実利用段階に入ったSTAMP/STPA
～Prologue to Autonomous System～

2018年12月3日

@NTT DATA 駒場研修センター

国立研究開発法人

宇宙航空研究開発機構

研究開発部門　研究領域総括・研究領域主幹

片平　真史/石濱　直樹

本日の内容

1. JAXAの技術研究・導入の動機
2. STAMP/STPA 導入手引き
   STPA Handbook の概説
3. Autonomous Systemの安全確保の試み

空へ挑み、宇宙を拓く

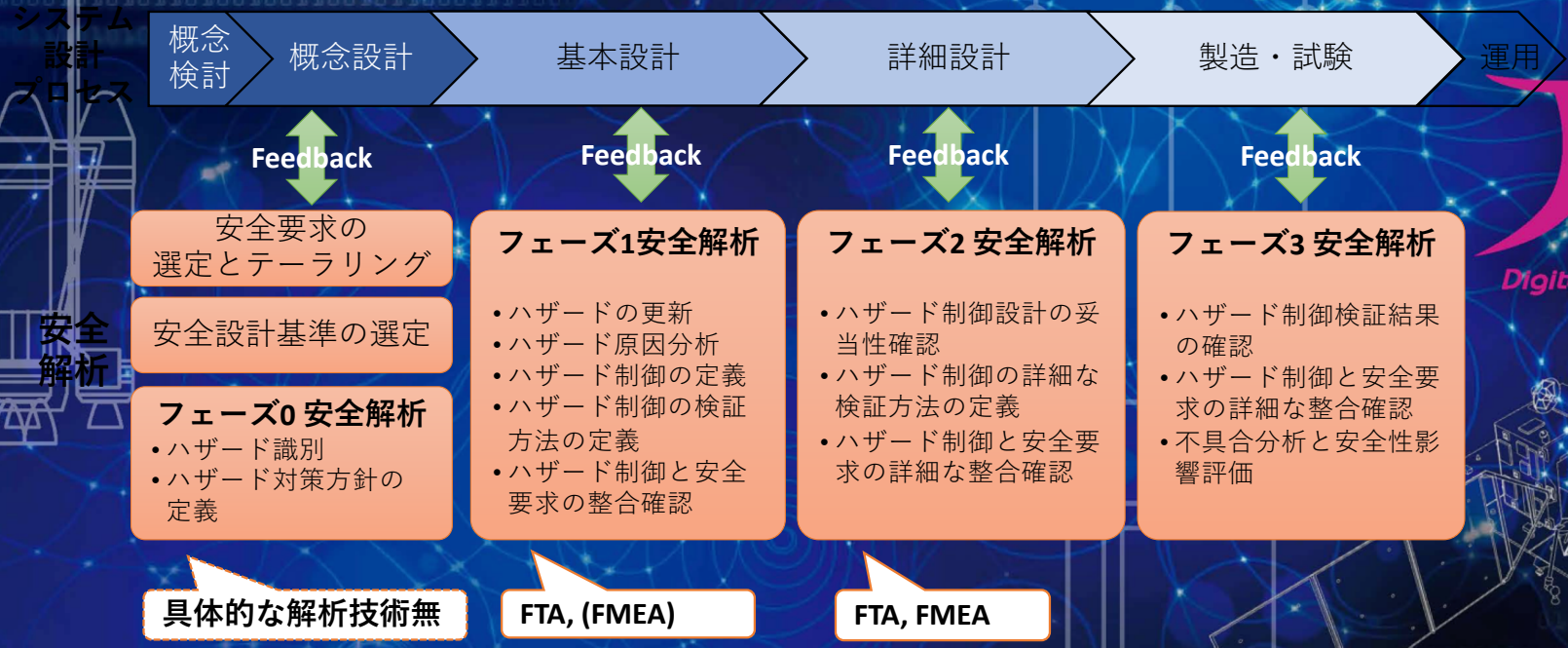
JAXA

1. JAXAの技術研究・導入の動機

JAXA's Engineering Digital Innovation Center

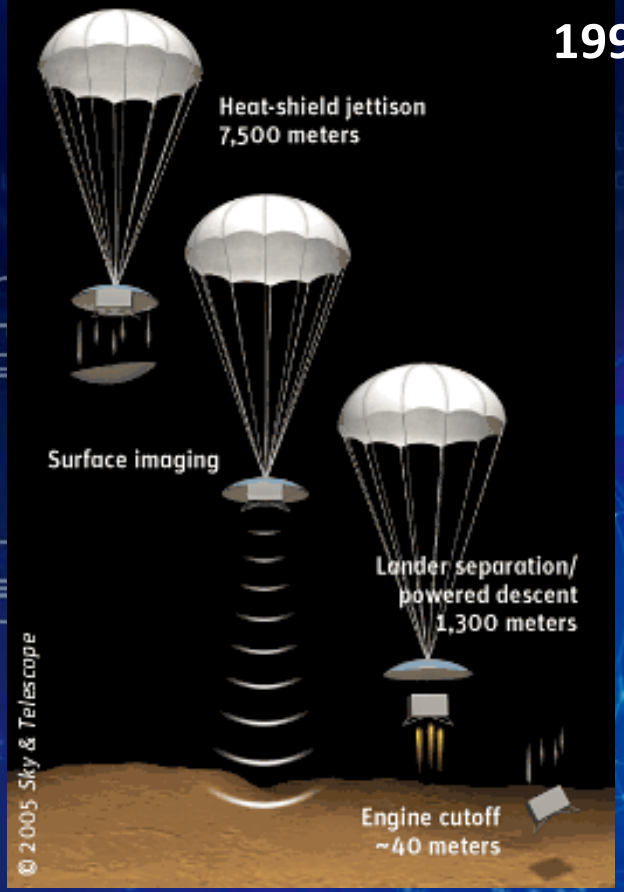
JAXAにおける安全設計プロセス（従来型）

- 安全化方針を各システム設計段階で分析し、システム設計に落とし込む
  - ハザード識別 → ハザード原因分析 → ハザード除去 or 制御設計 → 検証
  - 従来はFTAおよびFMEAを用いている



従来の安全設計プロセスの限界と課題

マースポーラランダーの事故 1999



故障はどこにもなかったが・・・

誘導制御ソフトウェアが降下用エンジンを40mで停止　→

事故

なぜ？

- センサが脚が展開されたときに、誤ったセンサ信号を出力。
- 地表40mで、ソフトウェアが着陸と認識し、エンジンを停止。

なぜ？

予算の圧縮、システム観点の不足、
検証の省力化、・・・

- End-to-End試験の不足
- ソフトウェアの要件定義のミス
  - 着陸検知ロジック（着陸直前に動く機能）が働くまでは、センサの値を無視する様に設計すべきだった・・・

このような設計解をどうやって早期に導くか？
コンポーネント間の相互作用を分析できる必要がある！！

非故障モードへの対応

事例:STPAとFTAの分析結果の特徴比較

	Identified by both STPA and FTA	Identified by STPA only
Controller	• ISS装置故障	• ISSクルーのコマンド発行時の誤操作 • ISSクルーのプロセスモデルの非一貫性
Activation Command	• アクチベーションコマンドの欠損/不適切	• アクチベーションコマンドの遅延
Controlled Process	• HTV装置故障 • HTV状態の時間経過による変化 • 物理的障害	• 範囲外の電波障害
Acknowledgment of Control Action		• t, xフィードバックの欠損/不適切 • t, xフィードバックの遅延 • t, xフィードバックの誤り • フライドモードフィードバックの欠損/不適切 • フライドモードフィードバックの誤り • 可視情報の欠損/不適切
Other Controllers		• JAXA/NASA GSからの間違った情報/指示

FTA、STPAとも
故障によるものは抽出できる

STPAだけが
タイミングなど非故障によるものを抽出できる

引用：**Hazard Analysis of Complex Spacecraft using Systems Theoretic Process Analysis** *AIAA Journal of Spacecraft and Rockets* , 2013.

新たな技法の研究・実用化

故障がない場合も含めた
安全に対する強固な解析体系が必要

STAMP/STPA

システムは何故失敗するのか？
失敗するのには理由がある
↓
それは安全制御(コントロールストラクチャー）の欠陥である
↓
その理由を見つけ、それが**起こらないような防御方法**を見つける

レジリエンス・エンジニアリング（FRAM）

システムは何故成功するのか？
成功するのには理由がある
↓
それはシステムの変化による適応の結果である
↓
その理由を見つけ強化する
それが**破たんするパターン（共鳴）**を見つける

それぞれの手法のメリット生かした統合解析体系を整える

STAMP/STPAの期待効果

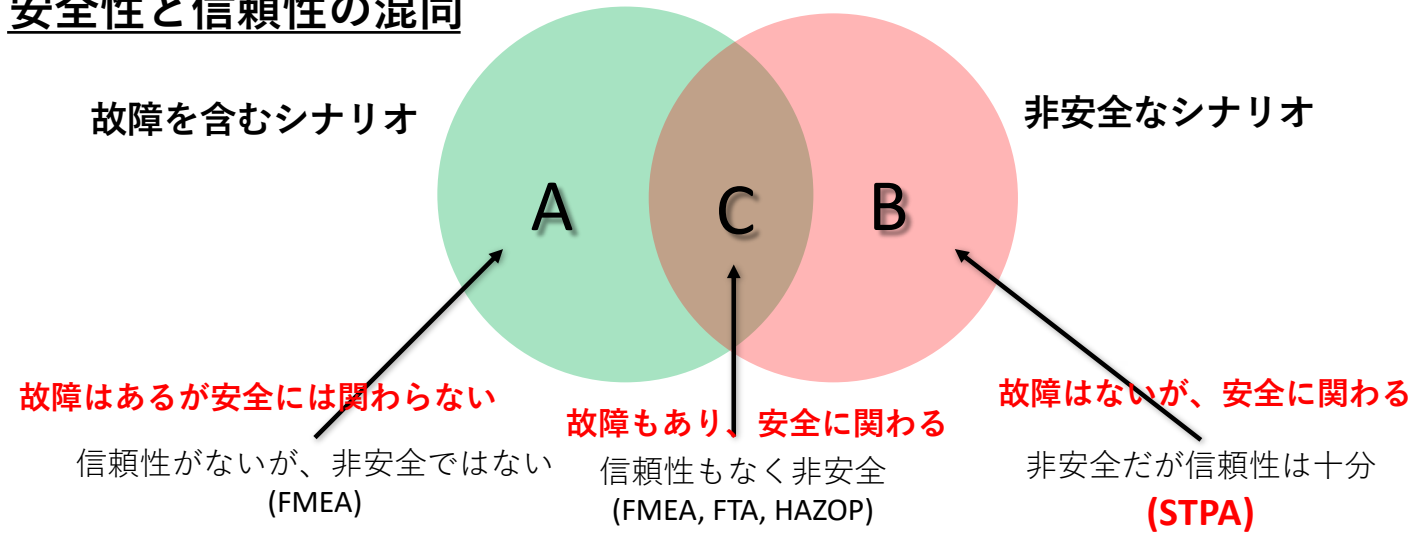
システム理論に基づく事故モデル: *WOCS 2016, Nancy Leveson Key Noteより*

STAMP (Systems-Theoretic Accident Model and Processes)

安全解析手法:

STPA (STAMP-Based Process Analysis)

安全性と信頼性の混同



コンポーネントや機能故障の防止だけでは十分でない

背景
～故障はないが安全に関わるとは？～

「もの」と「もの」とのコミュニケーションのやりとりが
うまくいなかった場合に問題（ハザード）に至る

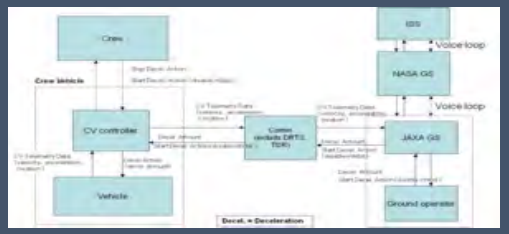
４つの側面

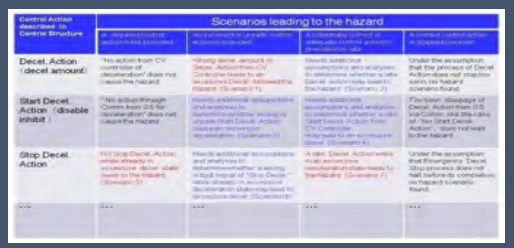
制御アクション	与えないとハザード	与えられるとハザード	早すぎ、遅すぎ、誤順序で、ハザード	早すぎる停止、長すぎる適用で、ハザード
コマンド1	来るべき制御アクションがないとどうなるか？	中身が誤った制御アクションが実行された場合にどうなるか？	意図しないタイミングで制御アクションが提供されるとどうなるか？	制御アクションが途中で止まってしまったり、長くかかってしまった場合はどうなるか？

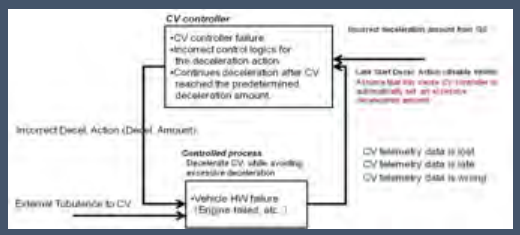
STAMP/STPAの方法論

システム理論に基づく事故モデル:
STAMP (Systems-Theoretic Accident Model and Processes)

ガイドワードを基に分析するのではない！







Step.0 ハザード制御に関わるControl Structureの作成

↓

Step.1 非安全なControl Actionの識別によるハザードシナリオの分析

↓

Step.2 Control Loopの作成によるハザード要因の分析

↓

Step.3 安全制約の識別

特徴

従来のハザード解析手法のような、個々のコンポーネント故障に着目するのではなく、複雑に連携するコンポーネント間の**相互作用**に関連するシステムレベルの問題に着目する

ガイドワードではない

４つの**側面**から、危険な状態を導くコントローラの動作（**非安全なコントロールアクション：UCA**）を識別する

Control Loop上の因果関係要因に注目し、UCAを識別する。特に、ソフトウェアやヒューマンに起因する要因として，コントローラの想定する**プロセスモデル**が，実際のプロセスの状態と矛盾することで起きる要因を識別する

ハザード要因を制御／除去するための安全制約を識別する

空へ挑み、宇宙を拓く

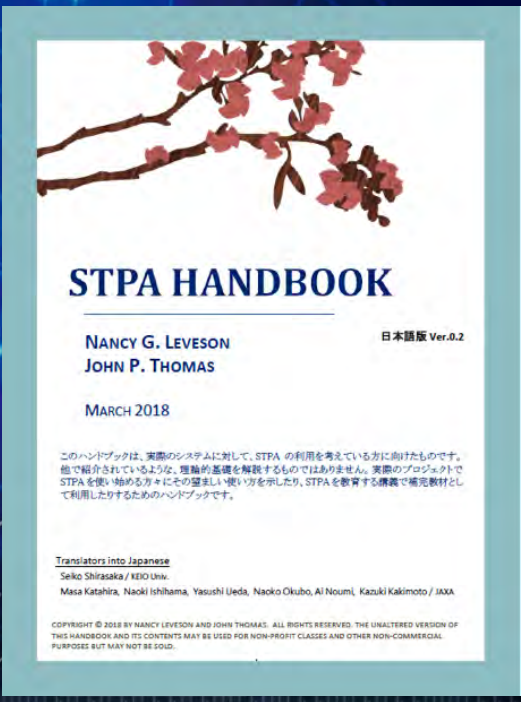
JAXA

2. STAMP/STPA 導入手引き
STPA Handbookの概説
(第2章より)

JAXA's Engineering Digital Innovation Center

STPA HANDBOOK

- 原書 （188頁） MIT Nancy Leveson, John Thomas 2018年3月
- 日本語版 （186頁） 2018年3月
  ボランティア翻訳 慶應大 白坂教授、JAXA 片平･石濱･植田･大久保･能美･柿本
- 英語版･日本語版とも
  http://psas.scripts.mit.edu/home/materials/ にて無料配布



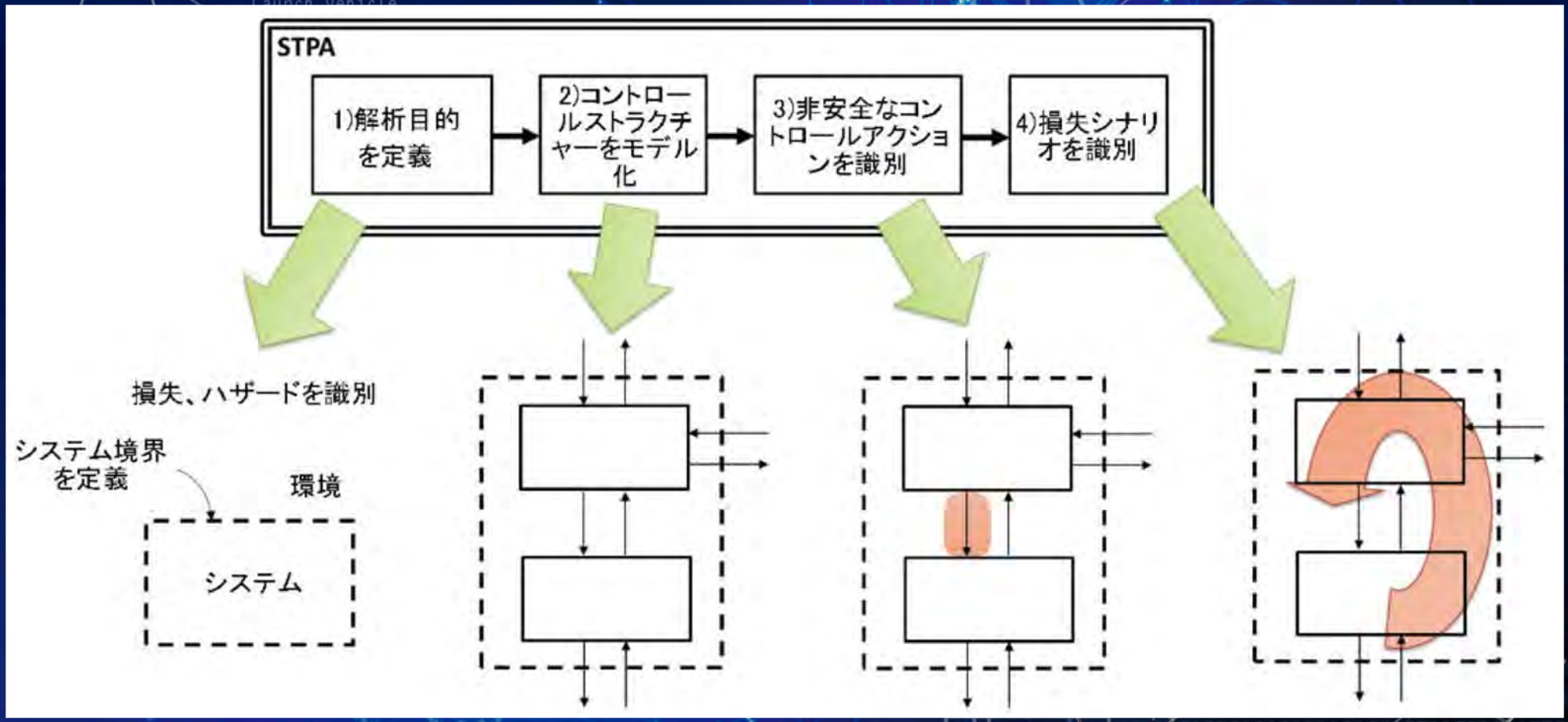
□STPAを使い始めたい方、単純なハザード解析でない目的で使用したい方向け

□多くの例が付録に掲載

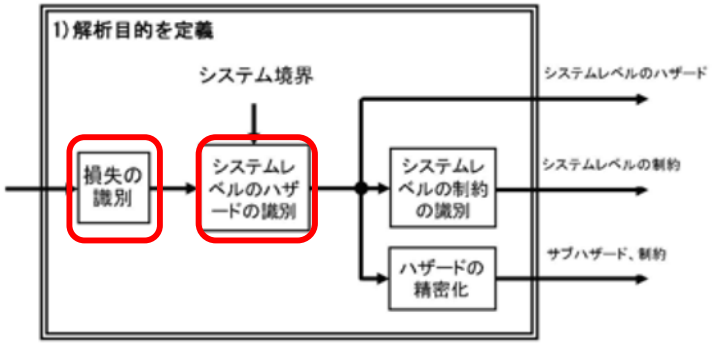
□ハンドブックの使用に必要となる、基本的な工学的概念を付録に解析

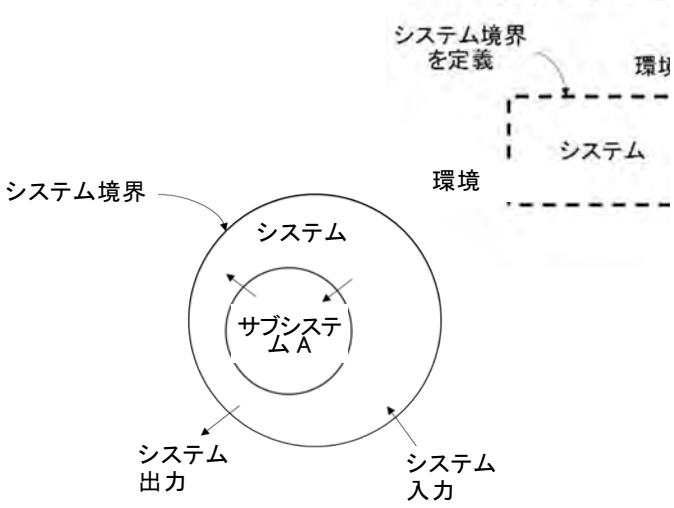
□ハンドブックの本編には、安全の新しい概念を紹介

STPAメソッド　基本ステップ



STPAメソッド　1）解析目的を定義





損失(Loss)の識別

損失は、利害関係者にとって価値のあるものに関わる。損失には、利害関係者に受け入れられないような、人命の喪失や人間の傷害、物的損害、環境汚染、ミッションの喪失、評判の喪失、機密情報の喪失もしくは漏洩、またはその他の損失が含まれる。**STPAの目的は損失を防ぐこと。**

例：**L-1:** 人命の喪失、**L-2:**車両の損害、 **L-4:**ミッションの喪失、**L-5:**顧客満足度の損失、 **L-6:**機密情報の喪失、等

システムレベルのハザードの識別

ハザードとは、一組の最悪ケースの環境条件で、**損失につながる**ような、システム状態、または、一組の条件である。

システムとは、いくつかの共通の目標、目的、または、その達成するために、一体として合わせて動くコンポーネントの集合である。システムには、サブシステムを含んでもよく、また、より大きなシステムの一部であってもよい。

例：**H-1:** 航空機が最小間隔の基準に違反する**[L-1,L-2,L-4,L-5]**

H-6:車両が障害物から安全な距離を保てない、

H-8:原発が危険な物質を放出する、等

STPAメソッド　1）解析目的を定義

システムレベルのハザードを識別する時の共通的な間違い

◆ハザードの原因とハザードの混同

◆不必要な詳細を含んだ多すぎるハザード

◆曖昧な表現　H-1: 航空機が<非安全な>飛行を行う[L-1]

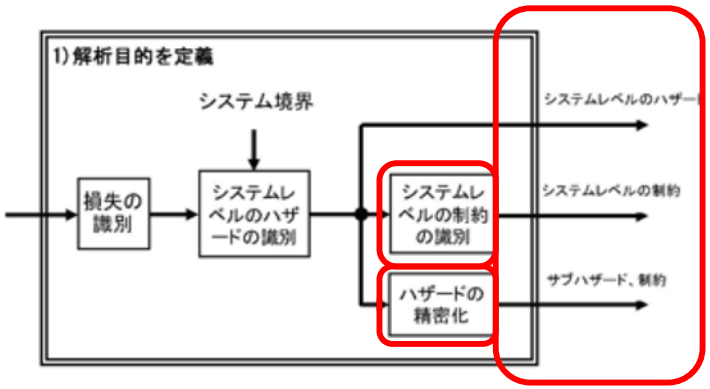
◯ H-1: 航空機が最小間隔の基準に違反する[L-1,L-2,L-4,L-5]

◆故障とハザードの混同

ハザードを識別する際の共通的なミスを防ぐためのヒント

- ハザードは、システムの個々のコンポーネントを参照すべきではない
- すべてのハザードは、システム全体およびシステムの状態を参照すべき
- ハザードは、システム設計者およびオペレータによってコントロールできる、またはマネージできる要因を参照する必要がある
- すべてのハザードは、それを防止するためのシステムレベルの条件を記述する必要がある
- ハザードの総数は、通常は7から10以上にせず、比較的小さくすべき
- ハザードは、「非安全な」、「意図しない」、「偶然の」などのような、あいまいな、または再帰的な言葉を含めるべきではない

STPAメソッド　1）解析目的を定義



システムレベルの(安全)制約の識別

ハザードを防ぐ（そして最終的に損失を防ぐ）ために満たす必要があるシステムの条件や動作を特定する。

H-1： 航空機が最小間隔の基準に違反する[L-1, L-2, L-4, L-5]

SC-1：航空機が他の航空機や物体からの最小間隔の基準を満足しなければならない[H-1]

SC-3：航空機が最小間隔に違反した場合に、違反が検出され、衝突を防ぐために対策が取られなければならない

注意：特定の解決策を初期段階に指定すると、後に潜在的に優れた解決策を見落とす

システムレベルのハザードの精密化（オプション）

必要な場合、サブハザードに精密化し、対応する制約を識別する。

例：H-4：航空機が、地上で他の物体に接近しすぎる[L-1、L-2、L-5]

サブハザード	制約の例
H-4.1：減速が、着陸時、離陸中止、または地上走行中に、不十分である	SC-6.1：減速は、着陸のTBD秒以内に、または最低TBD m/s^2で離陸中止する際に、行わなければならない
H-4.3：減速が、離陸時のV1ポイント後に発生する	SC-6.3：減速は、離陸時のV1ポイント後に与えられてはならない

STPAメソッド　2）コントロールストラクチャーをモデル化

混乱の共通点

- コントロールストラクチャーは、**物理モデルではない　→機能モデル**
- コントロールストラクチャーは、**実行可能モデルではない**（生成するために使用はできる）
- コントロールストラクチャーは、**従うことを想定していない（常に守られるという意味ではない）**
- 複雑さを管理するためには**抽象化を用いる**

モデル化する時の共通質問

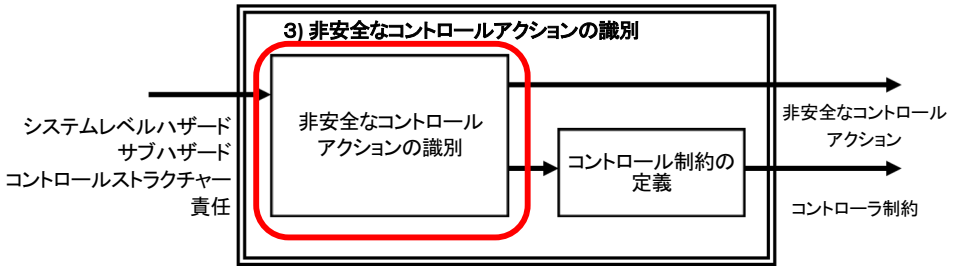
- コントロールストラクチャーは、先に進める前に完全である必要があるか？
- 矢印のラベルをどのように特定すべきか？
- コントロールストラクチャーには、全てのアクチュエータやセンサを入れるべきか？
- 物理的プロセス及び物理的相互作用をどのようにコントロールストラクチャーに入れ込むか？
- コントロールストラクチャーは、線形階層を必要としているか？
- どのようにして誰が誰をコントロールするのかを把握するか？
- コントロールストラクチャー図以外のものを文書化する必要があるか？

STPAメソッド　2）コントロールストラクチャーをモデル化

コントロールストラクチャーにおける共通の間違いを防ぐためのヒント

- ラベルが特定の物理的な実装ではなく、送られた機能的な情報を説明していること
- 情報の種類がわかっている場合には、単に「コマンド」または「フィードバック」のような曖昧で漠然としたラベルを避ける
- すべてのコントロール対象のプロセスは、1つ以上のコントローラによってコントロールされていること（必ずしも必要ではないが、しばしば間違いを指摘できる）
- 競合とのギャップのために（トレーサビリティを含む）責任をレビューする
- 責任を満たすために必要なコントロールアクション、フィードバックが含まれていること
  フィードバックが不明な段階（コンセプト開発に適用する場合）はオプション、のちのステップで欠落しているフィードバックを識別できる

STPAメソッド　3) 非安全のコントロールアクション(UCA)を識別



非安全なコントロールアクション(UCA)の４つの側面

1.コントロールアクションを与えないことがハザードにつながる
2.コントロールアクションを与えることがハザードにつながる
3.潜在的には安全なコントロールアクションを与えるが、
　早過ぎる、遅過ぎる、または間違った順序である
4.コントロールアクションがあまりにも長く続いている（連続コントロールアクションであり離散的なものではない）あるいは、あまりにも早く止まる

BSCU（ブレーキシステム制御ユニット）の例

コントロールアクション	与えられないとハザード	与えられるとハザード	早過ぎ、遅過ぎ、誤順序	早過ぎる停止、長過ぎる適用
ブレーキ	UCA-1： 着陸滑走中BSCUが作動している時、BSCU自動ブレーキがブレーキコントロールを出さない[H-4.1]	UCA-2： 通常離陸中にBSCU自動ブレーキがブレーキコントロールアクションを出す[H-4.3, H-4.6] UCA-5：着陸滑走中にBSCU自動ブレーキが、不十分なブレーキレベルでブレーキコントロールアクションを出す [H-4.1] UCA-6： 着陸滑走中にBSCU自動ブレーキが、方向性のある、または非対称のブレーキとなるブレーキコントロールアクションを出す [H-4.1, H-4.2]	UCA-3： 着陸後BSCU自動ブレーキがブレーキコントロールアクションを出すのが遅過ぎる(>TBD 秒) [H-4.1]	UCA-4： 着陸時にBSCU自動ブレーキが、ブレーキコントロールアクションを止めるのが早過ぎる(TBDタクシー速度に達する前) [H-4.1]

STPAメソッド　3) 非安全のコントロールアクション(UCA)を識別

UCAに関する一般的な質問

- UCAはハザードが常に発生することを保証しているか？　　→いいえ
- 既にセーフガードがある時、UCAを識別することがあるか？
  → STPAは、最悪の場合の解析方法であり、セーフガードが存在する時もUCAを省略しない
- UCAの最後の２種類は、両方ともタイミングである。この違いは何か？
  →３つ目は、早過ぎる、遅過ぎる、順序が違うといった、間違ったタイミングで与えられたコントロールアクション（CA）
  ４つ目は、ある継続時間でのみCAが適用されるもの、すなわち、連続または非ディスクリートなCA
- UCAの種類ごとに、正確に一つのUCAを特定する必要があるか？
  →いいえ、4種類すべてを考慮するべきではありますが、全ての場合にあるわけではない
- UCAには４つ以上種類があるか？別のカテゴリが必要か？
  →この4つのカテゴリは、おそらく全部揃っている。UCAには他のカテゴリは必要ない
  →しかし、サブカテゴリが考えられる。例えば、2. 与えるとハザード：
  アクションが決して安全ではないかもしれない状態/不足や過度のアクションが非安全になる状態/アクションの方向が非安全になる状態
- いずれのシステムレベルのハザードにも関連していない、重要なUCAを特定した場合、どうすれば良いか？
  →何かハザードを見逃している可能性がある

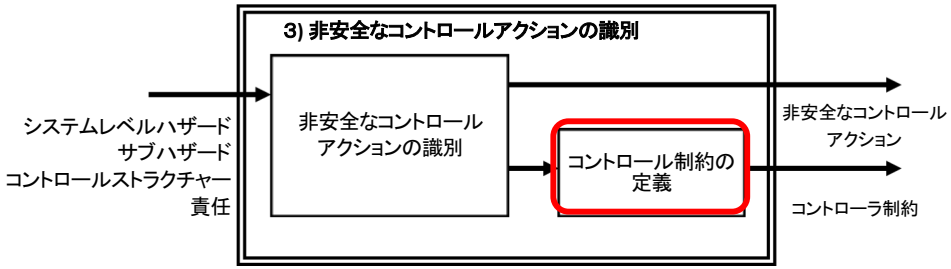
正しいUCA： 通常離陸時にBSCU自動ブレーキがブレーキコマンドを与える [H-4.3]
誤ったUCA： BSCU自動ブレーキはブレーキコマンドを出して衝突が生じる

STPAメソッド　3) 非安全のコントロールアクション(UCA)を識別

UCAを識別する際の一般的なミスを防ぐためのヒント(確認すべき事項)

- すべてのUCAが、コントロールアクションが非安全な状態になるコンテキストを特定すること
- UCAのコンテキストが、実際の状態についての潜在的に信じていることではなく、非安全にさせる実際の状態や条件を特定すること
- UCAのコンテキストが、明確に定義すること
- すべてのUCAが、1つ以上のハザードとリンクするようにトレーサビリティが文書化すること
- N/Aとなっているコントロールアクションの種類をすべてレビューして、それを回避すること
- パラメータを持つ連続コントロールアクションすべてに対して、それらのパラメータの過剰、不足や方向の間違いを考慮すること
- UCAの背後にある仮定や特別な推論が、すべて文書すること

STPAメソッド　3) 非安全のコントロールアクション(UCA)を識別



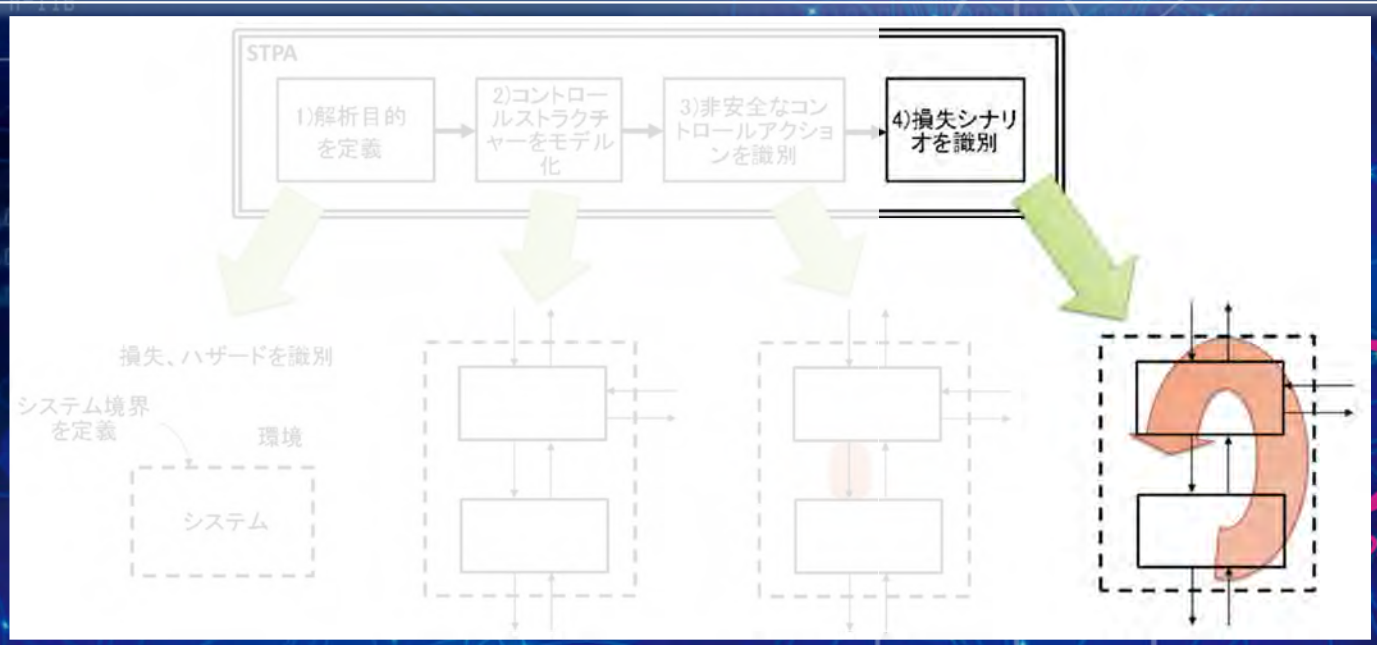
コントロール制約

コントローラの制約は、UCAを防ぐために満足されるべき
コントローラの動作を特定する

*UCAが識別されると、各コントローラの動作上の**制約**に変換することができる*

非安全なコントロールアクション	コントローラ制約
UCA-1： 着陸滑走中BSCUが作動している時、BSCU自動ブレーキがブレーキコントロールアクションを出さない[H-4.1]	C-1： 着陸滑走中BSCUが作動している時、BSCU自動ブレーキは、ブレーキコントロールアクションを**出さなければならない** [UCA-1]
UCA-2： BSCU自動ブレーキが通常離陸中にブレーキコントロールアクションを出す [H-4.3, H-4.5]	C-2： BSCU自動ブレーキは、通常離陸中にブレーキコントロールアクションを**出してはならない** [UCA-2]
UCA-3：着地後にBSCU自動ブレーキがブレーキコントロールアクションを出すのが遅すぎる（> TBD秒）[H-4.1]	C-3： 着地後に**TBD秒以内に、**BSCU自動ブレーキは、ブレーキコントロールアクションを出さなければならない [UCA-3]

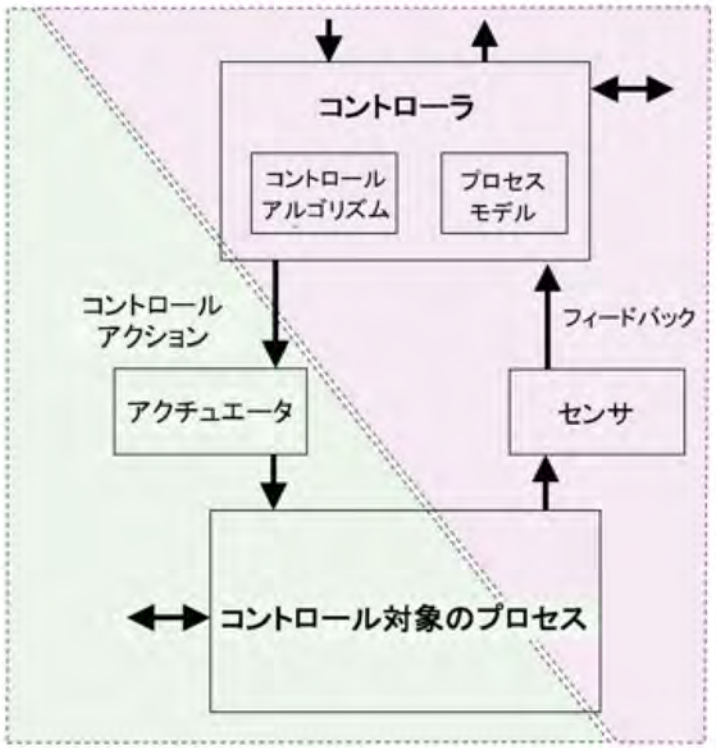
STPAメソッド　4）損失シナリオを識別



STPAメソッド　4) 損失シナリオの識別

a) なぜ、非安全なコントロールアクションが起こるのか？

b) なぜ、コントロールアクションは、不適切に実行される、または実行されないのか？

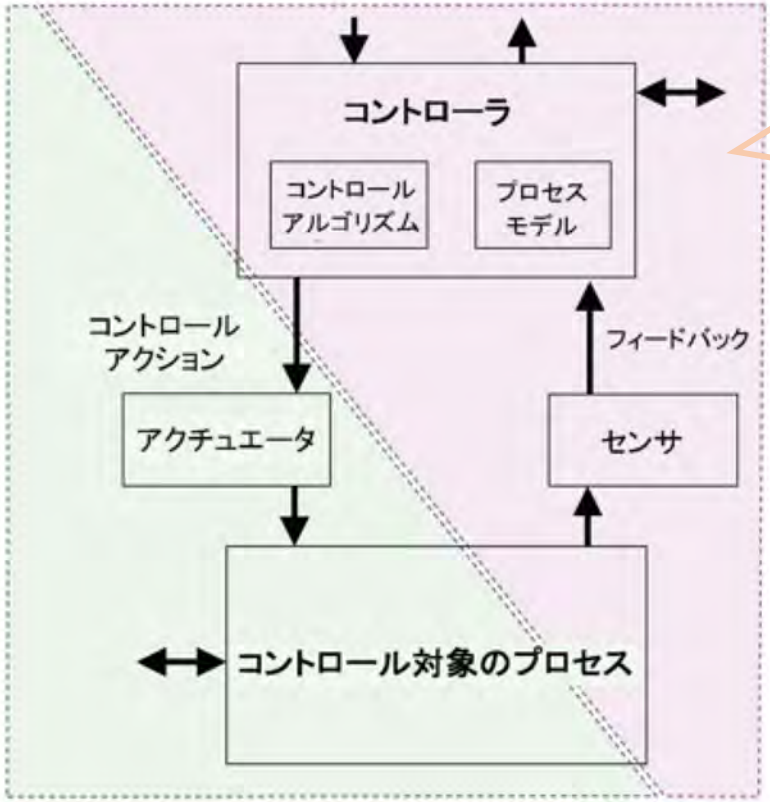


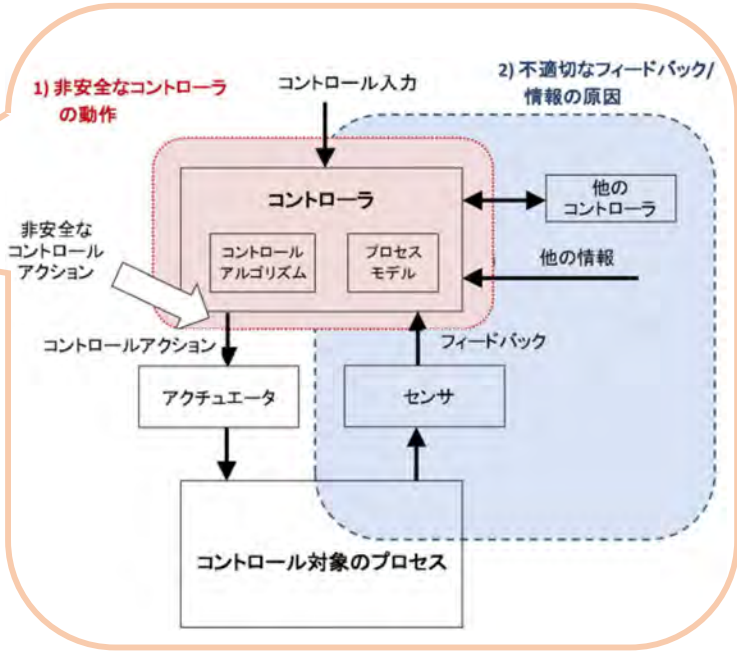
- コントローラに関連する故障（物理的なコントローラに対する）
  - コントローラ自体の物理的な故障
  - 電源の故障、など
- 不適切なコントロールアルゴリズム
  - コントロールアルゴリズム仕様の欠陥のある実装
  - コントロールアルゴリズム仕様に欠陥がある
  - コントロールアルゴリズム仕様は、時が経つにつれて、変更または劣化して不十分となる
- 非安全なコントロール入力
  - 別のコントローラから受けとったUCA（他のコントローラからのUCAを考慮する際に既に抽出されたもの）
- 不十分なプロセスモデル
  - コントローラが、間違ったフィードバック/情報を受けとる
  - コントローラが、正しいフィードバック/情報を受けとるが、誤って解釈する、または、それを無視する
  - コントローラが、必要な時に、フィードバック/情報を受けとらない（遅れる、または、全く受けとらない）
  - 必要なコントローラフィードバック/情報が存在しない

STPAメソッド　4) 損失シナリオの識別

a) なぜ、非安全なコントロールアクションが起こるのか？

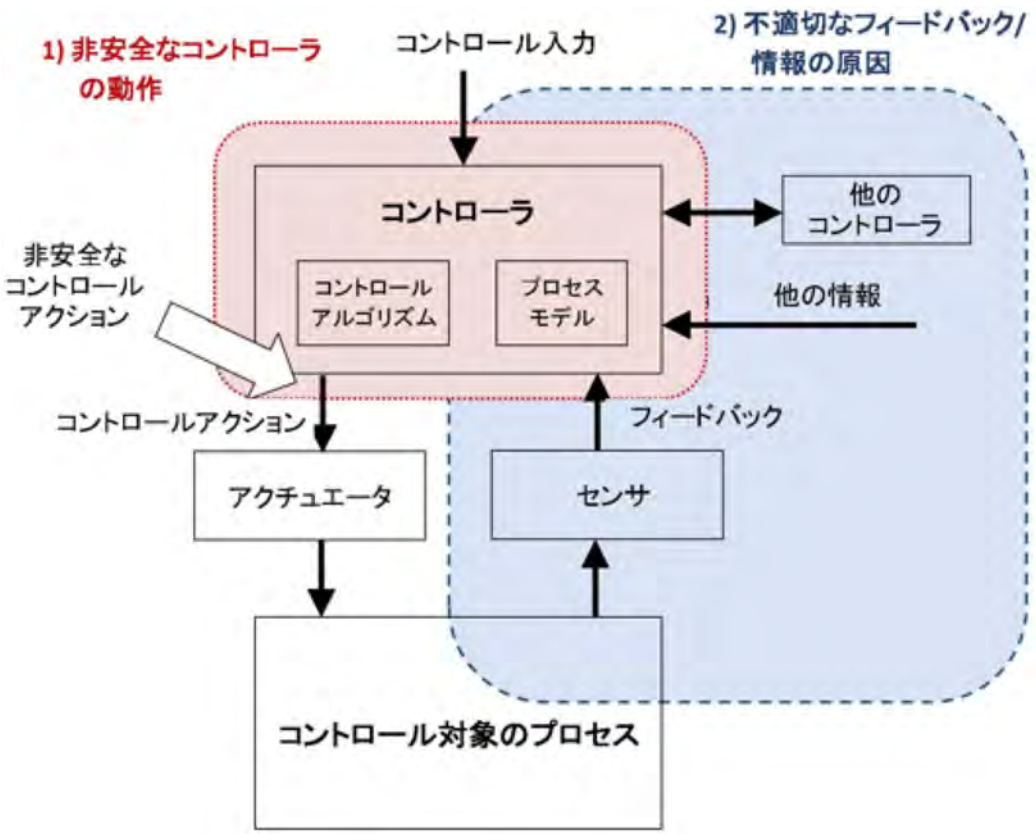
b) なぜ、コントロールアクションは、不適切に実行される、または実行されないのか？





STPAメソッド　4）損失シナリオの識別

a) 非安全なコントロールアクションにつながるシナリオ



1) 非安全なコントローラの動作

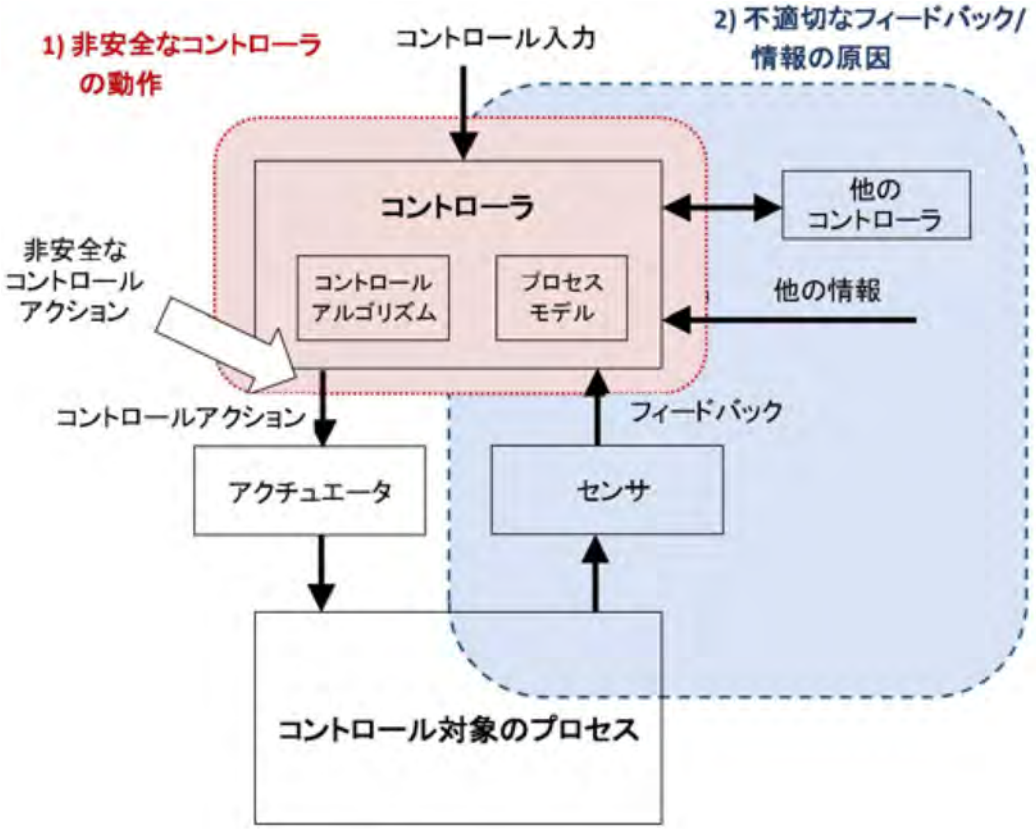
- コントローラを含む故障（物理的な）
- 不適切なコントロールアルゴリズム
- 非安全なコントロール入力（別のコントローラから）
- 不適切なプロセスモデル
  コントローラのプロセスモデルが現実とは一致しない場合

UCA-1： 着陸滑走中にBSCUが作動している時、BSCU自動ブレーキがブレーキコントロールアクションを与えない[H-4.1]

UCA-1のシナリオ1：BSCU自動ブレーキ物理コントローラが、ブレーキコントロールアクションが与えないことの原因として、着陸滑走中にBSCUが作動している時、故障する[UCA-1]。結果として、着陸時には不十分な減速が与えられるかもしれない[H-4.1]

STPAメソッド　4) 損失シナリオの識別

a) 非安全なコントロールアクションにつながるシナリオ



2）不適切なフィードバック/情報の原因

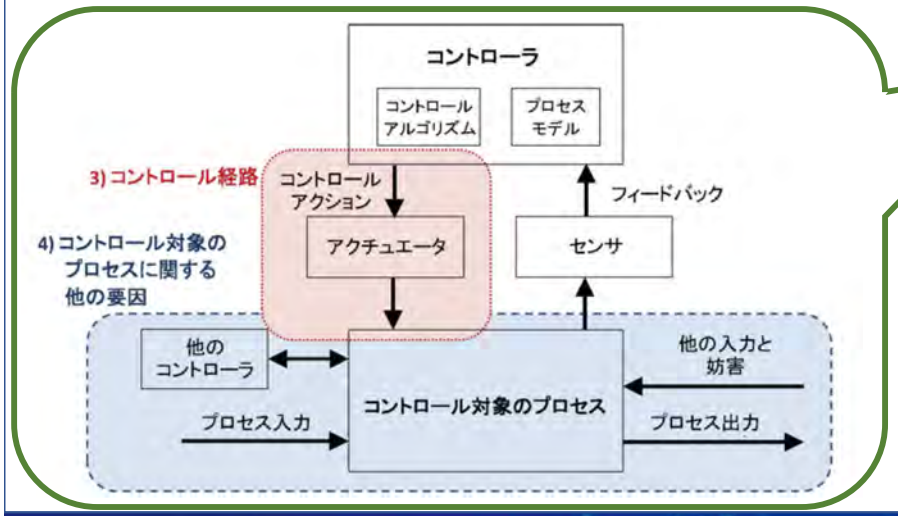
■ フィードバックや情報を受信しない
- センサはフィードバック/情報を送信したが、コントローラが受信しない
- センサはフィードバック/情報を送信していないが、受信される
- フィードバック/情報が受信されず、またはセンサに適用する
- フィードバック/情報はコントロールストラクチャーに存在しない、またはセンサーが存在しない

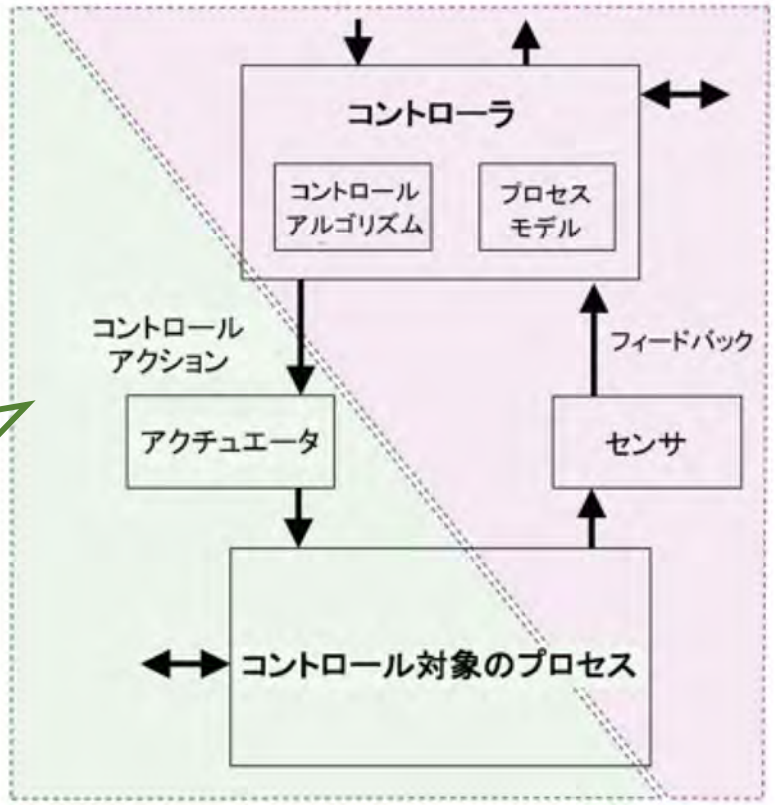
■ 不適切なフィードバックを受信する
- センサは適切に応答しているが、コントローラが不適切なフィードバック/情報を受信する
- センサが、（受信される、またはセンサに適用する）フィードバック/情報に対して不適切に応答する
- センサに必要なフィードバック/情報を提供するような性能がない、あるいは、そのように設計されていない

STPAメソッド　4) 損失シナリオの識別

a) なぜ、非安全なコントロールアクションが起こるのか？

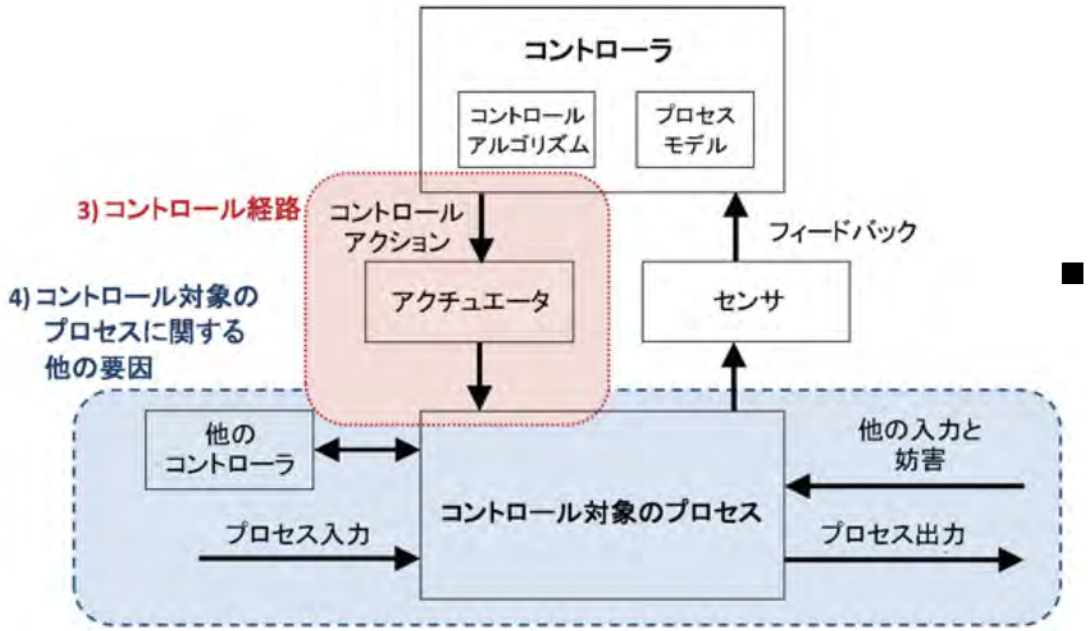
b) なぜ、コントロールアクションは、不適切に実行される、または実行されないのか？





STPAメソッド　4) 損失シナリオの識別

b)コントロールアクションが不適切に実行される又は実行されないシナリオ

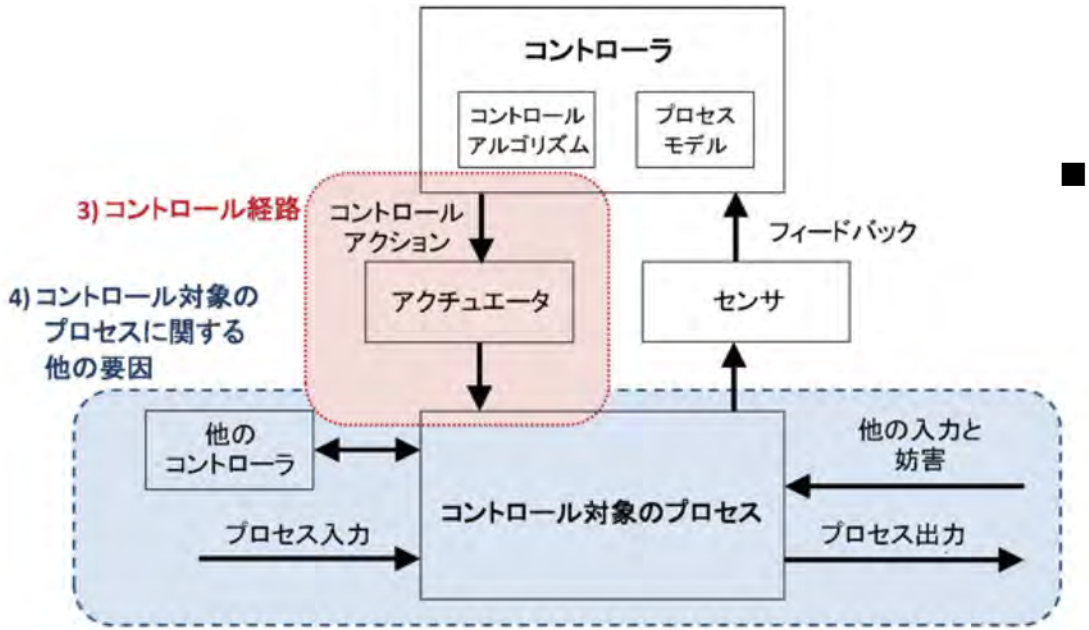


3) コントール経路

- コントロールアクションが実行されない
  - コントローラはコントロールアクションを送信したが、アクチュエータが受信しない
  - コントロールアクションをアクチュエータは受信したが、アクチュエータが応答しない
  - アクチュエータが応答するが、コントロールアクションが適用されない、あるいはコントロール対象のプロセスが受信しない
- コントロールアクションが不適切に実行される
  - コントローラはコントロールアクションを送信したが、アクチュエータは不適切な受信をする
  - コントロールアクションをアクチュエータが受信したが、アクチュエータの不適切に応答する
  - アクチュエータが適切に応答するが、コントロールアクションが不適切にコントロール対象のプロセスで適用または受信される
  - コントロールアクションをコントローラが送信していないのに、まるで既に送信しているかのようにアクチュエータまたは他の部分が応答する

STPAメソッド　4）損失シナリオの識別

b)コントロールアクションが不適切に実行される又は実行されないシナリオ

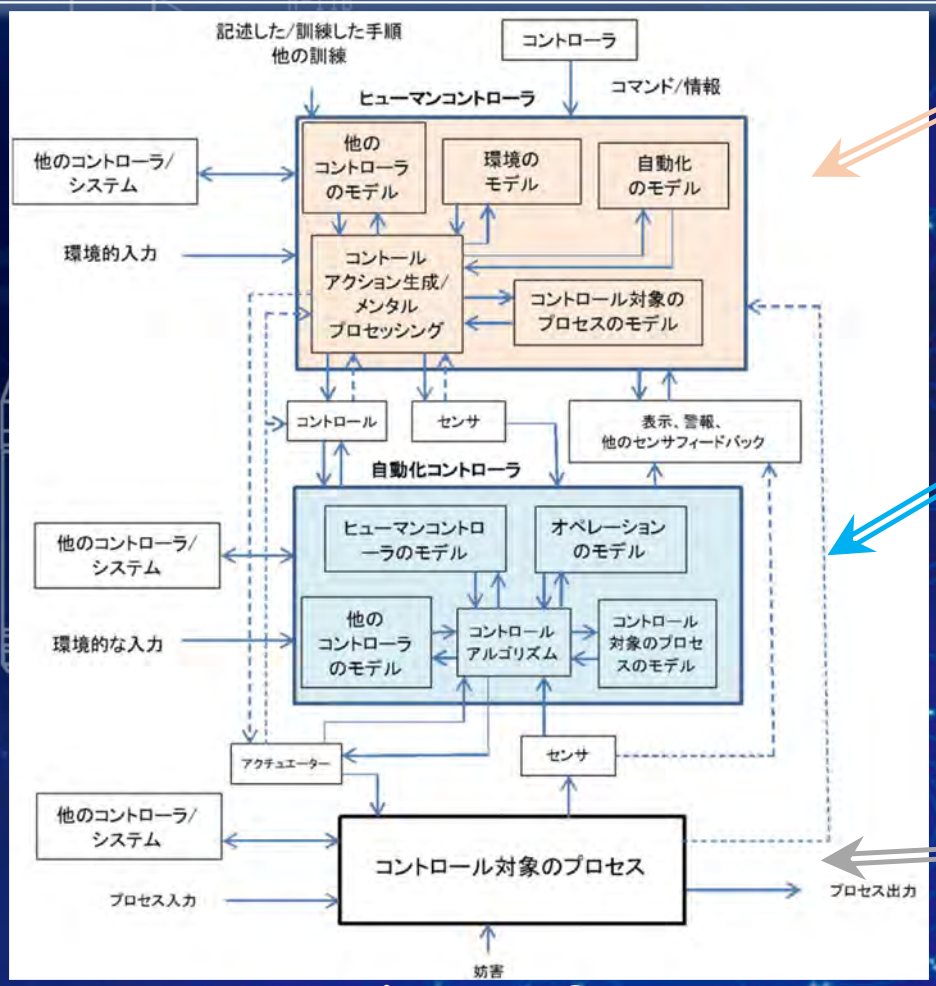


4）コントロール対象のプロセスに関する他の要因

■ コントロールアクションが実行されない
- コントロール対象のプロセスによって、コントロールアクションが適用される、または受信されるが、コントロール対象のプロセスは応答しない

■ コントロールアクションが不適切に実行される
- コントロール対象のプロセスによって、コントロールアクションが適用される、または受信されるが、コントロール対象のプロセスは不適切に応答する
- コントロール対象のプロセスによって、コントロールアクションが適用される、または受信されるが、まるで、コントロールアクションが適用または受信されたかのようにプロセスが応答する

因果関係Causal Factor シナリオ生成のための抽象モデル



ヒューマンコントローラ

- *コントロールアクションの生成／メンタルプロセッシング*
- *ヒューマンコントローラが使用するメンタルモデル*

自動化コントローラ

- *自動化コントローラからコントロールプロセスへの、アクチュエータを介したコントロール経路*
- *コントロール対象のプロセスからセンサを介する自動化コントローラへのフィードバック経路*
- *自動化コントロールアルゴリズム*
- *コントロール対象のプロセスのモデル*
- *オペレーションモードのモデル*
- *ヒューマンコントローラのモデル*
- *他のコントローラのモデル*
- *環境的入力*
- *他のコントローラ/システムのモデル*
- *オートメーションとその監督者の間の情報の伝達*
- *コントロールアルゴリズムを経由しない自動コントローラへの直接的な変更*

コントロール対象のプロセス

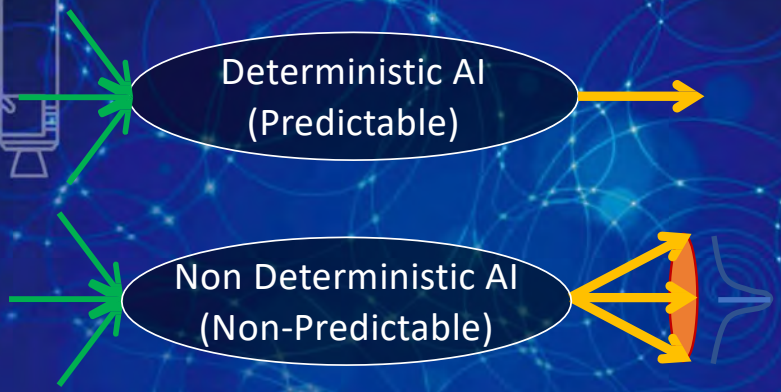
- *コントロール対象のプロセスコンポーネントの経年による故障あるいは劣化*
- *外乱*
- *コントロール対象のプロセスへの直接入力*
- *コントロール対象のプロセスのコントローラ*

3. Autonomous System の安全確保の試み

Autonomousシステム

- 人工知能などを用いたAutonomousシステムとなると
  - 機械学習、ニューラルネット、Deep Learning…
- 挙動の予測が難しい・わからない・・・
- まさにブラックボックスシステム・・・
- そんな状況で安全なシステムと言えるのか・・・？？？？

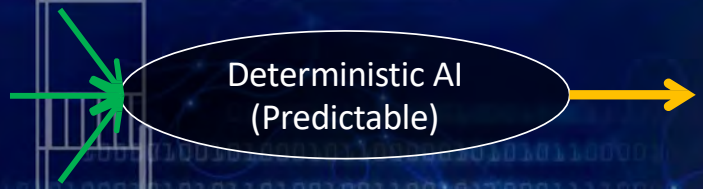
Autonomousシステムのタイプ



予測可能な出力
入力に対して再現性のある出力
入力に対して、一定の処理結果により挙動が決まり、結果が推測できるもの

予測困難な出力
入力に対して再現性のない出力
入力に対して、非決定論的動作をし、結果の推測が困難なもの

予測可能なシステムの動作保証



予測可能な出力

入力に対して再現性のある出力

入力に対して、一定の処理結果により
挙動が決まり、結果が推測できるもの

該当するシステム

- 知識ベース
- 最適化アルゴリズム(GA含む)
- データマイニング(成長なし)
- スパース推定

特徴

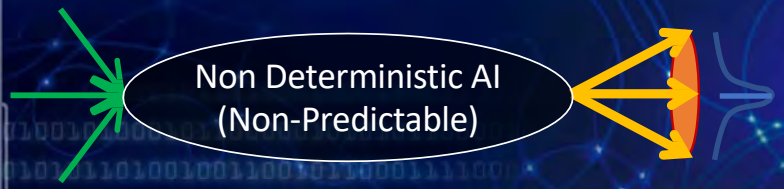
- 時間によらずデータに基づき再現性がある
- 予測は可能である

安全検証基本アプローチ

従来型の安全検証を踏襲する

- ハザード解析などにより、非安全状態を識別し、安全のためのクライテリアを満足するか試験等検証で確認し、IV&Vなどでクロスチェックする。
- FTA、SFTA、HOZAP、STPA、IV&Vなど。

予測困難なシステムの動作保証



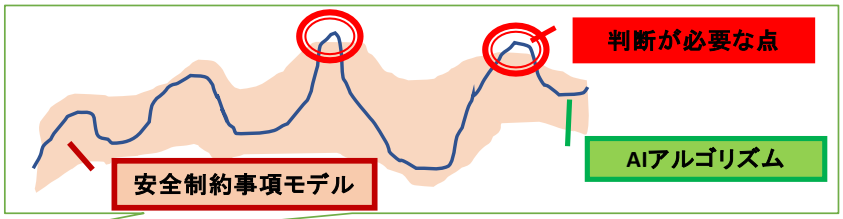
予測不可能な出力
入力に対して再現性のない出力
入力に対して、非決定論的動作をし、結果の推測が困難なもの

該当するシステム
- 人間
- ニューラルネット
- データマイニング(成長あり)

特徴
- 再現性が困難である
- 出力が一意に決まらない
- 予測が困難である

安全検証基本アプローチ
安全のためのクライテリアを一意に決定できないため、従来型の安全検証の方法では検証が困難となる。
しかし、システムの挙動は不確定を含むものの、その出口はある範囲内に包絡されると考えられるために、包絡域が安全包絡域の範囲内であることを立証する。
- STPA (SC領域分析)/FRAM(共鳴解析)+実証試験
- 自動車の場合、現行の人間の操作の不確定性を考慮した安全かシナリオが使える

予測困難なシステムの安全保証



AIアルゴリズム
安全制約事項モデル

AIアルゴリズムが安全制約の範囲内であれば正しいと判断する。
もし範囲を逸脱する場合は、無視していい値か、異常な事態になっているかを判断し、対処をする。
（局所解に陥ることがあるため、それは無視すれば成立することがあり得るため）
⇒（課題）ワーストケースをどう説明できるかがポイント

AI時代の制御システムの品質保証

- 対象：AIによる自律化システム（特にDeep Learningなど予測困難なシステム）
- プロダクト要件、プロセス要件、説明責任

プロダクト要件

<視点＞
全体システムが如何に安全に安全な挙動をするか？
<検討要素＞
・設計指針（安全技術要求、パターン）
　　安易な冗長要求は避ける必要あり
・安全解析・技術・評価（含、STAMP/STPA，FRAM）
・検証要件（方法、ケース）

プロセス要件

<視点＞
AIシステムが正しく作られているか？

<検討要素＞
・AIシステムのあるべき開発プロセス
アルゴリズム構築、学習データ最適化プロセス
・プロセス評価？

説明責任

<視点＞
如何に安全であるかを説明できるか？

<検討要素＞
・ルール、社会コンセンサス
・プロダクト要件、プロセス要件の客観的立証GSN

Deep Learningを搭載した自律ローバーを題材とした STAMP/STPAの適用事例

STEP1:　UNSAFE CONTROL ACTIONS（非安全なコントロールアクション）の抽出例

コントローラ	制御アクション	与えないとハザード	与えられるとハザード
オートパイロット	進路操作	UCA4: ローバーから対象物（障害物）までの距離が、限界を超えてしまう際に、オートパイロットが進路変更をしない。	–

重要

センサーの誤認識・予測ミスが最終的に問題となる

- **非安全なコントロールアクション（UCA）：**
  - 前方に障害物があるのに、オートパイロットが、直進走行をしてしまう。
- **シナリオ：**
  - カメラで環境情報を認識した際に、障害物があるのにも関わらず誤って障害物がないと判断してしまった。そのためオートパイロットは、直進してしまう（Deep Learningが誤認識してしまう）。その結果、障害物に衝突してしまう。

- **非安全なコントロールアクション（UCA）：**
  - 前方に急坂があるのに、オートパイロットが、直進走行をしてしまう。
- **シナリオ：**
  - カメラとレーダーで環境情報を認識した際に、前方が下り坂であるのにも関わらず、誤って緩やかな上り坂と判断してしまい直進可能と判断してしまう（Deep Learningが誤認識してしまう）。その結果、急な下り坂に直進してしまいコントロールを失いバランスを崩しレゴリス（砂）につかまってしまい走行不可能になってしまう。

最後に

■本日は、以下を紹介
- □JAXAの技術研究・導入の動機
- □STAMP/STPA 導入手引き（STPA Handbook の概説）
- □Autonomous Systemの安全確保の試み

■まとめ
- □いよいよ実用段階に！　MIT STPA Handbook活用のススメ
- □実用のためには、
  - ✓STPA 解析を行うことが目的ではなく、**「損失」を防ぐ**ことに目的を。
  - ✓4つの側面は単にUCA導出手段だけ、**システムレベルの損失シナリオ**に着目

■本講演のご質問は、

Katahira.Masafumi@jaxa.jp
Ishihama.Naoki@jaxa.jp