネントの機能

3.4.1 基本的な機能

センサ接続コンポーネントは、メッセージの最大滞留時間の軽減とネットワーク負荷の軽減を図ることを目的として、以下の機能を持つ。

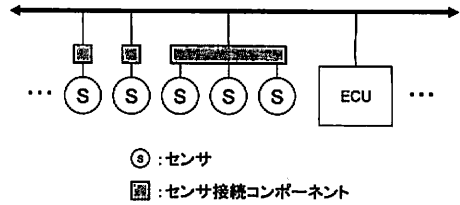


図2 本研究で対象とする車載ネットワーク構成

- センサのメッセージの周期送信
- センサ接続コンポーネント間の同期
- メッセージセットの切替え
- メッセージの間引き

センサ接続コンポーネント間の同期、メッセージセットの切替え、メッセージの間引きについては次小節以降で詳細に説明する。

センサ接続コンポーネントの基本的な機能はセンサの値を周期的にネットワークへ送信することである。各センサ接続コンポーネントは自ノードが所有するタイマを使い、各メッセージの周期、オフセットに従ってメッセージを送信する。同期を行う時点で、タイムマスタが送信する時刻より自ノードの時刻が早い場合、つまり、あるメッセージを送信後、同期によりその送信タイミングより以前の時刻に時刻合わせが発生した場合、そのメッセージを再送しない。同期を行う時点で、タイムマスタが送信する時刻より自ノードの時刻が遅い場合、つまり、同期を行った際に、メッセージの送信タイミングを超える時刻合わせが発生した場合は、そのメッセージを直ちに送信する。

3.4.2 センサ接続コンポーネント間の同期

本小節では、センサ接続コンポーネント間で同期をとるための仕組みについて説明する。センサ接続コンポーネント間の同期をとるために、同期用のメッセージを用意する。以後、これをリファレンスメッセージと呼ぶ。ネットワーク上のあるノードがリファレンスメッセージを全ノードに送信する。そのノードをタイムマスタと呼ぶ。タイムマスタは一定の間隔でリファレンスメッセージを送信する。リファレンスメッセージを受信したノードは自ノードが所有するタイマの値を調整する。リファレンスメッセージには、タイムマスタが管理するリファレンスメッセージの送信回数のがせられる。リファレンスメッセージを受信したノードは、その送信回数を利用してタイマの値を調整する。この仕組みにより、リファレンスメッセージを取りこぼした場合の被害を小さく抑えることができる。リファレンスメッセージによる同期の様子を、図3に示す。

リファレンスメッセージは、通常のCANメッセージである。従って、リファレンスメッセージは滞留する恐れがある。リファレンスメッセージが滞留し、各ノードへの到着が遅れると、タイムマスタと各ノードには時刻のずれが生じる。このずれを防止するために、タイムマスタは、リファレンスメッセージがネットワークに送信される瞬間に、リファレンスメッセージの滞留時間を計算し、その滞留時間をリファレンスメッセージにのせて送信する。受信側では、リファレンスメッセージの滞留

時間分だけ差し引いてタイマ値を調整する。

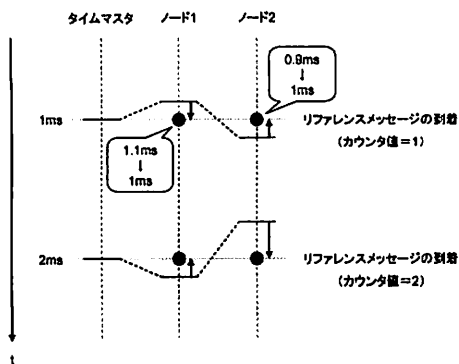


図 3 リファレンスメッセージを利用した同期

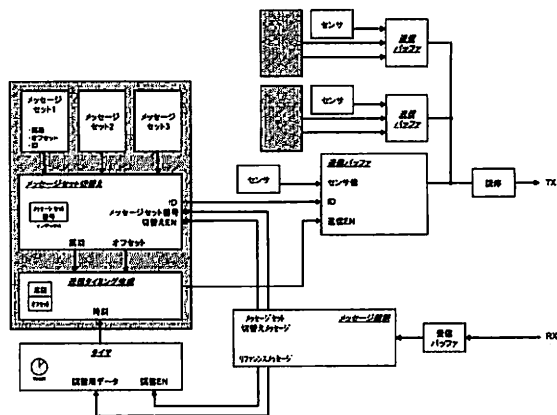


図 4 センサ接続コンポーネントの回路ブロック図

3.4.3 メッセージセットの切替え

次に、メッセージセットの切替えについて説明する。メッセージセットとは、各ノードが送信するメッセージの集合である。設計段階において、予め複数のメッセージセットが用意される。センサ接続コンポーネントは、他ノードからの要求があった場合に、メッセージセットの切替えを行う。これにより、メッセージの優先度、周期、オフセットを変更できる。

メッセージセットの切替えを各ノードに伝達するためのメッセージをメッセージセット切替えメッセージと呼ぶ。各センサ接続コンポーネントはメッセージセット切替えメッセージを受信すると、メッセージセットの切替えを行う。

3.4.4 メッセージの間引き

次に、メッセージの間引きについて説明する。前回送信したセンサの値と今回送信する値との間に、変更がない、あるいは、差分が小さい場合、メッセージをネットワークに送信しない。メッセージの間引きを行うか否かの判断はセンサ側へ任せ、センサ接続コンポーネントはその結果を受け取り、それ従ってメッセージの間引きを行う。メッセージの間引きの判断をセンサ側に任すことで、センサ接続コンポーネントの汎用度が増す。

3.5 センサ接続コンポーネントの設計

本節では、センサ接続コンポーネントの設計について説明する。

センサ接続コンポーネントの回路ブロック図を図 4 に示す。センサ接続コンポーネントには複数のセンサを接続できるように設計を行った。図 4 では、センサを 3 つ接続し、メッセージセットを 3 つ用意した場合を示している。センサ接続コンポーネントは、メッセージ解釈モジュール、タイマモジュール、メッセージセット切替えモジュール、送信タイミング生成モジュール、送信バッファモジュールの 5 つのモジュールで構成される。図 4 の中で、下線が引かれているものがこのモジュールに相当する。各モジュールの詳細を以下で説明する。

3.5.1 メッセージ解釈モジュール

センサ接続コンポーネントは CAN コントローラを介して、ネットワークに接続される。つまり、メッセージの送受信は、CAN コントローラを介して行う。センサ接続コンポーネント

は、メッセージを受信すると、そのメッセージを解釈する。センサ接続コンポーネントが受信するメッセージは、リファレンスメッセージとメッセージセット切替えメッセージの 2 つである。メッセージ解釈モジュールでは、受信したメッセージがリファレンスメッセージかメッセージセット切替えメッセージかを判断する。この判断は、ID で識別することができる。識別の結果、受信したメッセージがリファレンスメッセージであれば、必要な情報をタイマモジュールへ送る。受信したメッセージがメッセージセット切替えメッセージであれば、必要な情報をメッセージセット切替えモジュールへ送る。

3.5.2 タイマモジュール

タイマモジュールは、時刻の管理を行うモジュールである。センサ接続コンポーネントは独自にタイマを持ち、管理する。タイマはある一定の間隔でインクリメントされる。本研究では、インクリメントの間隔を 1 マイクロ秒とした。メッセージ解釈モジュールから伝達された情報を基に、タイマ値の補正を行う。この補正により、センサ接続コンポーネント間の同期を実現する。

3.5.3 メッセージセット切替えモジュール

メッセージセット切替えモジュールでは、主として、メッセージセットの管理を行う。設計時にあらかじめ用意された複数のメッセージセットを保持し、メッセージセット切替え番号により、適切なメッセージセットを選択する。選択されたメッセージセットにより、各センサの ID、周期、オフセットが決定される。

3.5.4 送信タイミング生成モジュール

送信タイミング生成モジュールでは、メッセージセット切替えモジュールで決定されたメッセージセットの情報を基に、メッセージの送信タイミングを生成する。そして、そのメッセージの送信タイミングと ID を送信バッファモジュールに送る。

3.5.5 送信バッファモジュール

送信バッファモジュールには、センサが接続される。ここで、送信バッファモジュールに接続されるセンサはメッセージの送信タイミングの発生時に、送信したい値をセンサ接続コンポーネントへ届けているものとする。送信バッファモジュールは、

送信タイミング生成モジュールの指令により、センサ値とIDをフレーム化し、CANコントローラへ送信する。複数のセンサが同時に送信タイミングを迎えた時、センサ接続コンポーネント内でメッセージの衝突が生じる。これに対応するため、送信バッファでは、IDによるメッセージの調停を行う。センサ接続コンポーネント内のメッセージ衝突も、CANでの衝突と同様に、優先度の高いメッセージが先に送信される。

3.6 SystemC言語による実装

本研究では、センサ接続コンポーネントをSystemC言語により実装した。SystemC言語はC++に基づいたシステム記述言語の一つで、Open SystemC Initiative(OSCI) [6] が標準化と普及を進めている。SystemC言語を用いることで、高速かつ柔軟なシミュレーションが可能となった。

4. 評価

本章では、提案機構の評価を行う。提案機構の性能評価には、SystemC言語で記述された車載ネットワークシミュレータを拡張して利用した。拡張を行ったシミュレータは、ボッシュ社が提供するC言語ベースのシミュレータと同等の動作を行う、SystemC言語で記述されたシミュレータである。本研究では、このシミュレータにバス負荷とメッセージの滞留時間を計測できるように変更を加えた。

性能評価の実験では、従来のネットワーク構成と提案機構を用いたネットワーク構成を比較した。具体的には、センサ接続コンポーネント間の同期の効果に関する実験とメッセージセット切替えの効果に関する実験を行った。

4.1 評価する車載ネットワーク構成

以下のネットワーク構成を基準に、評価を行った。評価の基準は、メッセージの滞留時間とバスの負荷率である。

本実験の基本ネットワーク構成を図5に示す。図5に示されるように、1本のCANバスに、ECUとセンサが複数接続される構成を基本として実験を行った。ECUの数は8個とし、センサの数は0~32個の中で可変とする。センサのネットワーク接続は、センサ接続コンポーネントを介して行われる。1つのセンサ接続コンポーネントに接続可能なセンサ数の上限を4として実験を行った。

また、バスの通信速度、シミュレーション時間は以下の通りである。

- バスの通信速度：500Kbps
- シミュレーション時間：実機での5秒

ここで、実機での5秒というのは、実機上での5秒間をシミュレーションするという意味である。

バスに接続されたECU、センサ接続コンポーネントを介してバスに接続されたセンサは、それぞれメッセージをネットワークに送信する。そのメッセージ構成を表1に示す。これらの他に、リファレンスメッセージとメッセージセット切替えメッセージを定義した。

4.2 同期の効果に関する実験

はじめに、同期の効果に関する実験について説明を行う。

本実験では、センサ数を可変とし、接続数を徐々に増やして

表1 メッセージ構成

	数	ID	周期 [ms]	オフセット
センサ	32 個	11~42	10	均等配分
ECU	20 個	110~129	20~100	なし
	30 個	130~159	100~500	なし
	40 個	160~199	500~1000	なし

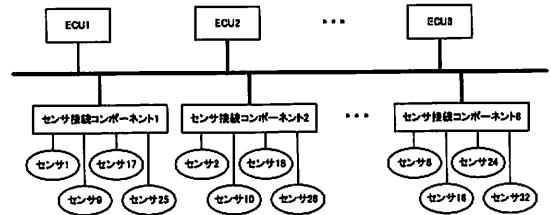


図5 評価する車載ネットワーク構成

いき滞留時間とバス負荷の動向を見る。

そして、本実験では、センサ接続コンポーネントのタイマのずれをパラメータとして指定した。センサ接続コンポーネントはこの値の範囲でタイマがずれる。そのずれを修正し、センサ接続コンポーネント間の同期をとるために、リファレンスメッセージを送信する。同期の頻度、つまり、リファレンスメッセージの送信周期はセンサのフレーム間隔をタイマ精度で割った値とした。

センサから送信されるメッセージの最大滞留時間をグラフにまとめたものを図6に示す。ECUから送信されるメッセージの最大滞留時間をグラフにまとめたものを図7に示す。最大滞留時間とは、滞留時間の中で最大のものを指す。

ここで、グラフの系列はタイマの誤差を示す。また、グラフ中の系列「±0%①」は全センサのメッセージに均等配分されたオフセットを付加した場合の実験結果を指す。均等配分されたオフセットとは、各センサのメッセージの送信タイミングが周期内で均等に発生するようなオフセット配分である。グラフ中の系列「±0%②」は全センサのメッセージにオフセットを付加しない場合の実験結果を指している。

センサ接続コンポーネント間で同期をとらない場合、ワーストケースとして全てのメッセージが衝突する場合は考えられる。この場合の最大滞留時間は、グラフの「±0%②」で示される。センサのメッセージ同士が衝突しないオフセット付けとセンサ接続コンポーネント間の同期により、センサの最大滞留時間を大幅に抑えることが、図6から確認できる。ただし、タイマ精度が悪化するにつれて、リファレンスメッセージの送信頻度は増す。従って、その場合は、その分だけ最大滞留時間は増加する。

ECUの最大滞留時間は、タイマ精度、つまりリファレンスメッセージの送信頻度に影響する。タイマ精度が悪化すると、リファレンスメッセージの送信頻度が増し、最大滞留時間が増加している。

本実験では、ECUのメッセージは全てのセンサのメッセージよりも優先度が低いため、システム全体の最大滞留時間は

ECUの最大滞留時間と等価になる。従って、システム全体の最大滞留時間の許容値を決定すれば、本実験のネットワーク構成における接続可能なセンサ数を予想することができる。

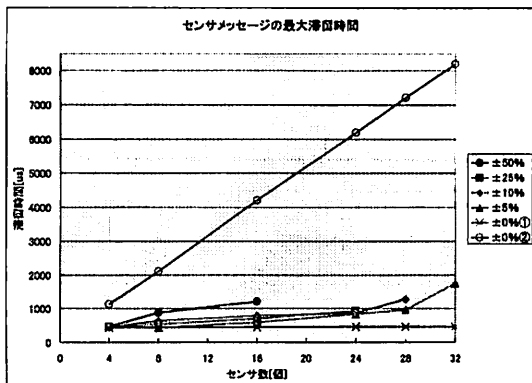


図6 センサメッセージの最大滞留時間

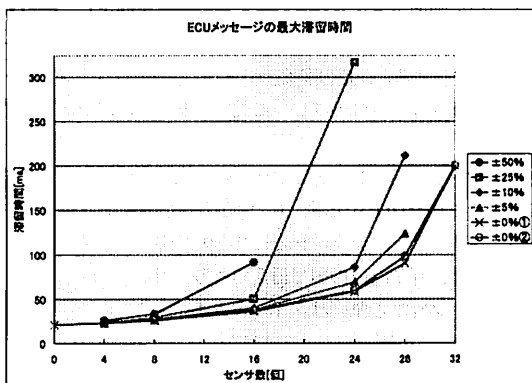


図7 ECUメッセージの最大滞留時間

4.3 メッセージセット切替えの効果に関する実験

次に、メッセージセット切替えの効果を評価するための実験について説明する。

メッセージセット切替えの効果に関する実験では、メッセージセット切替えの機能が有る場合と無い場合の比較を行った。

本実験では、センサ数は32個で固定とする。今回、車両の状況が2つあるものとしてメッセージセットを用意した。

メッセージセット切替えの機能がある場合には、状況に応じて適切なメッセージセットを選ぶことができる。メッセージセットA(状況A)では、優先度の高いメッセージ16個を10ミリ秒周期で送信し、優先度の低い残りの16個のメッセージを50ミリ秒周期で送信する。メッセージセットB(状況B)では、メッセージセットAとは反対に、優先度の高いメッセージ16個を50ミリ秒周期で送信し、優先度の低い残りの16個のメッセージを10ミリ秒周期で送信する。メッセージセット切替えの機能が無い場合は、すべてのセンサが10ミリ秒の周期で送信しなければならない。

状況の遷移の頻度、つまり、メッセージセット切替えメッセージの周期を50ミリ秒の場合と500ミリ秒の場合で実験を行った。

本実験では、メッセージセット切替えによる効果として、それぞれのメッセージの最大滞留時間とバスの負荷率を測定した。その結果を表2に示す。

表2 メッセージセット切替えによる効果

切替えの有無	センサの	
	最大滞留時間 [μs]	ECUの最大滞留時間 [ms]
あり (50ms)	8104.7	46.5
あり (500ms)	8104.7	46.5
なし	8190.7	200.0

メッセージセット切替えの機能がない場合、すべてのセンサを10ミリ秒周期で送信しなければならない。走行時の状況に応じてメッセージセットを切替えることで、センサのメッセージ量を削減することができ、ECUの最大滞留時間とバスの負荷率の軽減が図れた。また、この実験では、メッセージセットの切替えを行うことで、バスの負荷率を約30%減少できた。これは、10ミリ秒周期のセンサ12個分に相当する。

また、メッセージ切替えの頻度を変えても、最大滞留時間に変化は見られなかった。メッセージ切替えの間隔を小さくすると、メッセージセット切替えメッセージを多く送信しなければならないので、その分だけバス負荷は上昇する。

5. おわりに

本研究では、センサを直接ネットワークに接続した構成において、容易にリアルタイム性の確保が可能になるハードウェアの提案を行った。センサ間の同期の効果に関する実験では、センサが送信するメッセージの最大滞留時間を大幅に削減することができた。さらに、メッセージセット切替えの効果に関する実験では、バスの負荷率を約30%下げ、ECUが送信するメッセージの最大滞留時間を軽減できた。

今後の課題は、提案機構を実機上に実装し、より精度の高いシミュレーションを行うことである。

文献

- [1] T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, M. Walther, Robert Bosch GmbH, "Time Triggered Communication on CAN (Time Triggered CAN - TTCAN)," *Proceedings 7th International CAN Conference*, 2000.
- [2] F. Hartwich, B. Müller, T. Führer, R. Hugel, Robert Bosch GmbH, "CAN Network with Time Triggered Communication," *Proceedings 7th International CAN Conference*, 2000.
- [3] K. Tindell, A. Burns, A. Wellings, "Calculating Controller Area Network (CAN) Message Response Time," *Control Engineering Practice*, Vol.3, No.8, 1995.
- [4] 飯山真一, 高田広章, "システム構成を考慮したCANの最大遅れ時間解析手法" 情報処理学会論文誌: コンピューティングシステム, Vol.45, No.SIG1(ACS 4), 2004.
- [5] 飯山真一, 富山宏之, 高田広章, 城戸正利, 細谷伊知郎, "オフセット付きCANメッセージの最大遅れ時間解析," 情報処理学会論文誌: コンピューティングシステム, Vol.45, No.SIG11(ACS 7), 2004.
- [6] Open SystemC Initiative. <http://www.systemc.org/>