

自動車用組み込みソフトウェアのモデル化と安全化設計

2001MT057 近藤 広樹 2001MT077 野澤 昌史

指導教員 青山 幹雄

1. はじめに

本研究では、組み込みソフトウェアにおける自動車の安全性を問題点とし、安全性の高いソフトウェア開発を課題として取り組んだ。現在安全性の向上を目的とした交通システムに白線認識技術がある。そこで、白線認識技術を取り入れた自律走行を可能とするマイクロマウスに着目した。悪状況を想定したマイクロマウスの実験から得た結果を基に、安全性の高い走行パターンのモデル化を行った。

2. 組み込み技術を用いた自動車の安全性

2.1. 安全性における取り組み

近年 ITS は本格的な事業拡大の局面に移行し、安全運転の支援分野において研究成果が実用化されつつある。中でも道路環境認識に対する期待が高まっており、本研究では白線認識を用いた運転支援システムについて、様々な環境におけるマイクロマウスの走行実験を行い、より安全性の高いソフトウェアを実現する方法を研究する。

2.2. マイクロマウスにおける実装

2.2.1. マイクロマウスの特徴

- (1) マイクロマウスは完全自律型ロボットの一つであり、決められた白線をトレースすることを目標に走行する。
- (2) マウスに対し遠隔操作などを用いずに取り付けられているセンサからの情報を元に自律走行できる。

2.2.2. ライントレースの方法

- (1) 電源を入るとスタートマーカを探して前進する。
- (2) スタートマーカを見つけると停止する。モータモシャットダウン状態にし、センサも停止状態にする。
- (3) スタートスイッチを押すとセンサを動作させ、白線をトレースしながら走る。
- (4) ストップマーカを見つけると減速して停止する。

2.2.3. タスクの処理方法

図 1 において、STA, SW, STP がマウスのスタートからゴールまでのシナリオを担当し、必要に応じてその他の4つのタスクを起動・終了する。

仕様書に基づきマイクロマウスのタスクスイッチングについてシーケンス図を用いて図 1 に示す。

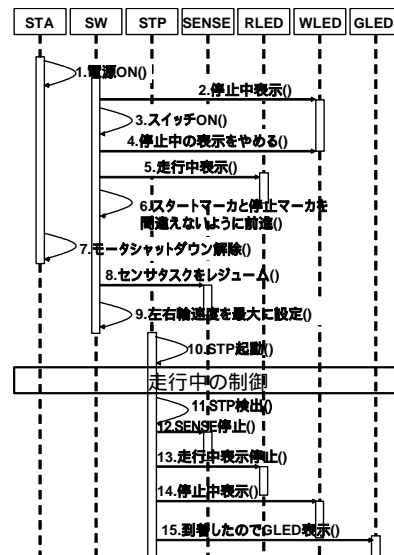


図 1 タスクスイッチング

3. マイクロマウスの白線認識と動作解析

3.1. マイクロマウスの走行制御

- (1) センサが直線と判断すると両輪ともに最高速度で走行。
- (2) 直線を緩やかに蛇行していると判断すると、外側の車輪を少しだけ加速して蛇行を停止。
- (3) コーナなどの急なカーブであると判断すると、内側の車輪を減速させて曲がる。
- (4) さらに急なカーブなどで白線上から大きくそれると判断すると、内側の車輪をさらに減速。

3.2. 走行制御における問題点

タスクスイッチングと走行制御の動作解析から、現状マイクロマウスの改善点として次の3点を抽出した。

- (1) 障害物を回避できない
- (2) 走行途中の速度変更ができない
- (3) 白線離脱後に迷走する

以上のすべての問題点の解決が望まれるが、本研究では(3)の問題点に着目した。

この問題点を改善する方法としてオブジェクト指向を用いたモデリングを行った。

3.3. 白線認識のモデル化

マウスのトレースの様子と各部を図2に示す。

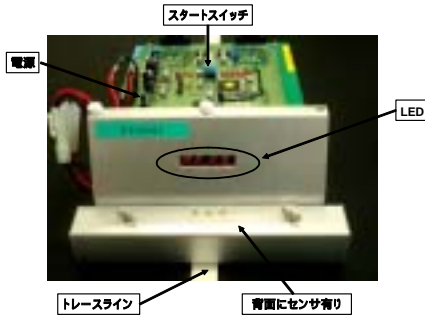


図2 マイクロマウス

3.3.1. 白線センサ

動作解析によってマウスの最前部の下面に取り付けられている4つのセンサによって白線を正確にトレースできることが分かった。そこで、4つのセンサを1つ1つ区別し、番号をつけ走行パターン別にどのセンサが反応しているかサンプルプログラムとマウスの動きから調べた。

センサの配置とその番号付けを図3に示す。



図3 センサの詳細写真

3.3.2. 白線センサのモデル化

センサの反応を走行パターン別に3つに分けることができた。センサの反応を走行パターン別に図4に示す。

- 1) 直線または、直線を少しずれた場合 2, 3
- 2) 直線をずれたまたは、ゆるいカーブを曲がる場合 1, 2 または 3, 4
- 3) コーナなどでカーブを曲がる場合 1 または 4

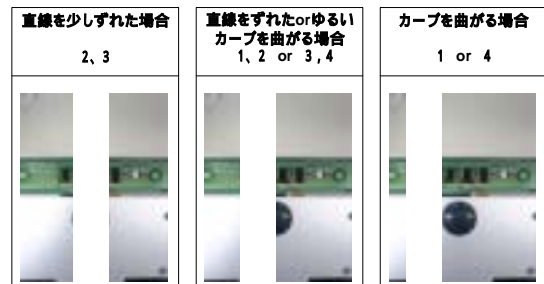


図4 走行パターン別におけるセンサの反応

3.3.3. 状態遷移図による白線認識のモデル化

図3, 4を踏まえてセンサタスクのモデル化を行った。センサタスクを状態遷移図で表し、CPUとモータとの関連を示す。この図を作成することでセンサが状況によって変化する白線に対してどのように反応し、CPUからモータにどのように信号が送られているのかが理解できた。

センサタスクの状態遷移図を図5に示す。

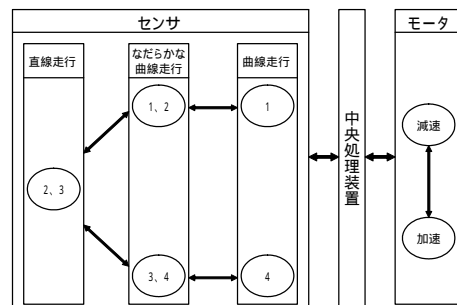


図5 センサタスクの状態遷移図

3.4. 白線認識の走行実験

マウスのサンプルプログラムを利用した動作解析からセンサタスクのモデル化を行った。続いて、走行パターンのモデル化を行うために走行実験を行った。

3.4.1. 実験の目的

悪状況下で走行する場合の安定性を実証することに着目した。白線間隔と助走距離を変化させて走行させ、センサの白線認識間隔とモータとの関連性を明らかにする。この結果に基づき、安全な走行を実現するシステムの設計・開発をする。

3.4.2. 実験方法

(1) 実験環境

全長7~9mの直線の白線。助走距離は1~3mである。

(2) 実験のねらい

現状マウスの走行性能を白線の間隔と助走距離

を変化させることで明確にする。

(3) 走行測定方法

1) 白線欠落間隔を 5cm ずつ広げる。

現状のマウスでは、どれくらいの白線の欠落まで走行できるのかを明確にする。

2) 助走距離を 1m ずつ広げる。

助走距離を広げることで、マウスの走行スピードやセンサの白線認識間隔の限界を測定する。

3) 助走距離ごとに白線欠落状態での走行を 20 回測定する。

(4) 結果の処理

白線欠落状態での走行を 20 回測定し、走行可能区間の最大値と平均値を求めた。

3.5. 白線認識の実験結果

実験結果を助走距離ごとに折れ線グラフを用いて図 6 に示す。助走距離の変化による安全性を満たす条件の値に変化はみられなかった。マウスのスピードと白線認識間隔は変化しないことが明らかになった。

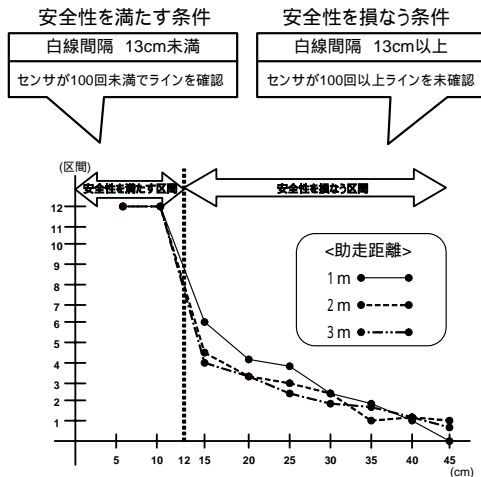


図 6 走行実験グラフ

3.5.1. 実験結果の分析

- (1) 白線間隔が 13cm 未満であれば、走行に問題が生じることがない。
- (2) 白線間隔が 13cm 以上の場合、走行可能区間にばらつきが目立った。
- (3) マウスの速度は 0.65m/s である。
- (4) センサの白線を感じるタイミングは 1/500 秒に 1 回。(0.13cm 進む毎に 1 回感知する)
- (5) 助走距離を 1m~3m に広げて実験した結果、白線認識に有意な差は見られなかった。

3.5.2. 安全性を損なう条件

- (1) センサの起動間隔 = 2msec 毎(1/500 秒) = 500 回/s

(2) 1 回のスキャンで進む距離 = 0.65m/500 回 = 0.13cm

(3) 走行可能区間のばらつき = 13cm

(4) 13cm でのセンサの白線認識回数 = 13/0.13 = 100 回

上記の(1)(2)(3)(4)の実験結果から、100 回連続してセンサが白線を認識できない場合に軌道を外れる。これが安全性を損なう条件である。

4. オブジェクト指向による走行のモデル化

4.1. センサタスクのモデル化と走行パターン

センサタスクのモデル化と白線認識の実験結果を踏まえた走行パターンを図 7 の状態チャート図に示す。

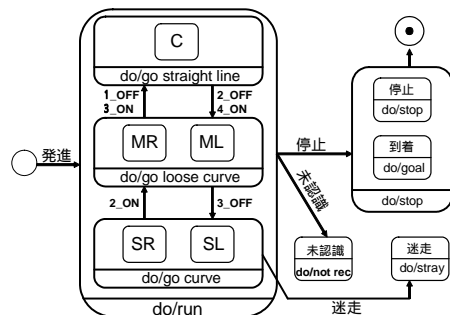


図 7 走行パターンの状態チャート図

モデル化したことで明確となった安全でない状態として以下の 2 点が抽出できる。

- (1) 一定時間白線を離脱した状態を未認識状態とする。
- (2) 未認識状態が一定時間継続した場合、迷走とする。

4.2. 改善方法の提案

走行実験よりマウスは 1 秒間で 65cm 進むことが明らかになった。そこで、白線離脱後の迷走状態を改善するため、本研究では 2 秒間(=130cm)を安全な停止間隔と判断し、モデル化を行う上で白線離脱後 2 秒間経過したら停止させるという処理を提案する。図 7 の走行パターンの状態チャート図に改善を実施すると図 8 の状態チャート図となる。

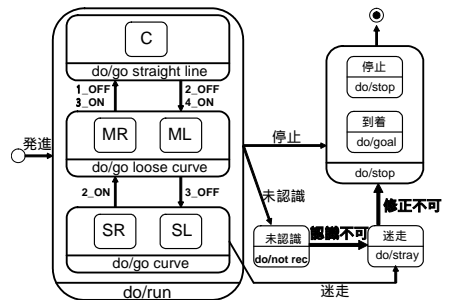


図 8 安全化した走行パターンの状態チャート図

5. 安全性を重視したシステム開発

5.1. マイクロマウスの走行制御モデル

第4章のセンサタスクのモデル化により、未認識状態と迷走状態ではマイクログマウスの走行が不安定になり、制御全体としての安全性を損なうことが分かった。そこで、改善点をさらに明確にするためにマイクログマウスの走行制御図についてシーケンス図を用いて表した。

現状マイクログマウスの走行制御モデルを図9に示す。

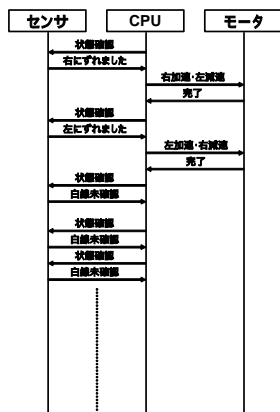


図9 現状マウスの走行制御モデル

5.2. 安全性を考慮した走行制御モデル

図9からセンサが白線を見失った場合、CPUとセンサ間の状態遷移が一定した処理の繰り返しとなることが分かった。そこで、この点を改善した走行制御図を図10に示す。

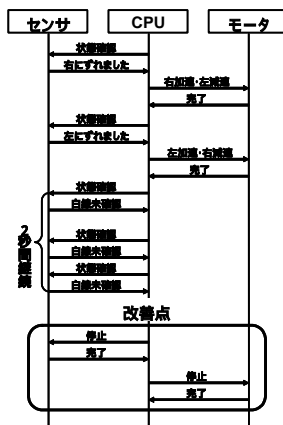


図10 安全性を考慮した走行制御モデル

5.3. 改善点の評価

マイクログマウスの安全性向上を目的としたモデル化を行ったことにより、シンプルで有用性の高い改善方法の提案をすることができた。具体的な変更後の評価としては以下

の点が挙げられる。

- (1)白線を離脱した場合に自動停止する点が、安全性の向上につながる。
- (2)経過時間で自動停止させることが最良な方法であるかを議論する必要がある。
- (3)経過時間で自動停止させる場合に、停止のタイミングを考慮する必要がある。

6. 考察と今後の課題

本研究では自動車の安全性に着目し、マイクログマウスの走行制御やセンサスイッチングのモデル化を行うことで、安全性の高いシステム開発を目指した。このシステムを開発する上で必要となることは、安全性を満たす制御方法を設計することであった。この設計方法として、マイクログマウスのセンサとCPUの状態遷移に着目し、一定時間同じ処理が繰り返された場合の自動停止を提案した。安全性において重要なのは、センサの起動タイミングとモータ制御である。本研究では、安全性追求のためにモデル化を行った。しかし、モデル化によって可能となる評価方法には限界があり、実装することで自動停止方法やタイミングについて検討する必要がある。

7. まとめ

本研究では、組み込みソフトウェア技術を用いた自動車の安全性に着目した。白線認識技術を用いたマイクログマウスを使用し、マイクログマウスの走行制御やセンサのモデル化を行った。さらに、マイクログマウスの走行実験から安全性を満たす条件と安全性を損なう条件を明確にした。次に現状のマイクログマウスの問題点を提起し、改善点を提案した。そして安全性を向上したマイクログマウスのモデル化を行った。モデル化は実装しなくても問題点を発見、評価することができる。モデル化は設計・開発において必須技術である。プログラム設計を行うためには、直ちにプログラムを変更するのではなく、モデル化を行うことで変更点を明確にすることが重要である。

参考文献

- [1] (社)日本システムハウス協会 エンベデッド技術者育成委員会(編・著), 組み込みシステム開発のためのエンベデッド技術, 電波新聞社, 2003.
- [2] 菅沼賢治, 自動車組み込みソフトウェアの現状と動向, 情報処理学会 ソフトウェア工学研究会資料, No. 134-13, pp. 47-54, 2001.10.
- [3] 富士通, マイクログマウスファームウェア仕様書第2.0版, 2000.5.