社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－124 (8)
2006／3／16

FlexRayのダイナミックセグメントにおけるメッセージの最大遅れ時間解析

村上 靖明† 村上 尚彦†† 冨山 宏之†† 高田 広章††

† 名古屋大学工学部 〒464-8603 名古屋市千種区不老町
†† 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
E-mail: †{yasuaki,murakami,tomiyama,hiro}@ertl.jp

あらまし 現在、自動車内の制御系ネットワークにおいては、新しいプロトコルであるFlexRayが注目を浴びている。このFlexRayを用いてメッセージの送信を行う場合には、スタティックセグメントとダイナミックセグメントのどちらかが用いられる。そのうち、後者を用いる場合には、バスを有効に使えるというメリットがあるが、リアルタイム性を保証するのが難しい。そこで、本研究では、ダイナミックセグメントを安全に使用するための解析を行う。最初に、遅れ時間がより長くなる条件を整理する。それをもとに、メッセージの最大遅れ時間の解析手法を提案する。この手法をランダムに作成した複数のメッセージセットに対して適用し、この手法の評価を行った。また、その結果を踏まえて、メッセージセットの厳しさの指標を導入した。

キーワード 車載LAN, FlexRay, 最大遅れ時間解析

Response Time Analysis for Messages in Dynamic Segment of FlexRay

Yasuaki MURAKAMI†, Naohiko MURAKAMI††,
Hiroyuki TOMIYAMA††, and Hiroaki TAKADA††

† School of Engineering, Nagoya University Furou-cho, Chikusa-ku, Nagoya, 464-8603 Japan
†† Graduate School of Information Science, Nagoya University Furou-cho, Chikusa-ku, Nagoya, 464-8601 Japan
E-mail: †{yasuaki,murakami,tomiyama,hiro}@ertl.jp

Abstract Nowadays, FlexRay attracts attention as a new protocol in automotive networks for control. Messages are transmitted in a static segment or a dynamic segment. In the latter case, it is difficult to guarantee real-time requirements though there is an advantage that the bus can be effectively used. In this research, we analyze for using dynamic segment safely. Firstly, we organize the conditions that response time is longer. And we proposed a method to analyze the worst-case response time of messages under these features. We applied it to message sets which are made randomly, and evaluated it. Moreover, we brought in the indicator of the severity of the message set based on the result.

Key words in-vehicle LAN, FlexRay, response time analysis

1. はじめに

現在、自動車内の制御系ネットワークにおいては、1990年代に提案されたバス型ネットワークのCAN (Controller Area Network)が、事実上の標準となっている。しかし、自動車に搭載される機能の増加に伴い、さまざまな問題が表面化してきた。このような問題に対応するため、新しいプロトコルであるFlexRayが誕生した。

FlexRayの特徴としては、タイムトリガ方式、高い柔軟性、冗長性、高信頼性、高帯域であることが挙げられる。FlexRayを用いて、メッセージの送信を行う際には、各メッセージごとにスタティックセグメントとダイナミックセグメントのどちらかが用いられる。スタティックセグメントでは、スロット長が固定であり、周期的な通信が保証されている。それに対して、ダイナミックセグメントでは、スロット長が可変であり、バスを有効に使えるというメリットがあるが、リアルタイム性を保証するのが難しい。そこで、本研究では、ダイナミックセグメントでの通信を安全に使用するために、通信が動的にスケジュー

リングされるダイナミックセグメント内でのリアルタイム性を検証することを目的として、ダイナミックセグメントを用いた際の最大遅れ時間の解析を行う。

解析では、最初に、既存の車載ネットワークである CAN においてメッセージの遅れ時間が最大となる条件をダイナミックセグメントの送信方法に照らし合わせて考え、従来手法では解決できない問題点を指摘する。また、各ノードごとのメッセージ長の上限値が全ノードで同じ場合と同じでない場合とを区別して考え、両者を比較しながら、遅れ時間がより長くなる条件の整理を行う。そして、その結果をもとにメッセージの最大遅れ時間を求めるアルゴリズムを提案し、評価する。

本論文では、2 章で本研究対象である FlexRay の概要、ならびにダイナミックセグメントを用いてメッセージを送信する際の仕組みや送信条件について述べる。3 章では、問題点を整理し、4 章で、その問題点を考慮したアルゴリズムの説明を行う。5 章では、4 章で提案したアルゴリズムを用いた結果を示し、考察する。最後に 6 章で、本論文のまとめと今後の課題について述べる。

2. FlexRay の概要

2.1 FlexRay プロトコルの概要

FlexRay の最新仕様 ver2.1 が、2005 年 5 月に FlexRay コンソーシアム [1] によって一般に公開されている。本節では、そのプロトコルの概要の内、本研究に関する部分について述べる。

FlexRay プロトコルでは、メディアアクセス制御は、0〜63 からなる通信サイクルが繰り返されることを基準にしている。通信サイクルは、FlexRay 内のメディアアクセス方式の基本的な要素である。図 1 に示すように、通信サイクルは、スタティックセグメント、ダイナミックセグメント、シンボルウィンドウ、ネットワークアイドルタイムで構成されている。このうち、直接メッセージの送信に使用されるのが、スタティックセグメントとダイナミックセグメントの 2 つである。スタティックセグメントでは、データ長が固定されており、静的にスケジューリングされる。一方、ダイナミックセグメントでは、データ長の変更を自由にすることができ、動的にスケジューリングされる。シンボルウィンドウは、バスのテストなどに使用するシンボルを送信するための時間である。最後に、ネットワークアイドルタイムは、すべてのデータ送受信が停止されたアイドル時間であり、各ノードが固有のタスク処理を行ったり、クロック補正のための計算及び補正を行ったりする。

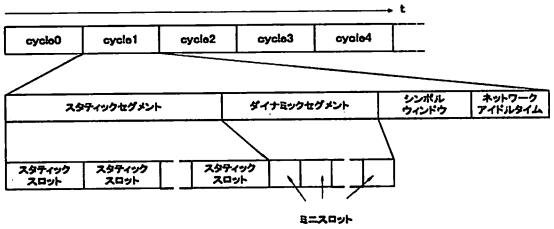

図 1 FlexRay のメディアアクセス

また、スタティックセグメントはスタティックスロット、ダイナミックセグメントはミニスロットで構成される。スロットとは、各ノードに割り当てられたバスを占有できる時間である。

2.2 ダイナミックセグメントにおける送信

本節では、ダイナミックセグメントを用いた際の送信方法や送信条件などについて述べる。スタティックセグメントについては、すでに研究が行われている [2]。

FlexRay の送信スケジュールはシステムの設計時に静的に決定される。各フレームには、ヘッダセグメントでフレーム ID が割り当てられている。フレーム ID は、フレームが送信されるべきスロットを明確にしている。また、FlexRay では、フレーム ID はコミュニケーションサイクル中で各々のチャネルで 2 回以上は使用されない。このフレーム ID とスロット ID が一致したときにフレームが送信される。スロット ID はスタティックセグメント、ダイナミックセグメント全体に対して、時間に沿って小さい順に割り振られる。現在のスロット ID を示すために、各チャネルごとにスロット・カウンタがある。スロット・カウンタは、ダイナミックセグメント内では、チャネルごとに別々に進められる。

図 2 にダイナミックセグメントにおける送信のモデルを示す。ダイナミックスロットは、1 つないし、複数のミニスロットで構成される。ダイナミックスロットの長さは、フレームの長さに応じて動的に変化する。フレームの送信がない場合には、1 つのミニスロット (空きスロット) に対応するアイドル時間の後、次のスロットに切り替わる。また、ダイナミックスロットの長さは各チャネルごとに決められる。さらに、ダイナミックセグメントでは、送信するフレームに対して、十分なミニスロットが確保できないとき、送信が行われないため、フレーム ID の大きいメッセージは必ずしも通信が保証されない。つまり、ダイナミックセグメントにおけるメッセージの送信では、フレーム ID がそのフレームの優先度を表しており、フレーム ID の小さいものほど優先度が高いといえる。

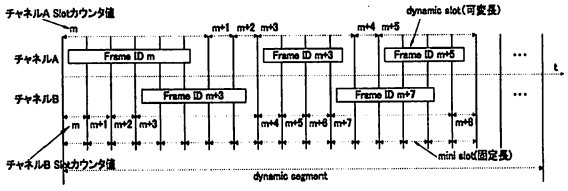

図 2 ダイナミックセグメントにおける送信

また、各ノードごとに、最終送信可能スロット (pLatestTx) が決められている。pLatestTx とは、ダイナミックセグメントにおいてフレームの送信を開始できる最後のミニスロット番号を表していて、メッセージ長の上限値に基づいて決定される。各ノードは、pLatestTx 以降では、いずれのダイナミックフレームも送信を開始することができないので、ダイナミックセグメント内に送信が完了するようなペイロード長の短いダイナミックフレームであっても送信は不可能である。

3. 問題点の整理

3.1 CANの場合との比較

FlexRayのダイナミックセグメントを用いてメッセージの送信を行う際には、既存の車載ネットワークであるCANにおける送信方法と同様に、優先度を持った可変長のメッセージを扱っている。そこで、CANにおけるメッセージの最大遅れ時間解析手法[3]を適用できないかどうか確かめる。

CANを用いてメッセージの送信を行う際に、あるメッセージの送信要求が発生してから実際に送信されるまでの時間がもっとも長くなる状況は、そのメッセージよりも優先度の高いメッセージがすべて同時に発生したときである。しかし、FlexRayのダイナミックセグメントの場合には、そういった状況のときに必ずしも遅れ時間が最も長くなるとは限らない。その例を図3に示す。ただし、簡単化のため、ダイナミックセグメントの開始のスロット・カウンタの値を1とし、メッセージのフレームIDもそれに合わせて表記するものとする。また、フレームIDの値がiであるメッセージをメッセージiと呼ぶ。図3において、メッセージ3の送信要求が発生してから、実際に送信されるまでについて考える。図の上段の場合(CANの場合のcritical instant)には、2サイクル目で送信される。一方、図の下段の場合には、空きスロットの影響により、3サイクル目で送信される。したがって、CANの場合の解析手法が適用できない。

また、CANにおける解析手法においては、対象メッセージ

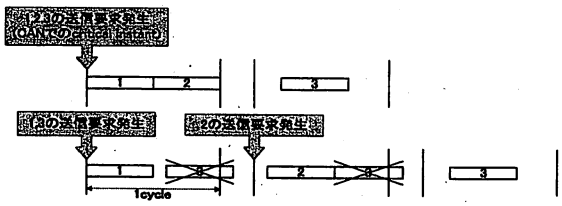

図3 CANの場合の手法が適用できない具体例

よりも優先度の低い異なるノードから送信されるメッセージについても考慮していたが、FlexRayのダイナミックセグメントの場合には、そのようなことは考慮しなくてよい。

3.2 遅れが最大となる条件

pLatestTxの値は、各ノードで設定できる。仮に、pLatestTxの値を各ノードでばらばらに設定したとすると、あるメッセージよりも優先度の低いメッセージが先に送信されるということが起こりうる。したがって、pLatestTxの値が全ノードで同じであるのか、同じでないのか、つまり、各ノードごとのメッセージ長の上限値が全ノードで同じなのか、同じでないのかを区別する。

全ノードでメッセージ長の上限値が同じ場合(a)と同じでない場合(b)についての、遅れ時間が長くなる条件を表1に整理する。ただし、遅れ時間が長いことを悪いといい、送信要求発生から実際に送信されるまでにかかる時間が最大になることを最悪になるという。また、表1の残りメッセージが多い方がより悪いかという項目は、残りメッセージの集合AとBが存在したときに、$A \supset B$が成立するとき、Aの方が遅れ時間が長くなるかということである。整理する際には、周期はサイクル長の整数倍とし、メッセージの送信要求はサイクルの最初にしか発生しないものと仮定する。

以下、表の各枠について見ていく。まず、Noの部分について、具体例を挙げて説明する。ただし、以下の具体例では、メッセージセットを((メッセージID,最大長),(メッセージID,最大長),…,(メッセージID,最大長),サイクル長)のように表記する。また、メッセージの最大長ならびにサイクル長の単位はミニスロット数とする。

	同時発生が最悪か？	最大長が最悪か？	残りメッセージ数が多い方がより悪いか？
(a)	No	Yes	Yes
(b)	No	No	No

表1 遅れ時間が長くなる条件の比較

(a)において、同時発生が最悪でない例としては、先ほど示した図3(メッセージセット:((1,5),(2,5),(3,5),10))がこの場合の具体例になる。また、(b)の方が(a)よりも条件が厳しいので、同時発生が最悪でないのは、必然である。

次に、(b)において、最大長が最悪でない例として、図4に

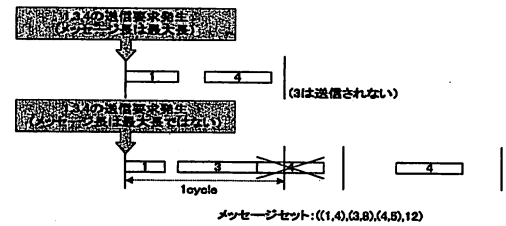

図4 (b)において、最大長が最悪でない例

具体例を示す。メッセージ4の遅れ時間について考える。ただし、メッセージ3と4は異なるノードから送信されるものとする。メッセージ1,3がメッセージ4と同時に発生したとすると、図の上段の場合(すべてのメッセージが最大長で発生)には、1サイクル目で送信できるが、図の下段の場合(メッセージが最大長ではない)には、最大長でないメッセージ1が送信されることにより、メッセージ3が送信され、メッセージ4が送信できない。したがって、メッセージが最大長のときに最悪とは限らない。

最後に、(b)において、残りメッセージ数が多い方が悪くない例として、図5に具体例を示す。メッセージ7の遅れ時間に

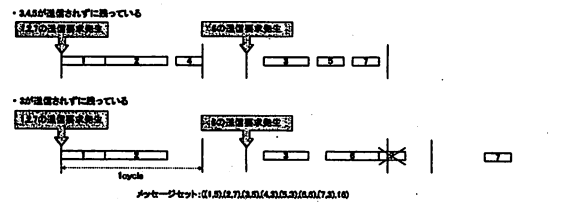

図5 (b)において、残りメッセージ数が多い方が悪くない例

ついて考える。ただし、メッセージ3と4、メッセージ6と7は、異なるノードから送信されるものとする。残りメッセージが多い図の上段の場合には、2サイクル目で送信されるが、図の下段の場合には、3サイクル目で送信される。したがって、残りメッセージが多いほうが、悪いとは限らない。

次にYesの部分について見ていく。証明をする前に、まず、以下のように状態を定義する。

状態定義：メッセージ i の最大遅れ時間について考えるとき、メッセージセットの状態を (Q,T) と表す。ただし、$Q=(q_1,q_2,..,q_i), T=(t_1,t_2,\cdots,t_i)$ とする。q_k とはメッセージ k について、送信要求が発生しているが、まだ送信が行われていないメッセージの個数のことである。(デッドラインミスが発生すると、q_k の値は2以上になる。) また、t_k とは、次回に送信要求が発生するまでにかかる最小のサイクル数である。(t_k の値が0になると、送信要求が発生する可能性があるだけで、必ず発生するわけではない)

以下では、この状態を用いるものとする。このとき、全ノードでメッセージ長の上限値が異なる場合には、メッセージの遅れ時間について以下の定理が成り立つ。

定理:メッセージ i がいつかは送信されるとき、状態 (Q,T) と状態 (Q',T') について、$\forall\, j, q_j \geqq q'_j$ かつ $\forall\, j, t_j \leqq t'_j$ ならば、状態 (Q,T) の方が悪くなる

証明:現サイクルで、2つの状態がともに、$t'_j=0$ であるいくつかのメッセージが発生したとする。このとき、送信可能なメッセージは、以下のように表すことができる。

・状態 (Q',T') : メッセージ $a_1, a_2, a_3, \cdots, a_m$

・状態 (Q,T) : メッセージ $a_1, a_2, a_3, \cdots, a_m, b_1, b_2, b_3, \cdots, b_n$

(ただし、$a_1 < a_2 < \cdots < a_m = i, b_1 < b_2 < \cdots < b_n < a_m$)

全ノードでメッセージ長の上限値が同じ場合、全ノードでpLatestTxも同じになるため、あるメッセージが送信されなければ、それ以降のメッセージIDの大きいメッセージは送信されない。すると、このサイクルで送信されるメッセージは

・状態 (Q',T') : メッセージ $a_1, a_2, a_3, \cdots, a_p$ (a_1〜a_p のメッセージはすべて送信される)

・状態 (Q,T) : メッセージ $a_1, a_2, a_3, \cdots, a_q, b_1, b_2, b_3, \cdots, b_r$ (a_1〜a_q、b_1〜b_r のメッセージはすべて送信される)

(ただし、$p \geqq q$、$1 \leqq p \leqq m$、$0 \leqq q \leqq m$、$0 \leqq r \leqq n$)

と表せる。

(ア) p＝m のとき

状態 (Q',T') では、メッセージ i が送信される。一方、状態 (Q,T) では、メッセージ i が送信されるとは限らない。また、2つの状態ともにメッセージ i が送信される場合には、$t_j=0$ かつ $t'_j \neq 0$ である j が状態 (Q,T) で、先ほどのメッセージ集合に加えて発生したとすれば、状態 (Q,T) では、メッセージ i は送信されないかもしれない。よって、状態 (Q,T) の方が悪くなる。

(イ) p≠m のとき

このサイクルでは、状態 (Q,T) も状態 (Q',T') もメッセージ i は送信されない。今回送信されなかったメッセージは、$p \geqq q$ であるため状態 (Q,T) の方が状態 (Q',T') よりも同じか多くなる。つまり、次サイクルでも $\forall\, j, q_j \geqq q'_j$ の関係が成立する。また、今回発生したメッセージ k については、$t_k = t'_k =$ (メッセージkの周期) が成り立ち、それ以外のメッセージについては、t の値は -1 され、次サイクルでも $\forall\, j, t_j \leqq t'_j$ の関係が成り立つ。そのため、次サイクルでも今回のサイクルと同様の関係が成り立つ。このようなことを繰り返すと、メッセージ i が送信されるとするなら、いずれは (ア) になるため、状態 (Q,T) の方が悪くなる。

したがって、状態 (Q,T) の方が悪くなる。(証明終)

この定理を用いて、Yesになる理由を示す。上定理の証明において、状態 (Q,T) と状態 (Q',T') の送信可能なメッセージは状態 (Q,T) の方が送信可能なメッセージが同じかより多く、また、状態 (Q,T) の方が悪くなっている。すなわち、あるサイクルで送信可能なメッセージが多いほうが悪くなる。

つまり、それ以前のサイクルでは、送信されずに残るメッセージが多い方が悪くなる。言い換えると、1サイクルで送信されるメッセージは少ないほうが悪くなる。メッセージが最大長のときに、送信するのに多くのミニスロットを使用し、1サイクルで送信されるメッセージ数が少なくなる。したがって、各メッセージが最大長のとき最悪になる。

4. 最大遅れ時間を求める解析手法

4.1 前提条件

全ノードでメッセージ長の上限値を同じに設定した場合には、通信効率が低下するが、メッセージ長の上限値の変更が生じた場合の最大通信遅延の変化を低減できるというメリットがある。そこで、全ノードでメッセージ長の上限値が同じ場合について解析を行う。

解析を行う前に、問題を簡単にする。先ほどの最悪になる条件の整理を行った際の前提に加えて、各ノードごとにメッセージの送信に十分なリソースがあることを仮定する。このとき、メッセージはそのメッセージよりも優先度の低い同一ノードから送信されるメッセージに邪魔されないことになる。

また、扱うチャネルは1つとし、目的メッセージの送信要求が発生するときに、それよりも優先度の高いメッセージの中に、送信要求が発生しているが、まだ送信されていないメッセージは存在しないものとする。

4.2 全数探索アルゴリズム

基本的な解析手法として、送信要求が発生するメッセージの組み合わせに対して、全数探索を行う。その際、前章で定義した状態を用いて、以下に示すアルゴリズムを提案する。ただし、ダイナミックセグメントの1サイクルのミニスロット数、各メッセージの周期 (送信要求が発生する最小の間隔)、各メッセージについてメッセージが最大長のときに送信に必要なミニスロットの数はあらかじめ分かっているものとする。

（1） 状態 (Q,T)、発生する送信要求の選択に使用する変数 a の初期化を行う。

$$(q_k, t_k) = \begin{cases} (1, \text{メッセージ } k \text{ の周期}) & (k = i) \\ (0,0) & (k \neq i) \end{cases}, a = 0$$

（2） 変数 a に応じて、送信要求を発生させる。変数 a を 2 進数で表したときに、右から第 j ビット目は、t の値が 0 であるメッセージの中でメッセージ ID が j 番目に小さいメッセージに対応している。値が 0 であれば、メッセージの送信要求が発生することになる。送信要求が発生したメッセージについては、q の値を +1 し、t の値をそのメッセージの周期にセットする。また、すべてのメッセージに対して、t の値を −1 する。(1 サイクル経過したことを表現するため)（ただし、t の値が 0 であるメッセージについては、t の値は 0 のままとする。）

（3） 送信を行う。メッセージ ID の小さいメッセージから見ていき、q の値が 1 以上であるメッセージがあれば、そのメッセージの送信を行い（q の値を −1 し、メッセージの長さ分だけミニスロットが使用される)、q の値が 0 であれば、1 つのミニスロットが使用される。これを pLatestTx に到達するまで行う。メッセージ i が送信されなかった場合は (4) へ、送信された場合は最悪かどうかの判定をし、(5) へ。(最悪なら最悪値の更新をする。)

（4） $a = 0$ として、次のサイクルへ ((2) へ)。次のサイクルの探索終了後、(5) へ

（5） $a = a + 1$ として、(2) へ。ただし、すべてのパターンを試していれば、このサイクルでの探索を終了する。1 サイクル目のすべての探索が終了したら終了。

4.3 枝刈り手法

全数探索を行うと、計算時間が膨大になるので、枝刈りを行い、効率的な解析の実現を目指す。そこで、以下に枝刈りに適用可能な 2 つの性質をあげる。

- 性質 1：たとえ、あるメッセージの送信要求が発生しなくても、そのメッセージを除いて、送信されるメッセージセットが変わらないとき（そのメッセージは送信されなくなる)、そのメッセージが発生しないほうが悪くなる。

証明:ある状態 (Q,T) において、以下の 2 つの場合を考える。

（1） メッセージ $a_1, a_2, \cdots, a_{k-1}, a_k, a_{k+1}, \cdots, a_p (a_1 < a_2 < \cdots < a_p)$ の送信要求が発生し、$q_j > 0$ であるメッセージ $b_1, b_2, \cdots, b_q (b_1 < b_2 < \cdots < b_q = i, a_k \neq b_x (x = 1, 2, \cdots, q))$ が存在するとする。このとき、この中から $a_1, a_2, \cdots, a_r, b_1, b_2, \cdots, b_s (k \leqq r \leqq p, s < q)$ が送信された。

（2） メッセージ a_k の送信要求が発生しない。これ以外に、(1) との違いはない。

このとき、次サイクルでは、(1) と (2) では、Q に違いはなく、T は t_k 以外は同じであり、t_k は (2) の方が値が小さい。よって、定理により、2 の方が悪くなる。(証明終)

- 性質 2:あるサイクルで送信要求が発生しても、送信されない場合には、その送信されないメッセージは次のサイクルで送信要求が発生するよりもこのサイクルで送信要求が発生した方が悪くなる。

証明:ある状態 (Q,T) において、以下の 2 つの場合を考える。

（1） メッセージ $a_1, a_2, \cdots, a_{q-1}, a_q, a_{q+1}, \cdots, a_p (a_1 < a_2 < \cdots < a_p)$ の送信要求が発生し、$a_1, a_2, \cdots, a_q$ が送信された。

（2） メッセージ $a_1, a_2, \cdots, a_{q-1}, a_q (a_1 < a_2 < \cdots < a_q)$ の送信要求が発生し、$a_1, a_2, \cdots, a_q$ が送信された。次のサイクルでメッセージ $a_{q+1}, a_{q+2}, \cdots, a_p$ の送信要求が発生した。

((1) と (2) にこれ以外の違いはないものとする)

このとき、現在のサイクルと次のサイクルにおいて送信されるものに違いはない。その次のサイクルでは、(1) と (2) では、Q に違いはない。一方、T については、(1) の方が値が小さい。よって、定理により、(1) の方が悪くなる。(証明終)

以降では、性質 1 を用いて枝刈りを行う方法を枝刈り手法 1 とし、性質 2 を用いて枝刈りを行う方法を枝刈り手法 2 とする。

5. 実　　験

5.1 適用対象

いくつかのメッセージセットを用意し、それらのメッセージセットに対して、提案したアルゴリズムを用いて最大遅れ時間の解析を行った。なお、メッセージセットを生成する際には、乱数を用いることとし、1 サイクルのミニスロット数、各メッセージの周期ならびに送信するのに必要なミニスロット数を正の整数で求めている。

5.2 実験結果

作成した複数のメッセージセットに対して、全数探索アルゴリズムのプログラムを実行したところ最大遅れ時間を求めることができた。

枝刈りを用いたときの効果を示す前に負荷率という概念についての説明をする。あるメッセージの負荷率とは、そのメッセージを送信するのに使用される時間の割合であり、複数のメッセージ（メッセージセット）に対する負荷率は、それぞれのメッセージの負荷率を合計したものとなる。

探索回数ならびに計算時間が膨大になる原因として、メッセージ数、負荷率、最大遅れ時間などがあげられる。そこで、枝刈りの効果を示すために、負荷率を固定し、メッセージ数を変化させて、メッセージ ID が最大であるメッセージの最大遅れ時間を求めた。そして、得られた最大遅れ時間が同一のものを抽出した結果の一部を表 2,3 に示す。なお、メッセージセットの負荷率はすべて約 45%のものを扱っている。

表 2 には、全数探索アルゴリズムを用いたときの結果を示している。表から分かるように、メッセージ数を増やすと、指

表 2　全数探索による結果

	最大遅れ時間：3 サイクル		最大遅れ時間：8 サイクル	
メッセージ数	探索回数	計算時間	探索回数	計算時間
8	1290	0.000s	27707	0.016s
10	22454	0.016s	486101	0.332s
12	263990	0.212s	8677978	6.756s
14	2803707	2.592s	172976925	2m31.681s
16	31913129	31.734s	3817739554	59m54.941s

表 3　枝刈り手法 1,2 を用いた結果

	最大遅れ時間：3 サイクル		最大遅れ時間：8 サイクル	
メッセージ数	探索回数	計算時間	探索回数	計算時間
8	819	0.000s	6283	0.004s
10	11814	0.012s	104901	0.076s
12	111505	0.088s	1918979	1.764s
14	1122134	1.120s	26702698	27.722s
16	11339458	13.049s	518723810	9m26.567s

数関数的に探索回数ならびに計算時間が単調増加している。そのため、現実的な時間で最大遅れ時間を求めるために、枝刈り手法が必要であることが確かめられる。

前章で示した枝刈り手法の効果について検討する。表 3 に枝刈り手法 1 と 2 の両方を用いたときの結果を示す。表 3 から分かるように、全数探索よりも大幅に探索回数ならびに計算時間を削減できている。特に、最大遅れ時間が長い場合に効果が大きく、最大で約 85%もの探索回数の削減に成功している。

次に、サイクル長だけを変化させることによって、負荷率を変化させて、メッセージの最大遅れ時間を求めた。すると、どのメッセージセットの場合にも、メッセージセット固有の負荷率を超えるとメッセージの最大遅れ時間が急激に長くなることが分かった。具体例として、メッセージ数が 14 である 2 つのメッセージセットに対して実行した結果を図 6 に示す。そのた

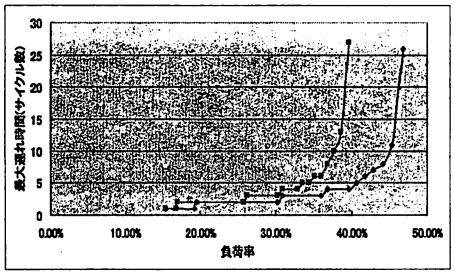

図 6　負荷率を変化させたときの最大遅れ時間の影響

め、ダイナミックセグメントを安全に使用するためには、負荷率がある値よりも低くなるようにメッセージセットを作成すればいいことが推測される。しかし、その負荷率の値はメッセージセットにより異なるため、一概に負荷率だけを指標に用いることはできない。そこで、メッセージセットの厳しさという概念を導入する。厳しさとは、サイクル長の中で実際にメッセージの送信に使用できる割合であり、以下の式で表す。ただし、サイクル長を c、平均メッセージ長を ave_len、最大メッセージ長を max_len、最大メッセージ ID を max_ID、平均送信メッセージ数を ave_n、メッセージ数を n と表す。

$$\frac{c-(max\_ID-ave\_n)-(max\_len-\frac{ave\_len}{2})}{c} \quad (1)$$

(ただし、

$$ave\_n = min(\frac{c-max\_ID-(max\_len-\frac{ave\_len}{2})}{ave\_len-1}, n) \quad (2)$$

)

この厳しさの指標、負荷率、最大遅れ時間の関係を図 7 に示す。図からわかるように最大遅れ時間が 10 以下の集合と 11 以上の集合の境界がほぼ直線になっていることがわかる。そのため、メッセージセットを作成する際には、この直線よりも右下側にくるように作成すればよいことが推測される。

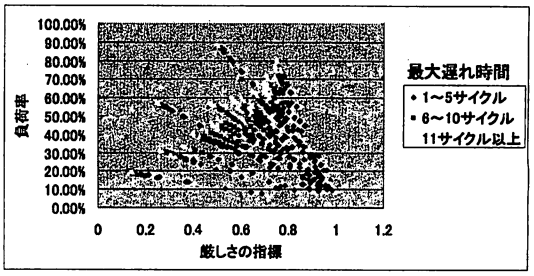

図 7　厳しさ、負荷率、最大遅れ時間の関係

6. おわりに

本研究では、FlexRay のダイナミックセグメントを用いた際のリアルタイム性を保証することを目的として、メッセージの最大遅れ時間の解析を行った。

まず、CAN の解析手法が適用できないことを指摘した。また、全ノードでメッセージ長の上限値を同じに設定するかしないかによって、遅れ時間が最大となる条件が異なってくるので、両者を区別して考え、遅れ時間がより長くなる条件を整理した。

その上で、本研究では、全ノードでメッセージ長の上限値が同じ場合について解析を行った。解析手法としては、まず、送信要求が発生するメッセージの組み合わせに対して、遅れ時間がより長くなる条件を考慮した全数探索アルゴリズムを提案した。また、枝刈り手法を提案し、現実的な時間での探索を目指した。その結果、最大で約 85%もの探索回数の削減に成功した。

また、負荷率を変更し実行すると、どのメッセージセットもそのメッセージセット固有のある負荷率を超えるとメッセージの最大遅れ時間が急激に長くなることが分かった。そこで、メッセージセットの厳しさという指標を提案し、負荷率、最大遅れ時間との関係を調べ、メッセージセットの作成をする際の指標を提案した。

今後の課題としては、更なる枝刈り手法の提案、本手法の拡張、実際の FlexRay のメッセージセットへの適用などが挙げられる。

謝辞

本研究は、トヨタ自動車との共同研究である。本研究を行うにあたり、有益なご助言を頂いた、トヨタ自動車の関係者の皆様に深く感謝致します。

文　献

[1] FlexRay Consortium. http://www.flexray-group.com/
[2] 村上 尚彦, 飯山 真一, 高田 広章, 城戸 正利, 細谷 伊知郎：自動車制御分散システムの静的スケジューリング手法（組込技術とネットワークに関するワークショップ ETNET 2005)
[3] 飯山 真一：自動車制御システムへのリアルタイムスケジューリング理論の適用（2004 年度豊橋技術科学大学大学院情報工学専攻博士学位論文）