

-自動運転制御モデル I/F ガイドライン-

**車両モデル入れ替えによる**

**I/F 評価事例 解説書**

**(Ver1.0)**

改訂履歴

Rev.	日付	内容	会社名
1.0	2023/2	初版	(株)両毛システムズ

## 目次

1	はじめに .....	5
1.1	著作権 .....	5
1.2	使用環境 .....	5
2	目的.....	6
3	IF の評価・検証.....	7
3.1	観点.....	7
3.1.1	IF から決定される機能配置の影響範囲評価.....	7
3.1.2	市販ツールの入れ替えやすさからの IF 評価.....	8
3.2	手法.....	9
4	入れ替え事例 .....	11
4.1	CarSim.....	11
4.1.1	入れ替え前後のアーキテクチャ.....	11
4.1.2	入れ替えのためのアダプタ .....	12
4.1.3	入れ替え前後のテストシナリオ走行結果.....	13
4.2	CarMaker.....	14
4.2.1	入れ替え方法の説明 .....	14
4.2.2	入れ替え後のアーキテクチャ.....	15
4.2.3	入れ替えのためのアダプタ .....	16
4.2.4	入れ替え前後のテストシナリオ走行結果.....	18
5	評価結果 .....	19
5.1	IF から決定される機能配置の影響範囲評価結果.....	19
5.2	市販ツールの入れ替えやすさからの IF 評価結果.....	20
6	APPENDIX .....	22
6.1	参考文献 .....	22
6.2	CarSim 所感 .....	22

6.3 CarMaker 所感 ..... 23

## 1 はじめに

本書に記載される内容は、両毛システムズが令和 3 年度に定義した「自動車開発における自動運転制御モデル I/F ガイドライン」における自動運転コントローラ I/F の妥当性を、同じく両毛システムズが令和 3 年度に作成した「I/F ガイドライン準拠ジェネリックモデル」を用いて検証したものである。

また、検証を行うにあたり、一般に使用されている車両モデルとして Mechanical Simulation 社の CarSim と IPG Automotive 社の CarMaker を用いさせていただいた。

### 1.1 著作

本ドキュメントの著作権は、著作者に帰属します。

著作者は、本文書の内容に関し、いかなる保証もするものではありません。万一本文書を利用して不具合等があった場合でも、著作者は一切責任を負いかねます。また、本文書に記載されている事項は予告なしに変更または廃止されることがありますので、あらかじめご了承ください。

### 1.2 使用環境

MathWorks MATLAB/Simulink 2021a

CARLA 0.9.13 ソース+追加マップ

Python3.7 / ライブラリ

Microsoft VisualStudio 2019

EpicGames UnrealEngine4.26 ソース

## 2 目的

車両モデルの入れ替え事例を通して、「自動運転開発における自動運転制御モデル I/F ガイドライン」で定義されている自動運転コントローラモデルと車両モデル間の I/F（以下図の赤枠部分）の妥当性を検証し、その結果から必要に応じて「自動運転開発における自動運転制御モデル I/F ガイドライン」を適切にブラッシュアップすることを目的とする。

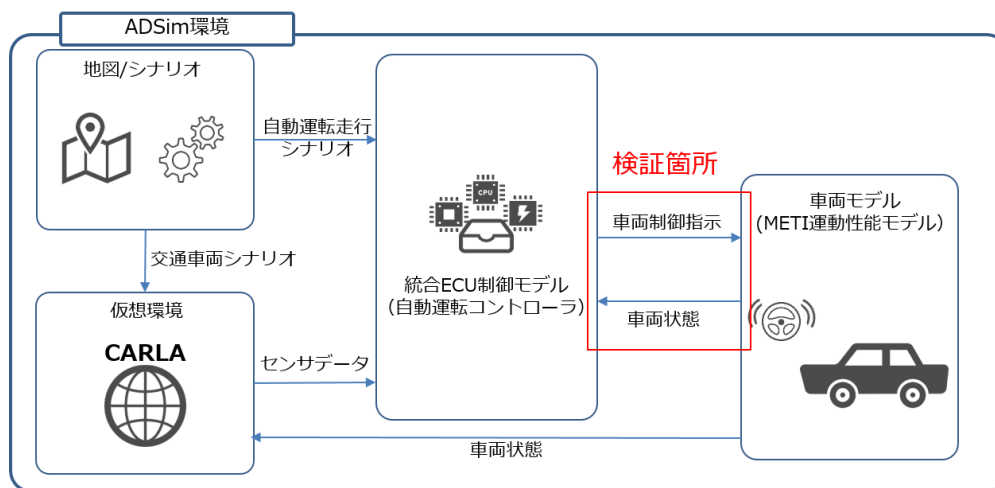


図 1 I/F 検証箇所

また副次的な効果として、「I/F ガイドライン準拠ジェネリックモデル」の拡張事例として本入れ替え事例を紹介することによる、「I/F ガイドライン準拠ジェネリックモデル」の利便性向上、利用範囲の拡大を見込んでいる。

### 3 IF の評価・検証

本項では、自動運転コントローラモデルと車両モデル間の IF 妥当性をどうやって評価、検証するかについて解説する。

#### 3.1 観点

今回の車両モデル入れ替え事例では、以下の 2 点を妥当性の評価観点とする。

- ・IF から決定される機能配置の影響範囲評価
- ・市販ツールの入れ替えやすさ評価

##### 3.1.1 IF から決定される機能配置の影響範囲評価

以下の図に示すように、システム全体として車両諸元や特性の異なる車両モデルに入れ替えるユースケースに対しての影響範囲（ADSim 環境のどこを変更しなくてはならないか？）を評価する。現行の自動運転コントローラモデルと車両モデル間の IF が適切であれば、本入れ替え事例に際して影響範囲が局所化されるはずである。

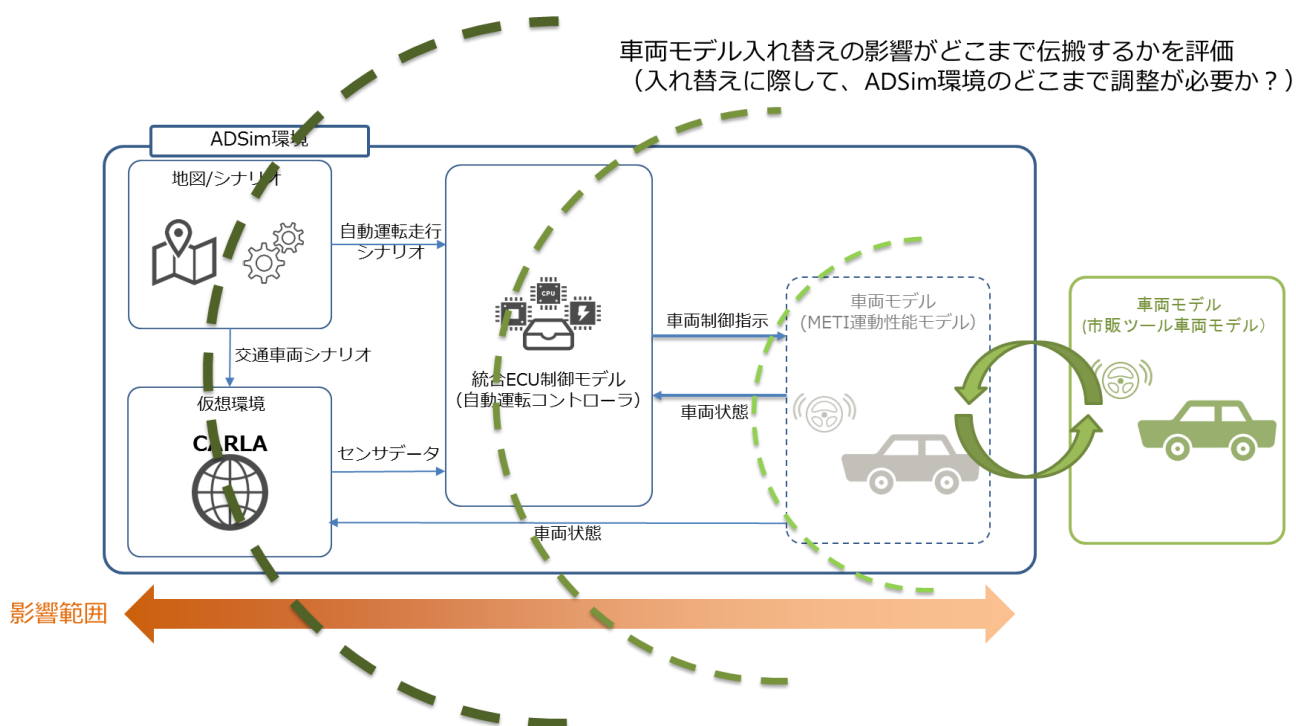


図 2 IF から決定される機能配置の影響範囲評価

### 3.1.2 市販ツールの入れ替えやすさからの IF 評価

今回市販ツールとして利用する、CarSim/CarMaker は自動車業界で広く利用されているメジャーツールとの認識である。よって、その市販ツール車両モデルで提供される IF はある程度世の中で利用されている IF になっているはずである。

本車両モデル入れ替え事例で、以下図のオレンジ点線枠で示すような IF 変換のためのアダプタ実装ボリュームが少ないほど、市販ツール IF をダイレクトに利用できることになり、結果として入れ替えやすさの評価観点とすることができる。

例えば、IF 変換アダプタがまったく必要なく市販ツール車両モデルと結合可能な IF になっていれば、世の中に流通している一般的な利用しやすい IF 定義になっていると推測できる。

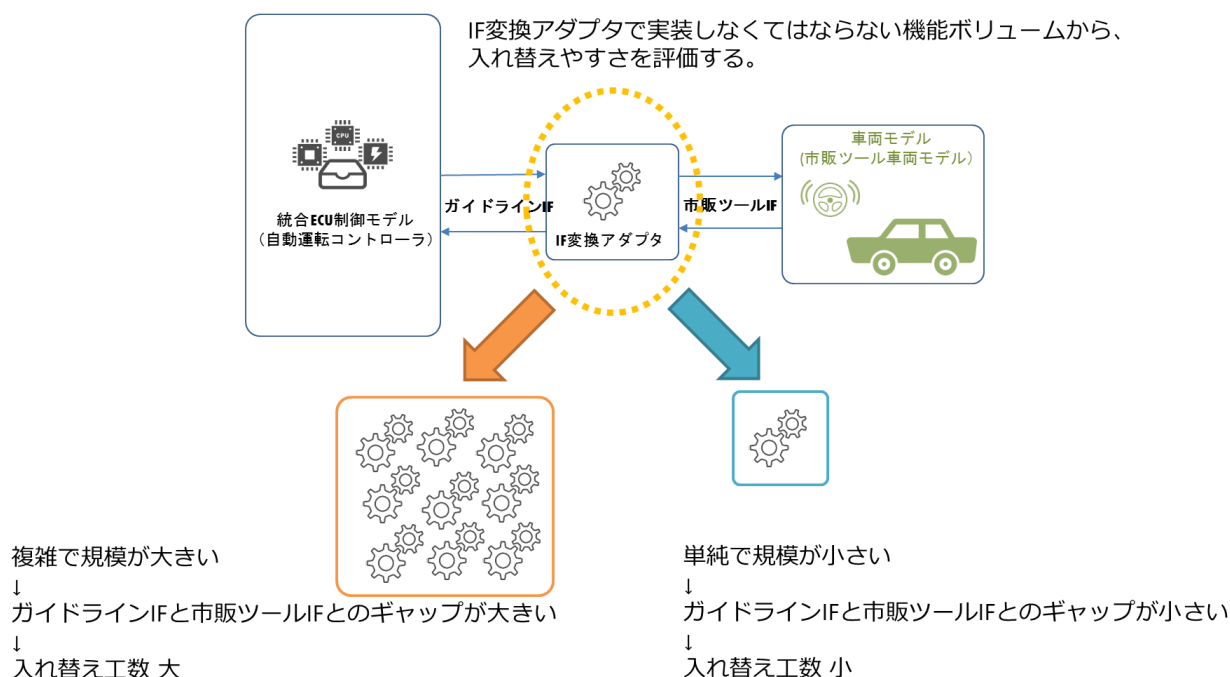


図 3 市販ツールの入れ替えやすさからの IF 評価



### 3.2 手法

令和3年度版「自動運転制御モデル IF ガイドライン準拠ジェネリックモデルシミュレータ環境」を用いる。

令和3年度版「自動運転制御モデル IF ガイドライン準拠ジェネリックモデルシミュレータ環境」の Simulink モデル (以降、ADSim モデル) で、車両プラントモデルとして使用されている METI 運動性能モデルを、市販ツールである CarSim/CarMaker 車両モデルに入れ替える。

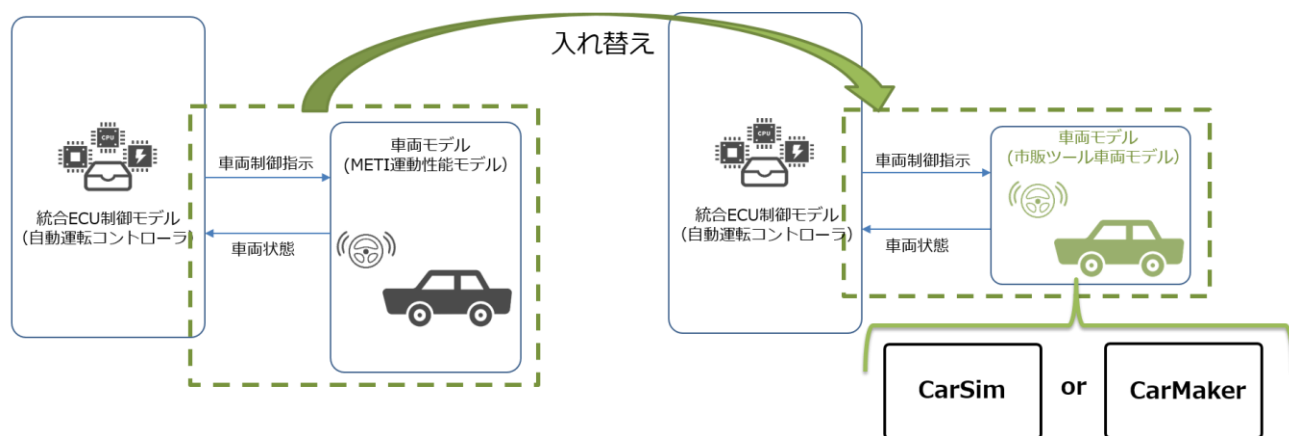


図 4 車両モデル入れ替え

ここで、入れ替え先である METI 運動性能モデルと、入れ替え元である CarSim/CarMaker 車両モデルで、なるべく同等の機能配置となるように、市販ツール CarSim/CarMaker 側の車両モデルの機能配置を調整する。

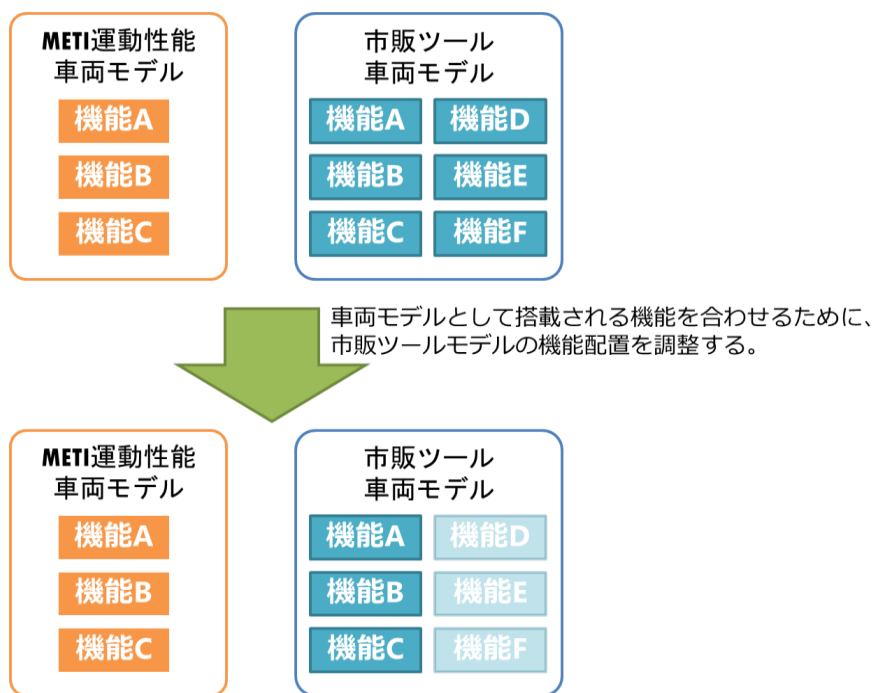


図 5 機能配置の調整

車両モデルの入れ替えがねらい通りに行えているかの確認には、シナリオ「CrossRoad\_OtherCar」を使う。

以下図に示すシナリオにおいて、車両諸元や車両挙動が異なる車両モデルに入れ替えた際も、自車両が概ねシナリオ期待通りの動きになる入れ替え事例となるように、モデルやパラメータ等の改修を行う。

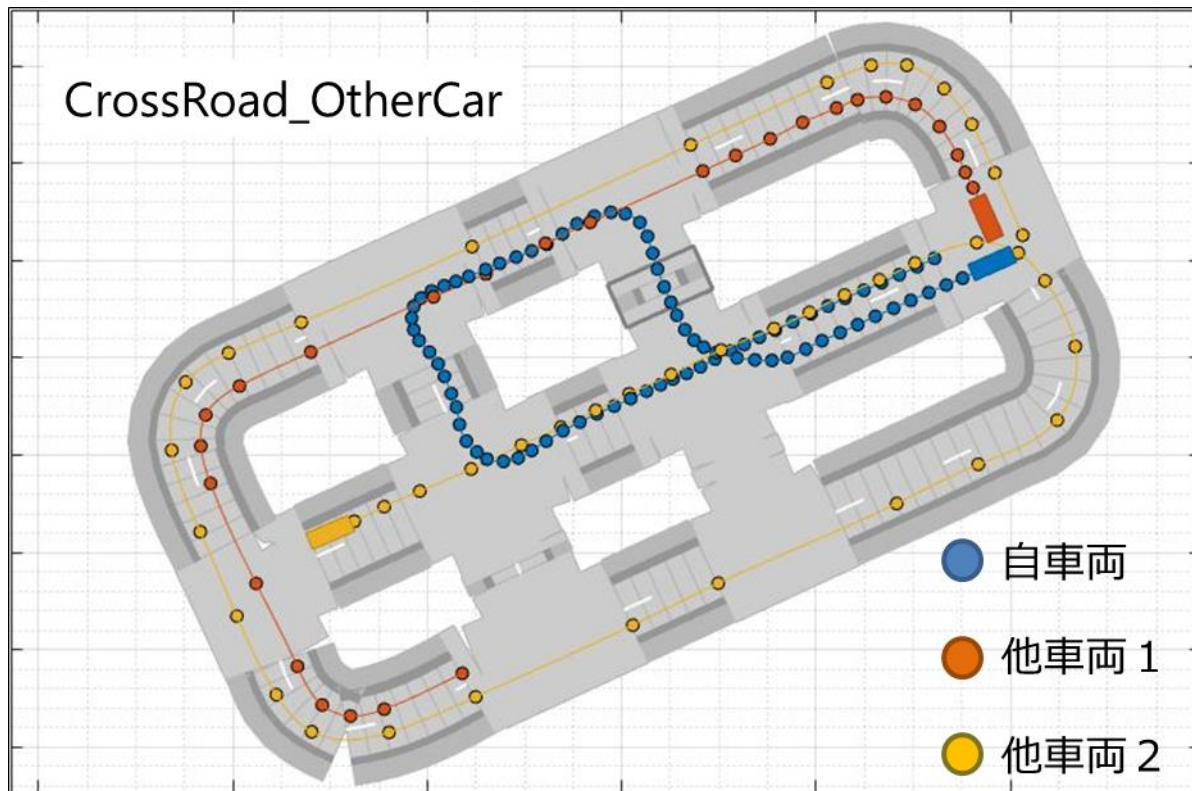


図 6 テストシナリオ CrossRoad\_OtherCar

ここまでの上述に示した車両モデル入れ替え事例を経て、ADSim モデルのどこを変更したかを観測することによって、3.1.1 で示した IF で区切られた自動運転コントローラモデルと車両モデルとの機能配分妥当性を評価する。

つまり車両モデルを入れ替えた際のモデル全体への変更影響が局所化されているかを調査し、影響が多岐に及ぶようなことがあれば、自動運転コントローラモデルと車両モデル間の IF について見直しを図る。

また、3.1.2 で示した市販ツールの入れ替えやすさからの IF については、後述の 4 入れ替え事例で IF 変換アダプタの実装ボリュームを実例として示す。

## 4 入れ替え事例

### 4.1 CarSim

ADSim モデルで使用する車両モデルを、Mechanical Simulation 社の CarSim に入れ替えた事例の解説をする。

#### 4.1.1 入れ替え前後のアーキテクチャ

入れ替え前後のアーキテクチャ図を以下に示す。

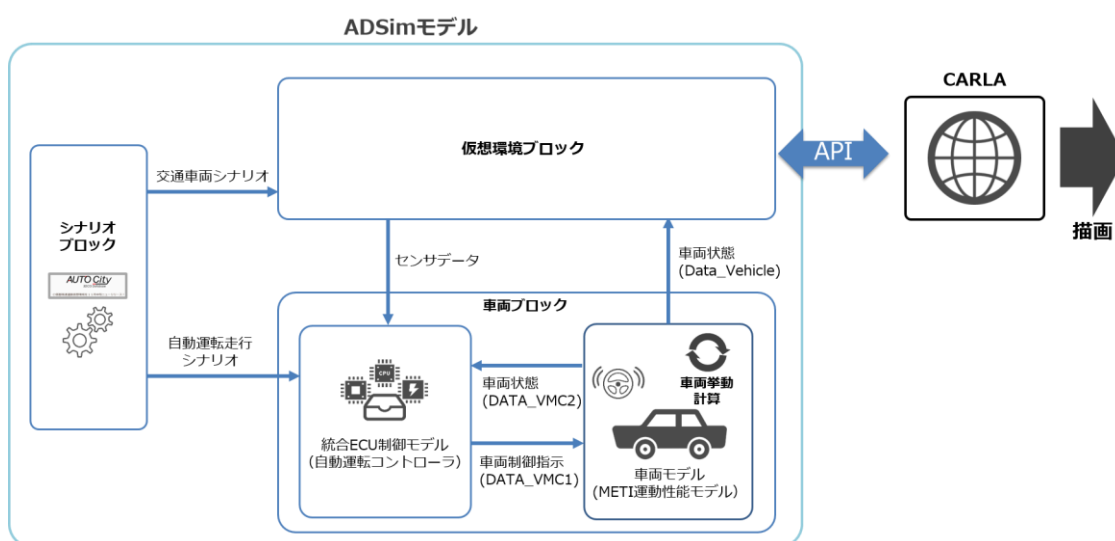


図 7 車両モデル入れ替え前のアーキテクチャ図

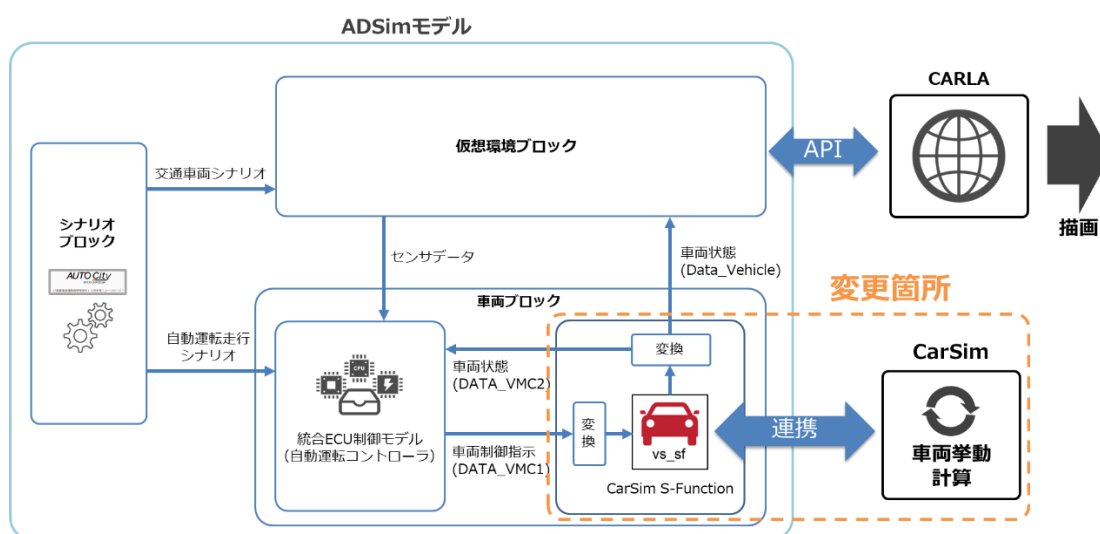


図 8 車両モデル入れ替え後のアーキテクチャ図(CarSim)

上図に示すように、CarSim の S-Function ブロックによる Simulink との連携手法を使って、車両モデルを入れ替えるアーキテクチャを採用した。

CarSim と Simulink との同期設定は非同期とし、推奨通り、Simulink の処理周期を CarSim 処理周期の半分に設定している。

### 4.1.2 入れ替えのためのアダプタ

自動運転コントローラと CarSim の S-Function との IF を接続するためのアダプタ仕様について解説する。

以下図に示すように、INPUT アダプタ部分については若干の変更が必要であったが、OUTPUT についてはアダプタの実装はほぼ不要な形となった。

INPUT アダプタの IF 変換については、ガイドラインに従った自動運転コントローラからの車両制御指示（目標加速度、目標曲率）を、CarSim S-Function INPUT として設定した前後トルク指示、ステアリング角に変換している。

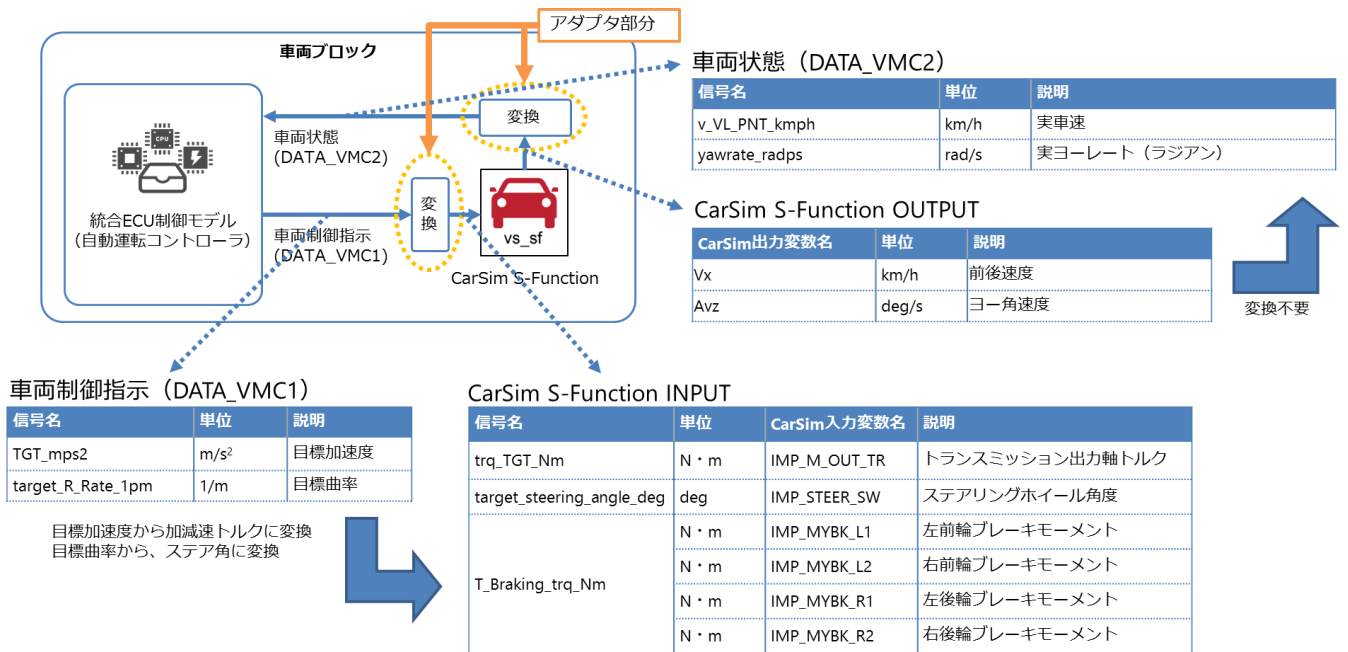


図 9 CarSim アダプタ

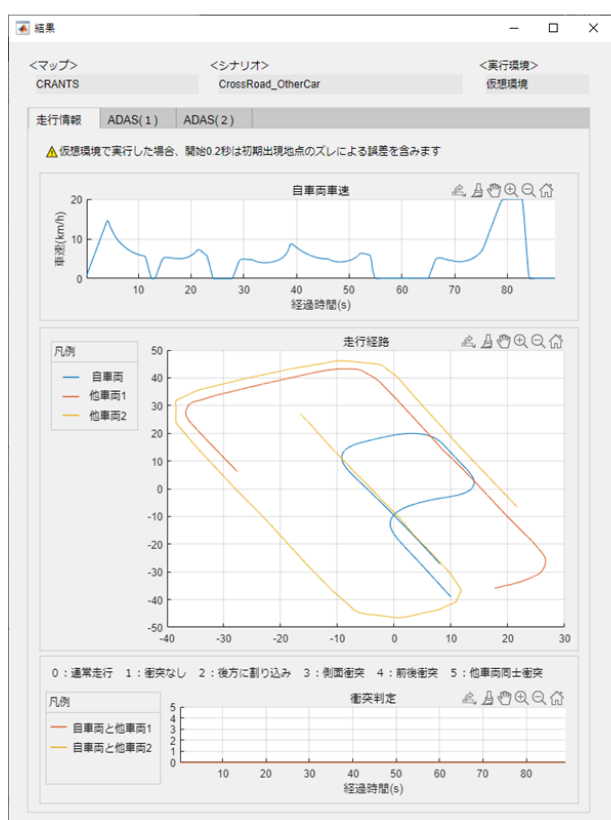
### 4.1.3 入れ替え前後のテストシナリオ走行結果

以下に、車両モデルを CarSim に入れ替え前後での、CrossRoad\_OtherCar シナリオでのシミュレーション走行結果を示す。評価結果のビューは「I/F ガイドライン準拠ジェネリックモデル」で実装されているモニタ機能を利用した。

結果としては、車両諸元（車重、パワトレ、ブレーキ Spec 等）や車両機能（キネマティクスの数式や解法等）異なる車両モデルにスワップしているため、同じシナリオでも、当然微小な車両挙動差異は発生している。

が、シナリオの動き（十字路での他車両回避、標識認識しての停車動作、等）や走行経路などのシナリオ本筋としては入れ替え前後でほぼ一致しており、シナリオの期待通りの車両挙動を入れ替え後でも示した。

METI車両モデル



CarSim車両モデル

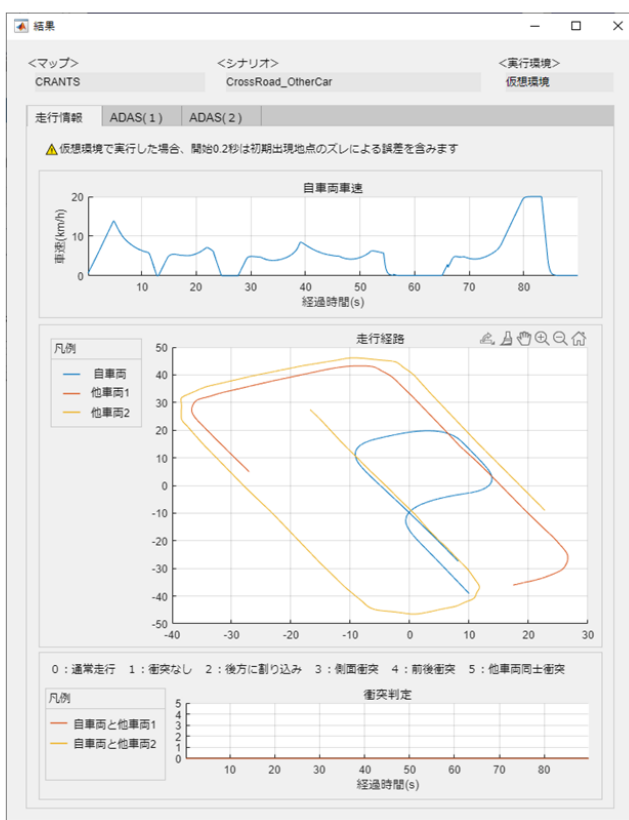


図 10 CarSim 入れ替え前後のテストシナリオ比較

## 4.2 CarMaker

ADSim モデルで使用する車両モデルを、IPG Automotive 社の CarMaker に入れ替えた事例の解説をする。

### 4.2.1 入れ替え方法の説明

CarMaker システムを Simulink 環境で実行できる機能を今回利用する。CarMaker 提供の Simulink モデル (generic.mdl) を改変し、同ファイル内で ADSim モデルと CarMaker モデルを動作させる。CarMaker モデルに制御指示信号を入力し、CarMaker 内部変数から必要な値を取得する。

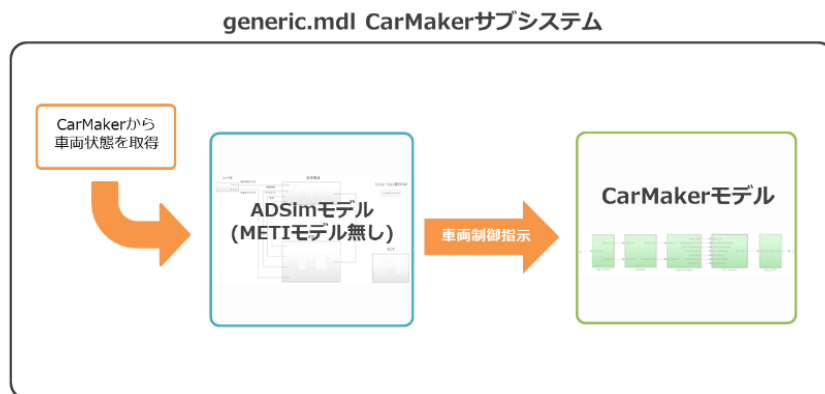


図 11 CarMaker モデル概略

また、以下に実際の CarMaker 提供の Simulink モデル (generic.mdl) を改修したものを示す。

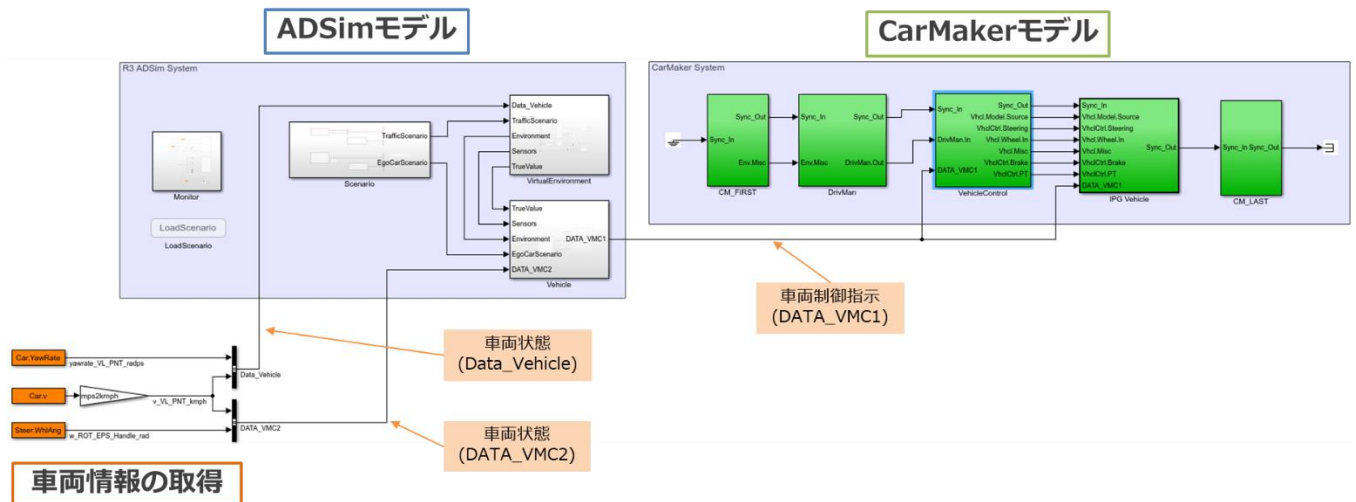


図 12 CarMaker Simulink モデル実際

### 4.2.2 入れ替え後のアーキテクチャ

車両モデル入れ替え後のシステムアーキテクチャを以下に示す。4.2.1 入れ替え方法の説明で概要説明した通り、CarMaker 提供の Simulink モデルに、ADSim モデルを追加する形で CarMaker 車両モデルの入れ替えを実現している。

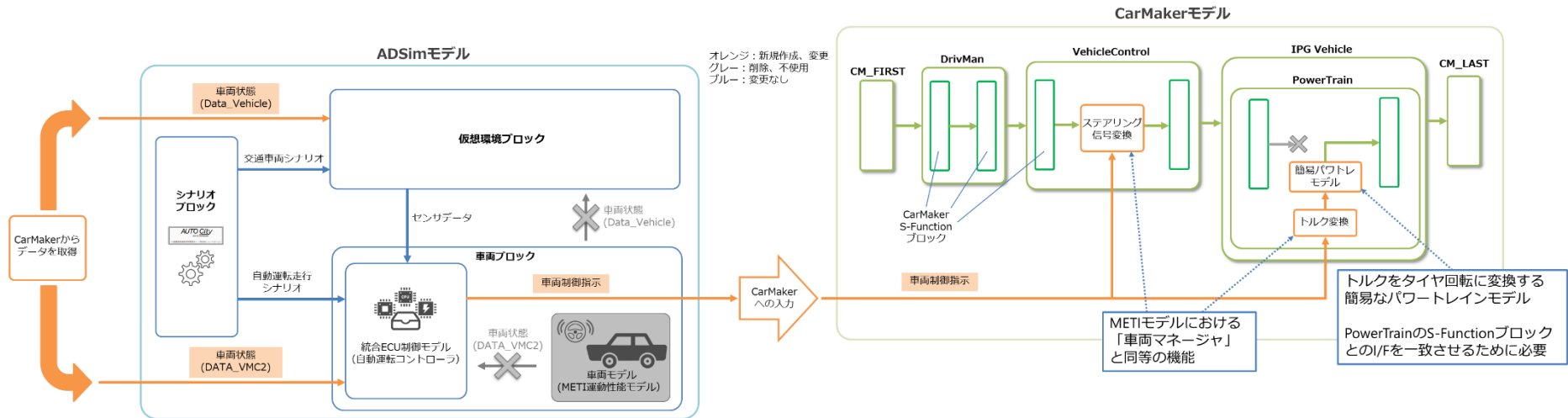


図 13 車両モデル入れ替え後のアーキテクチャ

### 4.2.3 入れ替えのためのアダプタ

自動運転コントローラと CarMaker との Simulink IF を接続するためのアダプタ仕様について解説する。

以下に示す図のように、INPUT アダプタ部分については若干の変更が必要であったが、OUTPUT についてはアダプタの実装はほぼ不要な形となった。

INPUT アダプタの IF 変換については、ガイドラインに従った自動運転コントローラからの車両制御指示（目標加速度、目標曲率）を、各輪のタイヤ回転速度、ステアリング角に変換して CarMaker 車両モデル内にバイパスしている。



# 車両モデル入れ替えによる I/F 評価事例 解説書

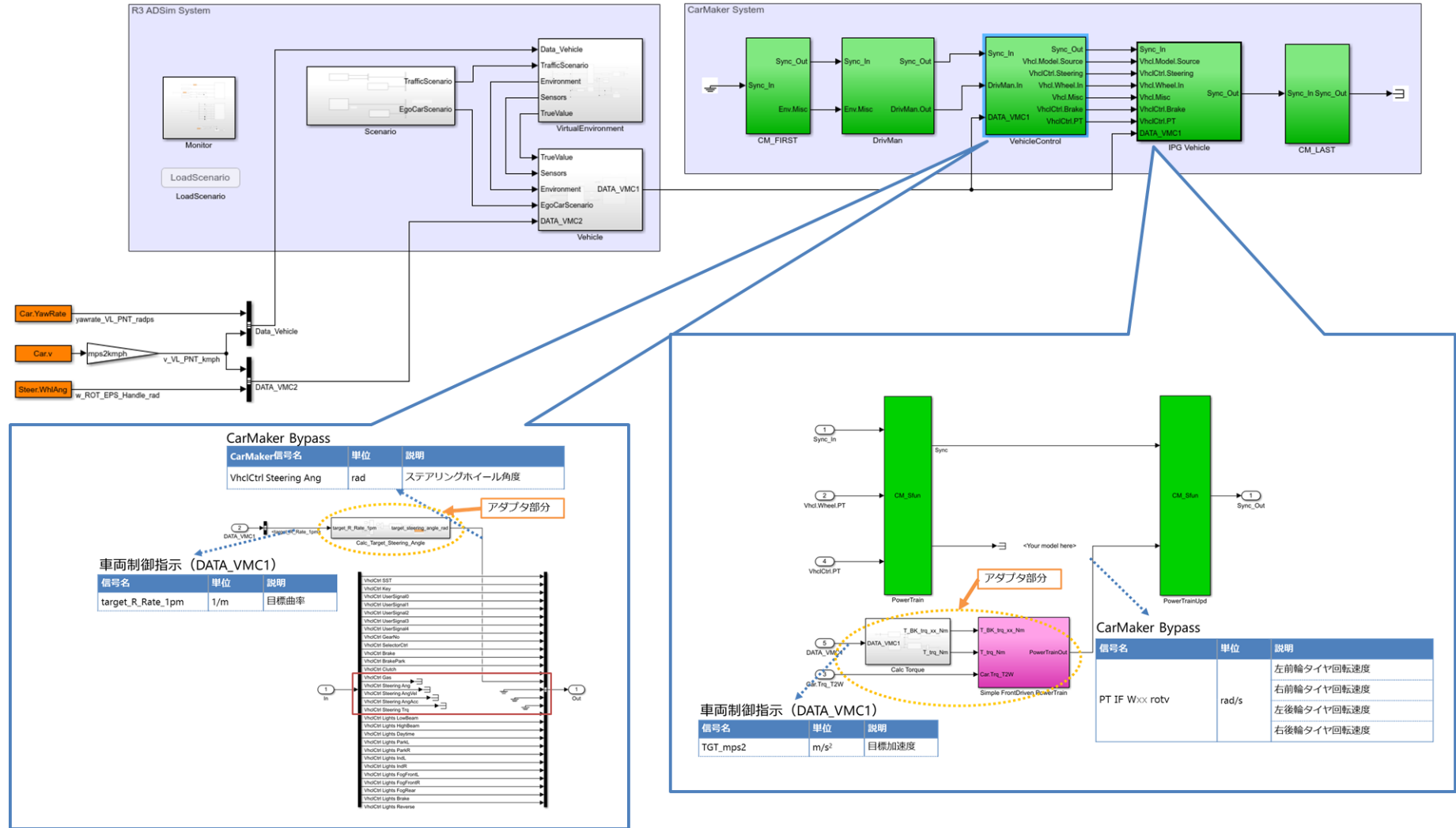


図 14 CarMaker アダプタ

#### 4.2.4 入れ替え前後のテストシナリオ走行結果

以下に、車両モデルを CarMaker に入れ替え前後での、CrossRoad\_OtherCar シナリオでのシミュレーション走行結果を示す。評価結果のビューは「I/F ガイドライン準拠ジェネリックモデル」で実装されているモニタ機能を利用した。

結果としては、車両諸元（車重、パワトレ、ブレーキ Spec 等）や車両機能（キネマティクスの数式や解法等）異なる車両モデルにスワップしているため、同じシナリオでも、当然微小な車両挙動差異は発生している。

が、シナリオの動き（十字路での他車両回避、標識認識しての停車動作、等）や走行経路などのシナリオ本筋としては入れ替え前後でほぼ一致しており、シナリオの期待通りの車両挙動を入れ替え後でも示した。

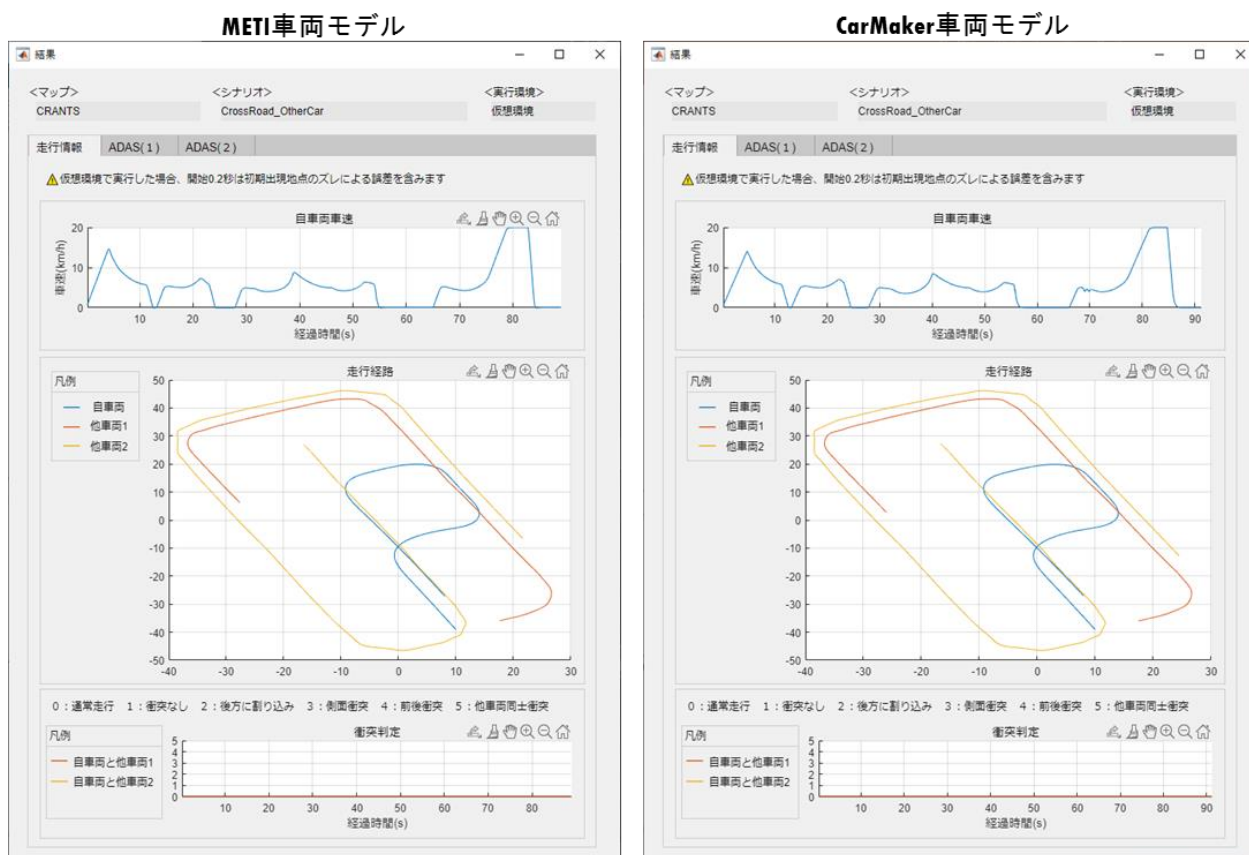


図 15 CarMaker 入れ替え前後のテストシナリオ比較

## 5 評価結果

### 5.1 IF から決定される機能配置の影響範囲評価結果

CarSim/CarMaker どちらの入れ替え事例でも、車両モデル入れ替えた上でテストシナリオを実行し、シナリオ通りの車両挙動にするための変更箇所は、車両モデル内のみになり自動運転コントローラの変更は不要であった。

よって IF ガイドラインで定めた自動運転コントローラと車両モデル間の IF 定義から決定される機能配置の凝集レベルは高く、本評価観点では IF ガイドライン定義に問題ないという結果となった。

自動運転コントローラ以下のシステムはパラメータチューニング含む変更なしで済み、車両モデル内だけに影響範囲は極所化されていた。

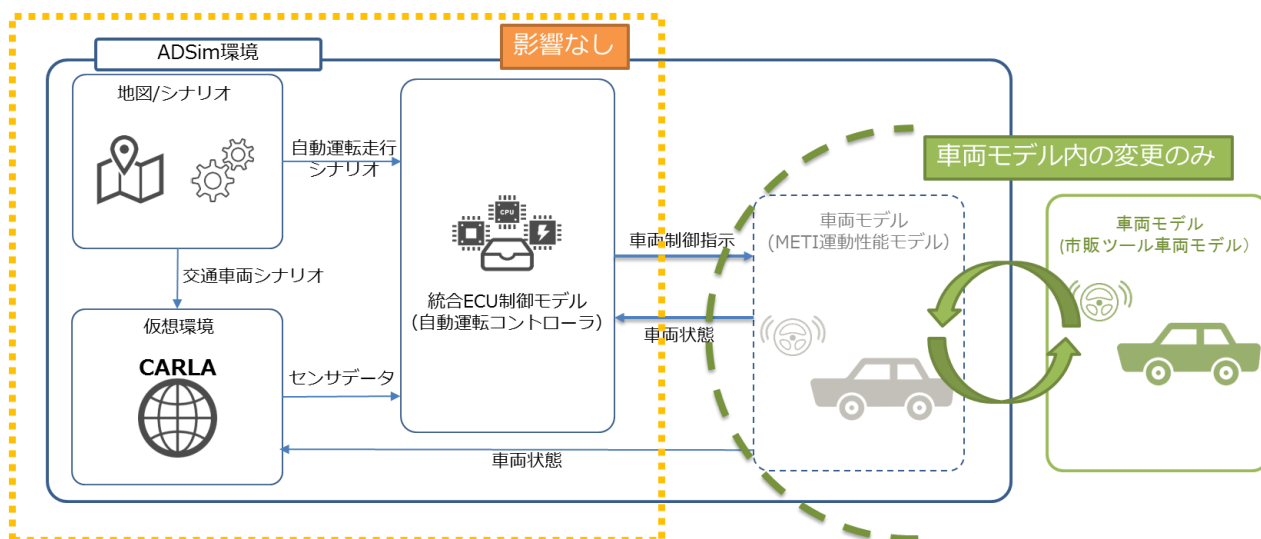


図 16 影響範囲結果

## 5.2 市販ツールの入れ替えやすさからの IF 評価結果

上述の入れ替えのためのアダプタ 4.1.2、4.2.3 で示した通り、CarSim/CarMaker で規模の若干の大小はあれど、車両モデル入れ替えに際して、INPUT の IF を合わせるためにある程度のアダプタ実装工数が必要な結果となった。

これは、基本的に今までの市販ツールのユースケースのメインストリームが、ドライバ操作 INPUT による車両挙動の模擬であることからドライバ操作を中心とする IF 設計になっているため、今回提案している自動運転コントローラ⇔車両コントローラ間の IF 定義と剥離しているのが原因と考えられる。

下図に示す通り、自動運転コントローラからの OUTPUT IF を運転操作 IF にすれば、市販ツールとの親和性という意味では向上する。が、5.1 IF から決定される機能配置の影響範囲評価結果で示した通り、車両諸元・特性による差異をアダプタ部分で吸収できているため自動運転コントローラに影響を及ぼさない IF として実現できている部分が、運転操作 IF に変更すると、車両諸元・特性に依存する機能が自動運転コントローラに配置されることになるため、凝集度が低下してしまう。

よって今回のケースでは、車両モデルを差し替えても自動運転コントローラに影響を及ぼさないことを優先させ、IF ガイドラインで定義したものを変更しない結論とした。

今後は、どのような機能単位でモデルを差し替え運用するか、自動車メーカーとサプライヤとの役割分担、供給部品の機能配置、車両のアーキテクチャ等、実ユースケースに応じて見直し検討していきたい。

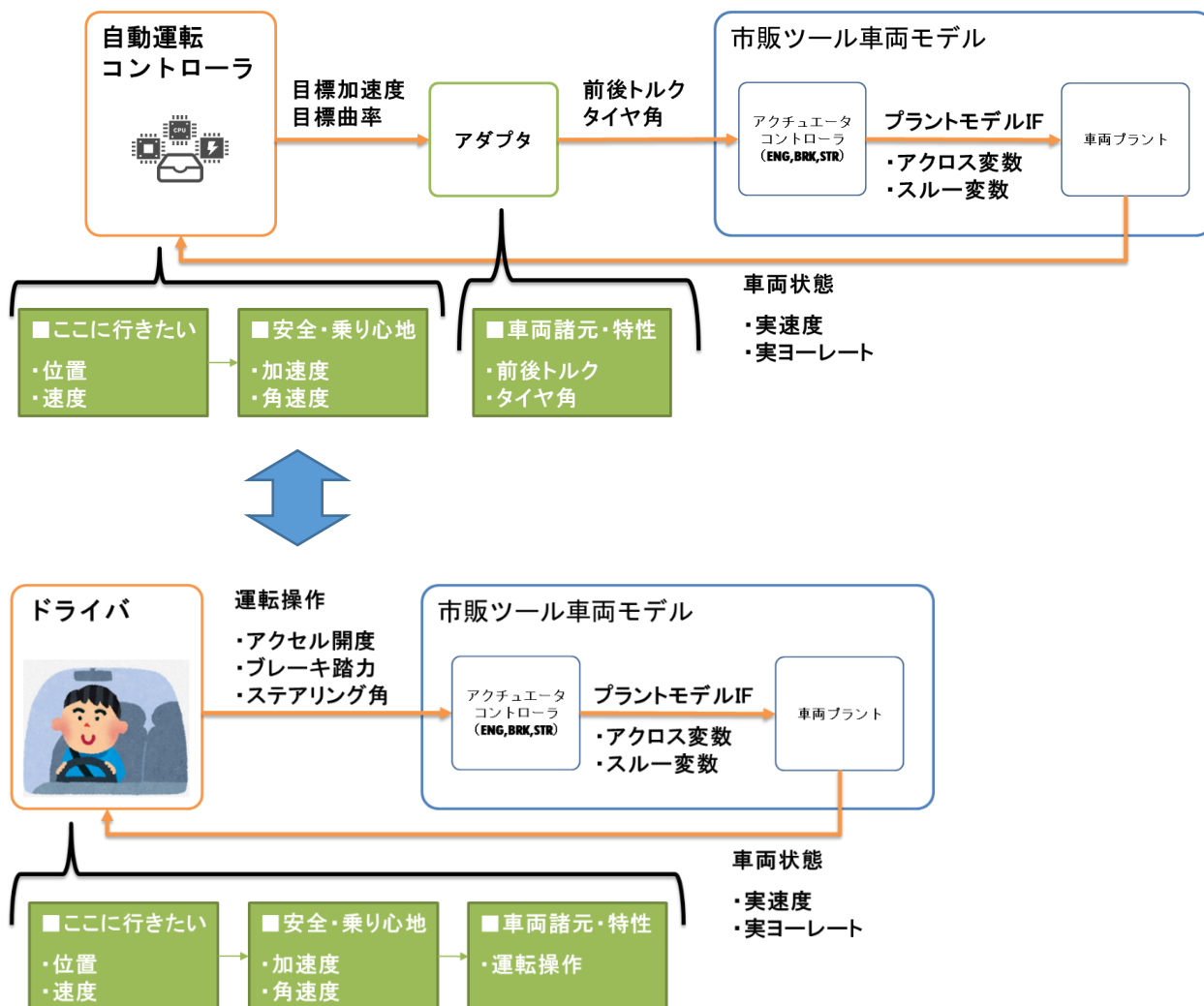


図 17 自動運転と手動運転での INPUT IF と機能配置の違い

## 6 APPENDIX

### 6.1 参考文献

- 1) 「自動運転開発における自動運転制御モデル I/F ガイドライン」

### 6.2 CarSim 所感

CarSim S-Function ブロックを用いての CarSim-Simulink 間連携は、下図のように一覧から選択した入出力がそのまま Simulink ブロックに反映されるので、操作が直感的で使い勝手が良かった。また、CarSim システムをブロックとして扱うことができるので、Simulink をベースとした本事例の環境に組み込みやすかった。

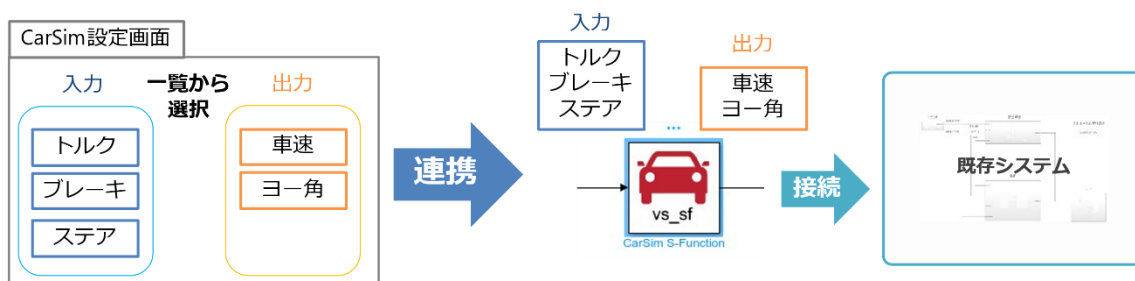
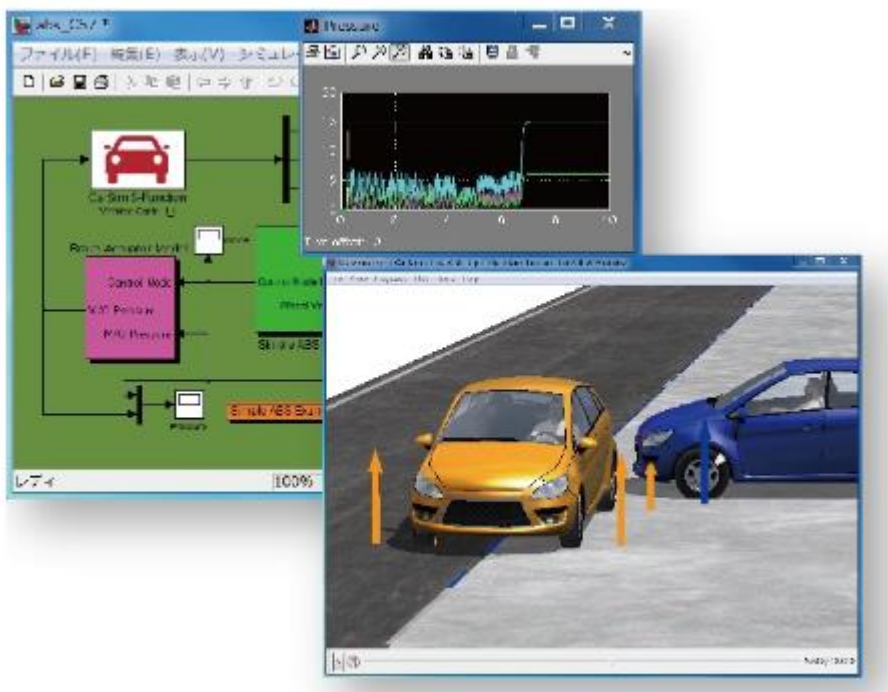


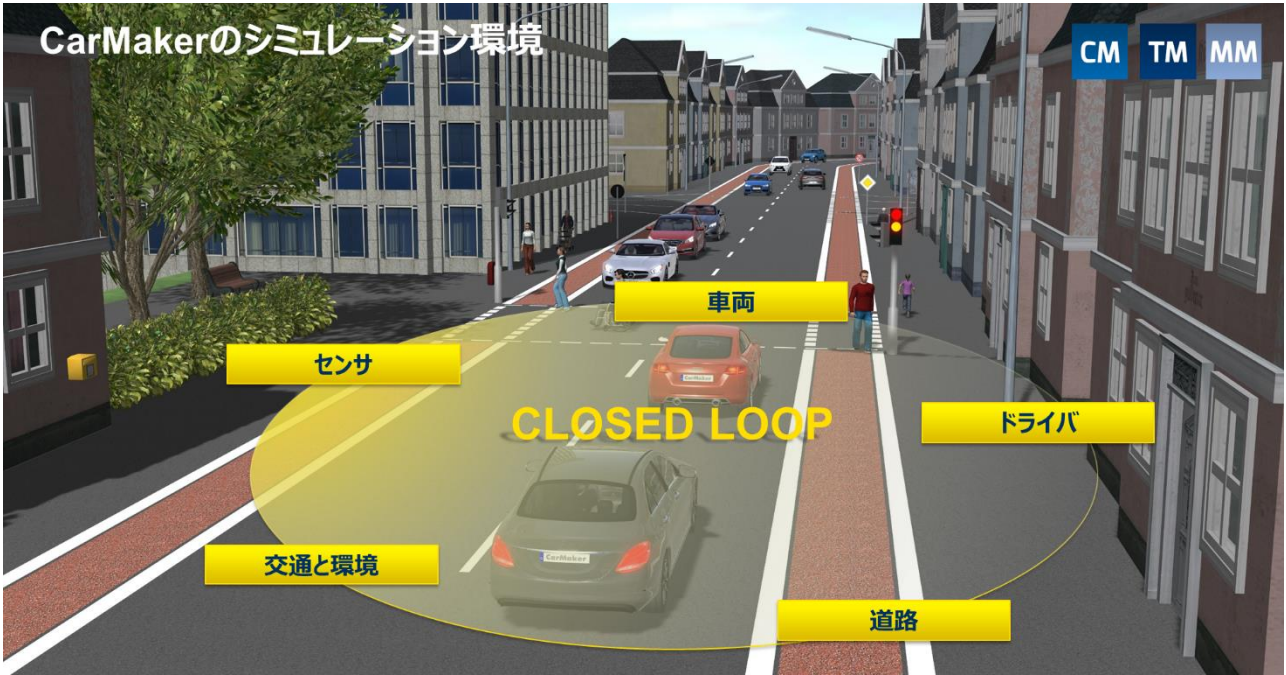
図 18 CarSim-Simulink 連携



出典 : [CarSim | VehicleSim | 製品を探す | バーチャルメカニクス | 自動車技術開発ソリューション \(vmc.jp\)](http://vmc.jp)

### 6.3 CarMaker 所感

CarMaker は設計思想が以下の図に示すように、交通環境等含めた統合的なクローズドループのシミュレーション環境を提供するツール思想となっている。そのため今回のようなすでに別システム環境に車両モデル機能を入れ替えて使うようなユースケースは、メインストリームではない。



が、4.2.1 入れ替え方法の説明で示すように、CarMaker システム全体を Simulink モデル環境で実行できる機能を今回利用することで、今回の事例にも対応することができた。内部機能へのアクセス、バイパスについては、以下図に示すように Simulink ベースで可視化されるため直感的に IF へのフック、バイパスを実装することができた。

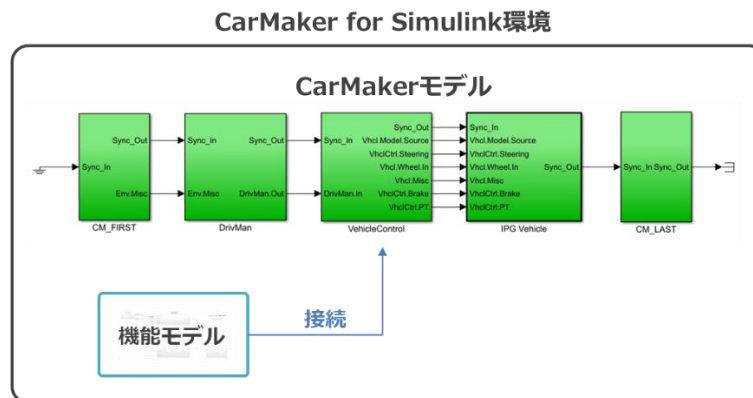


図 19 CarMaker-Simulink 連携

出典 : [CarMaker | IPG Automotive 株式会社 \(ipg-automotive.com\)](http://ipg-automotive.com)