



STAMP支援シミュレータ開発

— The simulation tool for STAMP/STPA —
(Sim4stamp)

2017年11月15日

安全仕様化WG

積田 恵一

Agenda



1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
3. Sim4stampの目的
4. Sim4stampによるモデリング事例
5. まとめ
6. 今後の課題

Agenda



1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
3. Sim4stampの目的
4. Sim4stampによるモデリング事例
5. まとめ
6. 今後の課題

JASA 安全仕様化WGの紹介



レジリエンス エンジニアリング

意図記述と仕様記述

FRAM

EARS

意図表記法

SLP

STAMP / STPA

仕様表記法

Sim4stamp

安全誘導型設計

Agenda



1. 安全仕様化WGの紹介
- 2. STAMP/STPAの課題**
3. Sim4stampの目的
4. Sim4stampによるモデリング事例
5. まとめ
6. 今後の課題



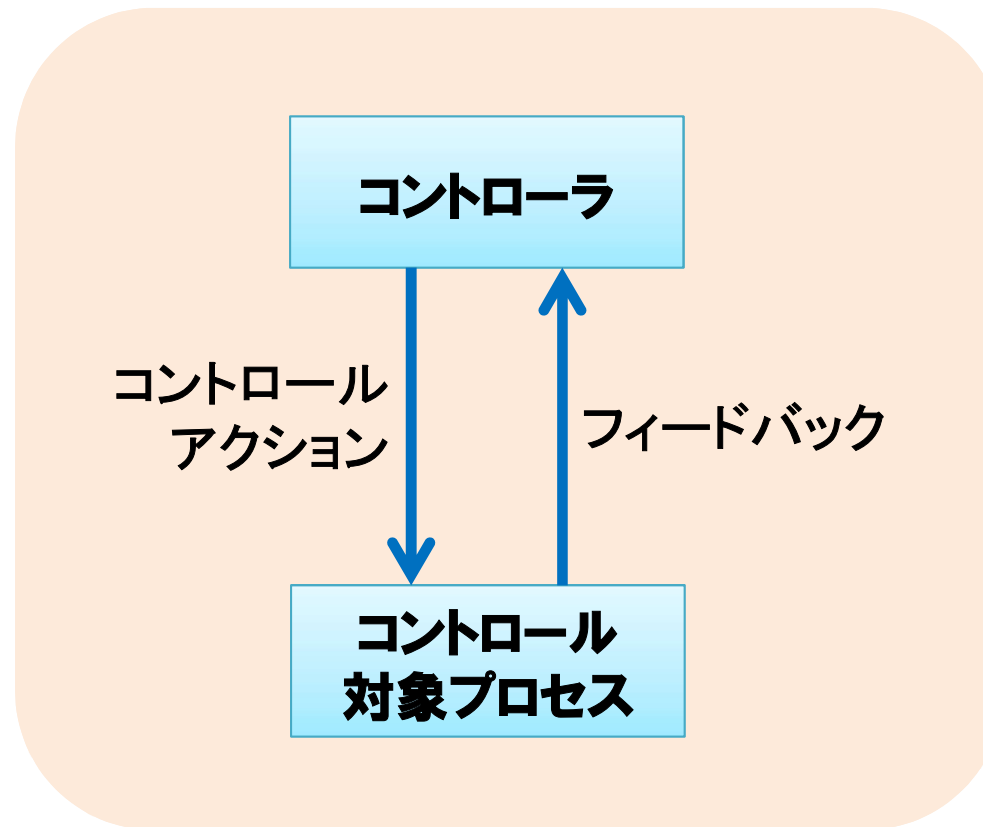
STAMPとは.....

- 事故はイベントの連鎖よりも複雑な動的プロセスを含む。
- 事故を(要素または部品)故障の問題ではなく、**コントロールの問題として捉える**。
- コンポーネントの振る舞いと相互作用の制約を課すことにより、事故を防ぐ。
- 事故のより深い原因を捉える。

STAMP : System-Theoretic Accident Model and Processes

STPA : System Theoretic Process Analysis

STAMPモデル



STAMPでは、「コントローラ」と「コントロール対象プロセス」に単純化し、それらの関係を考えるところがポイント！

STAMP/STPAの手順

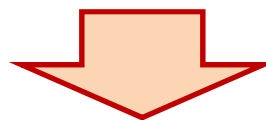


- Step-0:
制御構造図と**アクシデント・ハザード・安全制約**の定義。抽象的な機能に着目して**トップダウン**で階層的な**安全制御モデル**、プロセスモデルを作成
- Step-1:
四つのガイドワードを用いて非安全制御行為(**UCA: Unsafe Control Action**)を抽出
(N)Not providing、(P)Providing causes hazard、
(T) Inadequate timing、(D)Inadequate duration
- Step-2:
Control Loopのガイドワードを用いて、UCAごとに、**ハザード誘発要因**(**HCF: Hazard Causal Factor**)を分析し、**ハザードシナリオ**を導出

STAMP/STPAの課題



- トップダウン、システム指向、詳細未定のシステム開発早期でも使える。
- 抽象的な概念をガイドワードで具体化。



- ★ 想像力、想定力が必要。
- ★ 具体的なアーキテクチャなしでは想像し難い。

Agenda

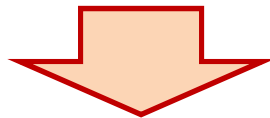


1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
- 3. Sim4stampの目的**
4. Sim4stampによるモデリング事例
5. まとめ
6. 今後の課題

STAMP/STPAシミュレーションの目的



- 抽象的な概念の具体化。
- 概念を「見える化」する。
- 事例で「深く理解」する。



有意義なハザードシナリオを確立する

Sim4stampとは



STAMP/STPAの思考過程をサポートし

抽象的なものを具体化、見える化し

試行錯誤を行ってアレコレ考えるための

**..... 誰でも自由に使える
STAMP/STPA専用シミュレータ**

Sim4stampとは？



**図で関係を
理解する**

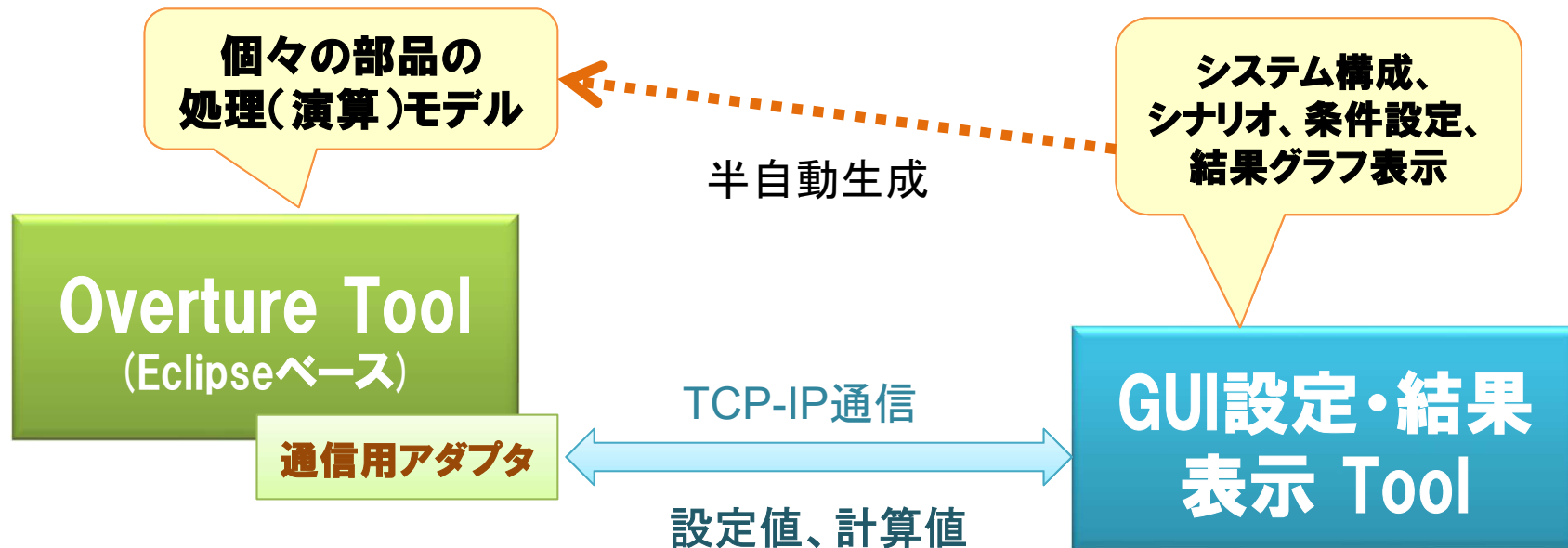
**VDM++で
仕様記述**

**オープンソース
ビジュアル シミュレーション ツール**

Sim4stampとは？



Overture IDEとの連携実行

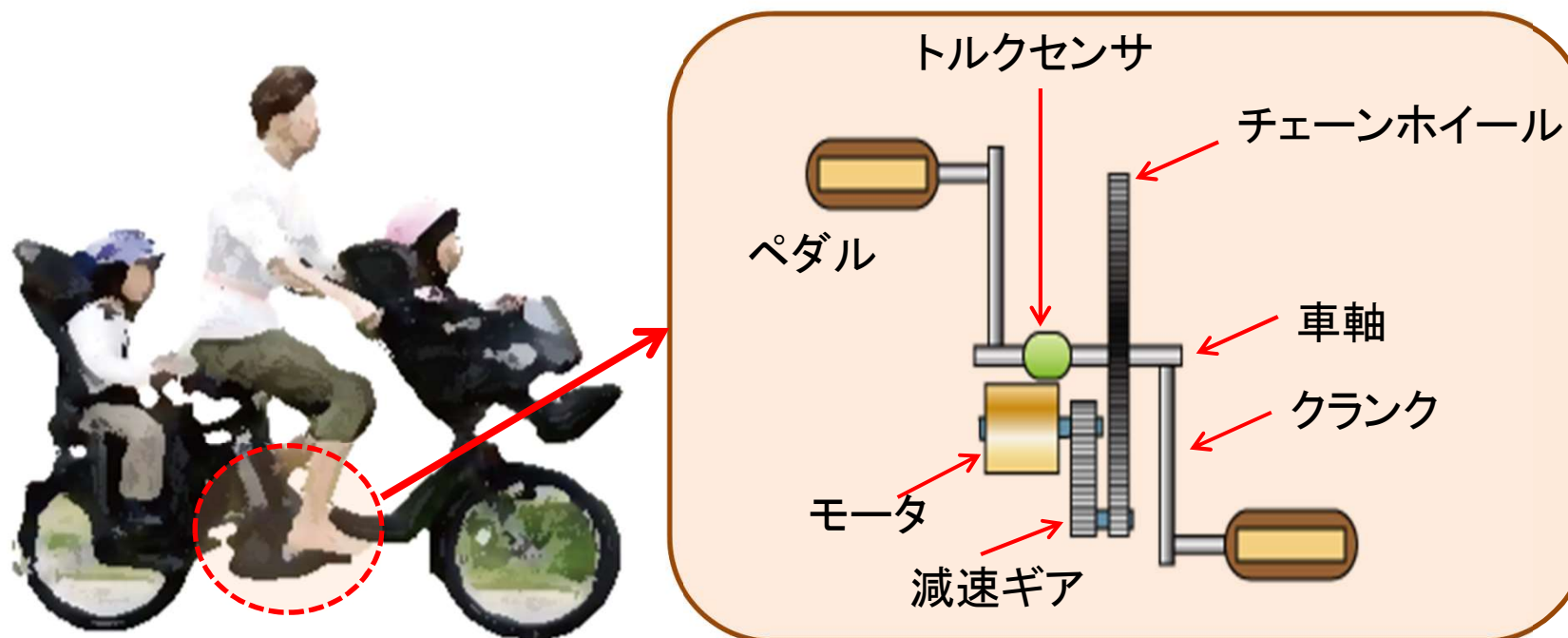


Agenda



1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
3. Sim4stampの目的
- 4. Sim4stampによるモデリング事例**
5. まとめ
6. 今後の課題

電動アシスト自転車で考えるSTAMP/STPA



- 電動アシスト自転車を題材にツールの適用方法を紹介
- 説明の都合上、ハンドル操作、ブレーキ操作は対象外

参考：電動アシスト自転車の事故



経済産業省によると、2014年の出荷台数は約47万台。11年にはオートバイを超え、10年で約2倍になった。鹿児島市など多くの自治体が、高齢者の車の事故防止や環境への配慮のため、電動自転車の購入に補助金を出しているほか、「レンタサイクル」での導入も進んでいる。

普及にともなって事故も増加した。警察庁によると、電動自転車による交通事故件数は15年に1394件。この10年で500件増えた。

公益財団法人「交通事故総合分析センター」(東京・千代田)の調査では、自転車事故の死者数は全体では11年に578人で01年から4割減ったが、電動自転車に限ってみると11年に50人となり、10年間で3倍と急増している。

特に、片足だけペダルに乗せ、もう片足で地面を蹴って勢いをつけて乗り出す「ケンケン乗り」での事故が目立つという。**体全体が自転車に乗りきっていない不安定な姿勢のままでモーターが作動し、急加速してしまうことがあるためだ。**

アシスト機能が優れていることで、**車体の重さに気付きにくくなる点も危険**という。車体は25～30キロの重量があり、自転車の専門家らでつくる「自転車の安全利用促進委員会」(東京・渋谷)の一員で自転車ジャーナリストの遠藤まさ子さんは「子供2人と親子で乗れば、全体で100キロを超える。**アシスト機能によりスピードも出るので、ぶつかった時の衝撃が大きい**」と指摘する。

出典:5/10/2016付けの日経電子版

AccidentとHazards



Accident

- 電動アシストが原因の転倒事故発生
- 電動アシストが原因の衝突事故発生

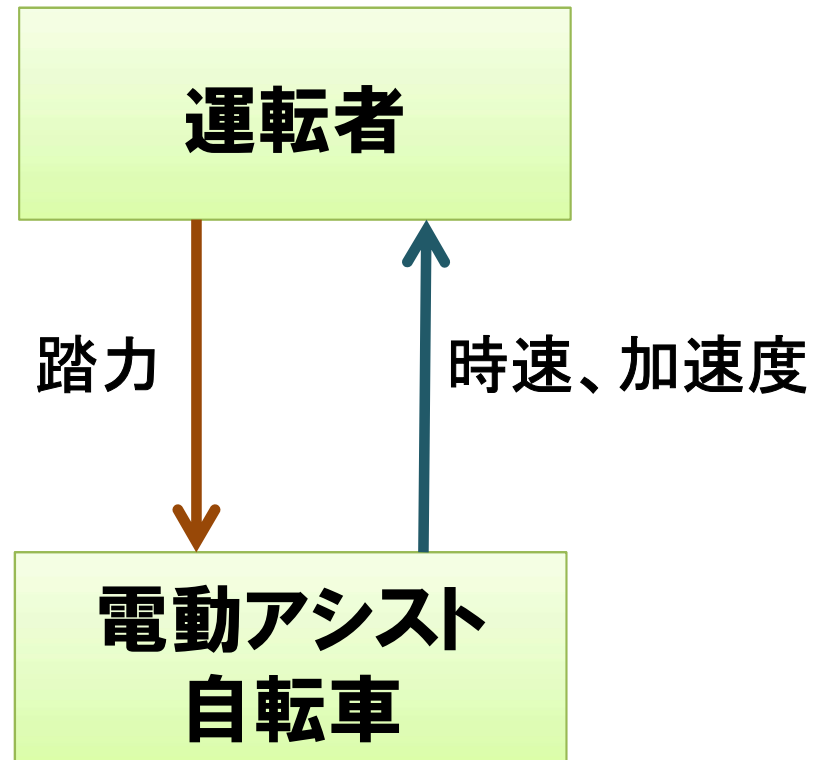
Hazards

- ・ 急加速の発生
- ・ 急減速の発生
- ・ 速度超過の発生

安全制約

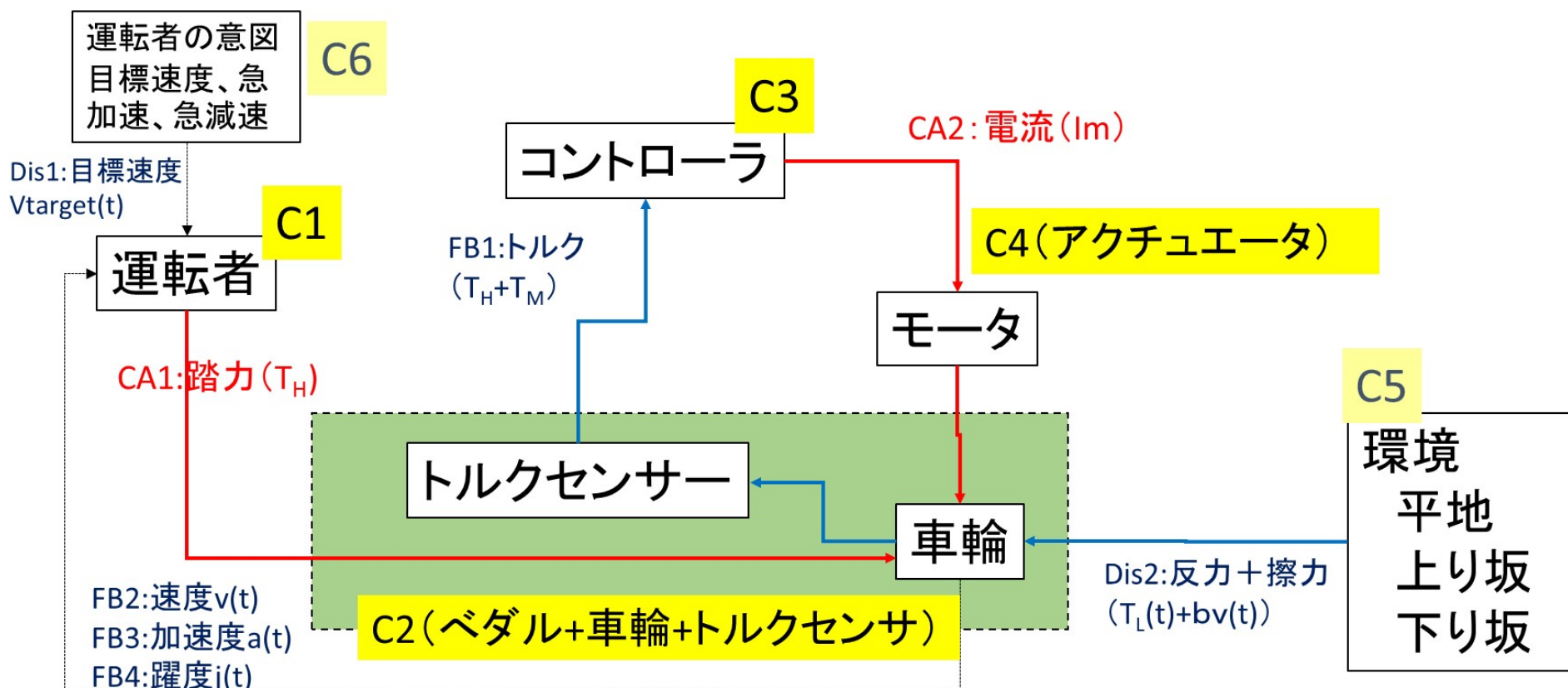
- ・ 急加速しないこと (SC1)
- ・ 急減速しないこと (SC2)
- ・ 速度超過しないこと (SC3)

STAMPモデル



注) この事例では ハンドル操作、ブレーキ操作は対象外とする

制御構造モデル



UCA(Unsafe Control Action)抽出例(部分)

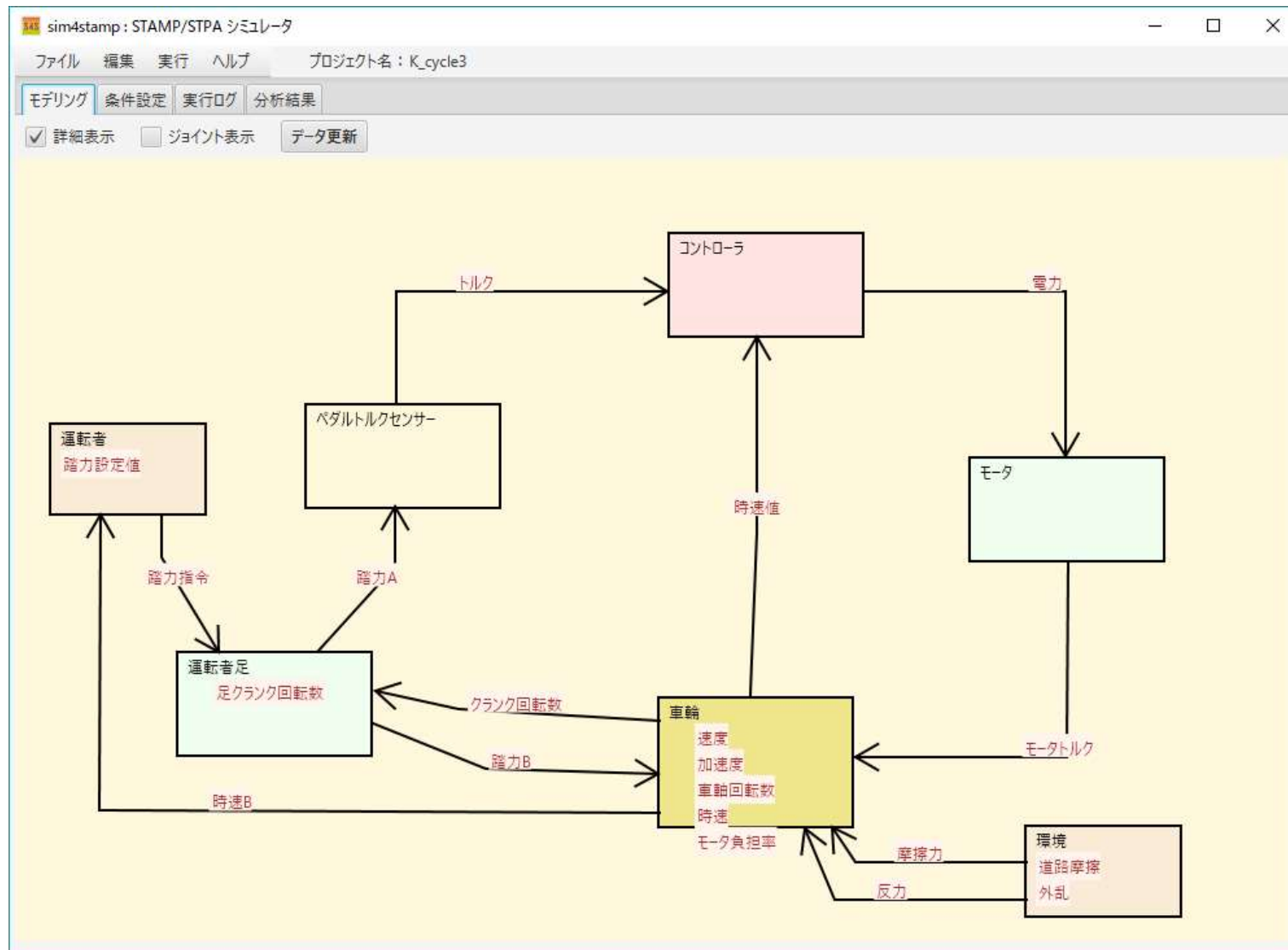


Sim4stamp未使用

#	Control Action	ガイドワードの種類	ハザードシナリオ	安全制約違反
1	ペダルを踏む	Not Providing	走らないので安全。	なし
		Providing causes hazard	ペダルを踏むと急加速が発生する。	SC1
		Too early / Too late	該当なし	なし
		Stop too soon / Applying too long	ペダルを踏み続けると速度超過する。	SC3

言葉だけで、定性的に考えることだけは、限界を感じる！

Sim4stampによるモデルの作成



シミュレーション パラメータの考察



- 質量: $80\text{kg} \leq \text{人間}(60\text{kg}) + \text{自転車}(20\text{kg})$
- 自転車のタイヤ 24インチ ($24 \times 25.4\text{mm} = 609.6\text{mm}$)
円周 $= 2 * \pi * 609.6\text{mm} / 2 = 1.9\text{m}$
- 自転車のギア比は 2.4~3.2のこと、ここでは 2.5 とする
- 定常スピードを10km/hとする $\rightarrow 10 * 1000\text{m} / (60 * 60) = 2.77\text{ m/s}$
- 通常加速度 : 停止から20秒で定常スピードとなると仮定
 $2.77\text{m/s} = \alpha * 20\text{s}$
 $\alpha = 2.77 / 20 = 0.1385\text{ m/s}^2$
- その時の力
 $80\text{kg} * 0.1385\text{ m/s}^2 = 11.08(\text{kgm/s}^2) \rightarrow 11.08\text{ N}$
- ペダル半径 : 160mm, ギア比2.5とすると
ペダル倍率 $= 1.9\text{ m} * 2.5 / (2 * \pi * 0.16\text{m}) = 4.75$
- 体重60kgの人間の体重による力は $60 * 9.8\text{m/s}^2 = 588\text{ N}$
- ペダル回転数 : 10km/h $\rightarrow 2.77(\text{m/s}) / 1.9\text{ m} = 1.457\text{ 回/s}$
ギア比2.5では、 $1.457(\text{回/s}) / 2.5 = 0.5828\text{ 回/s}$

シミュレーション パラメータの考察



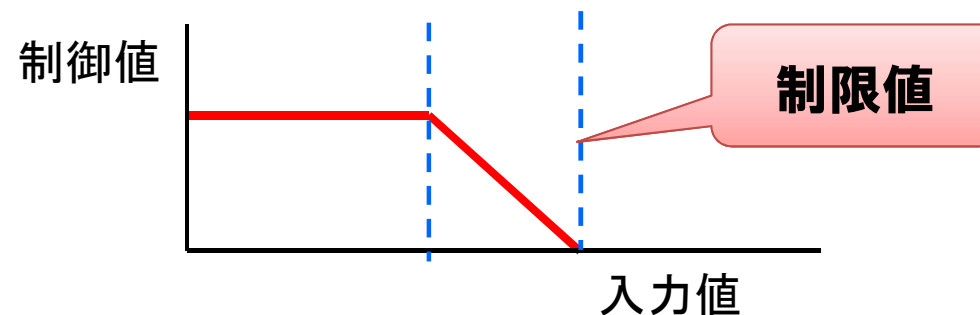
制約条件

- 人間が出せる力の最大値は体重程度：11 N 以内
ペダル倍率を考えると $1 / 4.75$ 倍となる
- ペダルに力を入れることが可能なペダルの回転数：最大で2回/s
- ユーザはアスリートではない
長い時間力強く漕ぐのはできない
せいぜいスピードは 25 km/h
アシストは24km/hでゼロにする。
- モータと踏力は単純合計とする
(エンジンブレーキの類のような現象は考えない)

VDM++側の記述ポイント

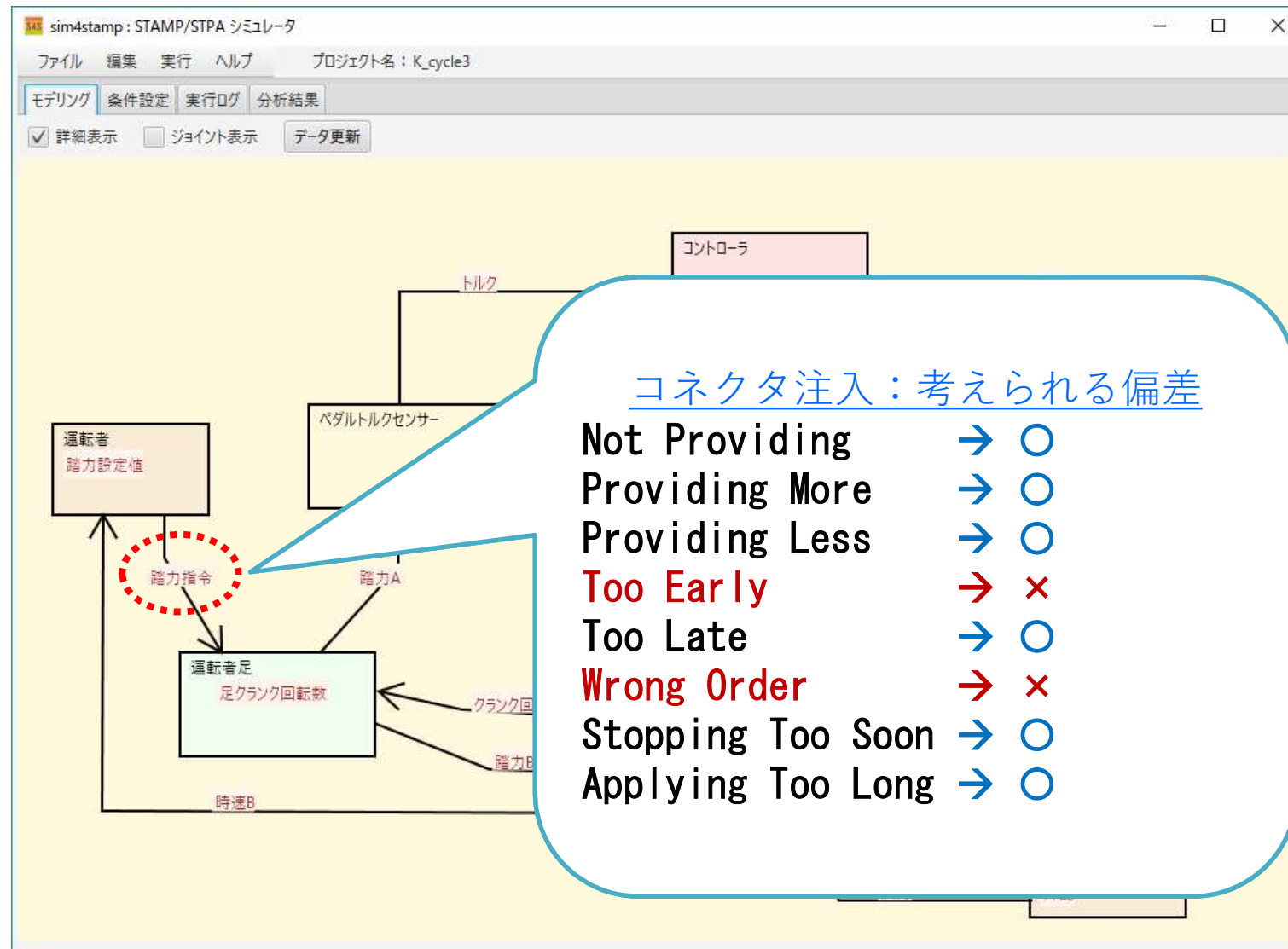


- コントローラに速度を入力し、最大アシスト速度を24km/hに抑える。
- 運転者に速度のフィードバックを掛けて、運転者の想定速度を超えた場合、踏力を抑える(最大15km/h)。
- 運転者の踏力はクランクの回転速度が一定値以上になると踏力が掛けられなくようにする(最大2回/s)。
- 制限値は、一定の猶予特性を持たせる。

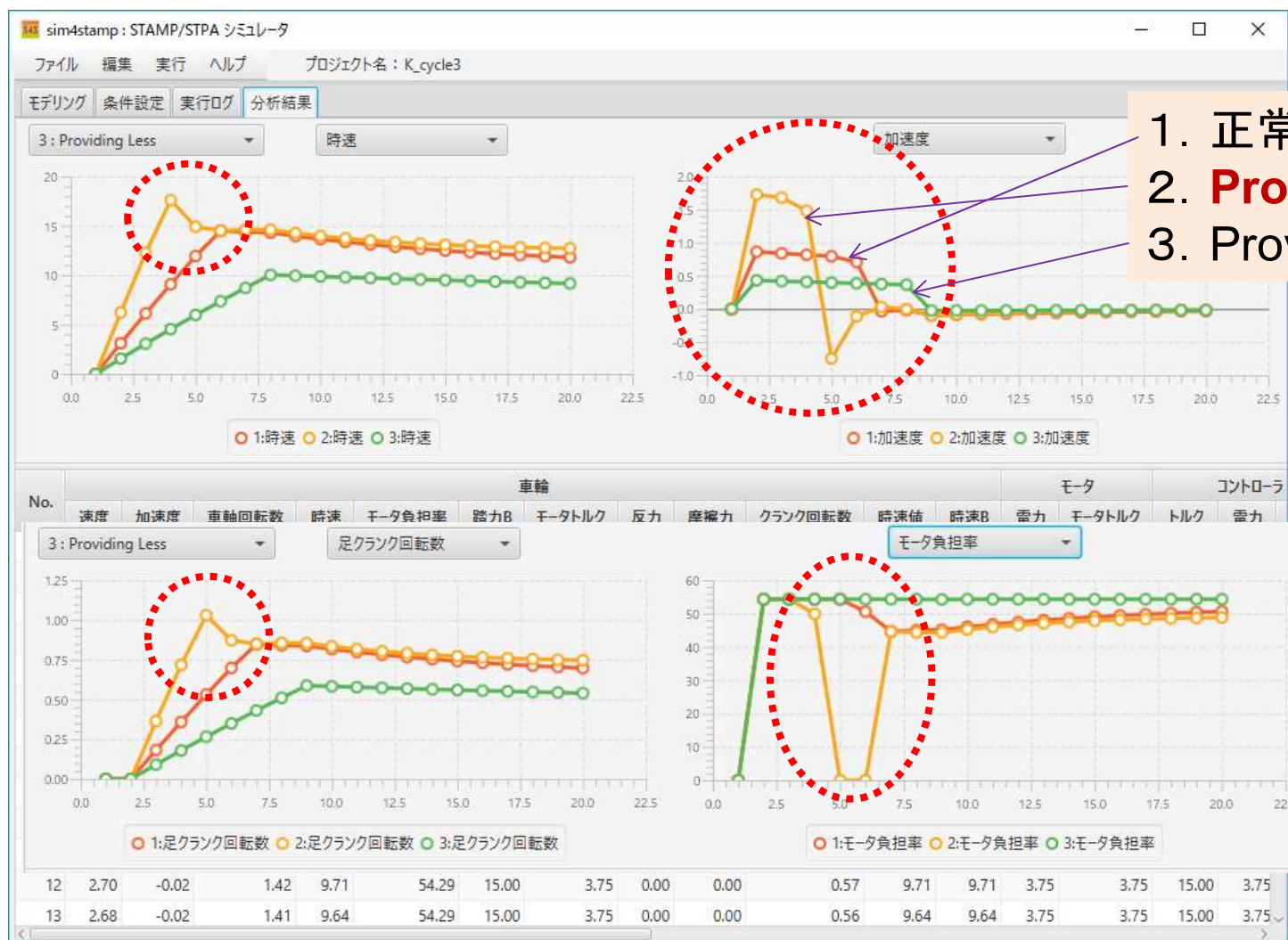


- 運転者への速度は 1秒遅れて入力される(1回遅れ)。

偏差注入一踏力

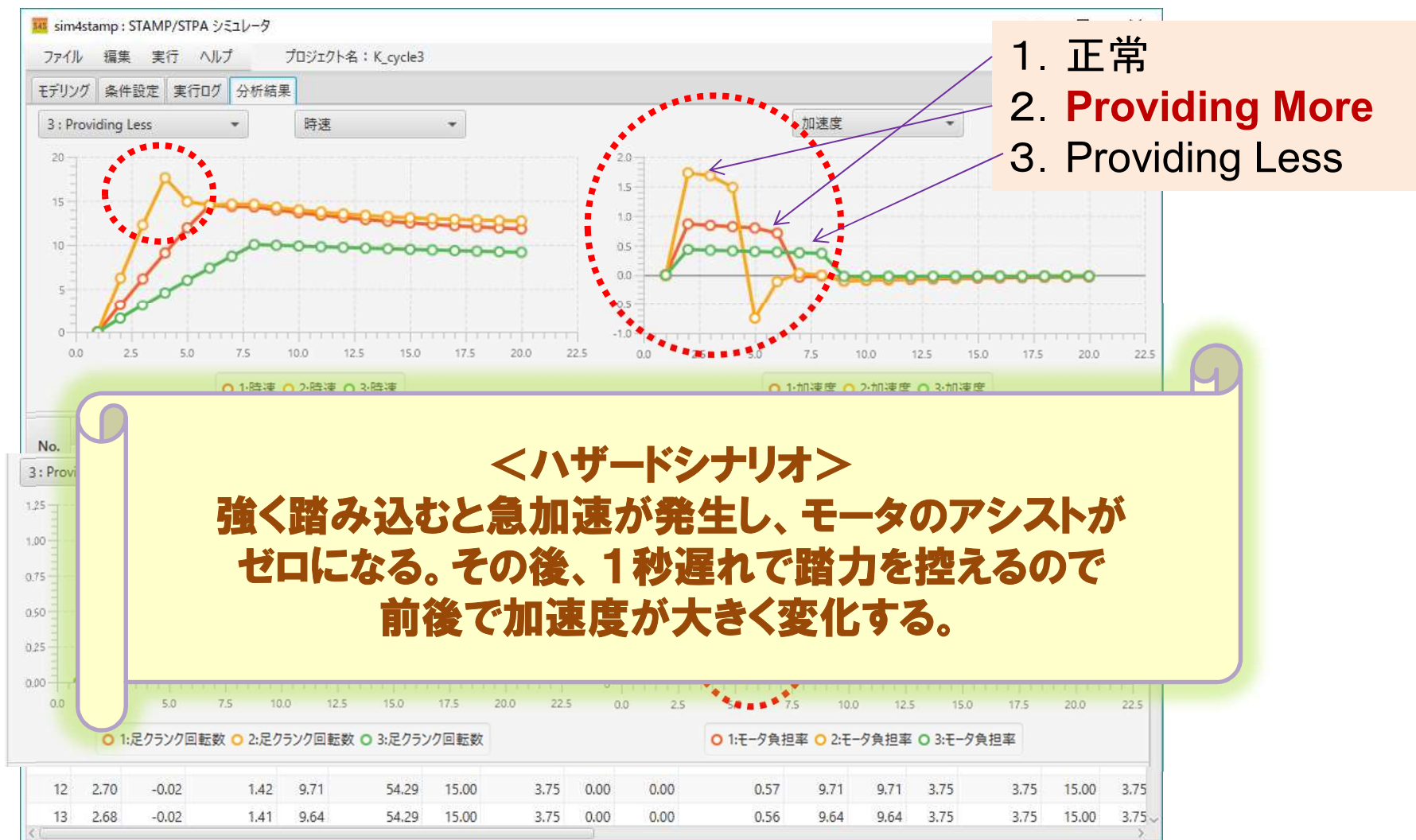


結果表示(偏差注入 - 踏力)



1. 正常
2. Providing More
3. Providing Less

結果表示(偏差注入 - 踏力)



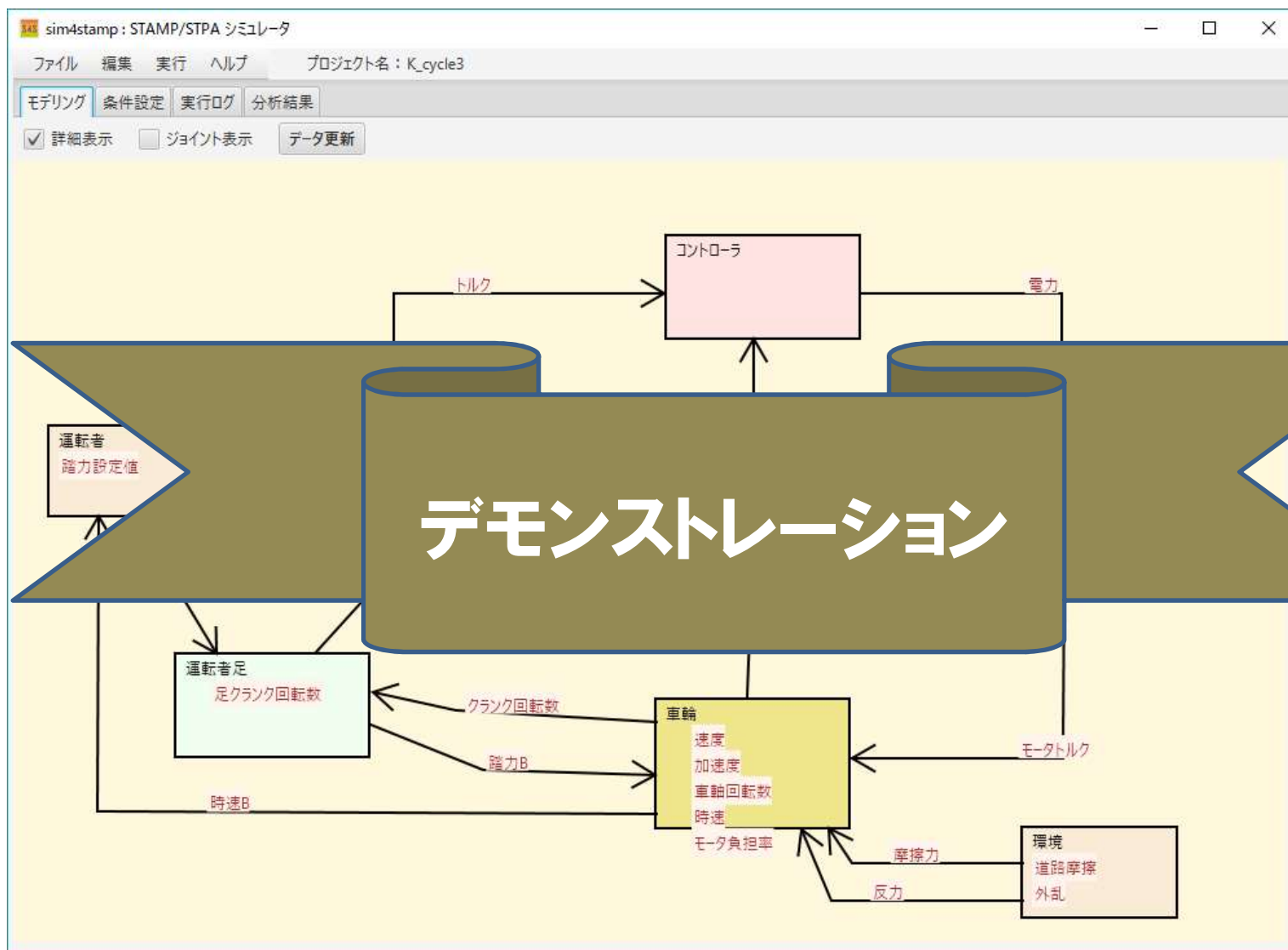
UCA(Unsafe Control Action)抽出例(部分)



Sim4stamp利用

#	Control Action	ガイドワードの種類	ハザードシナリオ	安全制約違反
1	ペダルを踏む	Not Providing	走らないので安全。	なし
		Providing causes hazard	ペダルを強く踏むと急加速し、慌てて踏むのを止めて急減速が発生する。	SC1とSC2
		Too early / Too late	該当なし	なし
		Stop too soon / Applying too long	<u>ペダルを踏むのをすぐに止めるとその時点で急減速する。</u> ペダルを踏み続けると速度超過する。	SC2 SC3

注) XXXXの部分がSim4stampによる分析で分かったこと。



Agenda



1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
3. Sim4stampの目的
4. Sim4stampによるモデリング事例
- 5. まとめ**
6. 今後の課題

モデリングから言えること



- シミュレーションしてみて、初めて気が付くハザードシナリオがある。
- 頭の中だけで定性的に予測したことが正しいこともありし、間違っていることもある。
- 試行錯誤が無ければ、ハザードシナリオは精密化できない。
(机上の空論となるかも！)
- モデル図と計算部を統合して作成できることでアイデアをすぐに試せる。



STAMPモデルのシミュレーションを行うことで...

- 対象の明確化、定義の厳密化。
 - 論理的に明確化、厳密化しないとそもそもシミュレーション出来ない!
- 具体的なシナリオの提示、気付き。
 - 具体的なシナリオでいろいろ気づきが発生する!
- 想定外を減らす。
 - シミュレーションは論理に正直!
- 対象の安全制約に対する根拠を与える。



- STAMP/STPAに特化した専用シミュレータ。
- 条件を変更し、シミュレーションを繰り返すことにより、いろいろなハザードシナリオを確認できる。
- モデル図を実行する。
絵を書けば、接続関係が自動的に生成される。モデル図を修正すると評価順やパラメータが自動的に変更される。
- 個々の要素部のロジックは、半自動生成したVDM++を、Overture IDEを使用して記述追加し、完成させる方式なので作成が容易。
(VDM++の文法に精通する必要なし)

Agenda



1. 安全仕様化WGの紹介
2. STAMP/STPAの課題
3. Sim4stampの目的
4. Sim4stampによるモデリング事例
5. まとめ
- 6. 今後の課題**

今後の課題



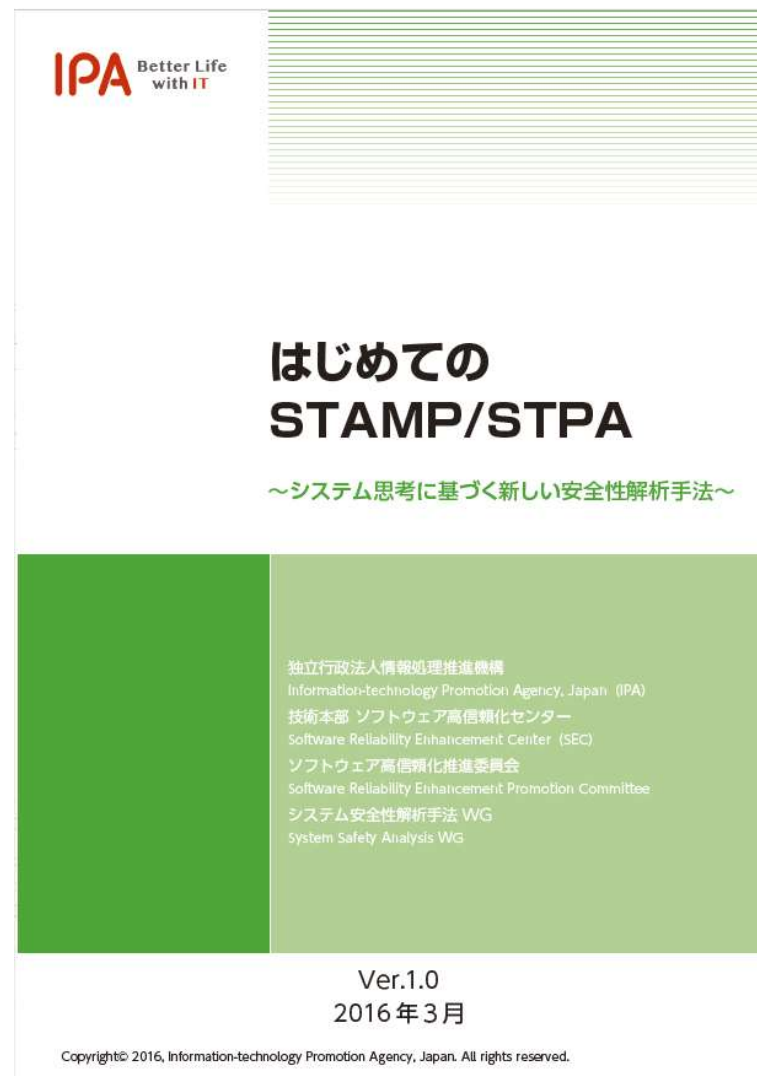
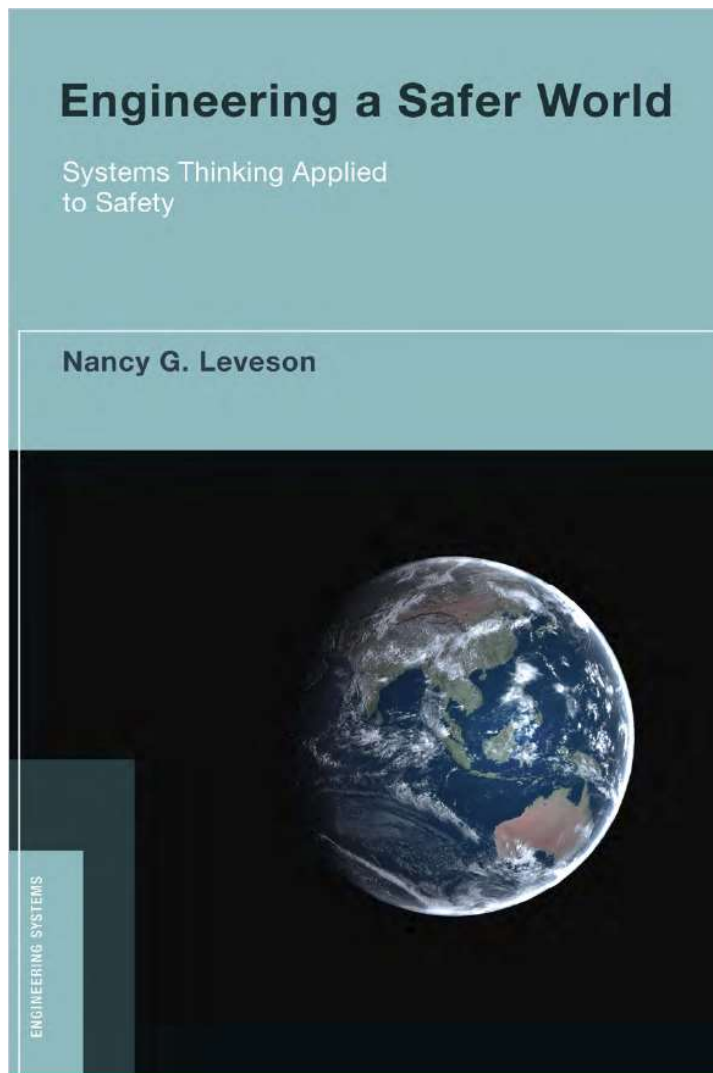
- モデリングでFRAMの手法を取り入れること。
 - ☆ 入力の分類ー 前提条件、時間、制御、リソースの扱い
 - ☆ シミュレーションパラメータの変動の扱い
- シミュレーション結果の保存方法
- IPA開発中ツールとの連携

Sim4stampの入手先、開発への参加

GitHubリポジトリ：

<https://github.com/KeiTsumuta/Sim4stamp>

付録 STAMP/STAPを学ぶには？





【STAMP支援シミュレータ開発】

2017/11/15 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋大伝馬町6-7
TEL: 03(5643)0211 FAX: 03(5643)0212
URL: <http://www.jasa.or.jp/>

本書の著作権は一般社団法人組込みシステム技術協会（以下、JASA）が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。