

オフセット付き CAN メッセージの最大遅れ時間解析

飯山 真一[†] 富山 宏之^{††} 高田 広章^{††}
城戸 正利^{†††} 細谷 伊知郎^{†††}

現在、自動車の制御系ネットワークでは、CAN (Controller Area Network) が事実上の標準となっており、CAN メッセージの最大遅れ時間を求める手法が提案されている。しかしながら、従来の手法では、メッセージの送信要求時刻がオフセットを持つ状況を取り扱うことができない。より広範な実システムに対応するため、本論文では、グループ分けされたメッセージの送信要求時刻がオフセットを持つメッセージモデルに対する解析手法を提案する。また、実際の車両への適用が検討されているメッセージセットに対して提案手法を適用し、提案手法の有効性を確認した。

Response Time Analysis for Grouped CAN Messages with Offsets

SHINICHI IYAMA,[†] HIROYUKI TOMIYAMA,^{††} HIROAKI TAKADA,^{††}
MASATOSHI KIDO^{†††} and ICHIRO HOSOTANI^{†††}

CAN (Controller Area Network) is a de-facto standard of automotive networks for control. Some methods to evaluate the worst-case response time of CAN messages have been proposed. However, these conventional methods cannot evaluate response times of messages with offsets. This paper proposes a method to decide the worst-case response times of grouped CAN messages with offsets. We also apply our method to a message set currently considered using for a control network of an actual automotive, and confirm the method to be effective.

1. はじめに

近年、自動車制御は、低燃費化、走行性の向上、排気ガスや安全性に関する厳しい規制をクリアするために、大規模化、複雑化している。これにともない、車両に搭載される ECU (Electronic Control Unit) の数は増加する一方であり、その制御ソフトウェアも複雑化、大規模化している。自動車内ネットワークの導入も本格化してきており、高いリアルタイム性が要求される制御系のネットワークにおいては、現在、Controller Area Network (CAN)¹⁾ が事実上の標準として広く使われている。

ハードリアルタイムスケジューリング理論の分野においては、CAN メッセージの最大遅れ時間に関する

研究が進められており、我々は、ノードのシステム構成を考慮したモデルに対する解析手法が提案している²⁾。従来の解析手法では、あるメッセージの遅れ時間が最大となるのは、その送信要求と同時に、すべてのメッセージの送信要求が同時に出される状況としていられる。しかしながら、実際のシステムでは、メッセージの衝突を回避するために、送信要求時刻を意図的にずらし、ネットワークのピーク負荷を抑える工夫することがある。このような状況を従来手法では取り扱うことができない。

そのため、我々は、メッセージごとに初回送信要求時刻 (オフセット) が与えられた場合について、最大遅れ時間を求めるための解析手法を提案する。具体的には、同じ送信ノード内にあるすべてのメッセージを、メッセージボックスに対応した 1 つ以上のグループに分け、そのグループの中でオフセットを持たせたメッセージモデルに対する解析手法である。本論文では、適切なオフセットの与え方については扱わず、オフセットが与えられたメッセージセットに対する解析手法について扱う。

本論文では、まず 2 章では、CAN 仕様の概要と想定するシステムの構造について述べる。3 章では、CAN

[†] 豊橋技術科学大学情報工学系

Department of Information and Computer Sciences,
Toyohashi University of Technology

^{††} 名古屋大学大学院情報科学研究科情報システム学専攻

Department of Information Engineering, Graduate
School of Information Science, Nagoya University

^{†††} トヨタ自動車株式会社統合システム開発部第 1 開発室

Development Department No.1, Integrated System Engineering Division, Toyota Motor Corporation

メッセージの最大遅れ時間を求める従来手法が、オフセット付きのメッセージを持つシステムには適用することはできないことを指摘する。4章では、グループ分けされオフセットを持つメッセージモデルについて説明し、5章で、そのモデルにおいてメッセージの遅れ時間が最大となる状況を示し、その解析手法を6章に示す。7章では、提案手法を実際のシステムに適用した事例について述べ、提案手法が有効であることを示す。

2. システム構成

本章では、CAN仕様の概要と想定するシステム構成について述べる。一般的に使用されるハードウェアやソフトウェアの構成とそれらの動作について述べる。

2.1 Controller Area Network (CAN)

現在、自動車内の制御系ネットワークにおいて、Controller Area Network (CAN)¹⁾ は事実上の標準となっている。CANは最高で1Mbpsの転送速度を実現できるブロードキャストバスである。CANバス上では、最大長を持つフレーム単位でデータがやりとりされ、1つのフレームあたり最大8バイトのデータを通信することができる。メッセージには11ビットのIDが割り付けられる。IDはメッセージの優先度を表し、IDの値が小さいメッセージほど高い優先度を持つ。詳細については、文献1)を参照されたい。

本論文では、通信エラーが発生した場合の挙動を含めた議論については行わず、つねに正常な通信が行われていることを仮定する。

2.2 ハードウェア構成

複数のノードが1つのCANバスに接続されたシステムを考え、図1に示すように、ノードはホストプロセッサとCANコントローラから構成される。

コントローラは、プロトコルコントローラ、受信フィルタ、メッセージバッファ、ホストインタフェースから構成される。コントローラには、CANメッセージを格納するためのメモリブロックであるメッセージボックスが固定的に用意されており、設計時にそれらを送信用と受信用とに割り付ける。通常、メッセージの受信漏れとリアルタイム性の弊害となる再送を抑えるために、コントローラ内のほとんどのメッセージボックスは受信用として使われる。

送信用のメッセージボックスが複数ある場合は、あらかじめノード内での調停が行われ、メッセージボックスに格納されたメッセージのうち最も高い優先度を持つ1つのメッセージのみがCANバスのアービトレーションに参加できる。バスに送信を開始された

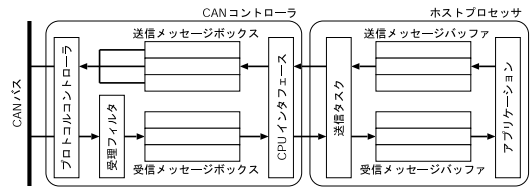


図1 CANノードの通信モデル

Fig. 1 The communication model of a CAN node.

メッセージは、そのメッセージの送信が完了するまで、送信メッセージボックスに存在する。送信完了後、コントローラは、送信メッセージボックスが空になったことをホストプロセッサへ割り込みを用いて通知する。受信フィルタは、バス上を流れるすべてのメッセージからそのノードが受け取るべきメッセージを選別し、受信メッセージボックスに格納し、受信したことをホストプロセッサへ割り込みを用いて通知する。

2.3 ソフトウェア構成

ホストプロセッサでは、CANコントローラを介して他のノードと通信を行う制御アプリケーションプログラムが動作している。制御アプリケーションプログラムは周期的にメッセージの送信要求を行う。

ホストプロセッサのメモリ上には、送信要求があったメッセージを格納する送信メッセージバッファと受信したメッセージを格納するための受信メッセージバッファがメッセージの種類ごとに固定的に確保されている。そのため、新しく到着した同じ種類のメッセージに上書きされない限り、メッセージがあふれてデータが消失することはない。

他のノードにメッセージを送信する場合は、送信メッセージバッファに送信したいデータを含んだメッセージを格納する。メッセージの送信要求ごとに起動される送信タスクは、コントローラの送信用メッセージボックスに空きがあれば、送信メッセージバッファ内で最も優先度の高いメッセージ送信用メッセージボックスに格納する。いったん、メッセージボックスに格納されたメッセージはその送信が完了するまで同じメッセージボックスに存在する。

複数のメッセージが送信メッセージボックスに格納されている場合、その中で最も優先度の高いメッセージが、バスのアービトレーションに参加でき、メッセージの送信が開始される。送信メッセージボックスに空きが発生した場合、割り込みを用いて送信タスクに通知し、新たなメッセージの格納を要求する。

受信され受信フィルタを通過したメッセージはコントローラ内の受信メッセージボックスに格納され、受信したことを割り込みにより受信タスクに通知する。通

知を受けた受信タスクは、受信メッセージボックスに格納されたメッセージをホストプロセッサ内の受信メッセージバッファに移動する。メッセージは、ホストプロセッサ内の受信メッセージバッファに格納された時点で、アプリケーションにより利用可能となる。

3. 従来手法

メッセージがオフセットを持つモデルに類似したモデルとして、マルチフレームタスクモデルがある。

マルチフレームタスクモデル³⁾は、タスクの実行時間があるパターンに従って大きく変動する状況を効率的に扱うための枠組みである。マルチフレームタスクモデルでは、Liu らにより提案された古典的な周期タスクモデル⁴⁾におけるタスクの各周期をフレームと呼び、各フレームの最大実行時間が大きい周期で変動するものとして扱われる。

一般化されたマルチフレームタスク (GMF タスク) は、マルチフレームタスクの前提を (1) 各フレームの長さが違う場合 (2) 各フレームにおけるデッドラインがそのフレームの終わりに一致しない場合を許すという 2 つの点で緩めたものである⁵⁾。

GMF タスクモデルは、複数の処理がオフセットを持ち 1 つのタスクとして実行すると見なすことができる点においては、複数のメッセージをグループ化して 1 つのメッセージとして見なす我々の扱うメッセージモデルと類似している。しかしながら、我々のメッセージモデルでは、異なる優先度を持つ複数のメッセージを 1 つのグループとして扱わなければならないのに対し、GMF タスクモデルでは、1 つのタスクの中の処理は同じ優先度であることを前提している。したがって、GMF タスクモデルでは、我々のメッセージモデルを適切に扱うことができない。

一方、CAN メッセージの最大遅れ時間を求める手法として、我々は、CAN ノード内のシステム構成を考慮した手法を提案している²⁾。しかしながら、この手法も含む従来手法では、あるメッセージの遅れ時間が最大となるのは、その送信要求と同時にすべてのメッセージの送信要求が同時に送られる状況であることを前提とされている。そのため、従来手法を用いて、オフセット付きメッセージの遅れ時間を適切に解析することはできない。

そこで本論文では、従来手法のさらに進んだ議論として、メッセージごとに初回送信要求時刻 (オフセット) が与えられた場合について、最大遅れ時間を求めるための解析手法を提案する。

解析には、GMF タスクセットの、プリエンプティ

ブな静的優先度スケジューリングによるスケジュール可能性の必要十分条件を効率的にチェックする方法として提案している maximum interference function (MIF)^{6),7)}を用いる。

GMF タスクセットでは、それに対する critical instant 定理より、あるタスクのあるフレームのスケジュール可能性をチェックするために、すべてのより優先度の高いタスクのどのフレームと同時に起動されたかを考える必要がある⁷⁾。MIF は、あるタスクがその時刻までに他のタスクを邪魔しうる最大時間を表しており、ある GMF タスクのあるフレームが他の GMF タスクへ与える影響を MIF を用いて表現することで、上記の組合せに対して、スケジュール可能性を効率良くチェックできる⁶⁾。

我々のメッセージモデルに従ったメッセージセットでは、GMF タスクセットの場合と同様に、あるグループのあるメッセージの最大遅れ時間を求めるために、すべてのグループのどのメッセージと同時に送信要求が発生したかを考えなければならない。本論文で扱うメッセージモデルは、GMF タスクモデルの前提を、各フレームの優先度が異なる場合を許すという点で緩めたものと見なせるため、メッセージの優先度を考慮した MIF を用いれば、GMF タスクモデルの解析手法である MIF も同様に適用できる。具体的には、あるグループのすべてのメッセージが他のグループのメッセージへ与える影響を MIF を用いて表現することで、上記の組合せに対して、メッセージの最大遅れ時間を効率良く求めることができる。

MIF の詳細は、文献 6) および 7) を参照されたい。

4. オフセット付きのメッセージモデル

従来手法では、メッセージの遅れ時間が最大となるのは、簡単には、そのメッセージの送信要求と同時に、すべてのメッセージも同時に送信要求が出されるという状況を仮定している。

しかしながら、実際のシステムでは、ノード内のメッセージの送信要求時刻を意図的にずらすことで、ネットワークのピーク負荷を抑える工夫をすることがある。ここでは、基準となる時刻 (典型的にはシステムのスタートアップ完了時刻) 以降からメッセージの最初の送信要求時刻を出すまでの時間をオフセットと呼ぶ。

4.1 オフセットによる効果

図 2 は、周期 2, 4, 8 の 3 つのメッセージ (周期が短いほど優先度が高い) がオフセット (周期 4, 8 のメッセージにそれぞれ 1, 3) を持って送信される状況を示している。従来手法では、つまりオフセットを

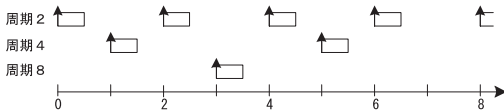


図 2 オフセット付きメッセージのスケジュール

Fig. 2 A schedule of messages with offsets.

考えないとき、周期 8 のメッセージは、その送信要求と同時に、より優先度の高い他 2 つのメッセージの送信要求が出された場合に、最大遅れ時間を持つ。この例で、周期 8 の最大遅れ時間は、3 つのメッセージの送信時間を足し合わせたものとなる。

一方、同図に示すように 2 つのメッセージの送信要求が互いに衝突しないように、オフセットを持たせた場合、メッセージの送信は互いに重ならないため、それぞれのメッセージの最大遅れ時間は、そのメッセージの送信時間と一致する。つまり、オフセットを持たせた場合は、メッセージが互いに邪魔する影響が小さくなり、結果として、メッセージの最大遅れ時間を小さく抑えることができると考えられる。

4.2 グループ分け

送信用のメッセージボックスの数は、設計時にノードごとに決定される。メッセージボックスが複数あるとき、それぞれのメッセージをどのメッセージボックスを経由して送信するかをメッセージのパラメータにより決める場合がある。たとえば、2 つのメッセージボックスがある場合、一方を優先度の高いメッセージ用、他方を優先度の低いメッセージ用に割り当てる方法などがある。メッセージの優先度に従って格納するメッセージボックスを決める方法は、優先度逆転を小さく抑えメッセージの最大遅れ時間を改善できる²⁾。

このように、すべてのメッセージは、設計時に何らかの基準に従い、いずれかのメッセージボックスに割り当てられ、同じメッセージが複数のメッセージボックス経由で送信されることはないものとする。このとき、1 つのメッセージボックスに割り当てられたメッセージの集合を 1 つのグループとして扱う。グループを区別するために、以下では、そのメッセージが所属するグループを自グループ、それ以外のグループを他グループと呼ぶ。

4.3 オフセットの決定方法

メッセージ間でオフセットを持たせるには、互いに同じ時間で同期していなければならない。CAN では、ノードごとに独立したタイマを持ち、ノード間で完全な時間同期ができない。したがって、ノードを超えたメッセージ間でオフセットを持たせず非同期に扱い、同じノード内のメッセージどうしでのオフセットを決

める。

モデルを単純にし解析を容易にするため、同じノードであっても、グループどうしは非同期として扱い、グループ内のメッセージ間において、送信要求時刻の衝突が起こらないようにオフセットを決める。非同期として扱うことは、最悪性能評価としては安全側にある。

あるメッセージのオフセットは、他のグループのすべてのメッセージに対して、その送信を邪魔する影響を最小限に抑えるように決定されることが望ましい。しかしながら、あるメッセージへの影響は小さくなるようにオフセットを求めても、他のメッセージへの影響が大きくなる場合がある。したがって、すべての他グループのメッセージに対しての影響を小さくするような最適なオフセットを求めることは単純ではない。

そのため、本論文では、オフセットの最適な決定方法については扱わず、オフセットがあらかじめ与えられているものとして議論する。ここでは、より適切なオフセットを求める指標について説明し、より具体的な方法については今後の課題とする。

最悪の状況において、他グループのメッセージの送信を邪魔する影響を小さくするためには、送信を平均化するようにオフセットを決めることが望ましい。しかしながら、それらのメッセージの送信要求周期の最大公約数が、設計時に与えられたタイマからの割込み周期以下である場合、どのようなオフセットを持たせてもいつかは衝突が発生する。そのような場合には、周期の最小公倍数が大きいメッセージどうしを同じオフセットを持たせ、平均的に衝突回数を減らす工夫を行うことも考えられる。

グループ間是非同期であり、メッセージの最大遅れ時間はグループ数にほぼ比例した影響を受けるため、可能な限りグループ数を減らした方がよい。これには、グループ数とメッセージボックス数は 1 対 1 に対応しているため、使用するメッセージボックスの数を少なくすることができるメリットもある。

5. メッセージの遅れ時間が最大となる状況

本章では、どのような状況において、グループ分けされてオフセットを持ったメッセージの遅れ時間が最大となるかを示す。メッセージの遅れ時間はそのメッセージの送信要求が発生してからその送信が完了するまでの時間である。遅れ時間が最大となる状況を示す前に、まず、対象のメッセージが置かれる状況を他グループから受ける影響のみに限定した状況を示したのち、自グループのメッセージから受ける影響も含めた

議論を行う。

以降では、メッセージの送信要求にジッタはなく、送信要求はあらかじめ決められた時刻に必ず発生することを前提とする。

5.1 メッセージボックス滞留時間の上限

あるメッセージがメッセージボックスに格納されてから、その送信を完了するまでの時間を、メッセージボックス滞留時間、もしくは単に滞留時間と呼ぶ。以下では、メッセージのメッセージボックス滞留時間の上限に関する定理を示し、その証明を行う。

定理 1 あるメッセージのメッセージボックス滞留時間は、そのメッセージがメッセージボックスに格納されると同時に、次のすべての条件を満たす状況から始めた滞留時間以下である (1) すべての他グループのそれぞれにおいて、より優先度の高いいずれかのメッセージの送信要求が発生した状況で、かつ、すべての送信においてそれぞれの最大送信時間を使いきる (2) より優先度の低い他グループのメッセージの中で最大の送信時間を持つメッセージのバスへの送信が、対象のメッセージの送信要求の直前に開始された。

証明 1 いったん、メッセージボックスに格納されたメッセージは、その送信を完了するまで同じメッセージボックスに存在し続け、また、1つのグループにつき1つのメッセージボックスが割り当てられるため、同じグループのメッセージに邪魔されることはない。したがって、メッセージの滞留時間は、そのメッセージ自身と他グループのメッセージのみにより決まる。(1) まず、1つの他グループからの影響を考える。メッセージ i が時刻 t でメッセージボックスに格納されたとき、時刻 t_{end} にその送信を完了すると仮定する。また、 t 以前で、より優先度の高いメッセージのいずれも送信されていない最も遅い時刻を t_0 とする。時刻 0 においては、どのメッセージも送信されていないので、 t_0 は必ず存在する。メッセージがメッセージボックスに格納される時刻を t_0 に変更したとしても、 t_0 と t の間は少なくとも1つのより優先度の高いメッセージが送信されているため、依然としてメッセージ i の送信は t_{end} に完了する。

次に、他グループのより優先度の高いメッセージのいずれかについて、 t_0 後の最初の送信要求時刻が t_0 に移動されたとする。このとき、メッセージの送信完了時刻は t'_{end} に遅延する可能性がある。すべてのメッセージが最大送信時間を使いきると仮定すると、送信完了時刻はさらに t''_{end} へと遅延される。

残りの他グループについても、同様の手順により、メッセージの送信遅れ時間が長くなることはあっても短く

ならないことが示される。

(2) メッセージは優先度に従って送信されるため、(t_0, t''_{end}) の間では、対象のメッセージより優先度の低いメッセージが送信されることはない。ただし、送信はノンプリエンティブに行われるため、より優先度の低い他ノードのメッセージは、時刻 t_0 に送信を開始した場合の、たかだか1回に限り送信される。したがって、 t_0 に他ノードの低優先度メッセージの中で最も長い送信時間を持つメッセージのバスへの送信が開始された場合に、対象のメッセージの送信完了時刻は最も遅れさせられる。

このとき、対象のメッセージは任意に設定した最初の状況と同じかより長い遅れ時間を持つ。また、条件 (1) と (2) は、必ずしも同時に満たされるとは限らない。したがって、メッセージの滞留時間は、これらの条件を満たす時刻 t_0 において、その送信が要求された状況から始めた場合の滞留時間以下となる。□

定理 1 より、各メッセージの滞留時間の上限は (1)、(2) の条件から始めたスケジュールを作ることによって求められる。このときの上限値を最大滞留時間とする。悲観的に見積もっているため、必ずしも正確ではないが、安全にそのメッセージの時間制約を満たすかどうかを確かめることができる。

5.2 遅れ時間の上限

オフセットを持つメッセージモデルにおいては、あるメッセージの送信が、同じグループで後に続いて送信されるメッセージの送信要求時刻を越えて、それらの送信を邪魔することが起こりうる。このような状況を取り扱うために、level- p busy period⁸⁾ の概念を適用した。Level- p busy period の概念は、もともと、単一プロセッサ上で動作するタスクモデルに対する概念であるが、本論文では、それをネットワーク上で通信されるメッセージモデルに適用する。メッセージモデルにおける level- p busy period の定義は、次のように書き直すことができる。Level- p busy period とは、どのメッセージも送信されていない時刻 t_0 以降に開始された優先度 p と等しいかより高い優先度を持つすべてのメッセージの送信が最初に終了する時刻 t までの区間 $(t_0, t]$ のことである。

Level- p busy period の概念を用いると、メッセージの最大遅れ時間は次の定理で示される。

定理 2 あるメッセージの最大遅れ時間は、そのメッセージまたは先行するメッセージが最大滞留時間を持つような状況から始まる level- p busy period の中にある。ここで、 p はそのメッセージと先行するメッセージの中で最も優先度の低い優先度を示している。

証明 2 あるメッセージ i の送信要求が時刻 t^i に発生し、時刻 t_{end}^i にその送信を完了すると仮定する。また、その送信が後に続くメッセージ j の送信要求時刻 t^j を越えたとき、メッセージ j は最大遅れ時間を持ち、時刻 t_{end}^j に送信を完了すると仮定する。また、 t^i から t_{end}^i の間には、 i 以上の優先度を持つメッセージが常に送信されているものとする。 t^i 以前で、 i より優先度の高いメッセージのいずれも送信されていない最も遅い時刻を t_0 とする。時刻 0 においては、どのメッセージも送信されていないので、 t_0 は必ず存在する。

(1) まず、メッセージ i の送信要求時刻 t^i を t_0 に変更したとしても、 t_0 と t_{end}^i には、少なくとも 1 つのより優先度の高いメッセージが送信されているため、依然としてメッセージ i の送信は t_{end}^i に完了する。

(2) 次に、 i より優先度の高い他グループの先頭にあるメッセージの送信要求時刻を t_0 に変更したとする (t_0 に移動したメッセージを先頭にしたスケジュールに従って、それ以外のメッセージの送信要求が発生する)。このとき、他グループのメッセージがメッセージ i を邪魔する回数は変わらないか増加するため、メッセージ i の送信完了時刻は $t_{end}^{i'}$ に遅延される可能性がある。これにともない、メッセージ j も直接的、また、 i により間接的にこれらのメッセージに邪魔される回数が変わらないか増加するため、メッセージ j の送信完了時刻は $t_{end}^{j'}$ に遅れる可能性がある。それぞれのメッセージは、最大送信時間を使いきると仮定すると、メッセージの送信完了時刻は $t_{end}^{i''}$ へとさらに遅延される。

(3) 同様に j より優先度の高い他グループの先頭にあるメッセージの送信要求時刻を t_0 に変更した場合、メッセージ i と j の送信完了時刻は変わらないか遅れる可能性がある。

(4) さらに、メッセージは優先度に従って送信されるため、 $(t_0, t_{end}^{j''}]$ の間では、対象のメッセージより優先度の低いメッセージが送信されることはないが、送信はノンプリエンティブに行われるため、より優先度の低い他ノードのメッセージは、時刻 t_0 に送信を開始した場合の、たかだか 1 回に限り送信され得る。したがって、対象のメッセージは、 t_0 において、他ノードの低優先度メッセージの中で最も長い送信時間を持つメッセージのバスへの送信が開始された場合に、対象のメッセージの送信完了時刻が最も遅れる。今、 t_0 は、メッセージ i の滞留時間を最大とするような時刻であり、 j は任意に設定した最初の状況と同じかより長い遅れ時間を持つ (この状況は、 i の滞留

時間を最大とする状況であり、 j にとってのそれではない)。

よって、メッセージ j の最大遅れ時間は、そのメッセージまたは先行するいずれかのメッセージの送信要求が、その滞留時間を最大となる状況から始まる level- p busy period $(t_0, t_{end}^{j''}]$ の中に存在する。□

この定理より、メッセージの最大遅れ時間が含まれる範囲を限定されうる。

6. 解析手法

本章では、前章で示した 2 つの定理に基づいたオフセット付きメッセージの最大遅れ時間の解析手法を示す。本手法は、CAN のようなメッセージが優先度ベースで、かつ、周期的に送信されるようなバス型ネットワークを前提としており、この前提が満たされていれば、同様のネットワークにも適用できる可能性はある。

6.1 メッセージボックス滞留時間の解析

メッセージの滞留時間は、定理 1 で示した条件から始めたスケジュールをすることで求められる。しかしながら、あるメッセージの最大遅れ時間を求めるためには、その送信要求と同時にすべての他グループのどのメッセージの要求が発生したかを考えなければならない。この組合せを効率良く表現する方法として MIF を用いると、あるメッセージ i のメッセージボックス滞留時間の上限は、次の手順により求めることができる。

- (1) すべての他グループのそれぞれにおいて、 i よりも優先度の高いメッセージの MIF を求める。
- (2) 求めたすべての MIF と他グループのより優先度の低いメッセージが持つ最も長い送信時間との飽和加算を行う。飽和加算は複数の MIF を 1 つの MIF に、MIF の性質を維持しながら足し合わせるための演算である。飽和加算の詳細については、文献 2) を参照されたい。
- (3) 飽和加算後の MIF において、その傾きが時刻 0 以降に初めて 0 となる時刻を t とする。この時刻は、バス上にいずれのメッセージも送信されていない最初の時刻であり、対象のメッセージが送信を開始できる時刻である。したがって、対象のメッセージのメッセージボックス滞留時間は、 $t + C_i$ となる。ここで、 C_i はメッセージ i を送信するのに要する最大時間である。

6.2 遅れ時間の解析

定理 2 より、あるメッセージの最大遅れ時間の上限が含まれる範囲が限定される。ここでは、遅れ時間に影響する要因をあげ、遅れ時間を求める具体的な方法

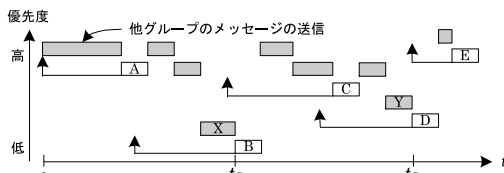


図 3 先行するメッセージの優先度による影響
Fig. 3 The affect of precedence messages.

について述べる。

あるメッセージの送信が、自グループの先行するメッセージのいずれからでも、その送信要求時刻を越えて邪魔されない場合は、自グループのメッセージからの影響を考慮しなくてもよい。つまり、そのメッセージの最大遅れ時間は、メッセージボックス滞留時間と一致する。

一方、先行するメッセージが送信要求時刻を越えた場合、他グループのメッセージから受ける影響が変わるため、その影響を見積もることが必要となる。

以下では、先行するメッセージの送信のすべてが順々に次の送信されるメッセージの送信要求時刻を越え、対象のメッセージにまで影響が伝播した場合についてのみ考える。つまり、先行するメッセージのいずれかが、その次のメッセージの送信要求時刻を越えて送信されない場合は対象としない。

6.2.1 中間の優先度を持つメッセージによる影響

先行するメッセージによる影響は、そのメッセージと対象のメッセージの優先度の関係により変わる。先行するメッセージのすべてが、対象のメッセージよりも優先度が高い場合、対象のメッセージはより優先度の高いメッセージによる影響だけを考えればよい。

一方で、先行するメッセージのいずれかが対象のメッセージよりも低い優先度を持つ場合、優先度逆転現象が発生し、両者の中間の優先度を持つ他グループのメッセージにより、対象のメッセージの送信完了時刻が遅れる可能性がある。

先行するメッセージがあることにより中間の優先度に邪魔される例を図 3 に示す。同図は、メッセージ A から D までの送信が、順にその次のメッセージの送信要求時刻を越えており、メッセージ E で初めて越えない状況を示している。

いま、対象のメッセージを C とすると、先行するメッセージの中により優先度の低い B が存在する。そのため、C の遅れ時間には、時刻 0 から B が送信を開始されるまでの時間 $(0, t_B]$ に、B が B と C の中間の優先度を持つ他グループのメッセージ X に邪魔される時間を加えなければならない。

次に、E を対象のメッセージとすると、先行するメッセージの中により優先度の低いメッセージ C が存在するが、C と E との中間の優先度を持つメッセージによる影響は、D と E との中間の優先度を持つメッセージによる影響に含まれるため、考慮しなくてもよい。

このときのメッセージ E の遅れ時間は、自身が E より優先度の高いメッセージに邪魔される時間に、D と E の中間の優先度を持つメッセージ Y が時間 $(0, t_D]$ で D を邪魔する時間と、B と D の中間の優先度を持つメッセージ X が時間 $(0, t_B]$ で B を邪魔する時間を加えたものとなる。

6.2.2 後続のメッセージの追越しによる影響

先行するメッセージが送信されないまま、対象のメッセージがさらに後続のメッセージの送信要求時刻を越えた場合で、かつ、後続のメッセージの優先度が対象のメッセージよりも優先度が高いとき、後続のメッセージが対象のメッセージを追い越して送信が行われる。

一方で、対象のメッセージがこの後続のメッセージと同じ状況である場合、先行するメッセージを追い越して送信される可能性がある。しかしながら、他グループのメッセージの送信時間を少し短くするなどして、先行するメッセージをメッセージボックスに格納させた状態で、対象のメッセージを邪魔させるようにした方が、より悪い状況を作りだすことができる。したがって、先行するメッセージに対しては、その優先度にかかわらず、対象のメッセージをかならず邪魔するものとする。

6.2.3 遅れ時間の解析

定理 2 では、メッセージの最大遅れ時間の上限が存在する範囲を示している。この定理より、あるメッセージの最大遅れ時間は、同じグループの先行するメッセージが最大滞留時間を持つような状況から、そのメッセージの影響が後続のメッセージに与えなくなるまでの間に含まれることが分かる。いい替えれば、定理 2 により限られた時間内で、定理 1 で示された状況におけるすべての組合せを試した中で、最も長い遅れ時間を最大遅れ時間とする。この組合せと前節までの遅れ時間に影響する要因を MIF を用いて表現すると、ある 1 つのグループにあるそれぞれのメッセージの最大遅れ時間の上限は、次の手順に従って求めることができる。

- (1) まず、そのグループ内にあるすべてのメッセージの周期の最小公倍数 (ハイパーピリオド) を求める。ハイパーピリオド内で送信されるメッセージの 1 つを先頭に (送信要求時刻を 0) したスケジュールを作る。

- (2) そのメッセージの滞留時間を求める．
- (3) そのメッセージの送信が次のメッセージ i の送信要求時刻を越えない場合は、 i を先頭にしたスケジュールに作り直し、手順 2 からを繰り返す．ハイパーピリオド内のすべてのメッセージを先頭にしたスケジュールについて試し終えた場合は、解析を終了する．
- (4) 送信要求時刻を越える場合は (1) すべての他グループのそれぞれにおいて、 i より優先度の高いメッセージの MIF (2) 自グループの先行するすべてのメッセージと後続の優先度の高いメッセージの MIF (3) 先行するメッセージがあることで間接的に邪魔される中間の優先度を持つメッセージの MIF を求める．
- (5) 求めたすべての MIF とより優先度の低い他グループのメッセージが持つ最も長い送信時間との飽和加算を行う．
- (6) 飽和加算後の MIF において、その傾きが時刻 0 以降に初めて 0 となる時刻を t とする．この時刻は、パス上にいずれのメッセージも送信されていない最初の時刻であり、対象のメッセージが送信を開始できる時刻である．したがって、メッセージ i の遅れ時間は、 $t - t_i + C_i$ となる．ここで、 t_i はメッセージ i の送信要求時刻である．
- (7) メッセージ i の送信がさらに次のメッセージの送信要求時刻を越える場合は、スケジュールの先頭のメッセージを変更せずに、次のメッセージを新しい i として、手順 (4) からを繰り返す．越えない場合は、次のメッセージを先頭にしたスケジュールに作り直し、手順 (2) からを繰り返す．

あるメッセージの送信が、自グループの先行するメッセージのいずれからでも、その送信要求時刻を越えて邪魔されない場合、最大遅れ時間は滞留時間と一致し、手順 (1) ~ (3) は、滞留時間を求める手順と同等である．送信要求時刻を越えて邪魔される場合は、さらに 6.2.1 項と 6.2.2 項において示した遅れ時間に影響する要因を考慮した手順 (4) 以降が行われる．

上記の手順をすべてのグループに対して行うと、すべてのメッセージについて、遅れ時間の候補がいくつか求まる．滞留時間を含めた候補の中で、最も長いものをそのメッセージの最大遅れ時間 (の上限) とする．

7. 適用

現在開発中の自動車への適用が検討されているメッ

セージセットに対して提案手法を適用し、メッセージの最大遅れ時間を求め、それらが時間制約を満たすかどうかを確認した．また、メッセージにオフセットを持たせた場合の最悪遅れ時間への効果を評価するために、文献 2) でのオフセットを持たない場合の評価式 (以下、評価式) を用いて、同一条件の下で比較を行った．

7.1 適用対象

CAN では最高 1Mbps のデータ伝送が可能であるが、電磁放射ノイズや安定性などを考慮して、高速な制御系のネットワークにおいても、半分の 500 Kbps 以下で用いられる．そのため、以降の適用においても、CAN の転送速度を 500 Kbps と想定する．

適用の対象とするメッセージセットは、16 のノード間で通信される約 80 種類の周期メッセージから構成されており、約 40% のバス負荷率を持つ．

7.2 オフセットによる効果

メッセージ間でのオフセットによる効果を評価するために、オフセット以外の条件に同一として、評価式を用いた場合との比較を行った．具体的には、まず、4.2 節の方法に基づいてメッセージのグループ分けを行った．ノードあたり 1~4 個のメッセージボックスを割り当て、メッセージセットを 28 のグループに分割を行った．

評価式を用いた評価では、メッセージボックスに関する条件を合わせるために、1 つのグループを 1 つのノードに対応させて、すべてのメッセージのオフセットを 0 とした．一方、本手法を用いた評価では、4.3 節に示す指標に基づいて、同じグループのメッセージとの衝突が起こらないようにオフセットを定めた．その際、他グループのメッセージへの影響ができる限り小さくなるようにオフセットを試行錯誤をしながら求めたため、必ずしも最適なオフセットではないことに注意されたい．より適切なオフセットを求めることで、今回示す結果よりも良くなる可能性は十分にある．

オフセットを持たせることで優先度逆転による影響を小さくできると考えられるため、優先度ごとのオフセットによるその効果を評価するために、メッセージを優先度により 3 つのグループに分けて行った評価を

表 1 オフセットによる効果
Table 1 The effects of message offsets.

	遅れ時間の減少率 (%)	メッセージ減少数
高優先度	71.2	27.9
中優先度	53.0	34.9
低優先度	46.8	43.0
全平均	58.3	35.1

表 1 に示す．同表には，オフセットがない場合からオフセットがある場合の遅れ時間の平均減少率と，邪魔されるメッセージの平均減少数を示している．平均減少率は，オフセットがある場合の遅れ時間とない場合の遅れ時間との比を 1 から引いたものと定義しており，値が大きいほどオフセットを持たせた効果が大きいことを示している．

今回のメッセージセットに対してオフセットを持たせることで，メッセージの最大遅れ時間を平均 58% 小さく抑えられる結果が得られた．優先度の高いメッセージが大きな減少率を持つのは，オフセットを持たせることで，優先度逆転現象による影響が小さくなるためと考えられる．

先行するメッセージが送信要求時刻を越えたことにより，遅れ時間が長くなったメッセージは 5 つあり，このときの遅れ時間は，最大で滞留時間の 2.3 倍となった．優先度の高いメッセージほど周期が短く，先行するメッセージの送信要求時刻との間隔が小さくなる傾向があるため，優先度の高いメッセージが先行のメッセージによる影響を大きく受けることが分かる．

以上により，最大遅れ時間を小さく抑えるために，メッセージの送信要求時刻にオフセットを持たせることが有効な場合があることが確認できたが，すべてのメッセージセットにおいて，最大遅れ時間が短くなるとはいえない．たとえば，送信周期が互いに素であるような，どのようなオフセットを持たせたとしても送信要求の衝突が発生するようなメッセージセットである場合，本質的に最悪ケースにおいてはオフセットの効果が見えない．しかしながら，周期が互いに素であっても，周期の変更が認められる場合には，オフセットを持たせられるように周期を変更することにより，オフセットの効果を得ることができる．このように，周期の変更が認められる場合について，オフセットの効果をも大きくするような周期の変更方法やオフセットの求め方については今後の課題である．

8. 結 論

本論文では，グループ分けされたメッセージ間で送信要求時刻のオフセットを持つメッセージモデルに対する解析手法を提案した．また，提案手法を実システムのメッセージセットに適用し評価を行い，このメッセージセットに対しては，オフセットを持たせない場合に比べ，メッセージの遅れ時間を平均で 50% 以下に抑えられる結果が得られた．これにより，メッセージのグループ分けを行い，送信要求時刻にオフセットを持たせることは，最悪時のメッセージの衝突回数を

減らし，遅れ時間を短くすることに有効であることが実際の例において確認できた．オフセットによる効果はメッセージセットに依存するため，他のメッセージセットに対しても同様の結果が得られるとは限らないことに注意されたい．

本論文では，メッセージのオフセットはあらかじめ与えられたものと仮定しているが，より適切なオフセットを与えることで，さらに衝突回数を減らし最大遅れ時間を短くすることのできるものと考えられる．最適なオフセットを与える方法については今後の課題である．

提案手法は，従来手法では取り扱うことのできない状況を扱うことのできるため有用であるといえる．本論文では，1 つのメッセージセットに対しての適用と評価を示したが，提案手法はオフセット付きのメッセージモデルに対する解析手法であり，オフセット付きのメッセージセットに対しては一般的な手法であるため，他のメッセージセットに対しても有効である．さらに多くのメッセージセットへの適用およびその評価は今後の課題である．

参 考 文 献

- 1) International Standards Organization: ISO 11898. Road Vehicles — Interchange of digital information — Controller area network (CAN) for high speed communication (1993).
- 2) 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).
- 3) Mok, A.K. and Chen, D.: A multiframe model for real-time tasks, *Proc. Real-Time Systems Symposium*, pp.22–29 (1996).
- 4) Liu, C.L. and Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J. ACM*, Vol.20, No.1, pp.46–61 (1973).
- 5) Baruah, S., Chen, D., Gorinsky, S. and Mok, A.: Generalized multiframe tasks, *Real-Time Systems*, Vol.17, No.1, pp.5–22 (1999).
- 6) 高田広章，坂村 健：マルチフレームタスクセットの静的優先度スケジューリングによるスケジュール可能性，信学技報 (1997 年実時間処理に関するワークショップ RTP'97), Vol.96, No.596, pp.29–35 (1997).
- 7) 飯山真一，高田広章，菅沼英明：エンジン制御システムのためのリアルタイム性検証手法，情報処理学会論文誌，Vol.43, No.6, pp.1715–1724 (2002).
- 8) Lehoczky, J.P.: Fixed Priority Scheduling of

Periodic Task Sets with Arbitrary Deadlines,
Proc. Real-Time Systems Symposium, pp.201-
209 (1990).

(平成 16 年 1 月 31 日受付)

(平成 16 年 6 月 17 日採録)



飯山 真一 (学生会員)

2002 年豊橋技術科学大学大学院
情報工学専攻修士課程修了。現在、
同大学院電子・情報工学専攻に在学。
リアルタイムスケジューリング理論
とその応用に関する研究に従事。修

士 (工学)。



富山 宏之 (正会員)

1999 年九州大学大学院システム情
報科学研究科博士後期課程修了。同
年より米国カリフォルニア大学アー
バイン校客員研究員。2001 年 (財)
九州システム情報技術研究所研究員。

2003 年名古屋大学講師。現在同大学助教授。EDA, コ
ンパイラ等の研究に従事。博士 (工学)。電子情報通
信学会, ACM, IEEE 各会員。



高田 広章 (正会員)

名古屋大学大学院情報科学研究科
情報システム学専攻教授。1988 年
東京大学大学院理学系研究科情報科
学専攻修士課程修了。同学科の助手,
豊橋技術科学大学情報工学系助教授

等を経て, 2003 年 4 月より現職。リアルタイム OS,
リアルタイムスケジューリング理論, 組み込みシステ
ム開発技術等の研究に従事。ITRON 仕様の標準化活
動に, 中心的メンバとして参加。博士 (理学)。IEEE,
ACM, 電子情報通信学会, 日本ソフトウェア科学会
各会員。



城戸 正利

1990 年トヨタ自動車 (株) 入社。
シャシー系/パワートレイン系/ボ
デー系の制御ソフト開発支援システ
ム開発を実施。現在, 主に車載 LAN
先行開発および開発支援システム開

発に従事。



細谷伊知郎

1984 年トヨタ自動車 (株) 入社。
東富士研究所においてパワートレイ
ン電子制御 (主にバイワイヤー) 開
発に従事後, トヨタ本社にてパワ
ートレインソフト開発, 車載通信開発

関連を実施。現在, 車載通信の先行開発に従事。