7K-08

デッドラインに基づくスケジューリングが可能なCAN通信ソフトウェア

橋本　優太†　　横山　孝典†　　兪　明連†

東京都市大学‡

1. はじめに

組み込み制御システム分野では複数の組み込みコンピュータをネットワークで接続した分散型の制御システムが多く用いられている．また，自動車や産業機器などではリアルタイム性のあるネットワークとして，メッセージに優先度を付加できるCAN(Controller Area Network)[1]が用いられている．しかし，一般にCANを用いる場合メッセージの種類ごとに固定したメッセージIDを与えることが多く，固定優先度に基づくメッセージ通信となっている．このため，優先度の低いメッセージは通信遅延が大きくなりがちであるとともに，時間経過によるメッセージの緊急性の上昇を反映できないという問題がある．そこで，デッドラインの近いメッセージを優先して送信する手法が求められている．その一つとして送信失敗時にメッセージIDを更新する手法が提案されているが，1度の失敗にしか対応していない[2]．

本論文では，デッドラインの近いメッセージを優先して送信するため，CANメッセージの優先度を表すメッセージIDにデッドラインを含め，デッドラインに基づくスケジューリングが可能なCAN通信ソフトウェアを提案する．

2. EDFに基づくCAN通信

2.1. CAN通信プロトコル

CANはCSMA/CR(Carrier Sense Multiple Access with Collision Resolution)と呼ばれる方式に基づいて通信を行う．CANメッセージは優先度を表すメッセージIDを含んでおりメッセージIDが小さいほど優先度が高い．複数のメッセージが同時に送信された場合は，アービトレーションにより優先度を比較し，優先度の高いものを送信する．

2.2. デッドラインの導入

本研究ではメッセージの優先度を表すメッセージIDにデッドラインを含めることで，デッドラインが近いほど優先度の高いEDF(Earliest Deadline First)に基づく通信を実現する．EDFに基づく通信により，固定優先度ではミスをしてしまうデッドラインを守れる場合がある．1メッセージの送信時間を0.25msecとし，周期5msecのメッセージを10，周期10msecのメッセージを10，周期15msecのメッセージを15送信する場合の例を図1,図2に示す．固定優先度では周期が短いほど優先度を高くするように設定している．図1に示す固定優先度の場合は，周期15のタスクは送信時刻15msecまでに5つのメッセージが送信できずデッドラインミスをしてしまう．一方，図2に示すEDFの場合はデッドラインを守ることが可能となる．

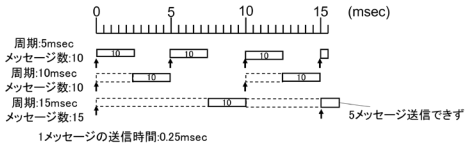

図1:固定優先度の場合の通信例

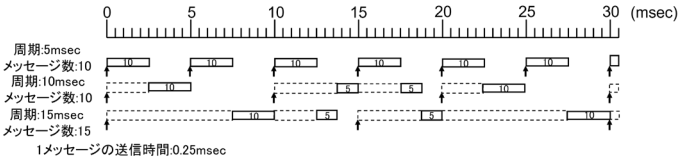

図2:EDFの場合の通信例

デッドラインを表現するには，時刻の絶対値で表される絶対デッドラインを用いる方法と，絶対デッドラインまでの時刻である相対デッドラインを用いる方法がある．本論文では，メッセージIDに絶対デッドラインを含める方法と相対デッドラインを含める方法の2つを提案する．

2.3. 絶対デッドラインを用いる方法

本研究では29bitのメッセージIDを使用する．図3に絶対デッドラインを用いる場合のメッセージIDのフォーマットを示す．下位11bitを従来の11bitメッセージIDを用いた場合のCANメッセージとの互換性を保つために使用し，上位18bitを絶対デッドラインに使用する．

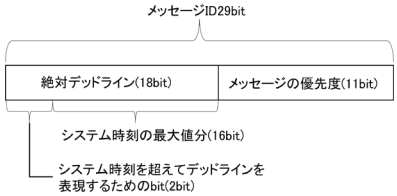

図3:絶対デッドライン使用時のメッセージID

CAN Communication Software with Deadline-Based Scheduling
† Yuta Hashimoto, Takanori Yokoyama and Myungryun Yoo
‡ Tokyo City University

絶対デッドラインはリアルタイム OS のシステム時刻で表現する．しかしながら，図 4 に示すようにシステム時刻リセット時をまたぐと,デッドラインの値が逆転する場合がある．そこで，図 5 に示すようにシステム時刻リセット後の再送信要求時に，当初のデッドラインからシステム時刻の最大値分を差し引くことでデッドラインの逆転を防ぐ．

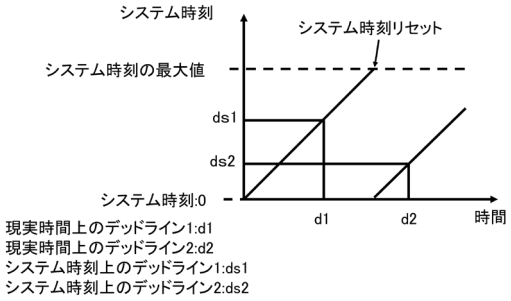

図 4:デッドライン逆転の例

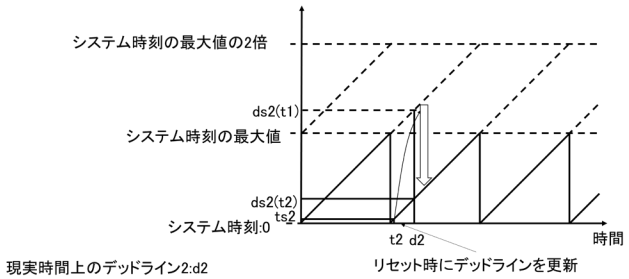

図 5:システム時刻リセット後の動作

また，正しくデッドラインに基づく通信を行うには，各ノードでのシステム時刻のリセットタイミングを一致させる必要がある．そこで，基準となるマスターノードを定め，マスターノードのシステム時刻がリセットされるタイミングで高優先度の同期メッセージをブロードキャストし，全ノードのシステム時刻の同期を行う．

2.4. 相対デッドラインを用いる方法

図 6 に相対デッドラインを用いる場合のメッセージ ID のフォーマットを示す．上位 18bit の内 16bit を相対デッドラインに使用する．残りの 2bit はメッセージに別の優先度を付加する際などに使用するなど，ユーザが定義可能な bit とする．

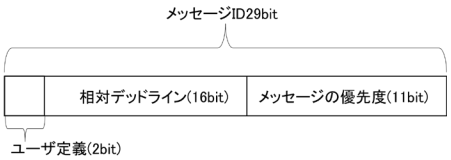

図 6:相対デッドライン使用時のメッセージ ID

図 7 に相対デッドラインを使用する場合の動作を示す．アービトレーションにより送信が中断された場合には，再送要求時に絶対デッドラインから現在時刻の値を差し引くことで相対デッドラインを更新する．

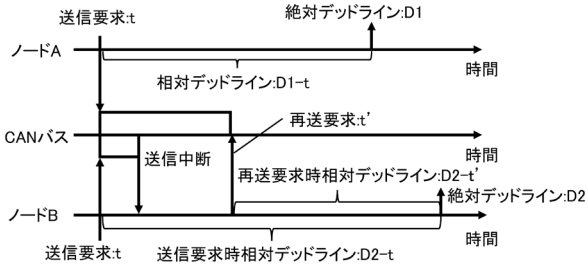

図 7:相対デッドライン使用時の動作

相対デッドラインを使用する方法は，アービトレーションによる送信中断時の相対デッドラインの更新によりオーバーヘッドが増大する．しかしながら，デッドラインまでの残り時刻で優先度が決まるため，時刻同期の精度の影響を受けにくいという利点がある．

3. 実装

実装には CPU として V850E2/FG4-L を搭載した評価ボードを用いる．システム時刻リセット時やアービトレーションによる送信中断時にメッセージ ID を更新するため，CAN コントローラによる自動再送機能は用いず，ソフトウェアでメッセージの再送要求を行う． TOPPERS プロジェクト[3]が開発した通信スタックである A-COMSTACK(Automotive COMSTACK)の CAN ドライバをベースに，提案した機能を実装した．

4. おわりに

本論文では，CAN メッセージの優先度を表すメッセージ ID にデッドラインを含めることで，デッドラインに基づくスケジューリングが可能な CAN 通信ソフトウェアを提案した．今後は提案した CAN 通信を用いた分散処理環境を開発する予定である．

謝辞

本研究で使用した TOPPERS の CAN ドライバの開発者に感謝する．本研究は JSPS 科研費 JP18K11225，JP21K1815 の助成を受けたものである．

参考文献

[1]. U Kiencke,"Controller Area Network-from Concept to Reality " Proc.1st International CAN Conference, pp.0-11-0-20 (1994)

[2]. Mohamad Reza Pourmoghadam, Yasser sedght, Ismail Ghodsollahee, “Improving the Fault Tolerance and Efficiency of CAN Communication Networks Based on Bus Redundancy”, 2020 10th International Conference on Computer and Knowledge Engineering(2020)

[3]. TOPPERS プロジェクト, TOPPERS プロジェクトホームページ, https://www.toppers.jp/index.html