

市街地での自動運転に向けた車両運動制御アルゴリズムと組み込み ECU での実装

広津鉄平^{†1} 横山篤^{†1}

概要: 市街地での自動運転では、他車や歩行者といった複数の動的障害物との衝突を回避し、かつ乗り心地の良い車両運動制御が求められる。本論文では、市街地の自動運転に適した車両運動制御方式、および本方式を組み込み ECU で高速実行する方法について述べる。車両運動制御には、動的障害物の将来位置や車両挙動予測に基づいた非線形モデル予測制御(NLMPC)を用いる。NLMPC では、車両運動制御の操作量をパラメータとし、衝突確率や乗り心地等の制御性能を出力する評価関数を設定し、評価関数に対する最適値問題を解く必要がある。最適値問題のソルバーには、評価関数の特徴や解の初期値に依存せず、安定して大域的な解を求めることが出来る Artificial Bee Colony(ABC)アルゴリズムを用いた。これにより、市街地の様な複数の動的障害物に囲まれた複雑な環境下において、衝突を回避し、かつ乗り心地のよい車両運動制御を安定して実行することが可能となった。さらに、本制御を組み込み ECU でリアルタイム実行するためにアルゴリズムの高速化を行った。本制御で用いる ABC アルゴリズムでは、評価関数の演算が処理の大半を占めている。そこで、評価関数演算を FPGA でハード化し、並列実行することで処理時間の短縮を行った。これにより、ABC アルゴリズムを活用した NLMPC を市街地の自動運転に適用した場合において、車両運動制御を 100ms の周期で実行可能とした。

キーワード: 自動運転, モデル予測制御, 並列処理, FPGA

Efficient Implementation of Non-Linear Model Predictive Control on an Embedded ECU for Automated Driving in Urban Environments

TEPPEI HIROTSU^{†1} ATSUSHI YOKOYAMA^{†1}

Abstract: In order to realize automated driving in urban environments, we need vehicle motion control which can avoid the collision against moving obstacles and have a smooth ride. In this paper, we show the vehicle motion control for automated driving in urban environments and the efficient implementation on an embedded ECU. For the vehicle motion control, we use Non-linear Model Predictive Control (NLMPC) based on the prediction of the positions of moving obstacles and the motion of ego-vehicle. In this NLMPC, the evaluation function, which has arguments of control outputs and measures the control performance such as collision probability and smoothness of the vehicle motion, is designed and the optimization problem for the evaluation function is solved. The evaluation function is optimized with Artificial Bee Colony (ABC) algorithm to gain the robustness against the non-linearity of the evaluation function and the initial value of the arguments. By using ABC algorithm, we realized stable vehicle motion control which can avoid collision under the complex conditions where multiple obstacles move at a time. In addition, we achieved the performance gain of the ABC algorithm to implement the algorithm into the embedded ECU whose performance is limited. We elaborated the ABC algorithm and found that the calculation of the evaluation function spends the large portion of the execution time of whole ABC algorithm. Therefore, we developed the dedicated hardware block with FPGA for the calculation of the evaluation function. Its calculation is parallelized by using multiple hardware blocks. Finally, we realized 100 millisecond control period on the embedded ECU for the NLMPC based vehicle motion control.

Keywords: Automated driving, Model predictive control, Parallel computing

1. はじめに

交通事故件数ならびに交通事故による死者数を削減することは、自動車社会において最重要な課題の一つである。このような背景から、自動車では、衝突時の被害を最小限にするためのシートベルト・エアバックのような衝突安全技術 (Passive Safety) や制動時のスリップを抑制する ABS (Antilock Brake System) や旋回時の車両挙動を安定化する ESC (Electronic Stability Control) のような予防安全技術 (Active Safety) が導入されてきた 1)。これらの衝突安全防止技術と予防安全技術は、死者数の削減や予事故件数の

削減に貢献しており、日本国内の場合では、1991 年以降、死者数は減少し、2005 年以降は事故件数も削減方向に転換している 2)。

更に、最近では、ドライバーの操作を代替する自動運転システムの開発が活発化し、日米欧の主要カーメーカーや欧州のメガサプライヤなどが、高速道路や駐車場などの限定した領域にてドライバー操作の一部を自動化する自動運転システムの製品化を発表している。一方、米国やドイツの交通局では、今まで明確にされていなかった自動運転の自動化レベルの定義を発表し、自動運転システムを社会的に受け入れる法的な整備も進展しつつある 3)4)。

このような状況から、国内では高速道路における公道試験走行が進んでいる一方、一般道では最近になって一部地

^{†1} (株)日立製作所
Hitachi Ltd.

域で公道試験走行が開始された段階である 5)。高速道路と比べて、一般道、特に市街地においては、歩行者・自転車などの複数の移動体に対する衝突回避が求められ、技術的な難易度が高いことが特徴である。そこで、本研究では、市街地での自動運転の実現を目的に、車両運動制御方式の開発、および組み込み ECU での実装を行った。

車両運動制御には、動的障害物の将来位置や車両挙動予測に基づいた非線形モデル予測制御(Non-linear Model Predictive Control: NLMPC)を用いた。MLMPC を用いることにより、複数の動的障害物に囲まれた複雑な環境下において、衝突を回避し、かつ乗り心地のよい車両運動制御を安定して実行することが可能となった。

さらに、本制御を組み込み ECU でリアルタイム実行するためにアルゴリズムの高速化を行った。本制御でボトルネックとなる演算を抽出、データの再利用や FPGA での並列処理により処理時間の短縮を行った。これにより、開発した車両運動制御を 100ms の周期で実行可能となり、市販車への搭載可能な見込みを得た。

本論文では、開発した車両運動制御方式、および本方式を組み込み ECU で高速実行する方法の詳細を述べる。

2. 関連研究

モデル予測制御を活用した自動運転の先行研究として、高速道路におけるレーンキープ、およびレーンチェンジの制御手法が開発されている 6)。この先行研究では、制御特性を評価する評価関数を二次式で定式化し、操作量の制約を一次不等式とすることで、二次計画問題に帰着させ、レーンキープ、およびレーンチェンジ時の最適な操作量を演算している。これを市街地での自動運転に拡張する場合、評価関数が複雑化し、二次計画問題では解けないという課題がある。

より複雑な環境下でモデル予測制御を行うには、非線形の評価関数を扱うことが出来る非線形モデル予測制御(NLMPC)が必要となる。NLMPC を活用した車両運動制御の先行研究 7)では、非線形の評価関数の最適化に、GMRES/C アルゴリズム 8)を活用している。GMRES/C アルゴリズムでは、最初に最適解の初期値を演算し、評価関数の微分から、最適解の差分を漸近的に演算することで、非線形の最適化問題を高速処理している。しかしながら、評価関数の大きな変動、例えば周囲環境の急激な変化、例えば動的障害物の急な速度変化等があった場合、最適解の初期値を再計算する必要があり、制御周期内に操作量が演算出来ないという課題がある。また、最適解の初期値が局所解に収束した場合、制御性能が悪化するがケースも有り得る。

本研究の目的は、一般道、特に市街地での自動運転が必要となる非線形モデル予測制御のロバスト性を向上し、さ

らにこれを車両制御に必要な制御周期(100msec)内で高速実行する方式を開発することである。

3. 非線形モデル予測制御を用いた経路生成

3.1 問題設定

図 1 に本研究が対象とする、市街地交差点における右折通過シーンを示す。

交差点右折シーンにおいては、自車は交差点の形状から求めた固定の軌道(Fixed path)を通過することを前提とするため、1次元の速度パターンが解くべき操作量となる。安全かつ、乗り心地の良い速度パターンを生成するためには、対向車(Vehicle1)、および複数の歩行者(Pedestrian1, 2)の動きを予測し、その予測に基づいた速度パターンの最適化を行う必要がある。

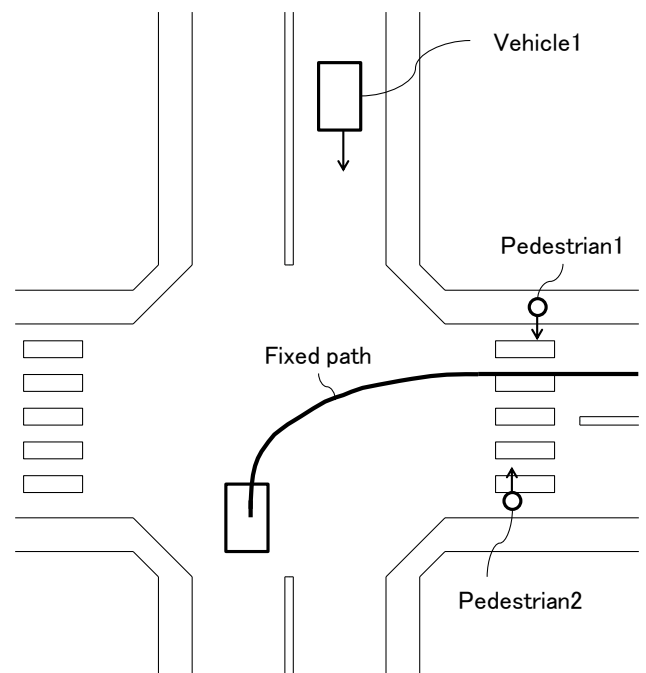


図 1 交差点通過シーンにおける問題設定

Figure 1 Problem setting for intersection environment.

3.2 経路生成手法概要

本経路生成手法を実装した経路生成部を含む自動運転システムの全体図を図 2 に示す。

ステレオカメラ等の外界認識センサが、対向車や歩行者といった動的障害物情報を検出し、検出した位置、速度情報を障害物確率分布予測部に入力する。障害物確率分布予測部は、一定時間の将来に渡る動的障害物の存在確率分布を生成する。動的障害物の存在確率分布の生成には、モンテカルロシミュレーションや、行動予測モデル等を用いる。経路生成部は、障害物確率分布予測部が生成した一定期間の将来に渡る動的障害物の存在確率分布を元に、安全かつ、乗り心地の良い走行パターンを計算し、その走行パターンを実現するための指令値をエンジン、ブレーキ、ステアと

いった各種車両制御用 ECU に分配する。

本報告で述べる経路生成手法は、経路生成部の走行パターンの演算方法に関するものである。

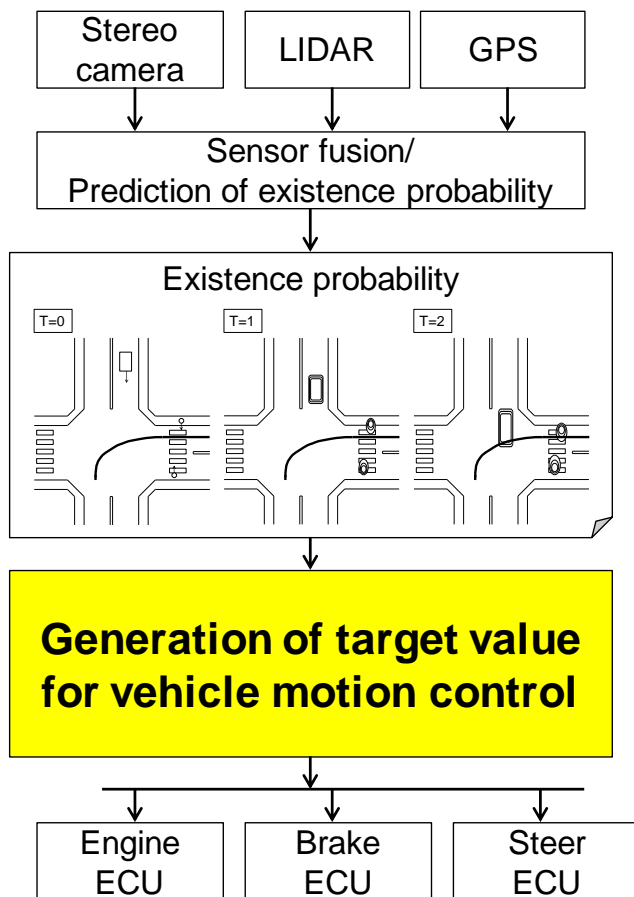


図 2 自動運転システムブロック図

Figure 2 Block diagram of autonomous driving system

3.3 評価関数の設計

最適化問題は、変数 x を入力とする評価関数 $H(x)$ が最小値となる時の x 値を求める問題である。ここで、 x を制御対象に対する操作量、評価関数 $H(x)$ を制御対象の所望の状態からの誤差として定式化し、最適化問題を解くことにより、最適な操作量を導出することが出来る。

本経路探索手法では、変数を加速度指令値 $a(k)(k=1,2, \dots, n)$ 、評価関数 $H(a)$ を式 (1) の通り定式化し、安全、かつ乗り心地の良い加速度指令を生成する。

$$H(a) = H_{\text{risk}} + H_v + H_a + H_j \dots \text{(式 1)}$$

$$H_{\text{risk}} = \sum_{k=1}^n P(k): \text{衝突確率項}$$

$$H_v = \sum_{k=1}^n (v(k) - v_{\text{ref}})^2: \text{速度項 (} v_{\text{ref}} = \text{目標速度)}$$

$$H_a = \sum_{k=1}^n a(k)^2: \text{加速度項}$$

$$H_j = \sum_{k=1}^n j(k)^2: \text{加加速度項}$$

なお、衝突確率項における $P(k)$ の演算は、時刻 k における動的障害物の存在確率分布 $\phi(x,y,k)$ を自車領域 S 内で積分することにより求める(図 3)。

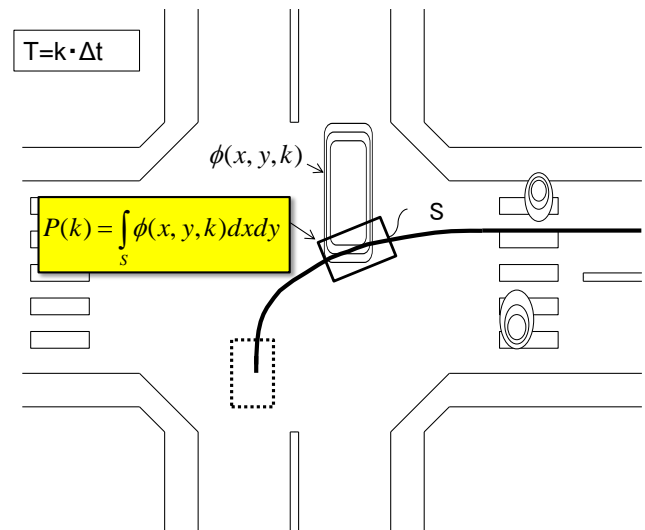


図 3 衝突確率の演算方法

Figure 3 Calculation of collision probability.

3.4 最適化ソルバーの選定

最適化問題を解くアルゴリズムである最適化ソルバーはいくつかの手法があるが、扱いが可能な変数量の多さ(スケラビリティ)、局所解問題の回避能力、処理速度、並列化の容易性を考慮し、人口蜂コロニーアルゴリズム(ABC: Artificial Bee Colony optimization)を選択した(表 1)。

表 1 最適化手法の比較

Table 1 Comparison of optimization problem solver.

	Iterative Methods	Heuristics		
		GA	PSO	ABC
Number of parameter	×	○	×	◎
Local minimum problem	×	○	×	○
Speed	◎	△	○	◎
Parallelization	×	○	○	○

GA: Genetic Algorithms
 PSO: Particle Swarm Optimization
 ABC: Artificial Bee Colony optimization

3.5 人口蜂コロニーアルゴリズム

人口蜂コロニーアルゴリズムは、蜂の捕食行動を模倣したアルゴリズムであり、初期値に依存せず、大域解を探索出来るといった特徴がある。また、蜂をプロセッサエレメントに対応させることで、専用ハードウェアによる並列化が可能である。

蜂の数=M、次元数=N のパラメータ の最適解を探索する ABC アルゴリズムの処理を以下に示す。

- (1) 蜂の初期化: 全蜂のパラメータを初期化
- (2) 収穫蜂フェーズ: 以下に従い, 蜂のパラメータを更新 (:更新後のパラメータ)
- (3) 偵察蜂フェーズ: 一定回数改善のない(評価関数値が下がらない)蜂を初期化
- (4) 規定回数 or 評価関数値収束まで(1)-(3)を繰り返す

図 4, 図 5 に本アルゴリズムで生成した速度パターンの一例を示す. 図 4 は, 本アルゴリズムで生成した速度パターン, およびそれに対応した固定軌道上の移動距離を示したものである. また, 図 4 で示す時刻①~④における自車, および動的障害物(車, 歩行者)の位置関係を図 5 に示す.

図 4, 図 5 に示す通り, すべての障害物が自車軌道を横切った後に, その交点を通過していることから, 直近の対向車だけでなく, その先の歩行者の動きも考慮し, 複数の動的障害物を回避する最適速度パターンを生成していることを確認した.

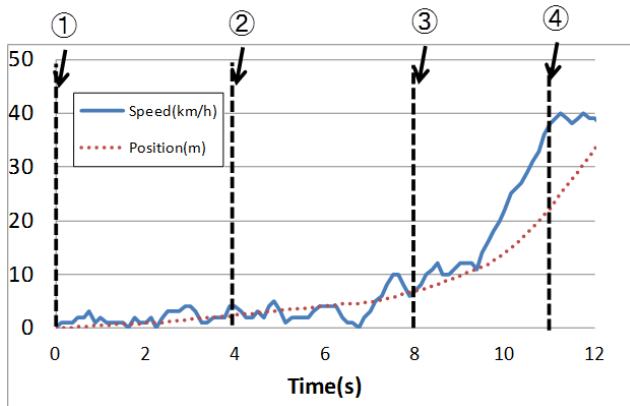


図 4 自車の速度, 位置プロフィール
 Figure 4 Speed and position of ego-vehicle.

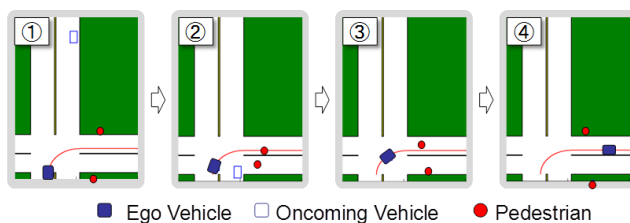


図 5 自車, および動的障害物の位置
 Figure 5 Location of ego-vehicle and obstacles.

4. 経路生成アルゴリズムの高速化手法

本章では, 3 章で説明した経路生成アルゴリズムの高速化手法の詳細を述べる.

4.1 ボトルネックの分析

表 2 に 3 章で説明した ABC による経路生成アルゴリズムの処理時間内訳を示す. 収穫蜂における評価関数の演算が 99% と処理のほとんどを占めている.

表 2 計算処理時間の内訳

Table 2 Breakdown of the calculation time.

Function		Ratio(%)
Initialization		1.03
ABC	Evaluation function	98.92
	Update x value	0.04

これは, 蜂の数 x イタレーション回数分, 評価関数演算処理が実行されること, また, 図 3 で説明した, 衝突確率項 H_{risk} の演算にて, 低速な外部メモリに格納される動的障害物の存在確率分布 $\phi(x,y,k)$ を積分することが原因である(図 6).

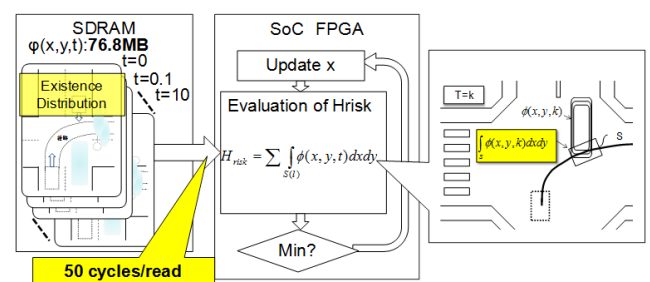


図 6 H_{risk} 演算の詳細
 Figure 6 Calculation for H_{risk} .

4.2 衝突確率演算の高速化

本アルゴリズムの最大のボトルネックである衝突確率演算の高速化手法を図 7 を用いて説明する. 4.1 節にて説明の通り, 従来のアルゴリズムでは, 衝突確率項 H_{risk} の演算を行うループ内で, 低速な外部メモリに格納される値を積分することで速度が低下する. 衝突確率 H_{risk} は, 固定の軌道上で演算されることに着目すると, 最適化ループ内で同じ演算を複数回実行するため, 演算結果の再利用による高速化が期待できる.

そこで, 時刻 t , 固定軌道上での自車移動距離を l , とした時の衝突確率をあらかじめ二次元テーブル $R(l,t)$ として計算しておき, 最適化ループ内で演算結果を再利用する. $R(l,t)$ は, 障害物存在確率分布 $\phi(x,y,t)$ に比べ容量を大幅に削減することが出来るため, 高速アクセス可能な内蔵 RAM に格納する. 最適化ループ内での衝突確率 H_{risk} の演算は, $R(l,t)$ の時間に関する総和を演算するだけで良く, 大幅な高速化が可能となる.

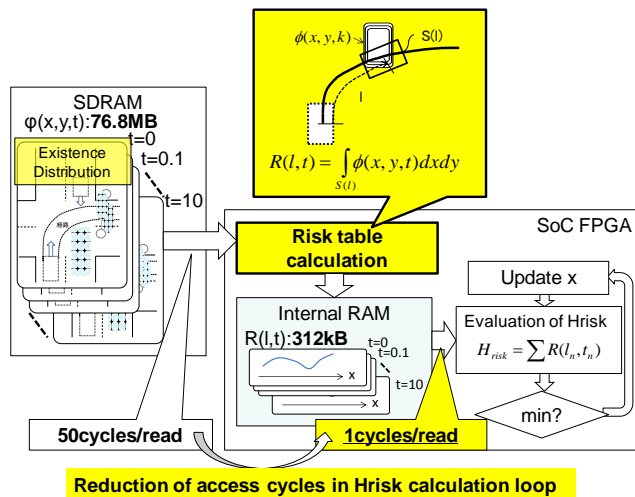


図 7 データ再利用による Hrisk 演算の高速化
Figure 7 Speed up method for the calculation of Hrisk.

4.3 CPU 内蔵 FPGA を用いた並列化

4.2 節で説明した高速化方式に加え、CPU 内蔵 FPGA を用いて、ABC アルゴリズムの並列化を行った(図 8)。ABC アルゴリズムにおける、蜂のパラメータ更新はハードウェアに不向きなランダム演算を含むため、CPU 内蔵 FPGA 内の CPU 上のソフトで実行し、更新されたパラメータを入力とした評価関数演算は、FPGA を用いた専用ハードで並列実行する。

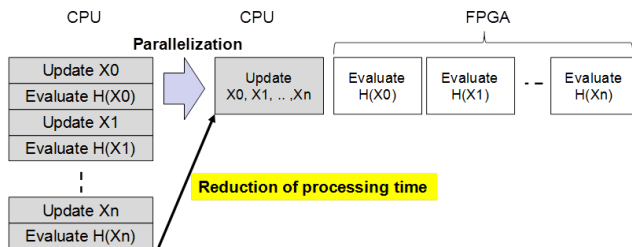


図 8 CPU 内蔵 FPGA を活用した並列化
Figure 8 Parallelization with CPU and FPGA.

4.4 性能評価結果

4.2, 4.3 節で説明した高速化手法の効果を図 9 に示す。再利用可能な衝突確率演算用中間データを内蔵 RAM に配置することにより、処理時間を 325(s)から 1.5(s)に短縮した。さらに、CPU と評価関数演算専用 FPGA を用いた並列化により、処理時間を 0.08(s)まで短縮し、一般的な車両制御における制御周期である 100ms 以下での処理を実現し、組込みレベルの ECU で本制御を実現する見通しを得た。



図 9 性能評価結果

Figure 9 Evaluation of execution time.

5. おわりに

安全、かつ乗り心地の良い自動運転実現のため、非線形モデル予測制御を活用した経路生成アルゴリズムを高速実行する技術を開発した。特徴は以下二点である。

(1) 固定軌道に着目した演算結果の再利用

本経路生成のボトルネックは生成した経路における動的障害物との衝突確率演算である。本手法では、固定軌道の特徴を生かし、衝突確率演算用の中間データを生成、高速アクセス可能な内蔵メモリに格納し、最適経路の探索するイタレーションでは、この中間データを参照することで演算の高速化を図る。

(2) CPU 内蔵 FPGA を活用した最適化ソルバーの並列化

最適化ソルバーは、求めたいパラメータ x に対し、その最適度を示す評価関数 $H(x)$ を演算し、 $H(x)$ が最小となる x を探索するイタレーションを実行する。このイタレーションを並列化するため、探索したいパラメータを CPU ソフトウェアで複数組生成し、この複数組のパラメータを用いて評価関数を FPGA 活用専用ハードウェアで並列演算し、高速化を行った。

以上(1)(2)により処理時間を 0.08(s)まで短縮し、一般的な車両制御における制御周期である 100ms 以下での処理を実現し、組込みレベルの ECU で本制御を実現する見通しを得た。

参考文献

- 1) EuroNCAP, <http://www.euroncap.com/home.aspx>
- 2) 永井正夫：自動車の予防安全技術と運転者の運轉行動モデル、システム情報制御学会誌 システム/制御/情報 V1.55, No.1, pp.26-31 (2011-1)
- 3) National Highway Traffic Safety Administration : Preliminary Statement of Policy Concerning Automated Vehicles, U.S. Department of Transportation Releases Policy on Automated Vehicle Development, NHTSA 14-13, (2013-5)
- 4) Tom M Gasser, Daniel Westhoff : BASt-study : Definitions of Automation and Legal Issues in Germany, Transportation Research Board -Road Vehicle Automation Workshop- ,German Federal Highway Research Institute, <http://onlinepubs.trb.org/onlinepubs/conferences/2012/Automation/presentations/Gasser.pdf>(2012)
- 5) 菅沼直樹, 他, 高齢過疎地域における自動運転自動車の市街地公道実証実験概要, 自動車技術会学術講演会予稿集, S070, pp. 390-394 (2015)
- 6) Seo J. et al: Robust Mode Predictive Control for Lane Change of Automated Driving Vehicle, SAE Technical Paper 2015-01-0317, 2015, doi:10.4271/2015-01-0317(2015)
- 7) Rosolia, U. et al: NLMPC for Real Time Path Following and Collision Avoidance, SAE Int. J. Passeng. Cars 2015, doi:10.4271/2015-01-0313(2015)
- 8) Ohtsuka T. , A continuation/GMRES method for fast computation of nonlinear recording horizon control, Automatica, Vol.40, No. 4, pp.563574(2004)
- 9) 森大輔: 主成分分析を取り入れた Artificial Bee Colony アルゴリズム, 電気学会論文誌 C, Vol.135 No.4 pp.423-435(2015)