

# 車載 ECU 向けソフトウェア更新アルゴリズム選択方式

杉本 俊輔<sup>†</sup> 染谷 一輝<sup>‡</sup> 鈴木 孝幸<sup>†</sup> 清原 良三<sup>†</sup>

神奈川工科大学<sup>†</sup>

## 1 はじめに

近年、自動運転システムや電気自動車の普及により 1 台の自動車に搭載される ECU(Electronic Control Unit)の数が増加している。それに伴いソフトウェアの不具合に起因するリコールが増えている。そのため、広域無線通信を利用する OTA(Over The Air)によるソフトウェア更新技術が開発され、実用化が目指されている。

OTA 更新では広域無線通信が可能な場所であれば、いつでもどこでも更新が可能になるため迅速な更新が期待される。しかし、更新中は自動車を使用することができないためユーザビリティの観点から更新時間の短縮が求められる。

差分更新方式などデータサイズを小さくすることによって更新時間を短縮する研究が数多く行われてきた。しかし、車載 ECU の更新を行う際には様々な制約条件があり、どの手法を用いることが最適なのかを判断することが難しい。

本論文では、車載 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。

## 2 関連研究

ECU の更新は車載ネットワークを用いて行われる。車載ネットワークは主に CAN(Control Area Network)が使用されている。CAN の伝送速度は広域の 4G や Wi-Fi などと比べて遅い。

ECU のソフトウェア更新はデータの転送時間とフラッシュメモリの書き換え時間に大きく分けられる。CAN の最大ビットレートは 1Mbps で、最大ペイロードは 8byte であり低速となっている。そのため、OTA 更新を行う場合、CAN 上の更新データの転送時間がボトルネックとなり、ソフトウェア更新時間の大部分を占める。

文献[1]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として、フラッシュメモリ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。この提案方式ではデータ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。

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[2]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。

これらの手法は決まった制約条件の元での最適な手法を提案しているが、現実に応用しようとした場合は、ソフトウェアの修正の程度などにも依存するため必ずしも最適な方式にはならない。本論文は現実の適用時の手法選択のためのプラットフォーム提案である。

## 3 提案手法

OTA に利用される圧縮アルゴリズムの制約条件の長所短所などを表 1 に示す。本論文では、想定される更新の流れ全体(図 1)を評価することを目的とする。CAN をシミュレータで再現しデータを流して実際の更新時間、必要 RAM、フラッシュメモリを評価する。とくに本論文では車載ネットワーク部で評価を行った。

表 1 データ削減と長所短所

アルゴリズム	長所	短所
差分更新	低速車載ネットワーク向け	消去ブロック以上の RAM 要
汎用圧縮	RAM の制約なし	高速車載ネットワーク向け
共有辞書圧縮	低速車載ネットワーク向け RAM の制約なし	フラッシュメモリの余裕容量に圧縮率が依存

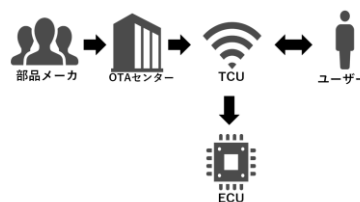


図 1 想定される更新の流れ

Confirmation Method of Human-Control mode during Level 3 Autonomous Vehicles

SHUNSUKE SUGIMOTO<sup>†</sup>, KAZUKI SOMEYA<sup>‡</sup>, TAKAUKI SUZUKI<sup>†</sup>, RYOZO KIYOHARA<sup>†</sup>

<sup>†</sup> Kanagawa Institute of Technology

<sup>‡</sup> Graduate School of Kanagawa Institute of Technology

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た。

- ECU の個数
- 伝送速度
- データの圧縮率

ECU の個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 の個数を 1,20,40,60,80 と設定しシミュレーションを行った。

CAN 構成は一般的に用いられるバス型と文献[3]をもとに設定したスター型の2種類で評価する。スター型はバス型と比べ、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接続が遮断され、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関連研究のような手法が多く提案されてきた。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。

以上の点から評価ポイントを以下の3つとした。

- 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

## 4 実験・評価

### 4.1 実験

本論文では、ネットワークシミュレータで CAN を仮想的に構成し、データを送り伝送時間を比較した。

CAN の構成はバス型とし、データサイズは 28742byte とする。また、更新の目標時間は電気自動車の急速充電にかかる時間をもとに設定した。電気自動車の急速充電は 80% の充電が完了するまでに約 30 分かかる。すべてのユーザが 80% まで充電せずに途中で自動車を利用することを想定し、目標時間は 15 分とした。

### 4.2 評価

ソフトウェア更新は、データの伝送時間とフラッシュメモリの書き換え時間に分けられる。伝送時間と書き換え時間のどちらにボトルネックがあるのかを判断するために実験を行った。結果を図 2 に示す。図 2 より伝送時間が大部分を占めていることが分かる。

次に、ボトルネックが伝送時間であることが確認できたため伝送時間を変えて実験を行った。結果を図 3 に示す。CAN は使用時には 500kbps に伝送速度が落ちる。ECU の数が 20 個の時点で目

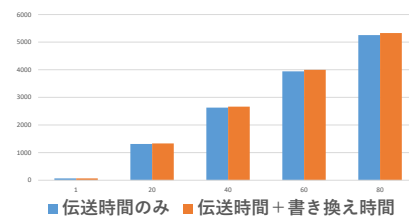


図 2 書き換え時間による変化

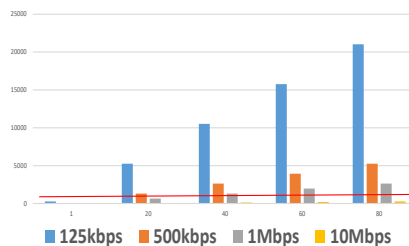


図 3 異なる伝送時間による違い

標時間を超え、今後 ECU の数の増加があれば、現在の CAN では OTA は難しい。

そこで、圧縮アルゴリズムを用いてデータサイズを小さくして実験を行った結果が図 4 である。bsdiff と共有辞書圧縮を用いた zstandard は目標時間を達成したため有効な手段だと確認した。しかし、bsdiff は今回の実験で用いた ECU の RAM 使用量を超えており、実際には使用できないため共有辞書圧縮を用いた zstandard が最適である。

## 5 おわりに

本論文では、CAN を仮想的に構成しパラメータを与えて更新時間の変化を確認しアルゴリズムの有用性を示した。

## 参考文献

- [1] 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), Vol.7, No2, pp. 41-50, 2017
- [2] 染谷一輝, 鈴木直希, 寺島美昭ほか, “車載 ECU 向け省メモリソフトウェア更新方式,” 情報処理学会論文誌, Vol. 62, No.1, 2021
- [3] 倉地亮, 高田広章, 手嶋茂晴ほか, “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,pp. 2643-2653(2009)

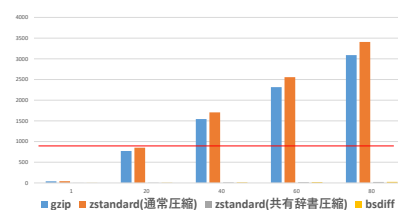


図 4 圧縮アルゴリズムを用いた場合の更新時間の違い