

車載 ECU 向けソフトウェア更新システムの評価環境

杉本俊輔^{†1} 染谷一輝^{†2} 鈴木孝幸^{†1} 清原良三^{†1}

概要 : OTA(Over-the-Air)更新を用いた車載 ECU のソフトウェア更新の需要は高まる一方で, CAN の伝送速度の限界や, ECU の書き換えにかかる時間が大きな課題となっている. データ圧縮アルゴリズムを用いた車載 ECU のソフトウェア更新や, CAN プロトコルの提案など, 様々な手法が提案されてきた. 本論文では, データ圧縮や CAN の構成によって更新時間にどの程度違いが出てくるのかを評価するためのプラットフォームを提案する. また, シミュレータを用いてソフトウェア更新時の CAN の伝送速度を評価した.

キーワード : OTA, Software Update, ECU, compression

An Evaluation Environment for in-Vehicle ECU Software Updates

SHUNSUKE SUGIMOTO^{†1} KAZUKI SOMEYA^{†1}
TAKAYUKI SUZUKI^{†1} RYOZO KIYOHARA^{†1}

1. はじめに

現在多くの車はブレーキやエンジンなどを ECU (Electronic Control Unit)によって制御している. 自動車に搭載される ECU の数が増加し, それに伴いソフトウェアの不具合によるリコールも増加している[1]. 2016年の時点で ECU は1台の車に平均 21.6 個搭載されていて, ECU の搭載数は 2025 年に平均で 30.4 個になると予想されている. そのため, ソフトウェアの不具合によるリコールはさらに増える予想される.

現在, ソフトウェアの不具合があった場合には, 自動車の所有者はディーラに持ち込み更新を行っている. 更新の間は自動車を使用することができず, ユーザビリティを低下させてしまうため短時間での更新が望ましい. またディーラのサービスの観点からは, 本来, 来て欲しい客まで混雑の関係で来店できなくなるのは好ましいことではない. 一方で, 客に足を運んでもらうメリットもあるので, 店頭で WiFi を利用したソフトウェア更新の必要性もある.

ユーザビリティの低下を防ぐために, OTA(Over The Air)が開発され, 車載 ECU のソフトウェア更新を広域無線通信およびインターネットによって行う試みがなされている. 現在一部の機能の更新を広域無線通信によって行っているメーカーもあり, 今後 5G の普及によってさらに OTA の需要は高まると予想される. OTA の需要が高まる一方で, 自動

車がインターネットに繋がるためサイバー攻撃の標的にされる危険性もある. そのため定期的にセキュリティのアップデートを行う必要もある.

一方, 多くの車に用いられている CAN(Control Area Network)の伝送速度は 4G に比べて遅い. CAN の伝送速度はすでに限界に達しており, データサイズを削減して送ることによって更新時間の短縮を行うという手法がとられてきた[2]. CAN に代わる次世代車載 LAN プロトコルとして FlexRay[3]が注目されていたが, 多くの CAN アプリケーションの再構成が必要となり, コスト面での課題があるため, CAN のシステムを FlexRay へ移行するにはまだ時間がかかる.

ECU は省メモリ・省電力・低下価格・リアルタイム性などさまざまな制約条件がある. 一方, ソフトウェア更新におけるデータ削減アルゴリズムも, 省メモリの方式から, メモリは必要だが, 圧縮率が高い方式などがあり, また適用する際のソフトウェアの変更量によってもどの方式が最適になるかは依存している. つまり, 工場では様々な方式による更新データを作成の上, ECU の制約条件が守れるかどうかの確認作業が必要となる.

そこで, 本論文では ECU の更新を行う際の最適な手法を選択するためのプラットフォームを提案する.

^{†1} 神奈川工科大学
Kanagawa Institute of Technology

^{†2} 神奈川工科大学大学院
Graduate School of Kanagawa Institute of Technology

2. 関連研究

CAN の最大通信速度は規格上 1Mbps とされているが、複数の ECU がバス型に接続されることによる複雑な反射の影響から実際の車載ネットワークでは 500kbps 程の速度となる。文献[4]では、10MbpsCAN プロトコルの提案と構成について述べられている。従来のバス型接続に加え、スター型接続を提案し、伝送路遅延を解決しようとしている。また、10MbpsCAN のシステムを考慮したゲートウェイにおけるメッセージの最大遅れ時間について提案し、プロトコルシミュレータの結果と比較することで提案された解析手法の有効性を確認している。

文献[2]では、CAN における更新時間の短縮のための、車載 ECU に適応可能な差分更新方式を提案している。Flash ROM の消去ブロックを基本単位として差分を生成するブロック差分方式を提案している。この提案方式では差分生成方式とアルゴリズムの変更により圧縮率を維持しつつ、ECU に搭載可能なレベルの省メモリ化を実現している。文献[5]などのこれまでの多くの差分更新技術は RAM が十分あることが前提となっている場合が多く、

差分更新ではフラッシュメモリの消去ブロックサイズ以上の RAM が必要であり、ECU の種類によっては、差分更新が適用できないものがある。文献[5]では、差分更新が適用できないような RAM が小さな場合にも単なるデータ圧縮に比べデータ量を削減する圧縮方式として、データを圧縮するための圧縮用辞書をあらかじめ出荷時にフラッシュメモリに記憶し、新版を作成する際にもこの旧版の辞書を再利用して圧縮することを提案している。

以下の表 1 に、メモリの使用量や圧縮率をまとめた。想定される ECU のスペックは文献[2]を参照した。差分更新を使う場合は、圧縮率が良いが、RAM 上に消去ブロック分のイメージの作成が必須となる。旧版の情報を利用するため少しずつ適用することができないためである。汎用的な圧縮は、ソフトウェアの更新として考えると圧縮率が低いかわりに、旧版のイメージを使う必要もないため、RAM 的

表 1 各アルゴリズムの圧縮率と RAM 使用量

	圧縮率	RAM 使用量	Flash メモリ使用量
gzip	△	○	◎
bsdiff	◎	△	○
Zstandard (通常圧縮)	△	△	○
Zstandard (学習圧縮辞書)	◎	○	△

にもフラッシュメモリの使用量としても良い。また、学習辞書を利用する場合は、圧縮率も良く、RAM 的にも効率が良いが、フラッシュメモリが余分に必要という欠点がある。そこで、適切なアルゴリズムを活用するためには、事前の評価が必要となる。

3. CAN

CAN はドイツの Bosch 社によって開発された通信プロトコルで 1985 年に完成した。CAN が初めて実用車に使われたのは 1990 年で、それ以前は、ECU 間を多くのハーネスでつないでいた。ハーネスを多く使用することによって、車重の増加や配線スペースの確保が問題となっていた。CAN の実用化により、ハーネスの必要数が減り、車重の減少や配線スペースが用に確保できるようになった。

CAN は通信速度によって高速な CAN である CAN-C や低速な CAN である CAN-B のように分けられる。また、SAE(Society of Automotive Engineers)では、以下の表 2 のように分類される。現状では、下記であるが、さまざまな場合を想定する上では、これらも含めて評価できる必要がある。

4. 提案手法

本論文では、想定される更新の流れ(図 1)全体を評価することを目的とし、一部シミュレータを用いる。本論文では CAN にデータを転送することで、更新時間のシミュレーションを評価する。

4.1 パラメータ

CAN のシミュレーションを行うにあたり、以下の 3 つをパラメータとして与える。

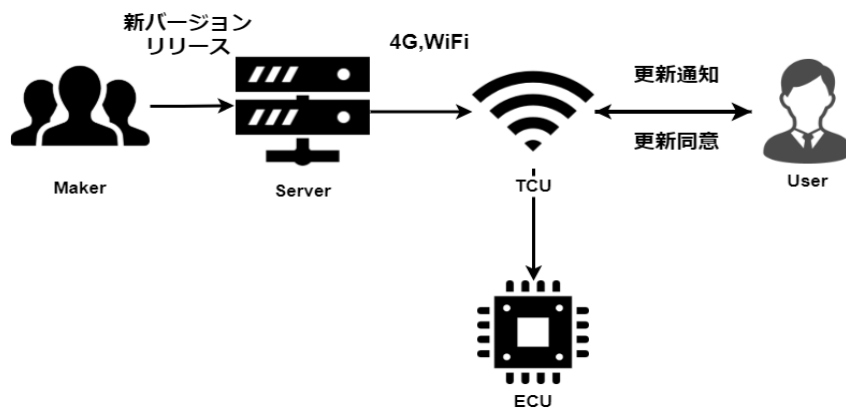
- ECU の個数
- 伝送速度
- ファイルサイズの圧縮率

4.2 評価ポイント

評価するポイントは以下の 3 点である。

表 2 CAN クラス

クラス	速度	用途
クラス A	~10kbps	ライト類、パワーウィンドウ、ドアロックなど
クラス B	10~125kbps	メータ、オートエアコン、故障などのステータス情報
クラス C	125kbps~1Mbps	エンジン、トランスミッション、ブレーキ制御などのリアルタイム制御系



- ECU の数を変えたときの更新時間の変化
- CAN の構成を変えたときの更新時間の変化
- 圧縮アルゴリズムを変えたときの更新時間の変化

ECU は 2016 年の時点で 1 台の車に平均で 21.6 個搭載されていた。ECU の数は今後増えていくと予想されるため、ECU の個数の変化による更新時間の変化を評価する必要がある。本研究では 2016 年の平均をもとに、ECU の数を 1, 20,40,60,80 と設定しシミュレートする。

CAN 構成は一般的に用いられているバス型(図 2)と文献 [2]をもとにパラメータを設定したスター型(図 3)の 2 種類で評価を行う。スター型はバス型に比べて、ハブ接続されているそれぞれの線が独立しているため障害耐性が高いが、ハブが故障した場合すべての通信が遮断されるため、ネットワークの障害耐性はハブに依存する。

CAN は伝送速度が遅いため、データサイズを小さくして送るという手法が多く提案されてきた。bsdiff や gzip な

ど圧縮アルゴリズムをもちいてデータを圧縮して送るという研究がある(文献[3])。圧縮アルゴリズムの違いによって送るファイルサイズは大きく変わるため、圧縮アルゴリズムの違いによってどの程度、伝送時間に影響してくるかを評価することが必要となる。

5. 実験・評価

本論文では、CAN シミュレータで CAN を仮想的に構成し、データを送り伝送時間を比較する。

5.1 バス型 CAN 伝送時間

CAN の構成はバス型とする。CAN のデータサイズは 28742byte とする。

文献[4]よりソフト更新のために割り当てられる帯域は少ないため、伝送効率は 10%が利用可能であると仮定した。

表 2, 表 3, 表 4 はデータの伝送速度をそれぞれ 1Mbps, 500kbps,125kbps とした。

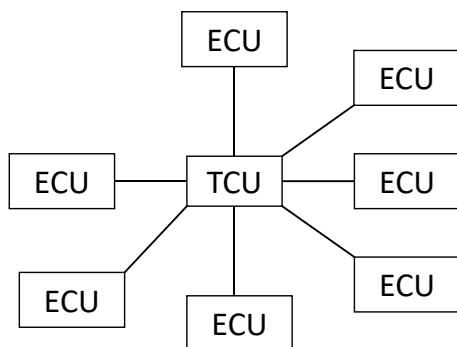


図 2 スター型結合の場合

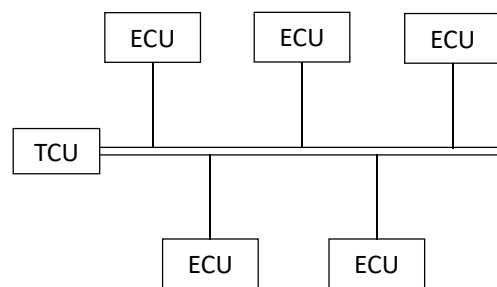


図 3 バス型結合 CAN

次に、圧縮されたファイルを送ることを想定しファイルサイズの圧縮率の変化による伝送時間の変化を比較した。データサイズは 28742byte と設定し CAN の伝送速度は 125kbps とした。圧縮率をそれぞれ 80%,60%,40%,20%とした。また、bsdiff を用いたデータ削減は圧縮率が 1%を下回るため、1%の圧縮率の場合の伝送速度もシミュレーションを行った。伝送効率文献[4]より 10%とした。結果をそれぞれ表 5,表 6,表 7,表 8,表 9 に示す。

表 2 伝送速度 125kbps

ECU 個数	1	20	40	60	80
伝送時間(秒)	18.4	368	736	1104	1472

表 3 伝送速度 500kbps

ECU 個数	1	20	40	60	80
伝送時間(秒)	4.6	92	184	276	368

表 4 伝送速度 1Mbps

ECU 個数	1	20	40	60	80
伝送時間(秒)	2.3	46	92	138	184

表 5 圧縮率 80%

ECU 個数	1	20	40	60	80
伝送時間(秒)	14.7	294	588	882	1176

表 6 圧縮率 60%

ECU 個数	1	20	40	60	80
伝送時間(秒)	11	220	440	660	880

表 7 圧縮率 40%

ECU 個数	1	20	40	60	80
伝送時間(秒)	7.4	148	296	444	592

表 8 圧縮率 20%

ECU 個数	1	20	40	60	80
伝送時間(秒)	3.7	74	148	222	296

表 9 圧縮率 1%

ECU 個数	1	20	40	60	80
伝送時間(秒)	0.2	4	8	12	16

5.2 評価

5.1 より、ECU の数を増やすことにより伝送時間は比例して増加していることがわかった。2016 年には平均で 21.6 個搭載されていた ECU は 2025 年には平均 30.4 個になると予想されている。

今後さらに ECU の数は増加していくと予想され、オリジナルのデータを送ることは更新時間の増加につながる。

圧縮アルゴリズムを用いて、圧縮されたファイルを送ったことを想定し、圧縮率を変えて伝送時間を測った結果を示した。圧縮率を 20%にした場合においては、500kbps でオリジナルデータを送る場合よりも早く送れるという結果になり、圧縮アルゴリズムの必要性を示せた。また、bsdiff を想定した圧縮率 1%の伝送時間は、CAN の最大通信速度である 1 Mbps の伝送時間より早くデータを送れることが分かり、bsdiff の有用性が示せた。

6. まとめ

本論文では、車載 ECU のソフトウェア更新時に最適なアルゴリズムを選択するための手法を提案した。CAN シミュレータをアルゴリズムや ECU の個数などのパラメータを変化させてシミュレートを行いまとめた。シミュレーションの結果から、ECU の個数による伝送時間の増加を評価した。また、圧縮率の違いによる伝送時間の変化をシミュレートし、アルゴリズムの有用性を示した。

参考文献

- [1] 国土交通省 自動車局, 平成 28 年度 リコール届出内容の分析結果について, <https://www.mlit.go.jp/jidosha/carinf/rcl/data.html>
- [2] 寺岡秀敏, 中原章晴, 黒澤憲一, “車載 ECU 向け差分更新方式”, 情報処理学会論文誌 コンシューマ・デバイス&システム(CDS), Vol7, No2, pp41-50, 2017
- [3] Lorenz, Steffen, “The FlexRay Electrical Physical Layer Evolution,” Automotive 2010. https://web.archive.org/web/20150216112537/http://www.hanser-automotive.de/fileadmin/heftarchiv/FLEX_10_ONL_NXP-Y.pdf
- [4] 倉地亮, 高田広章, 手嶋茂晴, 宮下之宏, ”10MbpsCAN プロトコルの設計と評価”, 情報処理学会論文誌 vol.50 No.11 2643-2653(Nov. 2009)
- [5] 染谷一輝, 寺島美昭, 清原良三, “圧縮辞書の再利用による車載 ECU 向けデータ圧縮方式” マルチメディア, 分散, 協調とモバイル(DICOMO2020)シンポジウム pp748-754