



Open SDV API Vehicle.Motion

バージョン : 202503a
発行日 : 2025年3月31日

Open SDV Initiative

ドキュメントの位置付け

ドキュメントの目的

- ▶ このドキュメントは、Open SDV Initiativeで作成を進めている Open SDV API の内、車両運動制御のためのAPIについて説明するものである

ドキュメントの完成度

- ▶ 現時点で、車両運動制御のためのAPIの基本的な設計ができているが、このAPIで車両を安全・円滑に制御できることは示せておらず、POC等を通じた評価が必要である
- ▶ APIの細部の設計も完了しておらず、未決定事項が多い

変更履歴

| バージョン | 発行日 | 備考 |
|---------|------------|----|
| 202503α | 2025年3月31日 | 初版 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

車両運動制御API

基本的な考え方

- ▶ 達成できる制御品質の低下を最小限にしつつ、車両の違いに依存せず、サービスコールの呼び出し周期を長くできるAPIとすることを目標とした

想定したアプリケーション

- ▶ 本API仕様を検討するにあたり、以下の機能をアプリケーションとして実現することを想定した
 - ▶ ACC (Adaptive Cruise Control)
 - ▶ LCA (Lane Centering Assist)
 - ▶ 自動運転
 - ▶ 自動駐車

縦方向の制御

- ▶ コアビークルAPIには、目標速度を指定して縦方向を制御する2種類のAPIと、目標停止位置を指定して縦方向を制御するAPIを用意する
 - ▶ 目標速度や停止位置の指定は、車両の違いに依存せず、最大でも10Hz程度で設定すれば十分と考えられる
 - ▶ それより先の制御(例えば、加速度・駆動力・制動力の算出・適用)はビークルOS内で行う
- ▶ 目標速度指定の2種類のAPIを1つのサービスコールで実現することも可能であるが、アクセス制御のために、別のサービスコールとする

縦方向の制御: API1

- ▶ 目標速度 v を指定して、縦方向を制御するサービスコールを設ける
- ▶ ビークルOSは、可能な限り迅速に目標速度に到達するように制御する
 - ▶ 目標速度に達した後は、その速度を維持するように制御を継続する
- ▶ アプリケーションは、サービスコールを周期的に呼び出して、目標速度を常に更新することを想定している
 - ▶ アプリケーションは、目標速度の更新周期および変化率を通じて、加速度を間接的に制御することができる
- ▶ 緊急ブレーキもこのサービスコールを利用する
 - ▶ さらなる検討により、緊急ブレーキが使用するサービスコールを別にする可能性もある

縦方向の制御: API2

- ▶ 目標速度 v と応答プロファイル p を指定して、縦方向を制御するサービスコールを設ける
 - ▶ 応答プロファイルは、「高速」、「標準」、「低速」の3段階のいずれかで指定する
 - ▶ **TODO**: 応答プロファイルを拡張・カスタマイズする方法を用意するかは、今後の課題とする
- ▶ ビークルOSは、応答プロファイルに従って、スムーズに目標速度になるように制御する
 - ▶ 目標速度に達した後は、その速度を維持するように制御を継続する

縦方向の制御: API3 (停止制御)

- ▶ 目標停止位置 x と停止プロファイル p を指定して, 縦方向を制御するサービスコールを設ける
 - ▶ 目標停止位置は, 現在位置からの走行距離で指定する
 - ▶ 停止プロファイルは, 「早さ重視」, 「バランス」, 「精度重視」の3段階のいずれかで指定する
 - ▶ **TODO**: 停止プロファイルを拡張・カスタマイズする方法を用意するかは, 今後の課題とする
- ▶ ビークルOSは, 停止プロファイルに従って, 目標停止位置に停止するように制御する
- ▶ 目標停止位置で停止できない場合には, エラーとする
 - ▶ 停止できる条件を車両構成記述に含める。具体的な記述方法はTBD

横方向の制御

- ▶ コアビークルAPIには、目標位置を指定して横方向を制御する2種類のAPIを用意する
 - ▶ 目標速度の指定は、車両の違いに依存せず、最大でも数Hz程度で設定すれば十分と考えられる
 - ▶ それより先の制御(例えば、実舵角の算出・適用)はビークルOS内で行う
- ▶ 到達できない目標位置を指定した場合には、エラーとする
 - ▶ 回転速度(回転半径)や回転加速度(曲率の変化率)の制約で到達できない場合がある
- ▶ 実舵角を指定して横方向を制御するサービスコールは設けない
 - ▶ 実舵角による指定では、車両の特性を知らないと制御できないため

横方向の制御: API1

- ▶ 距離 x 先における目標横位置 y を指定して、横方向を制御するサービスコールを設ける
 - ▶ 目標位置は、車両からの相対位置であり、車両が移動すると目標位置も移動する
- ▶ アプリケーションは、サービスコールを周期的に呼び出して、目標位置を常に更新することを想定している
- ▶ レーンキープ機能など、高速走行時に使用することを想定している

横方向の制御: API2

- ▶ 第1目標位置(x1, y1)とその次の第2目標位置(x2, y2)を指定して、横方向を制御するサービスコールを設ける
 - ▶ 目標位置は、車両との相対座標で指定する
 - ▶ ビークルOSは、サービスコールが呼び出された時点で対地座標に変換して保持し、車両が移動しても目標位置は移動しない
 - ▶ 第2目標位置の指定は省略できる
- ▶ ビークルOSは、スムーズに第1目標位置に到達するように制御する
 - ▶ 例えば、1つのクロソイド曲線(曲率の変化率が一定の曲線)に沿って走行するように制御する
 - ▶ 第2目標位置は、サービスコールの呼び出しの遅延や、第1目標位置での操舵の連続性に配慮するための情報であり、その使用方法はビークルOSに委ねられる

横方向の制御: API2 – 続き

- ▶ 第1目標位置に到達すると、第2目標位置が第1目標位置になり、第2目標位置は設定されていない状態になる
 - ▶ 第1目標位置に到達する前に、サービスコールを再度呼び出して、目標位置を更新しても良い
- ▶ 第1目標位置に到達し、第2目標位置が設定されていない場合、フェールセーフ制御(MRM)を行う
 - ▶ アプリケーションは、第1目標位置に到達する時点までに、第2目標位置を指定する必要がある
- ▶ 自動駐車など、低速走行時に使用することを想定している

ロック

- ▶ 縦方向の制御をロックする機能と、横方向の制御をロックする機能を設ける
 - ▶ 例えば、緊急ブレーキは、発動時に縦方向の制御をロックする

拡張ビークルAPI

- ▶ 自動駐車で使用するためのより高レベルなAPIは、必要であれば拡張ビークルAPIで用意する
 - ▶ 例えば、目標停止位置と目標停止位置における車両の向きや実舵角、最大速度、障害物の位置を指定して、縦方向と横方向の両方を制御するサービスコール