

清水孝一*
Koichi Shimizu
鈴木大輔†
Daisuke Suzuki
村松 竜‡
Ryo Muramatsu

毛利寿志§
Hisashi Mori

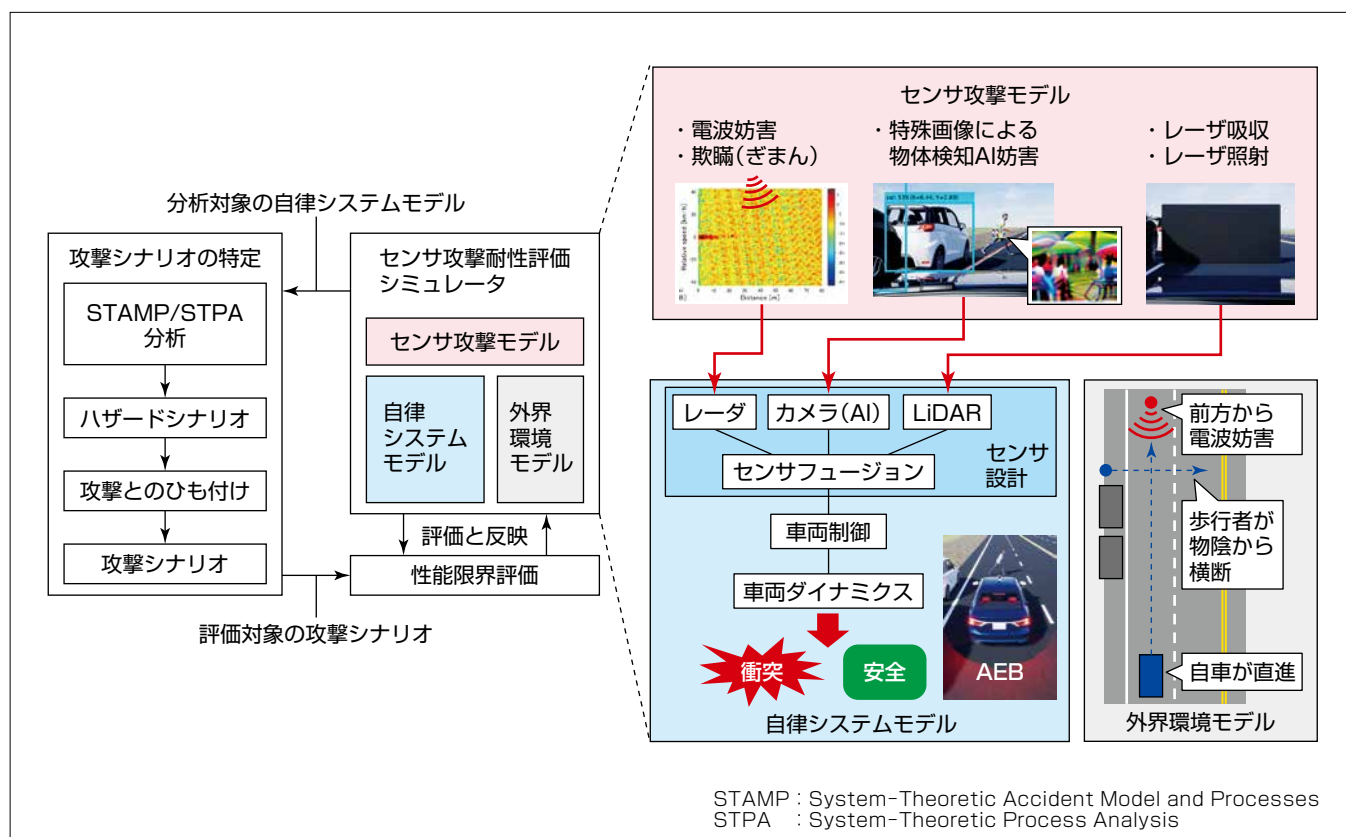
センサ攻撃耐性評価フレームワーク

Evaluation Framework for Sensor Attack Resistance

要 旨

自動運転車を始めとする自律システムは、様々なセンサによって周囲の状況を把握し、それに基づいてシステムを制御している。そのため、センサへの攻撃は安全性に深刻な影響を及ぼす。自動車の安全性に関連する国際規格のISO 21448ではセンサの性能限界を考慮した自動車の安全性を対象にしているが、セキュリティの観点は含まれていない。また、ISO/SAE(Society of Automotive Engineers) 21434では車載システムの開発プロセスでのセキュリティを対象にしているが、センサ攻撃のような特定の脅威は対象ではない。したがって、センサ攻撃に対する自動運転システムの安全性を評価する仕組みが必要である。実車での評価が安全性の観点で難しいことや、設計に反映させるために開発の早期段階での評価が必要なことから、STAMP/

STPA分析に基づくセンサ攻撃シナリオの特定と、自律システムに対するセンサ攻撃の影響のシミュレーション評価を組み合わせたフレームワークが有効である。具体例としてレーダ、カメラ、LiDAR(Light Detection And Ranging)を用いたAEB(Autonomous Emergency Braking: 衝突被害軽減ブレーキ)搭載車に対するシミュレータのプロトタイプを開発した。日本と欧州のNCAP(New Car Assessment Programme: 新車アセスメントプログラム)で規定されているAEB試験の走行シナリオを網羅し、走行車に対する様々なセンサ攻撃の影響を評価できるものである。これによって、攻撃耐性を考慮したセンサの性能要件の早期決定を可能にする。



センサ攻撃耐性評価フレームワークの概要

評価フレームワークは、STAMP/STPA分析に基づく攻撃シナリオの特定と、センサ攻撃耐性評価シミュレータによる評価から成る。対象の自律システムに対する攻撃シナリオを網羅的に特定し、センサ設計と攻撃のパラメータを変化させてシミュレーション評価を行うことによって、センサ攻撃に対するセンサの性能限界を評価できる。これによって、センサ設計者はセンサ攻撃耐性に必要になるセンサ設計パラメータを把握し、センサの設計に反映することが可能になる。

1. ま え が き

自動運転車などの自律システムは様々なセンサを用いて周囲の環境を把握し、それに基づいてシステムを制御している。本稿で述べるAEB搭載車では、レーダ、カメラ、LiDARを用いて前方の物体を検出して位置と速度を測定し、必要に応じてドライバーに警告を発したり自動的にブレーキをかけたりする。一方、センサに対する攻撃についてこれまでに数多くの報告がされており、センサ単体だけでなく、センサを利用する自律システムに影響を及ぼす例も知られている。したがって、センサ攻撃に対する自律システムの安全性を担保することが不可欠である。

自動運転車の場合、安全性の検証に必要とされる何百億km⁽¹⁾もの距離の走行テストを実世界で行うことは現実的ではないため、様々な走行シナリオを想定したシミュレーションによる評価が一般的である。しかし、無数のシナリオの中から評価すべきシナリオを絞り込むことが課題とされている。さらに、センサ攻撃の影響をシステムレベルで評価可能なシミュレータについてあまり報告されていない。

本稿では、STAMP/STPA分析⁽²⁾に基づく手法で評価すべきセンサ攻撃シナリオを網羅的に特定し、センサ攻撃の自律システムへの影響をシミュレーションで評価するフレームワークを提案する。また、シミュレータのプロトタイプを開発し評価を行った結果を述べる。

2. SOTIFに基づく評価フレームワーク

自動車の安全に関する国際規格として、機能安全規格ISO 26262⁽³⁾とSOTIF(Safety Of The Intended Functionality)規格ISO 21448⁽⁴⁾がある。機能安全は、故障等の不具合によって引き起こされるリスクがない状態を目指すものである。SOTIFは、意図した機能や性能の限界によるリスクがない状態を目指すもので、機能安全を補う比較的新しい安全性概念である。ISO 21448は、自動運転等の高度な機能がよりどころとしているセンサを意識している点が特徴であるが、セキュリティの観点は含まれていない。自動車のセキュリティに関する国際規格としてはISO/SAE 21434⁽⁵⁾があるが、開発プロセスでのセキュリティリスク管理が主眼であり、センサ攻撃等の特定の脅威は扱われていない。

ISO 21448では、システムの仕様を入力として、怪我(けが)などの損害につながる仕様の不備や性能限界を特定・評価することによって、仕様を修正しSOTIFを改善していくプロセスが規定されている。センサを組み込んだ自律システムがセンサ攻撃に対してどこまで耐え得るかを

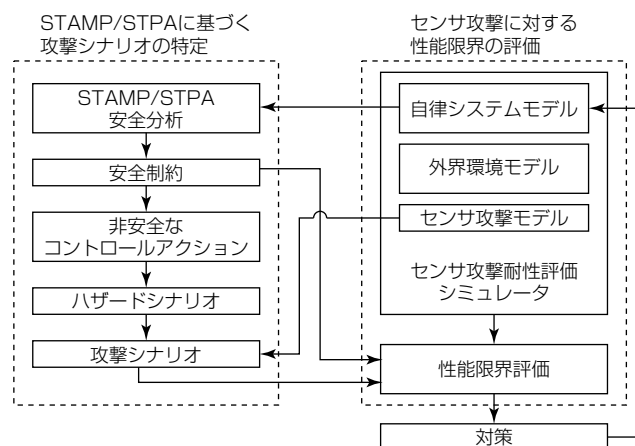


図1. センサ攻撃耐性評価フレームワーク

評価し、評価結果をセンサの設計に反映する目的に適したプロセスである。そこで本稿はSOTIFの改善プロセスに基づくセンサ攻撃耐性評価フレームワークを提案する(図1)。STAMP/STPA分析で損害につながるシナリオを特定し、その結果をセンサ攻撃とひと付けする方法で、性能限界の評価で考慮すべきセンサ攻撃シナリオを網羅的に特定する。性能限界の評価には、自律システムにセンサ攻撃と外界環境を組み込んだセンサ攻撃耐性評価シミュレータを用いて、各攻撃シナリオ下でのセンサ攻撃に対する性能限界を評価する。それぞれの詳細は3章と4章で述べる。

3. STAMP/STPA分析に基づく攻撃シナリオの特定

怪我や経済的損失など、防ぐべき損害はシステムによって様々であり、本稿ではアクシデントと呼ぶ。また、アクシデントが潜在するシステムの状態をハザードと呼ぶ。システムがハザードに陥るハザードシナリオを特定するための手法としてFTA(Fault Tree Analysis)、FMEA(Failure Mode and Effect Analysis)、STAMP/STPA等が知られている。FTAとFMEAでは主にコンポーネントの故障に起因するハザードに焦点を当てているのに対して、STAMP/STPAでは、たとえ故障がなくてもコンポーネント間の意図しない作用でハザードが発生し得るとの観点で分析する。本稿では、故障ではなくセンサ情報に基づくAEB制御と車両の相互作用が主眼であるためSTAMP/STPAを採用する。

AEB搭載車に対する攻撃シナリオ特定の流れと結果の一部を図2に示す。まず、システムが安全を実現する大まかな構造として、アクシデントとハザード及びハザードを制御するための安全制約を定義する。図2の最上段に定義の一例を示す。また、安全制約の実現に関係するコンポーネント(サブシステムや機器)と、コンポーネント間の相互作用(コントロールアクションやフィードバックデータ)を

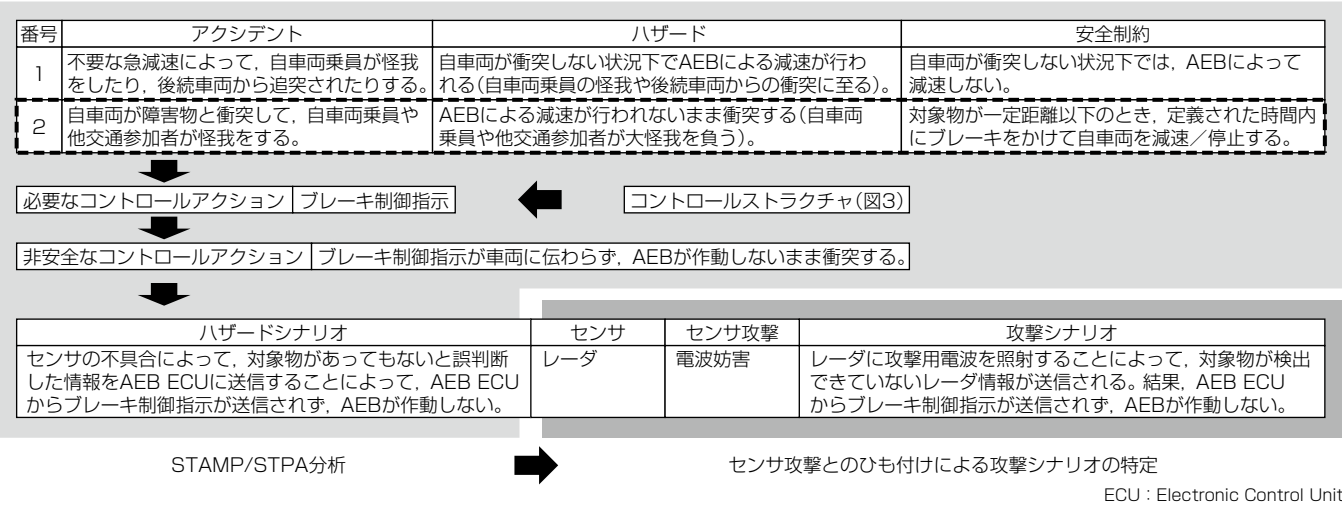


図2. AEB搭載車に対する攻撃シナリオ特定の流れと結果の一部

分析し、図3のようにコントロールストラクチャとしてまとめる。次に、コントロールストラクチャから、安全制約の実行に必要なコントロールアクションを識別し、ハザードにつながる非安全なコントロールアクションを特定する。そして、各非安全なコントロールアクションに対してハザードにつながるハザードシナリオを特定する。

最後に、ハザードシナリオでのセンサの不具合を、その原因になるセンサ攻撃と結び付けることによって、センサ攻撃シナリオを特定する。センサ攻撃については、攻撃の目的を物体検出の妨害と欺瞞の2種類に分類し、レーダ、カメラ、LiDARのそれぞれに対して現在知られている攻撃を網羅している(表1)。

図2の分析結果では、安全制約“対象物が一定距離以下のとき、定義された時間内にブレーキをかけて自車両を

減速／停止する”の実行に必要なコントロールアクションとして“ブレーキ制御指示”を識別し、ハザードにつながる非安全なコントロールアクション“ブレーキ制御指示が車両に伝わらず、AEBが作動しないまま衝突する”を特定し、対応するハザードシナリオと攻撃シナリオを特定している。

4. センサ攻撃に対する性能限界の評価

4.1 センサ攻撃耐性評価シミュレータ

センサ攻撃耐性評価シミュレータの構成を図4に示す。本稿のプロトタイプは、モデルベース設計で広く利用されているMathWorks社のMATLAB/Simulink^(注2)をプラットフォームとして、外界環境に関わるモデルをEpic Games社のUnreal Engine^(注3)で実装した。自律システムを構成するプラント、コントローラ、状態計測、認知・判断、センサについてはそれぞれのドメインで開発されるモデルを利用することを想定しており、今回はMathWorks

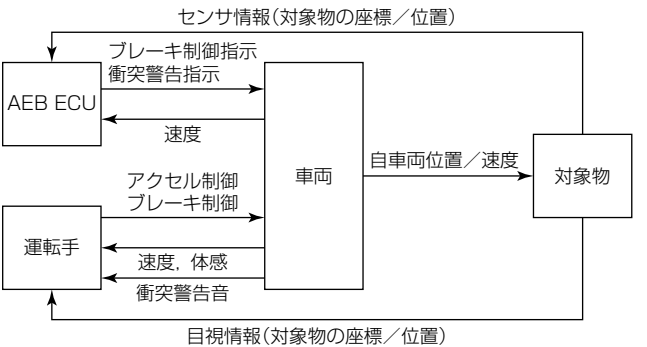


図3. AEB搭載車のコントロールストラクチャ

表1. サポートしているセンサ攻撃

センサ	妨害	欺瞞
レーダ	電波妨害	距離欺瞞 速度欺瞞 なりすまし攻撃
カメラ	パッチによる個体検出妨害 パッチによる全体検出妨害 路面標示改変 ^(注1)	投影による歩行者・車の誤検出 ^(注1)
LiDAR	照射攻撃 吸収攻撃	なりすまし攻撃

(注1) シミュレータでは未サポート

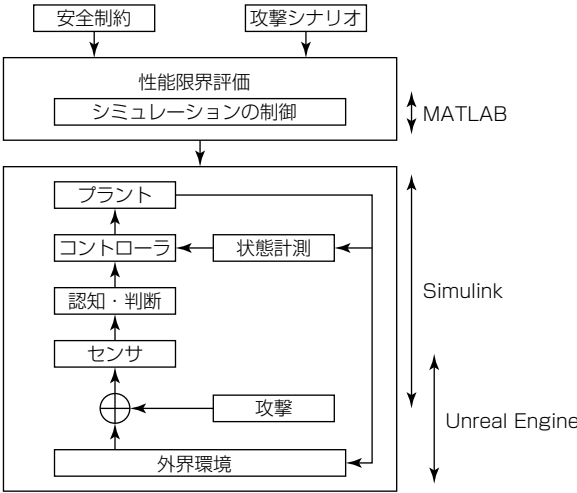


図4. センサ攻撃耐性評価シミュレータ

社が提供するサンプルモデルを組み合わせ、レーダ、カメラ、LiDARに基づくAEB搭載車のモデルを作成した。外界環境モデルでは、日本と欧州のNCAPで規定されているAEB試験の走行シナリオを網羅し、STAMP/STPA分析で特定される網羅的な攻撃シナリオとの組合せによって、走行中の車に対する様々なセンサ攻撃の影響を評価可能にする。攻撃モデルは三菱電機独自のものであり、サポートするセンサ攻撃は表1に示したとおりである。

(注2) MATLAB, Simulinkは、The MathWorks, Inc.の登録商標である。

(注3) Unreal Engineは、Epic Games, Inc.の登録商標である。

4.2 シミュレーション評価の例

センサによる物体検出で、誤検出を排除するための処理であるM-out-of-Nアルゴリズムについて、センサ攻撃耐性を評価する例を示す。M-out-of-Nアルゴリズムは、連続するN回の検出結果のうち少なくともM回以上同じ物体を検出した場合に検出を確定させる。定性的にはMを大きくしてNに近付けるほど検出結果の確度は高くなるが、ノイズの影響を受けやすくなる。(M, N)はシステムが満たすべき性能要件によって決定される設計パラメータと考えられるため、センサ攻撃に対するシステムの安全性も考慮する必要がある。

走行シナリオの例として日本のNCAPのCPNO(Car-to-Pedestrian Nearside Obstructed)を用いて、図2で特定したレーダに対する電波妨害の攻撃シナリオを重ね合わせる(図5)。攻撃者の設定として、位置は自車両の前方に固定し、信号強度と初期位置を変化させて、システムの安全性を評価する。この評価を、異なる(M, N)に対して行い結果を比較する。(M, N)=(2, 2)と(9, 12)に対して評価した結果を図6に示す。まず、攻撃者が自車両に近いほど、また攻撃者の信号強度が強いほど衝突しやすくなるという妥当な結果が得られることを確認できる。次に、センサの設計パラメータに関して、(M, N)=(2, 2)の方が安全性は高いが、ブレーキが早すぎる、つまり、歩行者との衝突のおそれがないタイミングでブレーキがかけられて

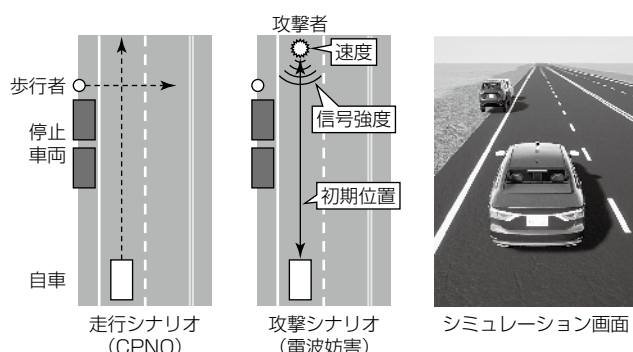


図5. CPNOシナリオでの電波妨害の例

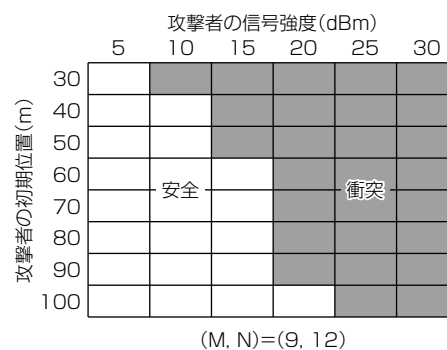
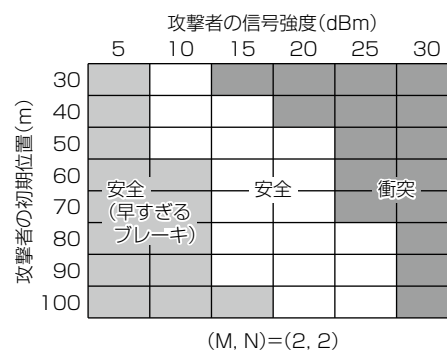


図6. 攻撃パラメータを変化させた場合の評価例

いるケースを確認できる。これは、図2に示したもう一つの安全制約“自車両が衝突しない状況下では、AEBによって減速しない”を考慮する必要があることを示している。

5. む す び

本稿では、自律システムのセンサ攻撃耐性を評価するためのフレームワークを提案し、レーダ、カメラ、LiDARに基づくAEB搭載車を例として、STAMP/STPA分析に基づくセンサ攻撃シナリオの特定と、シミュレータのプロトタイプによる評価の例を示した。自律システムが複雑化する中、今後は異なるシミュレータ間の相互接続が重要と考えられる。そこで、開発したセンサ攻撃モデルを他のシミュレータでも利用可能にするため、FMU(Functional Mock-up Unit)等の手法を用いたモジュール化を検討していく。

この研究の成果は国立研究開発法人 新エネルギー・産業技術総合開発機構(NEDO)の委託業務(JPNP16007)の結果得られたものである。

参 考 文 献

- (1) Nidhi, K., Paddock, S.M.: Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?. RAND Corporation (2016)
- (2) Nancy, L., John, T.: STPA HANDBOOK (2018)
- (3) ISO. Road vehicles-Functional safety. Standard ISO 26262:2018 (2018)
- (4) ISO. Road vehicles-Safety of the intended functionality. Standard ISO/FDIS 21448:2022 (2022)
- (5) ISO. Road vehicles-Cybersecurity engineering. Standard ISO/SAE 21434:2021(E) (2021)