東京大学
情報理工学系研究科 創造情報学専攻
修士論文

地域に最適化された遺伝的アルゴリズムベースのポテンシャル場による走行計画

Regionally Optimized Path Planning with Genetic Algorithm-Based Potential Fields

廣瀬稔彦
Toshihiko Hirose

指導教員　塚田 学 准教授

2021 年 1 月

概要

近年自動運転において Path Planning は盛んに研究されている．Path Planning とは，車が取る経路を決定するタスクであり，事故防止の観点から非常に重要なタスクである．そして内閣府の交通安全白書によれば，道路の形状（交差点，カーブ，直線路など）によって交通事故の起こる頻度は異なる．また見通しの悪さが異なっていたり，住宅街であることなどによって，同じ形状の道路であっても取るべき運転の方針は変わる．しかし Planning では道路の形状や周りの車の配置など，Perception によって得られた周辺のデータが似ている場合，似た運転をすることになる．そのため交通事故を防止するにあたり，Perception ではとらえきれない地域の特性によって運転の方針を様々に変更する研究が必要であると考えられる．しかし走行中に運転方針を様々に変更する研究は少なく，その地域に最適化された運転をする研究はない．そこでリアルタイムに運転の方針を変更し，地域に合わせたより安全な運転を行う手法を本研究では提案する．問題の解決のためには，リアルタイムでスムーズに運転方針を変更でき，かつ様々な地域に対する最適化をするために変更のバリエーションが広い手法を選ぶ必要がある．本研究では，パラメータ依存的な手法であり，パラメータを最適化することによって，地域固有の運転の最適化が可能と考えられる Artificial Potential Field（APF）を基礎として採用した．APF には，またパラメータが多く存在しており，各パラメータが持つ意味が明確であり，パラメータを最適化することによって本研究の目的を達成できる．本研究ではパラメータをある区間において遺伝的アルゴリズムを用いて最適化することによって，地域ごとに運転の最適化を行う．本研究では，車間距離を最適化しない場合よりも広く保つ運転を行う．他の車が一台存在する状況での実験の結果では，パラメータはある解に収束した．このパラメータを用いて走行したところ，車間距離はパラメータを最適化していない状態より大きく取れることが確認できた．

Abstract

In recent years, path planning has been actively researched in automated driving. path planning is the task of determining the path that a car will take, and it is a very important task from the viewpoint of accident prevention. According to the White Paper on Traffic Safety published by the Cabinet Office, the frequency of traffic accidents varies depending on the shape of the road (intersection, curve, straight road, etc.). In addition, the driving policy to be taken varies even for the same road shape, depending on whether the visibility is poor or the road is in a residential area. However, in Planning, if the surrounding data obtained by Perception, such as the shape of the road and the layout of the surrounding cars, is similar, the driver will drive in a similar manner. Therefore, in order to prevent traffic accidents, it is considered necessary to conduct research to change the driving policy in various ways depending on the characteristics of the area, which cannot be captured by Perception. However, there are few studies in which the driving policy is changed in various ways while driving, and there are no studies in which the driving is optimized for the region. In this study, we propose a method to change the driving policy in real time and to make driving safer according to the region. In order to solve this problem, it is necessary to select a method that can change the driving policy smoothly in real time and has a wide range of changes in order to optimize for various regions. In this study, we adopted the Artificial Potential Field (APF) as a basis, because it is a parameter-dependent method that can optimize region-specific operation by optimizing the parameters. The APF has many parameters, and the meaning of each parameter is clear, and the objective of this study can be achieved by optimizing the parameters. In this study, the parameters are optimized in a certain interval using a genetic algorithm to optimize the driving in each region. In this study, the distance between vehicles is kept wider than the case without optimization. The results of the experiment in the presence of one other car showed that the parameters converged to a certain solution. In the experiment, the parameters converged to a certain solution, and when the car was driven with the parameters, it was confirmed that the distance between the cars was larger than that without optimizing the parameters.

Contents

第 1 章　序論 1
1.1 背景 1
1.2 目的 2
1.3 構成 2

第 2 章　自動運転における Planning 3
2.1 自動運転の流れと Planning の位置づけ 3
2.2 VEHICLE MODELING 3
2.3 遺伝的アルゴリズム 4

第 3 章　関連研究 5
3.1 自動運転研究 5
3.2 APF を除いた Local Path Planning 5
3.3 APF 9
3.4 協調型通信を活用した Path Planning 11
3.5 各手法の長所と欠点 12
3.6 既存研究の課題 12
3.7 要求事項 13

第 4 章　提案手法 14
4.1 要求事項 14
4.2 提案手法にて用いる APF 14

第 5 章　実装 19
5.1 実装の概要 19
5.2 変数の定義 19
5.3 Vehicle model 22
5.4 遺伝的アルゴリズム 23

第 6 章　実験・評価 25
6.1 実験環境 25
6.2 実験概要 25
6.3 車 1 台の追い越しのシナリオ 25
6.4 考察 30

第 7 章　結論 33

7.1 まとめ 33
7.2 今後の課題 33

References 34

第 1 章

序論

1.1 背景

Path Planning とはスタート地点からゴール地点まで障害物を避けながら，たどり着くことを目的とした自動運転やロボティクスにおけるタスクである．高度な Path Planning は自動運転には不可欠であり，自動運転の実現によって事故防止，物流における運転車の解決，高齢者等の交通手段の確保などが可能とされていることから，Path Planning の研究が社会に与える影響は非常に大きいと考えられる．また本論文で大きく取り上げる自動運転による安全性の向上は，民間企業も政府機関も注目している．民間企業では，2017 年には Audi は世界初のレベル 3 の量産車である Audi A8 を発表した．またトヨタ自動車株式会社が 1990 年代から「交通事故死傷者ゼロを第一の目標として、予防安全と自動運転技術の研究開発に取り組んで」いる．[1]．ダイムラー，ボルボ，BMW，ホンダなども 2021 年 11 月現在，レベル 3 の自動運転車の開発に成功している．政府機関では国土交通省自動車局の資料の記述においては以下のようなものがある．「自動運転技術の実用化により、これらヒューマンエラーに起因する交通死亡事故が大幅に削減されることが期待されて」おり，「自動運転の実現において、『自動運転システムが引き起こす人身事故 2 がゼロとなる社会の実現を目指す』ことを目標として設定し、自動運転車の開発・普及促進を行う意義を明確にする。」[2].

このように安全な運転の実現は多くの団体から望まれており，実際に事故を防ぐ Path Planning の既存研究は数多く存在している．2010 年代後半からは機械学習と Path Planning を組み合わせた手法も開発されており，予測をアルゴリズムに反映させたようなものも産まれている．また協調型 ITS（以下この論文では CITS と呼ぶ）と呼ばれる手法も 2010 年代後半から研究が始まっている．CITS では自動運転車は周囲の車両や，道路に設置された機器と通信を行う．これによって自動運転車は周囲の車両の未来の行動や，観測した周囲の情報などを受け取ることができる．この情報に基づいた Path Planning の手法も産まれている．このように Path Planning の手法の発展は現在も著しい．

しかし，現在の研究には考えられていない問題がある．日本の事故についてまとめた令和 2 年に発行された内閣府の交通安全白書によると，令和元年の道路形状別交通死亡事故発生件数は一般道路での数が多いものの，カーブや交差点などにおいても数が多い [3]． この資料から，交差点など場所によって事故が起こりやすくなることが読み取れる．また，自動運転が発展し普及していくにしたがって，ヒューマンエラー的なミスが減り，機器の故障の他に場所の特性によって事故が発生することが想定される．そのため自動運転において，地域の特性を考慮した研究が必要になると考えられる．ところが Path Planning を用いて事故の発生を未然に防ぐ研究は数多く存在しているが，地域の特性に注目した運転を行い事故を防ぐことを狙いとした研究はない．

1.2 目的

本研究の目的は，地域に最適化した運転の方針を変更し，より安全な自動運転を実現することである．そのため，Path Planning のアルゴリズムの一つである Artificial Potential Field という手法のあるパラメータに対して遺伝的アルゴリズムを実行し，ある区間上でパラメータを最適化する手法を提案する．加えてパラメータを Road Side Unit という路側機を使ってパブリッシュする状況を想定する．提案手法を MATLAB/Simulink（バージョンは R2021b）を用いて実装し，最適パスと解を生成した．

1.3 構成

本論文の構成は以下である．2 章では研究を理解するために必要な知識と現在の Planning の課題を述べる．また 3 章では関連研究と既存研究の具体的問題について述べる．4 章では 3 章で述べられた課題解決のための提案手法を述べる．5 章では実装について述べる．6 章では実験と生成されたパスと解の評価について述べる．7 章では本研究のまとめと今後の課題を述べる．

第 2 章

自動運転における Planning

本章では提案手法の理解にあたって必要な知識を記す．

2.1 自動運転の流れと Planning の位置づけ

Path Planning はロボティクス分野でも扱われているタスクであり，自動運転の研究より以前から多くの研究が進んでいる．そして，Path Planning は大きく分けて Global Path Planning と Local Path Planning の二つに分かれる．Global Path Planning は事前に得た地図情報を使って Path Planning を行うものであり，Local Path Planning は動的な障害物を避けるために行われる．Local Path Planning はロボティクス分野や自動運転の分野で盛んに研究が行われているが，本論文ではその中でも Artificial Potential Field（APF）と呼ばれるものを取り扱う．この章では APF が自動運転においてどのように使われるのかを紹介する．一般的な自動運転車におけるタスクは Perception,Planning, Control に分かれる．各要素について説明する

■Perception　動物体の認識，および信号機の色認識などを行う．センシングデバイスとしてはカメラ，Light Detection and Ranging（LiDAR），ミリ波レーダーなどが用いられる．

■Planning　Perception で受け取ったデータをもとにゴールまでのルート計算から実際の走行軌跡生成までの処理を行う．Planning については二つに大別される．一つ目は事前に得た地図情報を元に最も適した Path を生成する Global Path Planning．二つ目は動的障害物を避けるために，リアルタイムに経路を再生成していく Local Path Planning である．

■Control　Planning で行われた軌跡を走るため，車両へ実際に行うコマンドの生成を行うこと．タイヤの角度コントロールや速度のコントロールはこのフェイズで行われる．

APF が直接関係するのは Local Path Planning である．

2.2 VEHICLE MODELING

Vehicel Modeling とは車両の運動を追うために車をモデル化したものである．車両の運動を記述する手法はいくつか存在する．[4] においては多数の方法が紹介されているが，y 座標と車体角度をコントロールするもの（Bicycle Model）などがある．また運用の際には状況に応じて，どのようなモデルを扱うかということが問題になる．

人工ポテンシャル場法に基づく経路計画は、計算が簡便であること、ハードウェアの実装が簡単であること、リアルタイム性が優れていることなどの利点から、ますます注目されている。

https://ieeexplore.ieee.org/document/9134006

2.3 遺伝的アルゴリズム

遺伝的アルゴリズム（Genetic Algorithm; GA）とは，自然淘汰の過程を参考に産まれた，最適化問題に応用されているアルゴリズムである．遺伝的アルゴリズムでは，ある評価関数を最小化するような遺伝子を探索することが目的となる．また遺伝的アルゴリズムは大域的探索に向いており，非線形性や多峰性（局所解が複数存在するもの）を持つ評価関数に対しても有効な手法となっている．

各遺伝子には遺伝子座（locus）と呼ばれる位置が一つ以上存在し，遺伝子座上に数値などの遺伝情報が記述されている．

GA の基本的な流れは以下である

1. 最初の世代の遺伝子を乱数などを使って生成する．
2. 評価関数により，現世代の遺伝子の適応度を計算する．
3. 終了条件を判定し，条件を満たしていた場合にはアルゴリズムを終了する．
4. 次世代の遺伝子を生成するため次の三つのいずれかの操作を行う．
   （a）個体を一つ選択して突然変異を行う
   （b）個体を複数選択して交叉を行う
   （c）個体を一つ選択して次世代にそのまま遺伝子をコピーする．
5. 次世代の遺伝子が一定の数に達するまで，上記の操作を行う
6. 2 以降の動作を最大世代数に達するまで繰り返す．そして最終世代において最も適合度の高いものを解とする．ただし，3 にて終了条件を満たしていた場合は最大世代数よりも前であってもアルゴリズムを中断して解を出力する．

第 3 章

関連研究

3.1 自動運転研究

自動運転のアイデアが現実となったのは，PROMETHEUS プロジェクトが開始された 1980 年代である [5]．またこの 15 年ほどの自動運転の発展は DAPRA Grand Challenge（DAPRA）と呼ばれるコンテストと関連があるとされている [6]．DAPRA では 2004 年・2005 年にオフロード（舗装されていない道路や、公道ではない脇道）を走るチャレンジが開催されており，2007 年には都市部を走るチャレンジが開催されている．また一般車両以外での自動運転のコンテストも開催されている．2019 年から毎年開催されている Indy Autonomous Challenge では，レーシングカーを自動運転している [7].

3.2 APF を除いた Local Path Planning

Local Path Planning においては Unmanned Aerial Vehicle（UAV）やロボティクスや自動運転の分野から様々な手法が提案されている．

Local Path Planning の手法の分類は [8] から抜粋したものに加え, 協調型 ITS（Cooperative Intelligent Transport System; CITS）を紹介する．

3.2.1 Graph Search Based Planners

この節において，グリッドと言う場合には格子の線または空間の区切り方を指し，セルと言う場合にはグリッドに囲まれたマスのことを指す．Path Planning というタスクは，車両を使ってスタートである地点 A から，ゴールである地点 B まで移動するため空間を移動していくタスクと言える．この空間はグリッドによって区切られたセルの集合として表現されることがある．この場合は車両は常にいずれかのセル上に位置しており，Path Planning において車両が複数のセルを占有しているという表現をする研究も存在する．そのためセルは環境内のある場所にどの程度の大きさの物体があるかを示していると言える．

A-star

車両は通常隣接するセルのみに移動できる．セルとして表現されたマップはグラフとして表現できる．複数の車両の運動を同時に最適化する問題は Multi Agent Path Finding(MAPF) と呼ばれ，これはグリッドベースで考えられることが多い．この MAPF は現実の問題に対しては倉庫作業の最適化において使われることがある．また時間を連続にしたもの・運動学的な制限を設けたもの・オークションの考え方を組み込んだものなど，前提を変える研究や特殊なアルゴリズムを使う研究が多い [9] [10], [11]．一方でセルとして表現さ

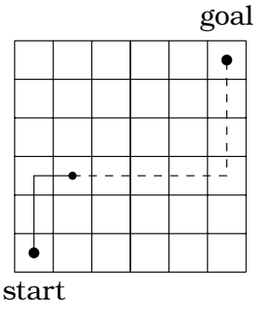


図 3.1: マンハッタン距離（実線は $g(v)$, 点線は $\hat{h}(v)$）

れたマップに対して，単一の車両の経路計画を最適化する場合には単一始点最短経路問題を解くこととなる．この問題に対しては一般的にダイクストラ法を用いる．

一方 Path Planning において使われることが多いのは A-star（A*）アルゴリズムである [12]．これはダイクストラのグラフ探索アルゴリズムに対してヒューリスティックな要素を加えたものとされる．次のように記号を定める

- $g(v)$: スタート地点からノード n までの最短距離
- $h(v)$: ノード v からゴール地点までの最短距離
- $f(v)$: ノード v を通る場合の最適な道のコスト

そして各関数に対してハットがついたものは推定値を表すとする．これによって各マスの最適なパスを選んだ場合の推定コストを

$$\hat{f}(v) = g(v) + \hat{h}(v) \tag{3.1}$$

と定義する．自動運転においては通常 $\hat{h}(v)$ はユークリッド距離における最短距離か，図 3.1 に示すようなマンハッタン距離を求める．A*は以下のような手順を踏むアルゴリズムである．

1. スタート地点の隣接したセルに対して，$\hat{f}(v)$ を定める
2. 計算済みかつ隣接コストを計算するセルとして，選択済みでないセルのうちコストが最も小さいセルを次のセルとして選択する（このノードを u とおく)
3. u の隣接セルのコストを計算する．
4. ゴール地点が計算されるまで 2 と 3 を繰り返す．

自動運転における A*の応用例には，駐車場などの衝突してはならない物体が多く，駐車などの特殊な運転に向けた研究が存在する [13]．この研究では (x, y, θ, r) の 4 つの変数に対して A*を改変した Hyblid A*を行う．（θ は車両の角度．r は $r = 0, 1$ であり，前進か後退のいずれであるかを表すフラグである．）この研究は二段階のフェーズで構成されている．第一段階では hyblid A*を用いて車両の運動学的に実現可能なパスを生成する．第二段階では非線形関数による最適化によって解の質を向上させる．さらにこれを拡張して事前知識も用いることで最終的な軌道を実現している．実験の結果，通常の A*のみを用いる Path-Planner と比較して，Waypoint の数と悪いパスを生成した後の再計算の面で優れた結果を示した．

State Lattice

また A*以外の Graph Search Based Planners では，State Lattice という手法も自動運転では使われる．State Lattice とは高次元の情報を離散化することによって無数の選択肢を，少ない有限個の選択肢に抑

える方法である．State Lattice を用いた手法を 2 つ紹介する．まず，Spatiotemporal State Lattice と呼ばれる State Lattice についてである [14]．これは時空間をまとめたグラフ上で State Lattice の考え方を用いて Path Planning を行う手法である．空間の次元に加えて時間の次元をまとめ，連続的な空間を離散化するため自動運転においては．3 次元空間上の離散グラフによって車が通る道は表現される．空間と異なり時間は正の方向にしか進まない性質を持つため，この離散グラフは Directed Acyclic Graph（DAG）よりもより強い条件を持つグラフとしてみなせる．進行方向の座標を l とし，それに直行する方向を r とおくと，l と r の時間に関する 1 階微分と 2 階微分に対して制約を設けることによって，グラフ上の移動に対しての適切な制約が設けられる．この研究では 2 つのシナリオにおいて実験が行われた．1 つ目はカーブしている道路での追い越しのシナリオ．2 つ目は交差点での合流シナリオである．

もう一つの研究は，角度に関する State Lattice をセッティングする [15]．この手法ではパスの始点と終点の x 座標（進行方向に関する座標）と終点と始点における車体の角度の差分を先に決めておく（これらを x_f,θ_f とおく）．x_f,θ_f を決めておくことによって，始点を基準として終点において角度 θ_f を取る円周型のパスが一意に定まる．そして，始点における車体の角度を 0 度として，15 度刻みでパスを作る．この複数生成されたパスの中から車両が通る道を選ぶ．この研究においては交差点での直進，左折，U ターンを行っていた．

1 つ目の研究では時空間を離散変数として，2 つ目の研究では角度を離散変数としていた．このようにパスを離散化されたものから生成することによって，素早いパス生成が可能になる．

3.2.2 Sampling Based Planners

Sampling Based Planners は，決定論的手法では対応できない時間的制約，すなわち高次元空間でのプランニングを解決しようとするものである [8]．この手法では，構成空間や状態空間をランダムにサンプリングし，Path をつないでいくという手法を取る．Sampling Based な手法はロボティクスの Path Planning においては，Probabilistic Roadmap Method（PRM）という手法と，Rapidly-exploring Random Tree（RRT）という手法が代表的である．自動運転では RRT を用いるものが多いため，RRT の説明のみ行う．

RRT は以下の手続きによって生成される手法である．

1. スタート（s とおく）とゴール地点を決める．$P = \{s\}$ として点集合 P を定める
2. 領域内にランダムに点を打つ（この点を r とおく）
3. そのランダムに打った点から，P の要素のうち最も近いものを p とする．p から r の方向へ線分を伸ばし，線分の p ではない側の点 q を P に含める．（一般に $q \neq r$）
4. これを目的地にたどり着くまで繰り返す．

このアルゴリズムの利点は高速かつ計算コストが低く，また実装が比較的簡単であることが挙げられる．一方欠点としては，最適なパスを生成するのが難しいことが挙げられる．自動運転において RRT を使う場合は，生成されたパスを滑らかにする操作が必要になる．ここで「滑らか」という言葉は，車体角度の変化が激しくなく，車体が振動していないことを指すこととする．

自動運転への応用例では，駐車への応用がある [16]．この研究では縦列駐車，直角駐車，進行方向に対して駐車場の角度が斜めであるような Echelon Parking の三つ全ての場合に対応している．RRT においては，新しい点を生成したとき点集合の中から最も近い点を選ぶ手続きがあるが，自動運転においてユークリッドノルムを使うべきではない．そのため，Reeds-Shepp 距離や Dubins 距離と呼ばれるものを使う [17] [18]．しかしこの距離の計算負荷は非常に大きいので，この研究では計算量を落として近似的に計算する方法が提案された．

3.2.3 Interpolating Curve Planners

Local Path Planning を行う際には Global Path Planning などによって前もって Waypoint が生成されている場合がある．実際の走行ではこのパスに対して，軌道の連続性，車両の制約，車両がナビゲートする動的な環境による影響を受けて新しいデータのセットを補完し生成する ．これを Interpolating Curve Planners という [8]．また動的な障害物が Waypoint 上に存在する場合には，それを避けるような経路を再生成する．補完の際に何を用いるかによって，この手法はサブカテゴリに分類可能である．

補完曲線は次のように分類される

1. 線と円
2. Clothoid Curves
3. Polynomial Curves
4. Bézier Curves
5. Spline Curves

各補完について簡潔に説明する．

線と円

線と円を組み合わせてパスを描画し，その後異なるセグメントに対しては補完を使うことによってパスを表現できる．

Clothoid Curves

クロソイド曲線とは曲率を一定の割合で変化させたときの曲線である．つまりハンドルを一定の角加速度で切った場合にできる曲線である [19].

Polynomial Curves

これは多項式による補完を指す．通常進行方向を x とし，それに直行する方向を y として，ある多項式関数 f によって $y = f(x)$ と，ある区間を補完する．本研究では Vehicle Model を使用するときに，この手法を用いる．

Bézier Curves

Bézier Curves は Bernstein 多項式によって表現される．この曲線はいくつかの制御点を決定すれば定義が可能になるため計算コストが低い．この特性を活かしてパス上に動的障害物が侵入した際に，パスを再生成し緊急回避を行う研究がある [20].

Spline Curves

Spline Curves は各区間上で Polynomial Curves または Clothoid Curves として定義できる区分的多項式パラメトリック曲線のことである．各区間の境界である接合部分は Knot と呼ばれ，通常 Knot 部分に滑らかさの制約を設ける．

3.2.4 Numerical Optimization

数値最適化と呼ばれる手法が存在する．

Function Optimization

関数最適化の手法では特定の関数の解を求めるか，特定の関数を最小化する．APF はこの手法に含まれる．APF については本研究の中心的な手法であるため，例外的に次の節で紹介する

遺伝的アルゴリズム

遺伝的アルゴリズムは広義には関数最適化に含まれる．Path Planning に遺伝的アルゴリズムを用いたものを紹介する．

まず交差点での交通に対して遺伝的アルゴリズムを用いたものがある [21]．この研究では交差点の位置を符号化し，またカーブに対しても定式化を行う．その後オートマトンを用いて車の位置を記述する．そして遺伝的アルゴリズムによって，交差点での面積に関する車両の平均占有率を最大化する．

3.3 APF

3.3.1 APF の概要

APF とは，古典力学における Potential Field のアナロジーから作られた Path Planning のためのメソッドである．APF では物体との距離や直進方向と直行する方向に関しての座標などを元に，各座標に対してポテンシャルの値を割り振る．この値は各座標に対して危険度を表現していると解釈が可能であり，また設計の段階では危険度が高いものに高い値を割り振れるようにしておく．通常は車線境界線，他の車両，車道の端，道路上の凹凸などに対して車が遠ざかるように周辺のポテンシャル値が高くなる．このように避けるべきものに対して設ける Potential Field を Repulsive Potential Field と呼ぶ．また車両はゴール地点に向かうべきなのでゴール地点などを中心に値を低く設定したり，また Global Path Planning で設けた Waypoint に対してもポテンシャルの値を低くすることがある．これを Attractive Potential Field と呼ぶ．そして各座標に対して Repulsive Potential Field と Attractive Potential Field の和を取ったものを，その座標におけるポテンシャルの値とする．最終的に各座標の勾配（Gradient）を取ったものを車に働く力とする．

APF に基づく Planning は，計算が簡便であること，ハードウェアの実装が簡単であること，リアルタイム性が優れていることなどの利点がある [22].

また APF は関数最適化の手法の一種である．そして関数最適化の手法の長所は，滑らかさや速さといった，パスや ego-vehicle に関する制限や他の車両についても，容易に考慮に入れることが出来る点にある．一方で関数最適化の手法の短所は計算時間が他の手法に比べて長いことである．それゆえに Time Horizon（最適化する際に考慮する現在時刻から最も未来の時刻）に制限がある．

3.3.2 APF を用いた研究

事故防止に関連する APF の研究を挙げると，2008 年の時点で高速道路において衝突回避を目的とした研究が存在する [23]．この研究では車の背後に楔形の領域をとって，車と楔形を合わせた五角形の図形のまわりに PF を設ける．また座標変換によって進行方向にゆがんだ PF を作り，後方の車に大きく車間距離を取るか，車線変更を強いる．本研究ではこの手法を改変したものを用いる．

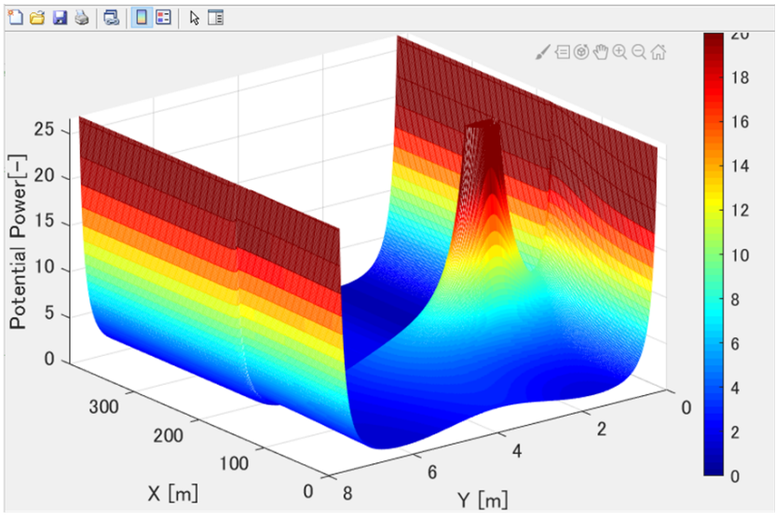


図 3.2: 生成した APF の一例

1. 車線境界線
2. 車道の端
3. 他車両
4. 速度

この論文では，上記 4 つに対して APF を設けている．この論文の手法によって生成された APF の一例は図 3.2.

y 方向の端の坂のような部分は車道の端周辺であり，画面手前でわずかに盛り上がっている部分は車線境界線であり，中央あたりの山のようになっているものは他車両による APF である．本研究ではこの論文による APF を用いる．

また同様に衝突回避を目的とした研究で，Waypoint を APF に活用した研究も存在する [24]．この研究では Waypoint Field と呼ばれる Attractive Potential Field を設けて，APF の中に waypoint 追従を組み込んでいる．筆者は Waypoint が離散的でありかかる力が急激に変化することを防ぐために，3 次多項式を用いて Waypoint をフィットする．この研究では 3 種類のプランナーと比較し，静的な障害物がある 1 つのシナリオと，動的な障害物がある 1 つのシナリオで実験を行った．実験の結果．他のプランナーよりも優れた結果を出した．

また 2016 年以降の APF の研究では Model Predictive Controller（MPC）を軌道生成に用いることが多い．MPC とは，「オープンループ最適化をサンプル時刻ごとに繰り返す」ような予測制御である [25]．精度は高いものの，サンプル時刻ごとに最適化を繰り返すため計算負荷が大きく，PC の性能の向上に伴い近年になって使われるようになった手法である．APF と MPC を組み合わせた研究は 2016 年に発表された [26]．この研究ではポテンシャル関数を MPC の制御問題にポテンシャル関数を考慮することによって，APF と MPC の融合を実現した．6 つのシナリオに対してこの手法が有効にはたらくことが，シミュレーションによって示された．

3.3.3 他分野と APF との融合

APF が現実世界のモデル化を行っているという性質上，他分野の手法を組み合わせてモデルに組み込む手法が近年提案されている．

機械学習と APF を組み合わせた研究も存在する [27]．この研究ではまず Long Short-Term Memory（LSTM）を highD データセットを用いて車の軌道予測を行う．そして Vechcle to Vehicle(V2V) 技術から抽出された情報を用いて，周辺車両の軌跡を予測する．その上で予測した軌道に基づき，実質的に APF と同等の Risk Assessment を行い Local Path Planning を行う．この研究では 2 つのシナリオに対してシミュレーションを行い，非線形入出力ニューラルネットワークと，Const-Model Predictive Controller（MPC）と比較実験を行った．一つ目のシナリオに関しては各手法で危険度に大きな違いはなかったが，二つ目のシナリオに関しては LSTM ベースの MPC が最も危険度が低かった．

流体力学と自動運転を組み合わせた研究も存在する [28]．この研究では 4 つの種類の流れを作りだし，それらを足し合わせる．各項は流れの速さを決めるパラメータを持つ．これらは実際の運転から取得したデータに，各点での車の向きができるだけ近づくように設定する．この研究で取り扱うポテンシャルの勾配は車両に働く力ではなく，ある点における速度を算出する．そして勾配によって車両に働く力を求めるものよりも，この研究では計算時間が少なくなっている．提案手法では 4 つのシナリオで実験をし，事前に決定したパラメータで有効に衝突回避ができるような軌道を生成できることを示した．

3.3.4 遺伝的アルゴリズムと APF を組み合わせたもの

次に APF と遺伝的アルゴリズムを組み合わせた手法を紹介する．このような研究は 2021 年 12 月現在非常に数が少ない．

まずロボティクス分野において，Path Planning に APF と GA を用いて複数のロボットを同時に動かす問題に対処したものがある [29]．この研究では通過する点を定めてから，その点同士を結んだものをパスとする．点を遺伝子としてパスを改良していく．この手法において遺伝的アルゴリズムを用いる際には，初期パスが必要である．通常これはランダムに生成されるが，この研究においては APF を用いて生成することによって計算時間を削減している．通過点の数はパスの表現能力を考えると，一定であるべきではないので，この手法では可変長の遺伝子を最適パスを見つけるという手法を取っていた．

また，はじめに遺伝的アルゴリズムを用いてグローバル軌道を最適化し，その後生成したグローバル軌道を基準に APF を生成するという研究もある [30]．この研究の遺伝的アルゴリズムでは最適化に時間変数が加わっているため軌道と速度・加速度曲線の同時最適化を可能にしている．これは実環境において必要な処置である．実験の結果，古典的な PF 軌道よりも遥かに滑らかな軌道を与えることができることが分かった．

3.4 協調型通信を活用した Path Planning

今まで述べてきたものは全て GPS などによって，グローバル軌道を前もってを生成する場合を除いて，通信を仮定していなかった．近年では V2X（Vehicle-to-Everything）通信を Path Planning に利用した研究も現れている．この場合複数の車両や計算機の間で情報が共有されることになる．

まず [31] においては，車両の将来の Path Planning やパスを定義し，複数車両の将来のパスの情報に基づく Future Path Sharing を用いた交差点での協調経路計画モデルを設計している．RSU は，共有された Future Path に従って，車両経路の潜在的な衝突や加速可能なタイミングを検出した場合，車両の速度を調

整する協調経路更新を生成する．この研究においては，2 台の車両を用いたシミュレーション実験が行われた．その結果，交差点において RSU を導入した場合の計算時間は，単独で自律走行する場合に比べて 23.0 %、28.1 % 短縮することができることが示された．

3.5 各手法の長所と欠点

ここまで様々な手法を見てきたが，それぞれの手法には長所と短所が存在する．これを以下にまとめる．

- A* Algorithm Family
  長所: 探索がヒューリスティックなため計算時間を削減できる
  短所: パスが連続でない．最適パスを見つけるのに時間がかかる
- State Lattice
  長所: 複数次元を取り扱える．動的なパスプランニングに適している．
  短所: 全ての解を評価しなければならないため，計算時間がかかる．
- RRT
  長所: 十分な時間があれば解が収束する．高次元において解が高速で見つかる．
  短所: パスが鋸歯状になる．
- Interpolating Curve Planners
  長所:CAGD という手法によって曲率と滑らかさに関しての最適化が行える．
  短所:Global Path Planning や Waypoints に依存している．動的な障害物がパス上に侵入した際に回避に時間がかかる．
- 関数最適化
  長所: 滑らかさや速さといった，パスや ego-vehicle に関する制限や他の車両についても容易に考慮に入れることが出来る．
  短所: 計算時間が他の手法に比べて長い．それゆえに Time Horizon（最適化する際に考慮する現在時刻から最も未来の時刻）に制限がある．

3.6 既存研究の課題

自動運転の研究に対する現状の課題を述べる．現状の Planning は一般的に Perception によって得た情報をもとに実行される．そして非決定的な手法を除き，周囲の車や自車両の速度や，道路の形状などの状況が同一であれば Planning の結果もほぼ同一になる．これによって Perception に問題がなく，Local Minimum などスタックに陥るような状況でなければ Planning は問題なく行われることが期待される．しかし現状の Planning は以下の問題を抱えている．

3.6.1 Perception の情報のみに依存

Perception で捉えきれない情報として，見通しが悪い路地と道路が合流していることや，子供が多く飛び出しの可能性が高いことや，路面に水が溜まりやすいことなどがある．これらの情報は地域に対して，個別に考慮すべきものである．しかしこのような Perception に反映されない情報を考慮した Planning は，車両の裏からの飛び出しを考慮するのみであるなど，非常に限定的な研究であることが多い．

3.6.2 細かい切り替えが想定されていない

地域によって運転の方針を細かく変えることは，事故の防止につながると考えられる．しかし現状の Planning は車間距離の詰め方などの，細かい部分での方針を想定していない．同じ直線での追い越しなどであっても，より細かく変更ができるべきである．

3.7 要求事項

本論文においては，地域に最適化された運転の実現を目標としている．既存研究においては，地域に最適化された運転の実現を目指したものは筆者が調べた限りでは存在しなかった．また運転の方針を場所によって細かく変更するという研究も少ない．そして既存研究の課題から手法を選ぶ際には以下に挙げる要求事項を満たすような手法を考える必要がある．

1. 特定の場所のみ運転方針を変更して，それ以外の場所では元に戻る．
2. 運転方針の変更は場所によって変更度合いを変えられる．

本研究では，歩行者は考慮せず他車両のみ考慮する．そこで車間距離をどれぐらい空けるかということについて焦点を当てる．車間距離を空けておくことによって緊急時の回避が可能になったり，また渋滞が発生しにくくなるというメリットがあるためである．また乗車者が快適に感じる運転は常に実現されるべきであるため，パスの滑らかさが損なわれないようにする．

各手法の短所と長所を踏まえて，次章においてこれらの要求事項を満たす手法を提案する．

第 4 章

提案手法

4.1 要求事項

前章にて述べた 2 つの要求事項については，連続的に運転方針を変えられるということに言い換えられる．この要求事項に対しては，実数パラメータによって運転の方針が決まることと，そのパラメータの持つ意味が明確であるような手法が望ましい．パラメータ依存の手法は関数最適化が当てはまる．そして，関数最適化の手法の中では APF に関してはパラメータが多く，またパラメータの持つ意味も多彩である．そのため本研究では要求事項を満たす手法として APF を採用する．加えて APF の走りを調節する方法として，APF のパラメータに対する遺伝的アルゴリズムを採用する．遺伝的アルゴリズムを採用した理由としては，遺伝的アルゴリズムはコスト関数の設計によって，何を最適化するかを決定できるためである．

4.2 提案手法にて用いる APF

今回使用した APF は関連研究の章で紹介した [23] である．この APF では車線境界線，車道の端，他車両，速度の四つの対象から APF を設ける．設ける APF を順番に $U_{\mathrm{lane},j}, U_{\mathrm{road},i}, U_{\mathrm{car},k}, U_{\mathrm{vel}}$ と置く，添え字 i, j, k はそれぞれの対象が複数存在する場合に，各々から APF を設けるためである．座標は (x, y) で表し，x は進行方向を表し，y はそれに直行する方向を指す．$U_{\mathrm{lane},j}$ は Gaussian 関数から作られている．

$$U_{\mathrm{lane},j} = A_{\mathrm{lane}} exp\left(-\frac{(y - y_{c,i})^2)}{2\sigma^2}\right) \tag{4.1}$$

この式において，$y_{c,i}$ は車道の端を除く，車線境界線の y 座標である．また σ はポテンシャルの増減の速さを決定し，車線幅に比例する形で決める．

$U_{\mathrm{road},i}$ に関しては車道から原則として車は外れてはいけないので，車道の端では無限大に発散するように設定する．

$$U_{\mathrm{road},i} = \frac{1}{2}\eta\left(\frac{1}{y - y_{0,j}}\right)^2 \tag{4.2}$$

η はスケーリング因子であり，$y_{0,j}$ は j 番目 $(j \in \{1, 2\})$ の端を表す．

$U_{\mathrm{car},m}$ に関しては他の車両と接触してはいけないことから，他の車両の近くでは無限大に発散する．そこで、物理学で用いられる湯川ポテンシャルを用いる [32].

$$U_{\mathrm{car},m}(K) = A_{\mathrm{car}}\frac{e^{-\alpha K}}{K} \tag{4.3}$$

ただし K は次のように定義される．ここで B は障害物を構成する点の集合である．q は Ego Vehicle（Path Planning を行う車両のこと）の点のうち，最も他車両に近い点である．元の論文では他車両の相対位置に

よって K を変更しているが，本研究ではそれは考慮しない．

$$K = \min_{b \in B} \|q - b\| \tag{4.4}$$

APF 全体を U_{total} とし，他の車両に仮想的にかかる力を F_{total} とする．このとき，

$$\begin{aligned} U_{\text{total}} &= \sum_i U_{\text{road},i} + \sum_j U_{\text{lane},j} + \sum_m U_{\text{car},m} \\ F_{\text{total}} &= -\nabla U_{\text{total}} \end{aligned} \tag{4.5}$$

このうち，他の車両との接触については $U_{\text{car},m}$ の項のみがかかわっている．

4.2.1 A_{car} に対するパラメータ最適化

$U_{\text{car},m}$ の項は A_{car} というパラメータを持っている．このパラメータの大きさは他の車両によって生じるポテンシャルフィールドによる力の大きさを表していると解釈できる．そのため，このパラメータを x 方向の関数として遺伝的アルゴリズムを用いて最適化する．各車両に対して独立して最適化を行う方法も考えられるが，本研究では A_{car} を変化させる際には，全車両の A_{car} を変化させるに留める． これは解が非常に多くなる可能性があることと，少しの状況の違いで解が大きく変わる可能性があるためである．

まず，この研究においては 3 つの新たなパラメータ (a_0, a_1, a_2) を用いて，A_{car} を最適化していく．ここで．a_0 を含めて計算を進めた場合には探索空間が広すぎるため，a_0 はハイパーパラメータとする．そして，A_{car} は事前に定められた区間 $[a-b, a+b]$ のみ最適化を行う．この区間の意味としては以前に事故が起こった場所の周辺や，事故が起こりやすい区間などを想定している．A_{car} は各区間において以下のような値を取る．

$$A_{\text{car}} = \begin{cases} \text{const} & (x < a-b\,,\ x > a+b) \\ a_2(x-a)^2 + a_1(x-a) + a_0 & (a-b \le x \le a) \\ a_2(a-x)^2 + a_1(a-x) + a_0 & (a < x \le a+b) \end{cases} \tag{4.6}$$

A_{car} は $x = a$ に対して対称な値を取る．ここで，三次以上の項を用いなかった理由は，A_{car} は安全性の観点から緩やかに変化させるべきであることが挙げられる．すなわち変化率を小さく保つべきである．急激に A_{car} を変化させた場合，車両に仮想的にかかる APF が急激に変わり，車両は振動すると考えられる．三次以上の項を設けた場合，A_{car} が急激に変化する懸念から，三次以上の項は式 4.6 から除外した．

4.2.2 コスト関数

$\mathbf{a} = (a_1, a_2)$ を最適化していくに当たって，コスト関数を考える必要がある．コスト関数の作成に当たっては，Path Planning という問題全体を関数最適化（パラメータが連続的な値を取る実数値関数の最適化）に置き換えたものと，遺伝的アルゴリズムを用いてパス生成を行った研究を参考にする [29] [33]. [29] では，以下の要素がコスト関数に含まれている

1. 躍度の変位（躍度: 加速度の時間微分）
2. 加速度の変位
3. 速度の変位
4. ヨー角度の微分の変位
5. 車線から車両の重心までのオフセット距離

[33] では以下の要素が含まれている

1. パスの滑らかさ

2. パス上の点から最も近い障害物までの距離

これらの要素のうち考えるべきものを絞る．今回の研究では車両の速度は一定である．また車線から車両の重心までのオフセット距離は Road Potential によって調整されるので，他の車両による APF で調整すべきではない．またヨー角度の微分の変位とはパスの滑らかさを考慮していることになる．そのため以下の三つの要素を考える

A). 他の障害物（今回の場合は車両）までの距離の最小値
B). ヨー角度のサンプル時間ごとの差分をとったものの総和
C). y 方向のサンプル時間ごとの差分をとったものの総和

パスの滑らかさは速度が大きく，有人走行である自動運転において非常に重要な要素である．そのため，ヨー角度だけではなく y 方向の差分に関しても考慮に入れた．また本研究では安全性を考慮したパラメータ最適化を行うため，他の障害物までの距離を考慮したものを考えた．

この三つの要素を足し合わせたものを $J(\mathbf{a})$ とおく．$J(\mathbf{a})$ の具体的な表式を記述するために，変数を以下のように定義する

- δt : サンプル時間の幅
- N_T : $\delta t * N_t$ が車両が目的地についた時刻となるような整数
- M : 障害物（ここでは他の車両）の個数
- P_i : 車両の時刻 $t_i = \delta_t * i$（i 番目のサンプル時間を指す）における座標の位置
- O_i^j : 障害物 j の時刻 t_i における座標の位置
- ψ_i : 時刻 t_i における車体の角度

このとき，(A),(B),(C) の要素は順に次のように表される．

$$\min_i \min_j d\left(P_i, O_i^j\right)^2 \quad \left(i = 0, 1, \cdots, N_T \quad j = 1, 2, \cdots, M\right) \tag{4.7}$$

$$\sum_{i=0}^{N_T-1} \left(\psi_{i+1} - \psi_i\right)^2 \tag{4.8}$$

$$\sum_{i=0}^{N_T-1} \left(y_{i+1} - y_i\right)^2 \tag{4.9}$$

.

よって $J(\mathbf{a})$ は次のように表される

$$J(\mathbf{a}) = w_1 \min_i \min_j d\left(P_i, O_i^j\right)^2 + w_2 \sum_{i=0}^{N_T-1} \left(\psi_{i+1} - \psi_i\right)^2 + w_3 \sum_{i=0}^{N_T-1} \left(y_{i+1} - y_i\right)^2 \tag{4.10}$$

ただし，w_i はパラメータを最適化せずに車両を走らせたときの，三つの項の値の逆数である．そのため，パラメータを最適化せずに車両を走らせた場合，$J(\mathbf{a}) = 1 + 1 + 1 = 3$ となる．

このことからコスト関数を計算するためには，走行中に次の情報を保存する必要がある．

- 各時刻における y 座標
- 各時刻における車体の角度
- 各時刻における他の車までの距離

4.2.3 遺伝的アルゴリズム

a は実数パラメータであるため，遺伝子型が実数であっても適用できるような実数型遺伝的アルゴリズムを用いる必要がある．実数型遺伝的アルゴリズムの中で代表的なものは，単峰性正規分布交叉 (Unimodal Normal Distribution Crossover;UNDX) [34] の手法を多親交叉へと発展させた UNDX–m と，シンプレックス交叉（Simplex Crossover;SPX）[35] である．

今回は SPX を用いる．理由を説明する前に，遺伝的アルゴリズムにおいて考慮すべき性質について説明する．関数最適化を行う遺伝的アルゴリズムは変数間依存性と悪スケール性と呼ばれる性質を持つことがある [36]．変数間依存性は「複数の変数を同時に変化させなければ目的関数の改善を効率的に行えないような性質」のことである．また悪スケール性（ill-scale）は「座標系のスケールの取り方が異なるために，目的関数に対する感度が変数によって大きく異なる性質」のことである．

悪スケール性の影響を受けにくいが，変数間依存性を持つ BLX-α という手法がある．そして SPX は BLX-α 変数間依存性を改善することを目的としたものである．現在では SPX をさらに改善した手法も産まれているが，今回はこれらを用いない．

または UNDX という手法は変数間依存性の影響を受けにくいが，悪スケール性を持つ．これを改善したものが UNDX-m である，今回は二次項に関するパラメータと一次項に関するパラメータを最適化するため，悪スケール性の影響が大きいと考えられる．そのため本研究では SPX を用いる．

SPX は以下のようなアルゴリズムである．探索空間の次元数を n とおく．

1. n 個の親個体を選択する．
2. 親個体の重心 $\mathbf{G}$ を計算する
$$\mathbf{G} = \sum_{i=0}^{n} \mathbf{P}_i \tag{4.11}$$
3. そして，$\mathbf{x}_k$ と $\mathbf{C}_k$ を次のように定める．
$$\begin{aligned} \mathbf{x}_k &= \mathbf{G} + \epsilon(\mathbf{P}_k - \mathbf{G}) \quad (k = 0, \cdots, n) \\ \mathbf{C}_k &= \begin{cases} 0 & (k = 0) \\ r_{k-1}(\mathbf{x}_{k-1} - \mathbf{x}_k \mathbf{C}_{k-1}) & (k = 1, \cdots, n) \end{cases} \end{aligned} \tag{4.12}$$
ここで，ϵ は正のパラメータで拡張率と呼ぶ．通常 $\epsilon = \sqrt{n+2}$ と設定する．本研究においてもそれに従う．また r_k は区間 $[0,1]$ の一様分布乱数 $u(0,1)$ を次式で変換して得られる乱数である．
$$r_k = (u(0,1))^{\frac{1}{k+1}} \tag{4.13}$$
ただし計算の便宜上，これ以外の範囲の k に対して次の定義をしておく．
$$r_k = \begin{cases} 0 & (k < 0) \\ 1 & (k \geq n) \end{cases} \tag{4.14}$$
4. 子個体 $\mathbf{C}$ を次式によって得る．
$$\mathbf{C} = \mathbf{x}_n + \mathbf{C}_n \tag{4.15}$$

4.2.4 リアルタイム性の問題

この手法では遺伝的アルゴリズムを用いてパラメータを計算する必要がある．本研究では直線路に対して提案手法を用いるため，車両の速さが非常に大きい．また遺伝子に対して走行を評価するために，遺伝子の数だけ車両走行のシミュレーションを行う必要がある．車両が高速で移動すること，特定の場所で運転方針を変化させること，また計算量が多いことからリアルタイムで計算を行うことが難しい．

4.2.5 リアルタイム性の解決のための RSU

本研究では遺伝的アルゴリズムによって運転の最適化が行われるが，リアルタイムでパラメータを最適化するのは難しい．そのため，**a** を様々な状況に対して事前に計算しておき，路側に設けられたコンピュータである Road Side Unit(RSU) に保存しておくことが考えられる．そして類似した状況が表れた場合，事前に計算した **a** を車両へ送信する．つまり状況とパラメータの対応を RSU に保存する．状況の「近さ」の表現方法だが，これは車両自身の速度，Ego Vehicle から見た他車両の相対速度，他車両の相対位置の三つを考慮に入れるべきだろう．本研究では，事前に計算した状況と，初期状態が全く同じ状況が実現されたという理想的な状態を仮定して実験をする．そのため同一の (a_0, a_1, a_2) を渡した場合，全く同じ走行をすることになる．

関連研究の章にて紹介した LSTM の研究では，事前に交通データを学習しておき，予測を Risk Assessment に反映するという手法を取っていた [27]．本研究においても RSU に事前学習の結果を保持しておき，車両から送信されたデータをもとに，即座に適切なパラメータを返すという手法が有効であると考えられる．

第 5 章

実装

本研究では MPC を用いた最適化を行う APF に対して，遺伝的アルゴリズムを用いるという実装を行う．そのために，Simulink/matlab に対して遺伝的アルゴリズムを行うような拡張を行う必要がある．

5.1 実装の概要

本研究で実装したシステムのうち，APF の全体図は図 5.1 にて示す．遺伝的アルゴリズムも含めた全体図は図 5.2 にて示す．本研究では，同じシナリオに対して任意回シミュレーションを行えるようにする必要がある．そのため，一度車両が走行を終えた後に，そのパラメータに対して評価が可能になるよう必要なデータを取得する必要がある．取得する必要のあるデータは以下である．

- 各時刻における y 座標
- 各時刻における車体の角度
- 各時刻における他の車までの距離

APF による自動運転の流れとしては以下のようになる．

1. APF によって Waypoint を生成.
2. waypoint の補完を行う三次曲線の係数を求める．(Curve Fitting)
3. 求めた係数から $\mathbf{x}$ の現時点での $\mathbf{x}_{\text{state}}$ を求める．
4. 前もって求めているフィードバックゲイン K_{gain} によって，Bicycle Model を用いて入力ベクトル $\mathbf{u}$ を求める
5. $\mathbf{u}$ によって次の時刻の車の位置，角度を確定する．
6. ここまでの操作を 1 から目的地にたどり着くまで繰り返す.

そして，これを一回として遺伝的アルゴリズムを使う．

5.2 変数の定義

この章にて出てくる変数に対しての定義は以下である．

- XYZ: グローバル座標系
- xyz: 車における座標系
- ψ: グローバル座標系における車体のヨー角度

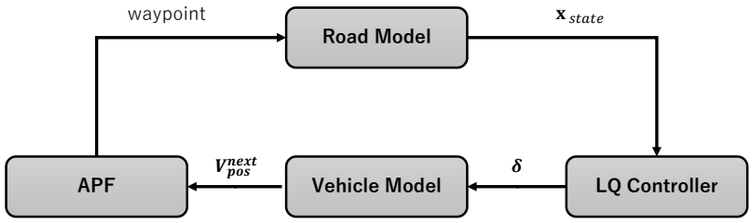


図 5.1: APF の全体図

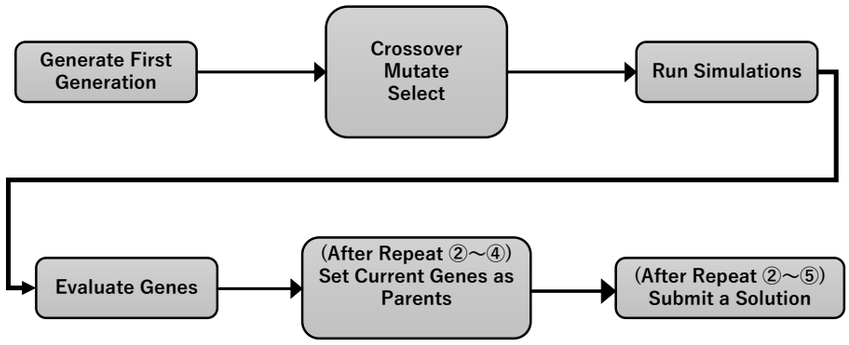


図 5.2: 遺伝的アルゴリズムの全体図

- V_x : 重心における車両の進行方向の速度
- V_y : 道路方向と直行する方向の速さ
- α : (α_f, α_r) 前輪，後輪のスリップ角
- C_α : $(C_{\alpha f}, C_{\alpha r})$ 前輪，後輪におけるコーナリング係数
- l : (l_f, l_r, l_w, l_{fr})　順番に前輪，後輪から車両の重心までの進行方向に関する距離，車両の幅，車両の長さ
- δ_f : 前輪の角度
- m : 車両の質量
- I_z : ヨー角に関する慣性モーメント
- T_s : サンプル時間同士の最少の差分

- L: 前方注視距離．角速度コマンドを計算するために、車両が現在の位置からパスに沿って読み取る距離のこと
- $e_y = y - y^d$: 参照している点からの y 方向の誤差
- e_{yL} : x を前方注視距離としたときの補完多項式の点の y 座標から，現在車体が向いている方向の直線までの距離.
- e_ψ : 道路に対する角度の誤差

5.2.1 Curve Fitting から $\mathbf{x}_{\text{state}}$ の算出

この小節の論理展開は [24] に従う．生成された waypoint はそのままではヨー角度の変化が急激になってしまう可能性がある．そのため，多項式を使って waypoint の間を補完する必要がある．これは以下の多項式によって行われる．

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \tag{5.1}$$

c_i を求める手順を述べる．ここから大文字で変数 XYZ を書く場合にはグローバル座標系を表し，小文字 xyz で書く場合には，ローカル座標系を表すことに注意．まず N 個の点（$N > 4$），P_i は次のように表される．

$$P_i = \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (i = 1, 2, \cdots, N) \tag{5.2}$$

そして，この P_i をローカル座標系に変換する．これはアフィン変換を用いる．

$$\begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\theta\text{だけ回転する}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{(d_x, d_y)\text{ だけ移動する}} \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix} \tag{5.3}$$

ただし，$\theta = -\psi, d_x = -X_{\text{Ego}}, d_y = -Y_{\text{Ego}}$

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix}}_{\mathbf{F}} \tag{5.4}$$

ここでベクトル $\mathbf{Y}$ を次のように定める

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{5.5}$$

ここで，$\mathbf{c}^*$ を最適解とし次の式に最小二乗法を用いることによって，$\mathbf{c}^*$ を求める

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} J(\mathbf{c}) \tag{5.6}$$

ここで $J(\mathbf{c}) = \frac{1}{2}(\mathbf{X}\mathbf{c} - \mathbf{Y})^\top W (\mathbf{X}\mathbf{c} - \mathbf{Y})$ である，ただし，$W = \text{diag}(w_1, w_2, \cdots, w_N) > 0$, $(N > 4)$．これによって，最適解 $\mathbf{c}^*$ は次のようになる

$$\mathbf{c}^* = \left(\mathbf{X}^\top W \mathbf{X}\right)^{-1} \mathbf{X}^\top W \mathbf{Y}. \tag{5.7}$$

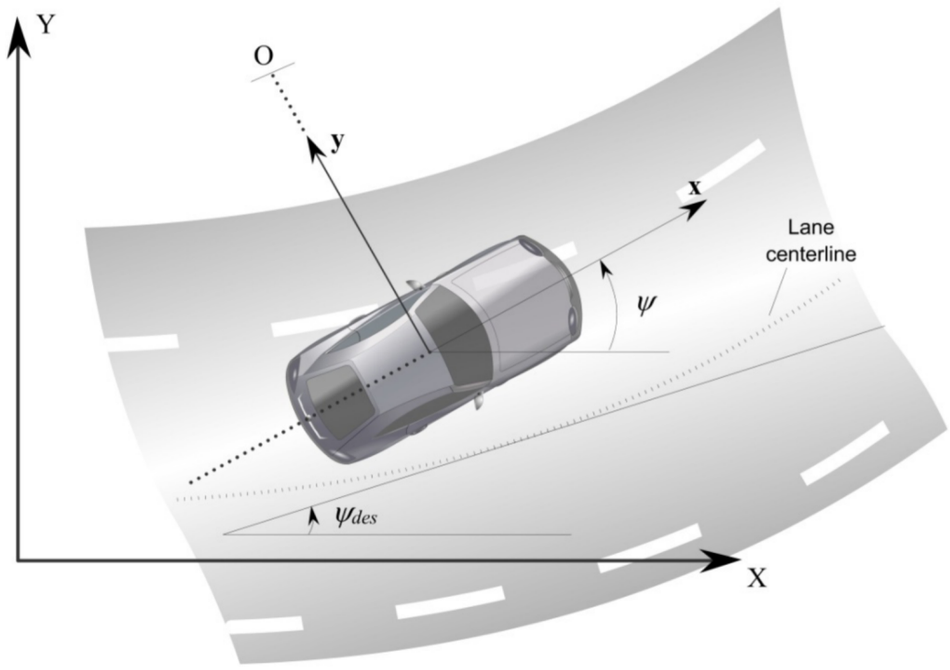


図 5.3: Lateral vehicle dynamics（出典: [37]）

この式 5.7 を用いることによって，Bicycle Model の $e_{yL}, \dot{e}_y, e_\psi$ を求めることができる．これは次のような式によって導かれる．L は先読み距離である．

$$e_{yL} = f(L) = c_0 + c_1 L + c_2 L^2 + c_3 L^3 \quad \text{[m]} \tag{5.8}$$

$$\dot{e}_y = -V_y + V_x e_\psi \approx -\frac{l_r}{l_f + l_r} V_x \delta_f + V_x e_\psi \quad \text{[m/s]} \tag{5.9}$$

$$e_\psi = \left.\frac{d}{dx}\right|_{x=0} f(x) = c_1 \quad \text{[rad]} \tag{5.10}$$

$\dot{\psi}$ は慣性計測装置（Inertial Measurement Unit; IMU）を使用すれば求めることが出来る．

5.3 Vehicle model

$\mathbf{x}_{\text{state}}$ を求められたので，モデルを用いてモデル予測制御を行う．（モデル予測制御とはサンプル時間ごとに再計算を行い，未来の状態を予測して最適化を行う制御手法である [38].）本研究では Bicycle model と呼ばれるモデルを用いる．これは車が高速であるときに用いるモデルである [37]．車両が高速である場合，前輪後輪それぞれの速度の向きはその車輪の向きとは一致しない．その仮定をモデルに組み込んだのが Bicycle Model である．

この Model においては図 5.3 のように，横方向 (y) と車体角度 (ψ) を内部状態とする．そして現在の状態から推定した未来の状態の誤差に基づいて，前輪の車輪の角度 u を制御する．ただし本研究では y 方向そのものではなく，先読み誤差 e_{yL} などを用いる．

目標はフィードバックゲイン K_{gain} を求めることである．フィードバックゲインとは，状態ベクトルの目標値に対する誤差に対して，どの程度入力をするかを決めるベクトルである．

表 5.1: 変数の値

変数	値
C_f	66900
C_r	62700
m	1270 kg
I_z	1536.7 Nm
l_f	1.015 m
l_r	1.895 m
V_x	20 m/s

状態ベクトルを $\mathbf{x}$ とおくと $\mathbf{x}$ は次のようになる．

$$\mathbf{x} = \begin{bmatrix} e_{yL} \\ \dot{e}_y \\ e_\psi \\ \dot{\psi} \end{bmatrix} \tag{5.11}$$

Bicycle Model は以下のような式になる．

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}-l_f-2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2l_fC_{\alpha f}-2l_rC_{\alpha r}}{I_zV_x} & 0 & -\frac{2{l_f}^2C_{\alpha f}+2{l_r}^2C_{\alpha r}}{I_zV_x} \end{bmatrix}}_{A} \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2l_fC_{\alpha f}}{I_z} \end{bmatrix}}_{B} \delta \tag{5.12}$$

今回の実験では V_x, I_z は定数であるため，A, B は定数値行列とみなせる．これらの定数に割り当てられた値は表 5.1 にて記載．

そして，フィードバックゲインは線形最適制御（Linear Quadratic Regulator;LQR）を使って求める．LQR から導かれるリッカチ方程式を解くことによって，フィードバックゲインは以下の数値になる．

$$K_{\text{gain}} = \begin{bmatrix} -0.0372 \\ 0.0372 \\ -0.8685 \\ 0.0293 \end{bmatrix} \tag{5.13}$$

このモデルによって，車が向かう角度を適切に修正できる．第一成分，第二成分を現在の車両の位置として，第三成分を現在の車の角度とする変数を $\mathbf{V}_{\text{pos}}$ と表す．次のサンプル時間後も同様の変数 $\mathbf{V}_{\text{pos}}^{\text{next}}$ を考える．$\mathbf{V}_{\text{pos}}^{next}$ は以下の式によって求められる [39].

$$\mathbf{V}_{\text{pos}}^{\text{next}} = \mathbf{V}_{\text{pos}} + T_s v_{\text{vech}} \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{1}{l_{fr}\tan\delta} \end{bmatrix} \tag{5.14}$$

ただし現在時刻が初期値の場合には，$\mathbf{V}_{\text{pos}}^{\text{next}}$ を初期位置 $\mathbf{V}_{\text{origin}}$ に設定する．

5.4 遺伝的アルゴリズム

この小節までに述べたことによって，車両の走行テストが可能である．次に繰り返しの走行によって，APF に対する遺伝的アルゴリズムを実装する．これは生成した遺伝子を APF を生成している部分に送り込み，シ

Algorithm 1 親世代から子世代を生成する

Require: $parents \leftarrow$ 親世代の遺伝子

$DIM \leftarrow$ パラメータの数
$N \leftarrow$ 各世代の遺伝子の数
$M \leftarrow N/2$
$selected \leftarrow$ 上位 50% の遺伝子のリスト
for i=1:(M/3) **do**
  $random \leftarrow selected$ から既選択のものと重複しないように 3 個体選択
  $child \leftarrow crossover(random)$ （crossover は SPX 交叉を行う）
end for
$result \leftarrow selected$ に $child$ を一つにまとめる

Algorithm 2 遺伝的アルゴリズムの全体

Require: $GENERATION \leftarrow$ 最大世代数

$genes \leftarrow M$ 個だけ乱数によって遺伝子を作成
for 1:GENERATION **do**
  $genes \leftarrow$ Algorithm 1 によって，親と子をまとめたリストを生成する
  for gene = i 番目の遺伝子 **do**
    if $gene$ が未評価 **then**
      $gene$ に基づき走行させる
      $gene$ に走行データから算出したコスト関数を付加する
    end if
  end for
end for

ミュレーションを行い，その結果を保存することによって可能である．遺伝的アルゴリズムにおいては各遺伝子のコスト関数を算出する必要がある．コスト関数の算出のためには以下のデータを持っておく必要がある．

1. 各時刻における Ego Vehicle の y 座標
2. 各時刻における Ego Vehicle の角度
3. 各時刻におけるすべての車両の座標

本研究では遺伝的アルゴリズムのはエリート方式を用いる．上位 50% の遺伝子を複製し，それらを下位 50% の遺伝子とする．そして下位 50% の遺伝子に対して，遺伝子交叉を行い，それらを一定確率で突然変異させる．親世代から子世代を生成する擬似コードは Algorithm 1 である．この Algorithm 1 を用いて，任意の世代数だけ遺伝的アルゴリズムを続けることのできるプログラムを組む．この擬似コードは 2 にて記載する．コスト関数の値が評価されていない遺伝子が来るたびに，前章までで述べた実装にしたがい，車両の走行シミュレーションを行う．そしてコスト関数の値を計算する．

第 6 章

実験・評価

6.1 実験環境

最初に実験環境について述べる．使用した PC は一台である．使用した PC のスペックを表 6.1 にて示す．

また利用したソフトウェアは MATLAB/Simulink であり，MATLAB のバージョンは R2021a. Simulink は 10.3 を使用した．

6.2 実験概要

本研究では遺伝的アルゴリズムによる学習の結果の確認と，最も適合度の高かった解と最適化を行わなかった場合との比較が目的である．後半部の適合度の高かった解を使って自動運転を行う実験は，RSU に事前に計算結果を保存しておき，車両にパブリッシュする実験に当たる．

今回の研究では MATLAB のみを用いて，実機での実験は行わない．実験に関わる重要な変数は前章の表 5.1 にてまとめる．

今回測定した項目は以下である．

- Ego-Vehicle の軌跡
- Ego-Vehicle の向いている向き
- Ego-Vehicle の車輪の向き
- Time Horizon（ここでは $H_p = 7$）までのパスの添え字で並べる
- 曲率
- 遺伝的アルゴリズムを使って最終的な解を算出するまでに経過した計算時間

実験のシナリオは以下の通りである．

- 直線路における車 1 台の追い越し

この実験では適切に運転の方針を切り替えることができるかどうかを確認する．

6.3 車 1 台の追い越しのシナリオ

6.3.1 初期条件

まず Ego Vehicle を $x = 0$ から $x = 400$ まで走らせるシナリオで実験をした．Ego Vehicle の前方には他の車両がある．各車両の初期状態は表 6.2 である．速さは定数であることに注意．

表 6.1: 使用した PC のスペック

項目	PC
CPU	Core(TM) i9-10980HK CPU @ 2.40GHz 3.10 GHz
GPU1	Intel(R) UHD Graphics
GPU2	NVIDIA GeForce RTX3080 Laptop GPU
RAM	64.0 GB
OS	Windows 10 Pro

表 6.2: 初期値

項目	ego	other1
初期速さ	25 m/s	15 m/s
初期位置	(0,5)	(100,4)

表 6.3: 遺伝的アルゴリズムの設定

項目	値
最大世代数	20
各世代の個体数	18
遺伝子の値の範囲	$\left(-\frac{1}{100}, -\frac{1}{100}\right) \leq (a_2, a_1) \leq \left(\frac{1}{100}, \frac{1}{100}\right)$
a_0	20

また遺伝的アルゴリズムにおける諸条件は表 6.3 である.

6.3.2 遺伝的アルゴリズムの結果

この遺伝的アルゴリズムの計算が終わるまでの時間は 316.0 s であった.

最終世代の遺伝子

まず遺伝的アルゴリズムの最終世代の遺伝子の値をまとめる. 最終世代での遺伝子は表 6.4 のようになった.

上位遺伝子のスコアの遷移

次に各世代における上位 50 %（18 個体のうち上位 9 個体）のスコアの平均の推移は図 6.1 のようになる. 第 1,2 世代を含めると 3 世代以降の変化を見るのが難しくなるため, 3 世代目以降のみデータを取っている.

上位 3 つの遺伝子に対するパス

上位三つの遺伝子に対するパスを見る.

図 6.2 から読み取れることとしては, パスはほぼ同じものに収束していることである. そのためパラメータがほぼ同一ならば, パスはほぼ同一になることが推測される. そして一次項は二次項ほどは大きな影響を与えないことが分かる.

表 6.4: 最終世代の遺伝子の値とコスト関数の値

a_2	a_1	コスト関数の値
0.002265	0.013331	2.069786
0.002227	0.010906	2.073265
0.00224	0.010408	2.073973
0.002259	0.010306	2.074326
0.002262	0.010292	2.074393
0.002243	0.01	2.074606
0.002251	0.01	2.074698
0.002296	0.01	2.075866
0.002315	0.01	2.076697
0.002214	0.007915	2.077521
0.001524	0.01	2.176521
0.002922	0.009962	2.210188
-0.000326	0.01	2.582876
0.003752	0.010452	2.690808
-0.001841	0.010287	2.758083
0.007681	0.010296	1413.536166
0.005913	0.006833	4995.803547
0.005501	0.01	61577.32037

6.3.3 最適化をしない場合の走行

図 6.3 にパス，車体の向き，そして予測パスを並べたものをまとめた．予測パスを並べたものとは，このモデルでは Time Horizon を 7 として設定しており，一度に 7 つ先の Waypoint まで生成する．それらの点を $p_1, p_2, \cdots, p_7$ としておき，各 p_i を配列 P_i（はじめ P_i は空配列）の末尾に加えておく．そして (x, y) 平面上に各 P_i の要素を並べたものが，図 6.7c である．この図によって予測と実際に走ったパスが，どれほど異なっていたのかを知ることが出来る．

6.3.4 最も適応度の高い遺伝子の場合の走行

次に $(a_2, a_1) = (0.002265, 0.01331)$ とした場合の走行を示す．

6.3.5 最適化を行った場合と行っていない場合の走行の比較

この小節では二つの走行を比較する．二種類の運転を特徴づけるパラメータとその値は表 6.5 にて記載．

二つの運転を見ると，最大曲率は最適化を行わない場合の方が小さく，他の車両まで最も接近したときの距離は最適化を行った場合の方が小さい．（ego-vehicle のパスは縦軸の表示範囲が異なる．）また二つの運転のグラフを比較する．（表 6.3,6.4）最適化を行った場合の運転は滑らかでない箇所がある．特に $x = 300$ 周辺においては最適化行っていない場合に比べ，違いが顕著である．

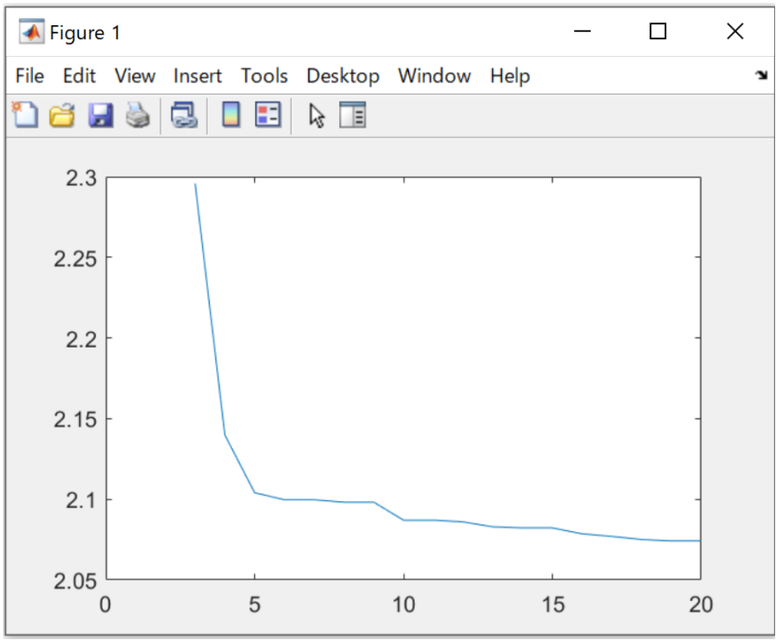
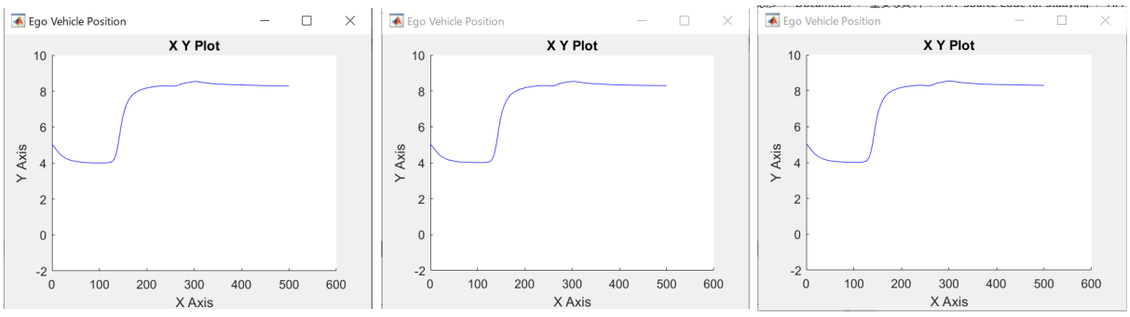


図 6.1: 上位 50% の個体のスコア平均の推移（横軸は世代数）



(a) 一番目の遺伝子 **(b)** 二番目の遺伝子 **(c)** 三番目の遺伝子

図 6.2: 上位三つの遺伝子のパス

表 6.5: 二種類の運転の比較

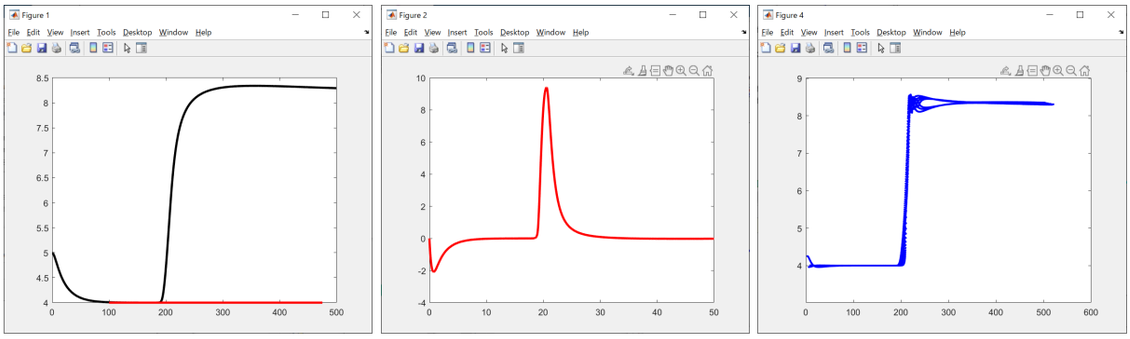
項目	最適化なし	最適化あり
最大曲率	43.4098	66.0222
他の車までの最少距離 [m]	18.7633	18.8878

6.3.6 より滑らかさに対する条件を強めた場合

コスト関数を以下のように変更し，経路がより滑らかになるようにする．

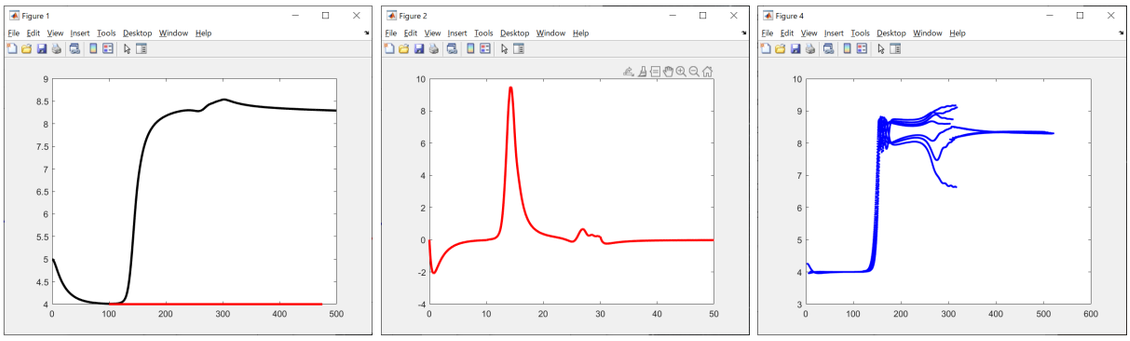
$$J(\mathbf{a}) = w_1 \min_i \min_j d\left(P_i, O_i^j\right)^2 + 6w_2 \sum_{i=0}^{N_T-1} \left(\psi_{i+1} - \psi_i\right)^2 \tag{6.1}$$

車体角度の変更に対して，大きく重みづけしている．初期条件を先ほどの実験と同一のものとして，同じ実験を行う．



(a) パス (黒線が Ego-Vehicle) **(b)** 車体の向き **(c)** 七つの予測パス

図 6.3: 最適化を行わない（$A_{\mathrm{car}} = 10$）場合のデータ



(a) パス (黒線が Ego-Vehicle) **(b)** 車体の向き **(c)** 七つの予測パス

図 6.4: 最も適応度の高い遺伝子の場合の走行

6.3.7 遺伝的アルゴリズムの結果

この遺伝的アルゴリズムの計算が終わるまでの時間は 300.9 s であった.

最終世代の遺伝子

まず遺伝的アルゴリズムの最終世代の遺伝子の値をまとめる．最終世代での遺伝子は表 6.6 のようになった.

上位遺伝子のスコアの遷移

次に各世代における上位 50 %（18 個体のうち上位 9 個体）のスコアの平均の推移は図 6.5 のようになる．第 1,2 世代を含めると 3 世代以降の変化を見るのが難しくなるため，3 世代目以降のみデータを取っている.

6.3.8 最も適応度の高い遺伝子の走行と，コスト関数を変更する前の走行との比較

次に $(a_2, a_1) = (0.002060, 0.008094)$ とした場合と，$(a_2, a_1) = (0.002265, 0.01331)$ とした場合での比較を行う．パス，車体の向き，予測したパスについては図 6.6 と，図 6.7 にて示す．また曲率と最小距離に関しては表 6.7 にて示す.

二つの走行を比較すると，図に現れる範囲ではあまり違いは見られない．また曲率と最小距離に関しては，コスト関数を改変した後の方が，わずかに曲率が低くなっており，最小距離に関しては大きな変化はない.

表 6.6: 最終世代の遺伝子の値とコスト関数の値

a_2	a_1	コスト関数の値
0.00206	0.008094	5.663608
0.002057	0.002934	5.6638
0.002045	0.003817	5.664797
0.002136	0.007226	5.665994
0.002027	0.01	5.666842
0.002022	0.002197	5.667497
0.002021	0.004624	5.667664
0.00202	0.01	5.667892
0.002006	0.006106	5.670208
0.002004	0.005475	5.670566
0.002004	-0.002655	5.670625
0.002171	-0.002276	5.67194
0.002597	0.01	5.991568
0.000017	-0.01	7.65857
-0.00023	0.006094	7.873326
0.00421	0.008585	11.095031
0.004263	0.01	11.3112
-0.005833	0.006726	459022.7062

表 6.7: 二種類の運転の比較

項目	コスト関数改変前	コスト関数改変後
最大曲率	66.0222	64.7352
他の車までの最少距離 [m]	18.8878	18.8825

6.4 考察

6.4.1 最適化を行ったものと行わなかったものの比較

こちらはパスの滑らかさが最適化を行ったものと比較して損なわれていたが，車間距離は僅かに大きく取れていた．滑らかさが損なわれた理由としては以下が考えられる．

1. 車間距離を大きくとることによってコスト関数の値が小さくなるために，多少パスの滑らかさが失われても車間距離を大きくとる遺伝子が上位になってしまっている．コスト関数の性質上，複数の要素がある場合，いくつかの要素が多少大きくなっても，他の要素が小さければ，上位の遺伝子となる．
2. A_{car} の変動のさせ方が不適切．パラメータの変化が急激であったり，A_{car} の値が大きいためにパスが滑らかでなくなった．

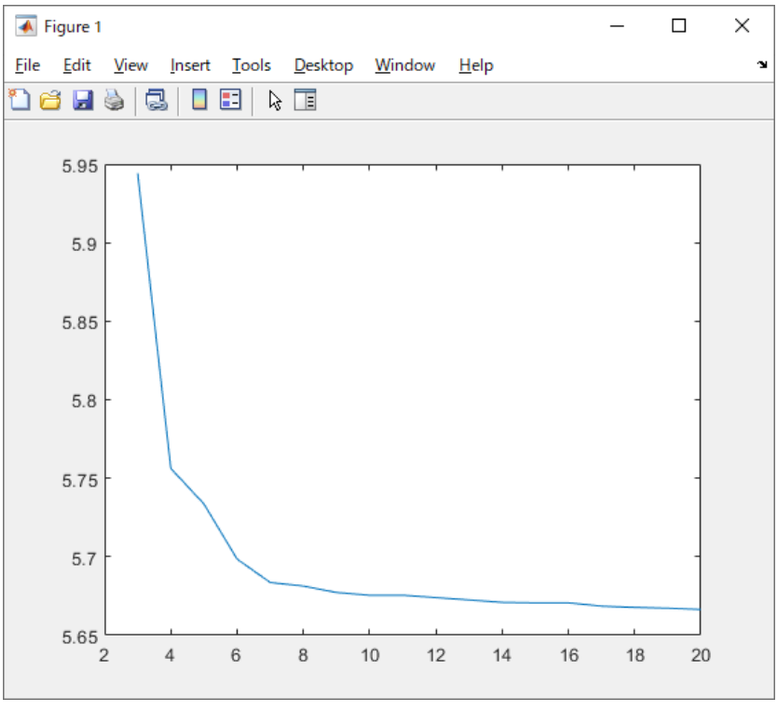
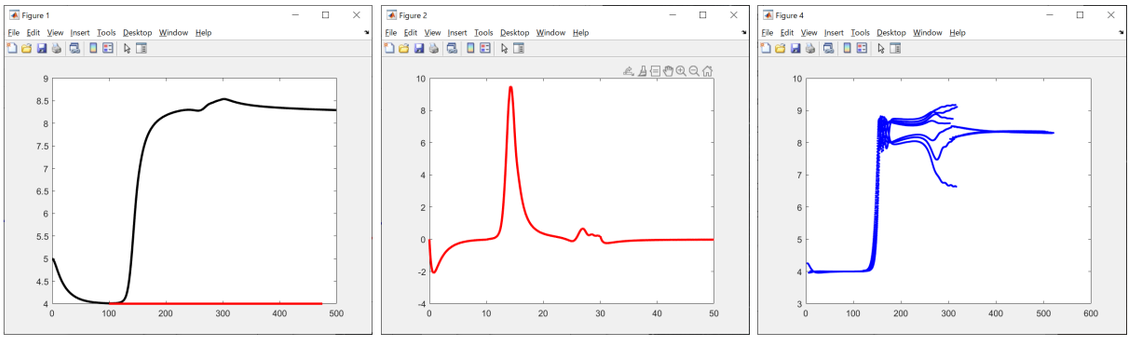


図 6.5: 上位 50% の個体のスコア平均の推移（横軸は世代数）

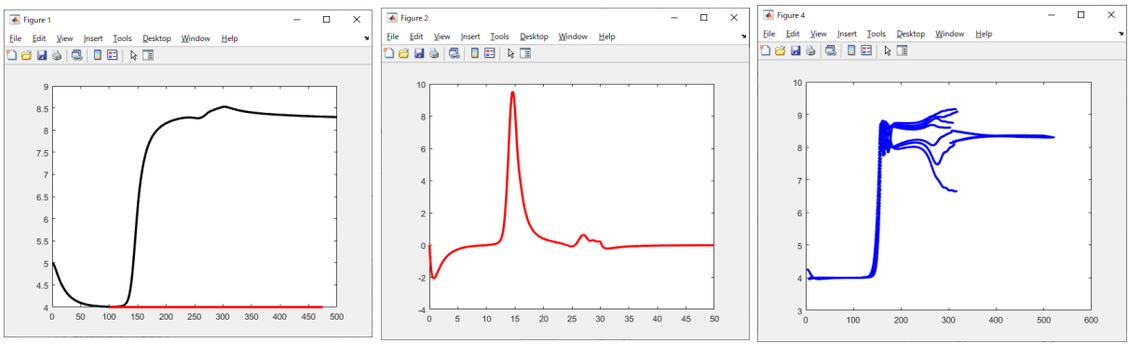
6.4.2 コスト関数を改変する前後の比較

滑らかさをより重視したコスト関数を作成し，比較を行った．しかしコスト関数を改変した前後で，あまり走行に変化は見られなかった．これは放物線によって最適化した部分に問題があった可能性がある．二つのパラメータのうち a_2 の方が a_1 よりも最適化に与える影響は大きい．しかし，二つの走行を比較すると，a_2 の値は変更前が 0.002265 となり，変更後が 0.002060 となっており，大きな差がない．ここから放物線による表現能力に限界があったために，最適化が上手くいかなかった可能性が考えられる．



(a) パス (黒線が Ego-Vehicle) **(b)** 車体の向き **(c)** 七つの予測パス

図 6.6: コスト関数を変更する前の走行



(a) パス (黒線が Ego-Vehicle) **(b)** 車体の向き **(c)** 七つの予測パス

図 6.7: より滑らかさを考慮したコスト関数の場合の走行

第 7 章

結論

7.1 まとめ

本研究では，自動運転において地域に最適化された Planning の必要性を述べた．自動運転の大半の Planning は Perception の情報に基づいているが，地域ごとに最適な運転が異なることに注目し，地域ごとの最適化を目指した研究が存在していなかった．その解決策として本論文では APF の活用を提案した．特に APF の特性のうち，パラメータの変更によって走りが大きく変わること，パラメータが多くパラメータごとの役割が明確なことに着目し，パラメータ最適化によって，車間距離を最適化する以前よりも広く保ち，滑らかさを同程度に保ち，局所的により安全な運転を実現することを目指した．具体的な手法としては APF のパラメータを遺伝的アルゴリズムを使うことによって，リアルタイムでパラメータを二次関数に沿って変更し，コスト関数に応じたパラメータ最適化を行った．

提案手法の実装のため，APF を MATLAB/Simulink を用いて実装した後，任意回車両の走行が可能なフレームワークを作成し，遺伝的アルゴリズムによる学習を可能にした．

実験では，1 台の他車両がある場合に対して実験を行い，パラメーター最適化の結果を反映した場合と，反映していない場合との比較を行った．またコスト関数をより走行が滑らかになるように変更を行った．最適化を行った場合と，そうでない場合を比較すると，パラメーター最適化の結果を反映した場合は，他車両に対して僅かに大きく距離を取る運転が実現できた．車間距離の最小距離は最適化を行わない場合は最適化を行った場合に対して， 0.66% だけ距離を大きく取ることが出来た．そしてコスト関数を改変した前後では走行に顕著な違いが見られなかった．

7.2 今後の課題

本研究では車両一台の場合，コスト関数を変更していても最大曲率は最適化を行った場合 152% に増加しており，滑らかさが低減していた．今後の課題としては以下のものが考えられる．

1. 放物線以外のものを使って，最適化を行う
2. APF 以外の手法に対して遺伝的アルゴリズムを用いた最適化を行う

References

[1] 人とクルマはチームメイト－自動運転の実現に向けて | トップメッセージ | 企業情報 | トヨタ自動車株式会社公式企業サイト. https://global.toyota/jp/company/messages-from-executives/details/pratt.html, 12 2019. (Accessed on 11/22/2021).

[2] 国土交通省自動車局. 自動運転車の安全技術ガイドライン. https://www.mlit.go.jp/common/001253665.pdf, 9 2018. (Accessed on 11/19/2021).

[3] 内閣府. 令和２年版交通安全白書　全文（pdf 版）. https://www8.cao.go.jp/koutu/taisaku/r02kou_haku/pdf/zenbun/1-1-1.pdf, 2020. (Accessed on 11/19/2021).

[4] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, Boston, MA, 2012.

[5] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Mach. Vis. Appl.*, Vol. 1, No. 4, pp. 223–240, December 1988.

[6] Chris Urmson, Chris Baker, John Dolan, Paul Rybski, Bryan Salesky, William Whittaker, Dave Ferguson, and Michael Darms. Autonomous driving in traffic: Boss and the urban challenge. *AIMag*, Vol. 30, No. 2, pp. 17–17, June 2009.

[7] History - indy autonomous challenge. https://www.indyautonomouschallenge.com/history, 2019. (Accessed on 11/16/2021).

[8] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. Vol. 17, No. 4, pp. 1135–1145, April 2016.

[9] Ofra Amir, Guni Sharon, and Roni Stern. Multi-agent pathfinding as a combinatorial auction. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. aaai.org, 2015.

[10] Anton Andreychuk, Konstantin Yakovlev, Dor Atzmon, and Roni Stern. Multi-Agent pathfinding with continuous time. January 2019.

[11] Wolfgang Hönig, T K Satish Kumar, Liron Cohen, Hang Ma, Hong Xu, Nora Ayanian, and Sven Koenig. Multi-agent path finding with kinematic constraints. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*. aaai.org, 2016.

[12] František Duchoň, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, Tomáš Fico, and Ladislav Jurišica. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, Vol. 96, pp. 59–69, January 2014.

[13] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Rob. Res.*, Vol. 29, No. 5, pp. 485–501, April 2010.

[14] Julius Ziegler and Christoph Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1879–1884. ieeexplore.ieee.org, October 2009.

[15] S Pothan, J L Nandagopal, and G Selvaraj. Path planning using state lattice for autonomous vehicle. In *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, pp. 1–5. ieeexplore.ieee.org, December 2017.

[16] Long Han, Quoc Huy Do, and Seiichi Mita. Unified path planner for parking an autonomous vehicle based on RRT. In *2011 IEEE International Conference on Robotics and Automation*, pp. 5622–5627. ieeexplore.ieee.org, May 2011.

[17] Cedric Pradalier, S. Vaussier, and Peter Corke. Path planning for a parking assistance system: Implementation and experimentation. 01 2005.

[18] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P. How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 5, pp. 1105–1118, 2009.

[19] James McCrae and Karan Singh. Sketching piecewise clothoid curves. *Computers Graphics*, Vol. 33, No. 4, pp. 452–461, 2009.

[20] Long Han, Hironari Yashiro, Hossein Tehrani Nik Nejad, Quoc Huy Do, and Seiichi Mita. Bézier curve based path planning for autonomous vehicle in urban environment. In *2010 IEEE Intelligent Vehicles Symposium*, pp. 1036–1042, 2010.

[21] Luis Cruz-Piris, Miguel A. Lopez-Carmona, and Ivan Marsa-Maestre. Automated optimization of intersections using a genetic algorithm. *IEEE Access*, Vol. 7, pp. 15452–15468, 2019.

[22] Xin Lin, Zhan-Qing Wang, and Xu-Yang Chen. Path planning with improved artificial potential field method based on decision tree. In *2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, pp. 1–5, 2020.

[23] Michael T Wolf and Joel W Burdick. Artificial potential functions for highway driving with collision avoidance. In *2008 IEEE International Conference on Robotics and Automation*, pp. 3731–3736, May 2008.

[24] Pengfei Lin, Woo Young Choi, and Chung Choo Chung. Local path planning using artificial potential field for waypoint tracking with collision avoidance. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, 2020.

[25] 大嶋正裕, 小河守正. モデル予測制御-i：基礎編：発展の歴史と現状. システム／制御／情報, Vol. 46, No. 5, pp. 286–293, 2002.

[26] Yadollah Rasekhipour, Amir Khajepour, Shih-Ken Chen, and Bakhtiar Litkouhi. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 5, pp. 1255–1267, 2017.

[27] Hong Wang, Bing Lu, Jun Li, Teng Liu, Yang Xing, Chen Lv, Dongpu Cao, Jingxuan Li, Jinwei Zhang, and Ehsan Hashemi. Risk assessment and mitigation in local path planning for autonomous vehicles with lstm based predictive model. *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2021.

[28] Naoki Shibata, Seiji Sugiyama, and Takahiro Wada. Collision avoidance control with steering using velocity potential field. In *2014 IEEE Intelligent Vehicles Symposium Pro-*

ceedings, pp. 438–443, 2014.

[29] Milad Nazarahari, Esmaeel Khanmirza, and Samira Doostie. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, Vol. 115, pp. 106–120, 2019.

[30] Jean-Baptiste Receveur, Stéphane Victor, and Pierre Melchior. Autonomous car decision making and trajectory tracking based on genetic algorithms and fractional potential fields. *Intelligent Service Robotics*, Vol. 13, , 04 2020.

[31] Mai Hirata, Manabu Tsukada, Keisuke Okumura, Yasumasa Tamura, Hideya Ochiai, and Xavier Défago. Roadside-assisted cooperative planning using future path sharing for autonomous driving. *CoRR*, Vol. abs/2108.04629, , 2021.

[32] R. Volpe and P. Khosla. Manipulator control with superquadric artificial potential functions: theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 6, pp. 1423–1436, 1990.

[33] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for bertha — a local, continuous method. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 450–457, 2014.

[34] 喜多一, 小野功, 小林重信. 実数値 ga のための正規分布交叉の多数の親を用いた拡張法の提案. 計測自動制御学会論文集, Vol. 36, No. 10, pp. 875–883, 2000.

[35] 隆英樋口, 茂義筒井, 雅幸山村. 実数値 ga におけるシンプレクス交叉の提案. 人工知能学会論文誌 = Transactions of the Japanese Society for Artificial Intelligence : AI, Vol. 16, pp. 147–155, nov 2001.

[36] 小林重信. 実数値 ga のフロンティア. 人工知能学会論文誌, Vol. 24, No. 1, pp. 147–162, 2009.

[37] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[38] モデル予測制御 - matlab & simulink. `https://mathworks.com/discovery/model-predictive-control.html`. (Accessed on 12/15/2021).

[39] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099, 2015.

謝辞

本論文の執筆にあたり，様々な方から多くのご指導，ご協力，ご助力，また研究生活のためのヒントを頂きました．ここに心より感謝申し上げます．

指導教官として研究の方向性，資料作成や，読むべき論文など様々な方面から労を惜しまず指導をしてくださり，また中々テーマが決まらない中でも温かくご指導くださった塚田学准教授に深く感謝いたします．

合宿などの発表の場において，有益で興味深い指摘をくださった江崎浩教授と，落合秀也准教授に感謝申し上げます．

協調型 ITS のグループにおいて，テーマが非常に近かったため，非常に具体的で役立つアドバイス，基礎的な質問を回答を頂き，また拙い自分の英語にも温かい態度で接していただいた Pengfei Lin 氏に深く感謝申し上げます．また先輩として鋭い指摘をくださった，水谷将也氏，増田英孝氏，平田真唯氏，辻尾康平氏に感謝申し上げます．またグループの同期として励ましあった，浅部佑氏，有井慎平氏，Vishal Chauhan 氏，Nicholaus D Yosodipuro 氏に感謝申し上げます．

研究生活の様々な面でバックアップをしていただいた岩井愛映子秘書，高橋富美秘書に感謝申し上げます．

研究内容と直接関係はありませんが，自分が毎日研究を継続する習慣を作る際の考え方について，深く自分に影響を与えた「SEKIRO: SHADOWS DIE TWICE」を制作してくださった方々に感謝申し上げます．

また，修士時代とコロナ禍が重なってしまったため，コミュニケーションの機会が少なくなっていた時に，くだらない雑談に付き合ってくれた友人達に感謝します．

最後に自分が勉強や部活動を行う際に，経済的な援助を惜しまず，修士卒業まで育てて支えてくれた両親に深く感謝します．本当にありがとうございました．