オブジェクト指向を用いた自動車用組込みソフトウェアの安全化設計

2002MT016 市村 尚規 2002MT070 小山内 秀輔

指導教員 青山 幹雄

1. はじめに

本研究は，オブジェクト指向技術を用いて自動車組込みソフトウェアの安全性の向上を目的とする．マイクロマウスの白線認識をモデルとし，リアルタイム性を考慮して安全性を向上する機能拡張方法を提案する．

2. 自動車における組込みソフトウェア

2.1. 自動車組込みソフトウェアの現状

現代の自動車では，様々な機能が電子制御されている．安全性，環境負荷の低減要求などから，ECU（Electronic Control Unit）の役割や機能が増大している．

2.2. 安全性における取り組み

安全性向上に対するアプローチとして「危険を回避する機能を追加する」「障害が発生しないようソフトウェアの品質を向上する」の二つが挙げられる．本研究では前者をマイクロマウスに適用し安全性の向上を図る．

2.3. 問題点と解決策の提案

安全性を向上する機能開発においても組込みソフトウェアの複雑化，巨大化は問題視されている．本研究では設計開発にオブジェクト指向技術を導入する．

3. 組込みソフトウェア開発へのオブジェクト指向技術導入

3.1. オブジェクト指向技術導入の利点と問題点

オブジェクト指向技術を導入することで分析，設計などの上流工程において正しさを確認，検証する開発を実現する．また信頼性と再利用性の高い設計が可能になるため，生産性の向上も可能である．OMG（Object Management Group）により標準化がされている UML(Unified Modeling Language)を用いたモデリングによりシステムの開発プロセスを整合的に進めることができる．

一方，問題点として組込みソフトウェアでは必要不可欠な時間の取り扱い方についての概念が不十分である．このためリアルタイム性を考慮した設計を行う必要がある．

4. マイクロマウスのオブジェクト指向設計

4.1. マイクロマウスの特徴

マイクロマウスは完全自律型のロボットの一種であり，連続した一本の白線をトレースしながら走行する．マイクロマウスの外観を図 1 に示す．遠隔操作を必要とせず，4 つのセンサからの白線情報をもとに自律走行をする[1].

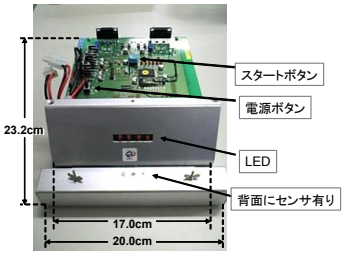


図 1 マイクロマウス

白線認識に必要なセンサの写真を図 2 に示す．走行に用いられるコースの板を図 3 に示す．センサ番号は進行方向かつ下面から見て左から 4.3.2.1 とする．

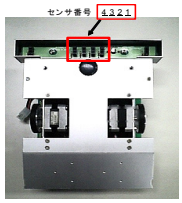


図 2 センサ

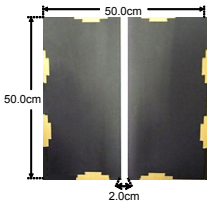


図 3 走行に用いられる板

4.2. 構造分析

マイクロマウスの動作を以下に示す．

(1) 電源を入れると白線を辿りスタートラインまで徐行．
(2) スタートマーカを検出すると一時停止．
(3) スタートボタンを押すと走行を再開．
(4) センサ値により両車輪を制御し，白線を辿り走行．
(5) ストップマーカを検出するとしばらく直進し，停止．

マイクロマウスのクラス図を図 4 に示す．

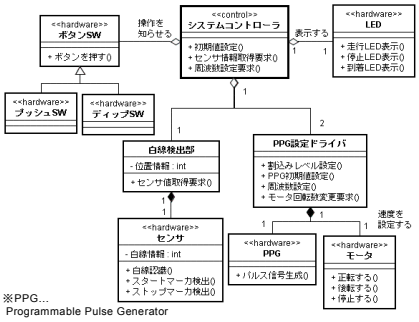


図 4 マイクロマウスのクラス図

動作分析やソースコード解析より 9 つのクラスを抽出した．システムコントローラはメッセージの起動制御を担うコントロールオブジェクトである[2]．白線検出部はセンサ情報を保持し，センサが現在のセンサ値を取得するとPPG 設定ドライバへ周波数の設定を要求し，PPG でパルス信号を生成することでモータ回転数が変更される．

4.3. 車輪制御の仕組み

マイクロマウスは白線を光センサで認識し，センサ情報より車輪の制御を行う．図5にリアルタイム性を考慮し，センサタスク内の処理の流れを表したシーケンス図を示す[3]．

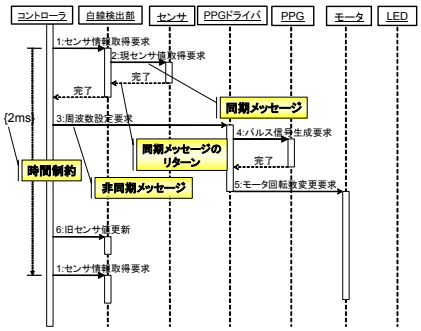


図 5 センサタスクのシーケンス図

センサタスクの仕組みを図 6 のアクティビティ図に示す．車輪制御要求内の「減速 1」は現在の速度×0.8，「減速 2」は×0.5 の減速となる．

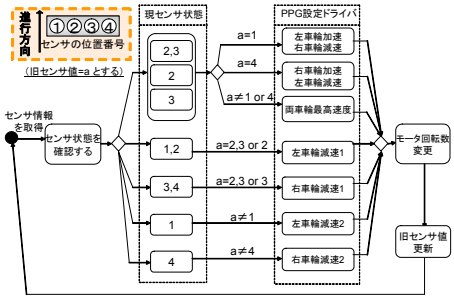


図 6 センサタスクのアクティビティ図

4.4. 走行における問題点

マイクロマウスの走行における問題点は「白線離脱後に迷走する」ことである．白線未認識時は白線を再認識するまでモータ回転数を変更できないため迷走状態に陥る．

5. マイクロマウスにおける安全化設計

5.1. 設計方法の提案

ソフトウェアの安全性を保証するように機能拡張する方法を提案する．安全性の低い状態を回避することで安全性を保証するアクティブセーフティ，安全性が低い状態の被害を最小限に抑えることで安全性を保証するパッシブセーフティの 2 つの考えから問題の解決を図る．図 7 に走行における状態遷移図を示す．

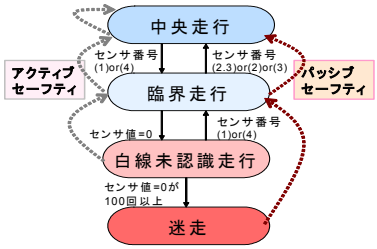


図 7 安全化設計による状態遷移

中央走行はセンサ位置番号(2)(3)(2.3)，臨界走行は(1)(4)の場合である．昨年の卒業研究より「迷走」状態の定義は「白線未認識状態が 0.2 秒以上続いた」場合であるのでセンサ値=0 が 100 回を超えた場合は迷走状態に陥る[4]．

5.2. アクティブセーフティ

迷走を未然に防ぐために，以下のようなセンサタスク処理の追加と変更を行う．

5.2.1. センサタスク処理の追加

第 1 に図 6 のセンサタスクの 7 つの処理に加え，白線を離脱する可能性が高い場合の処理を 2 つ追加する．状態を細分化することにより車輪制御の精度を上げる．

1) 現センサが 4 のみ ON の場合

旧センサが 4 のみ ON の場合，左車輪加速，右車輪減速

2) 現センサが 1 のみ ON の場合

旧センサが 1 のみ ON の場合，左車輪減速，右車輪加速

5.2.2. センサタスク処理の変更

第 2 にカーブ進入時に減速する機能を追加する．白線離脱の確率が低下し，白線認識率が向上する．処理内容はセンサ位置(1)or(4)が白線を認識した場合に減速する．図 8 にセンサタスクのアクティビティ図を示す．

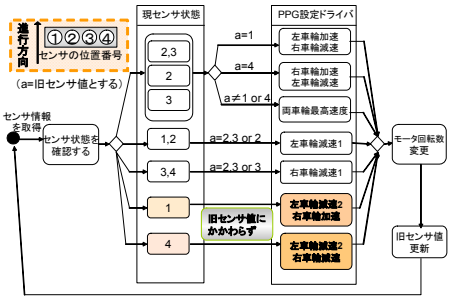


図 8 カーブ進入時に減速させるセンサタスク

5.3. パッシブセーフティ

迷走時の動作変更を行い，それに伴って方向転換と自動停止機能を追加する．

5.3.1. 迷走時の動作変更

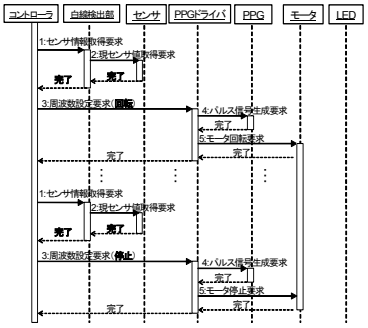
迷走状態に陥る前に一時停止させる．モータ停止時間を 0.2 秒とする．後退時は通常速度の 0.5 倍で走行する．0.5 倍は，急カーブの場合には図 6 に示す「減速 2」(通常速度×0.5 倍)の処理をすることから，0.5 倍で走行することで白線認識率が向上する．以下に動作の流れを示す．

(1) 0.2 秒間白線未認識状態が続くとモータ停止
(2) モータを逆転させセンサが白線を再認識するまで後退
(3) センサが白線を再認識した場合にモータ停止
(4) モータを正転させ，通常動作(0.5 倍の速度)に戻す

5.3.2. 方向転換機能の追加

5.3.1 の拡張機能は一度白線を離脱した場所に同じ車体の向きでリスタートする．そのため車体の向きを進行方向に回転する機能を追加する．以下に動作の流れを示す．

1)後退中にセンサが白線を認識し，一時停止
2)センサ値を呼び出し車体の向きを変える
3)進行方向に車体を向け，一時停止
4)通常動作に戻す



2)でセンサ番号が(1)の場合は左車輪停止・右車輪回転，(4)の場合は左車輪回転・右車輪停止する．3)でセンサ番号(2,3)(2)(3)の場合に回転を停止する．方向転換時の処理の流れを図 9 に示す．図 9 方向転換のシーケンス図

5.3.3. 自動停止機能の追加

後退中に白線を認識できない場合に自動停止する機能を追加する．後退走行は白線を認識するまでに最低 0.4 秒(13.0cm)が必要であるので，自動停止時間は 0.4 秒以上となる．コースの板上であれば安全性が保証できると仮定し，白線中心から板の端までの 25.0cm を安全走行距離とした．25.0cm は 0.4 秒での走行距離 13.0cm を超えているため 13.0 から 25.0cm が安全性を保証できる走行区間である．

自動停止時の処理の流れを図 10 に示す．

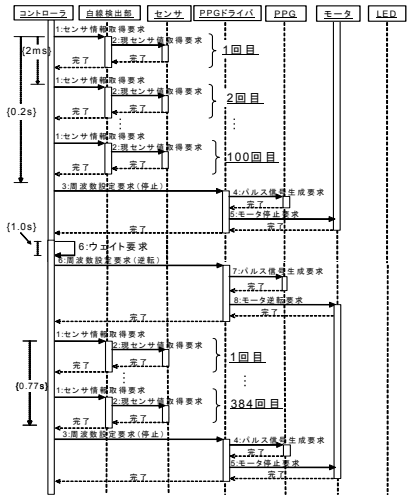


図 10 には実行時間を明確にするため時間制約を付加した．センサ情報取得要求の最初 100 回と後退時 384 回のレスポンスメッセージは全て白線未認識メッセージである．

図 10 自動停止のシーケンス図

5.4. 拡張機能の動作分析

拡張機能により走行状態は中央，臨界，後退，方向転換，白線未認識，迷走の 6 つに分けられる．後退走行は白線未認識回数が 1 回以上 384 回以下，白線未認識は1回以上 100 回以下，迷走は 100 回を超えた場合である．方向転換は白線認識回数が 1 回より多く 200 回以下である．よって安全性が高い順に「中央走行>臨界走行>方向転換>後退走行>白線未認識>迷走」となる．表 1 に安全化基準表を示す．これにはマイクロマウスから白線までの距離，白線認識率，センサ状態を付加させた．拡張機能を含めた動作の状態遷移を図 11 に示す．

表 1 安全化基準表

	状態名	距離(cm)	認識率(%)	センサ状態	センサ処理状態
1	迷走	1.7-14.0	0-87.9	センサ値=0	未認識回数>100
2	白線未認識	1.7-14.0	0-87.9	センサ値=0	1≦未認識回数≦100
3	後退走行	1.7-14.0	0-87.9	センサ値=0	1≦未認識回数≦384
4	方向転換	0-1.7	87.9-100	センサ値≠0	1<認識回数≦200
5	臨界走行	0.4-1.7	87.9-97.1	センサ番号=(1)(4)	
6	中央走行	0-0.4	97.1-100	センサ番号=(2,3)(2)(3)	

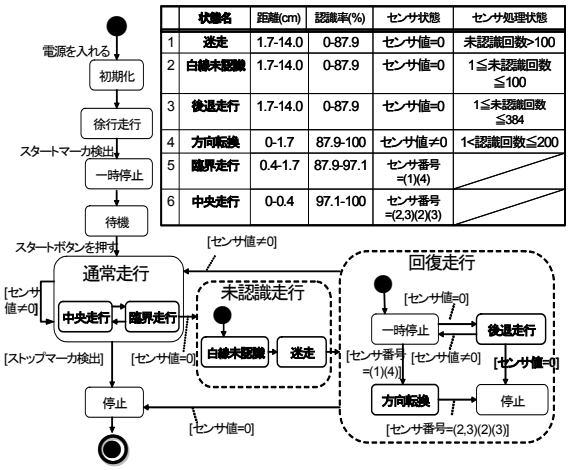


図 11 制御全体の処理の流れ

6. 拡張機能の検証

6.1. 検証方法

図 7 の状態遷移図の各状態に遷移確率を導入することで安全性の保証を検証する．拡張機能と安全性の評価を行うために以下の走行実験を行った．

(1) コースを直線 60%，曲線 40%の割合で作成
(2) マイクロマウスを 6 台用いて走行
(3) 1 台 20 回(6 台で計 120 回)走行
(4) 完走した回数，コースを外れて迷走した回数を記録

6.2. 実験結果

表 2 実験結果

	完走(回)	迷走(回)	迷走確率(%)
a	19	1	5.0
b	20	0	0
c	19	1	5.0
d	18	2	10.0
e	18	2	10.0
f	20	0	0
合計	114	6	5.0

走行結果を表 2 に示す．表中の a～f はマイクロマウスを指し，それぞれの完走と迷走回数，迷走確率を表している．実験より迷走する確率は 5.0%である．

6.3. 状態における確率

実験では全てカーブで白線を見失い迷走状態に陥った．また「白線未認識状態に遷移した場合は全て迷走状態に陥る」と仮定をする．中央走行はコースの直線の割合である60%であり，迷走する確率は 5.0%である．臨界走行は35(100-(60+5))%となる．後退走行は迷走と同じ 5.0%である．後退走行は白線を再認識することを前提として追加した機能であり，白線未認識率は低い．よってマイクロマウスの迷走状態に陥る確率と同じ 5.0%(全体で 0.25%)となり，方向転換は 4.75%(5-0.25)となる．表 3 に確率を付加した安全化基準表を示す．図 12 に各状態遷移に確率を付加した状態遷移図を示す．

表 3 安全化基準表

走行状態名	状態名	確率(%)
未認識	迷走	5.0
	白線未認識	5.0
回復	後退走行	5.0
	方向転換	4.75
通常	臨界走行	35.0
	中央走行	60.0
停止	自動停止	0.25

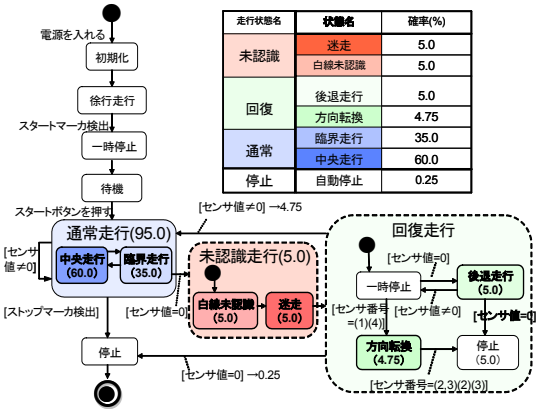


図 12 状態遷移図を用いた状態ごとの確率

7. 評価

図 12 より迷走状態に陥った場合，拡張機能により通常走行に戻る確率は 99.75%(95.0+4.75)である．機能拡張前と比べ全体として安全性が 4.75%向上した．迷走し続ける確率は 5.0%から 0%となった．さらに通常走行に戻る確率は 4.75%になった．よって迷走状態の 5.0%と比べると前者は 100.0%，後者は 95.0%の安全性が向上した．図 13 に拡張前後の状態における確率の遷移を示す．

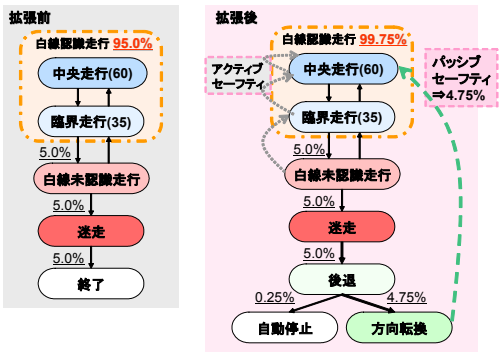


図 13 拡張前後の状態における確率の遷移

図 13 からも分かるように，拡張後は迷走後に終了することはなくなり，通常走行に戻る確率が向上した．よって本研究で提案した拡張機能の安全性の向上が確認できた．

この実験はアクティブセーフティの機能は実験結果に含まれていない．アクティブセーフティの機能を追加したマイクロマウスを使用して今回と同様の実験を行なった場合，迷走確率は今回の実験結果である 5.0%未満になり，白線認識走行の確率が向上すると考えられる．

8. 考察と今後の課題

本研究では，安全性向上のための機能拡張を UML を用いて動作解析をし，評価と検証を行った．しかし UML の図的表現によって可能となる評価方法には限界がある．白線を離脱しない動作や停止時間，実行のタイミングなど UML 上では表現できない問題が存在する．そのような問題を解決するには実装をしてマイクロマウスを実際に動作させる必要がある．同様にオブジェクト指向技術を導入した場合の利点も現状と比較して検証する必要がある．

9. まとめ

本研究では自動車組込みソフトウェアの設計・開発段階においてオブジェクト指向技術を導入し，安全性を向上する設計方法を提案した．モデルとしてマイクロマウスを使用し，パッシブセーフティ，アクティブセーフティの両面から安全性の向上を図った．安全性向上機能の評価方法として走行状態の状態遷移図に確率を導入して検証を行った．しかし UML を用いた検証方法では限界があるので，実装することで拡張機能における動作や実行のタイミングを検討する必要がある．

参考文献

[1] 富士通，マイクロマウスファームウェア仕様書 第 2.0 版，2000.

[2] 青山幹雄ほか，オブジェクト指向に強くなる，技術評論社，2003.

[3] B. Douglas，リアルタイム UML 第 2 版，翔泳社，2001.

[4] 近藤 広樹ほか，南山大学数理情報学部情報通信学科 2004 年度卒業論文「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」，2005.