2017年度版

STAMP支援シミュレータ開発 ～開発経緯と結果～

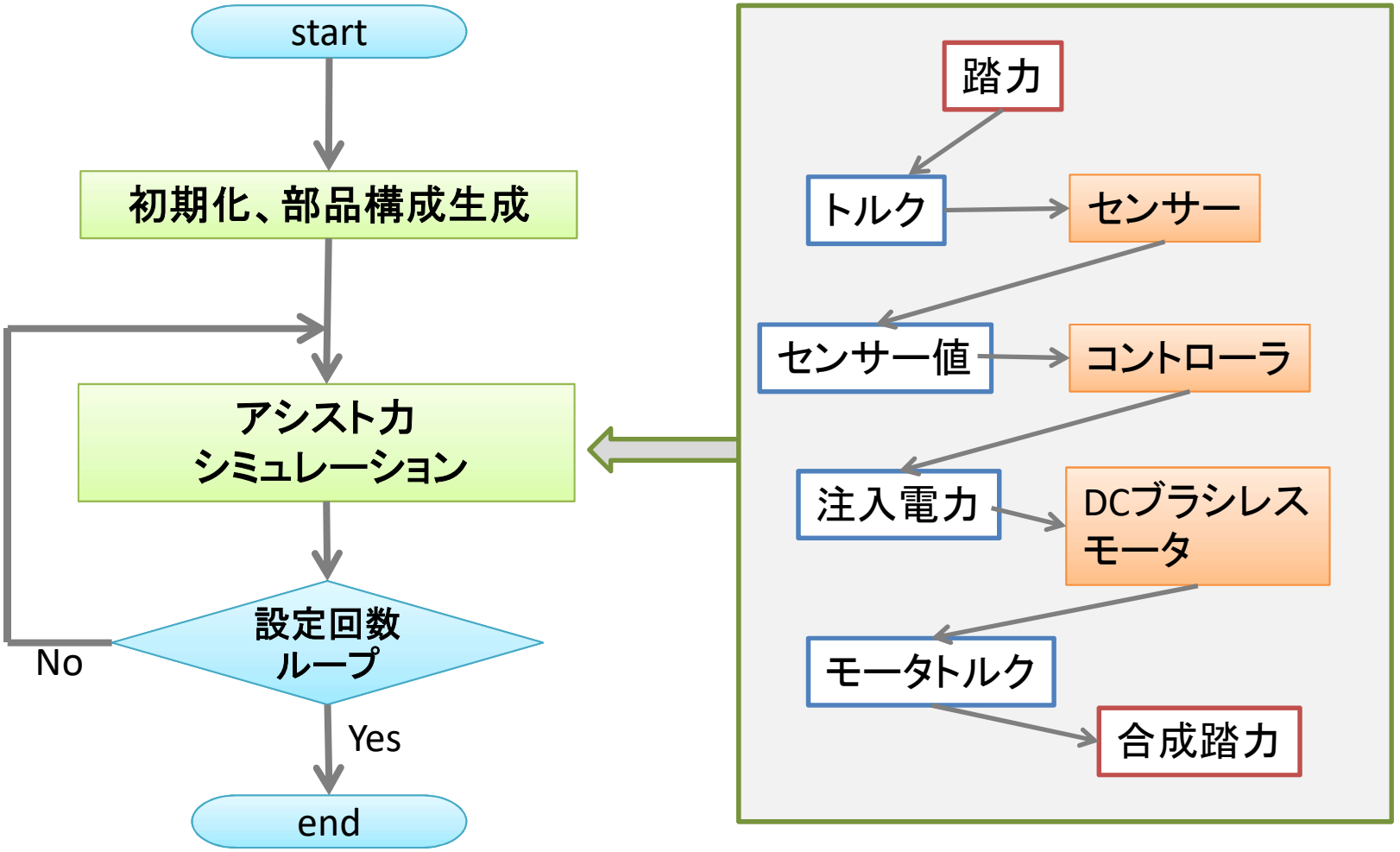
― The simulation tool for STAMP/STPA ― (Sim4stamp)

安全仕様化WG
積田 恵一

Sim4stamp開発経緯

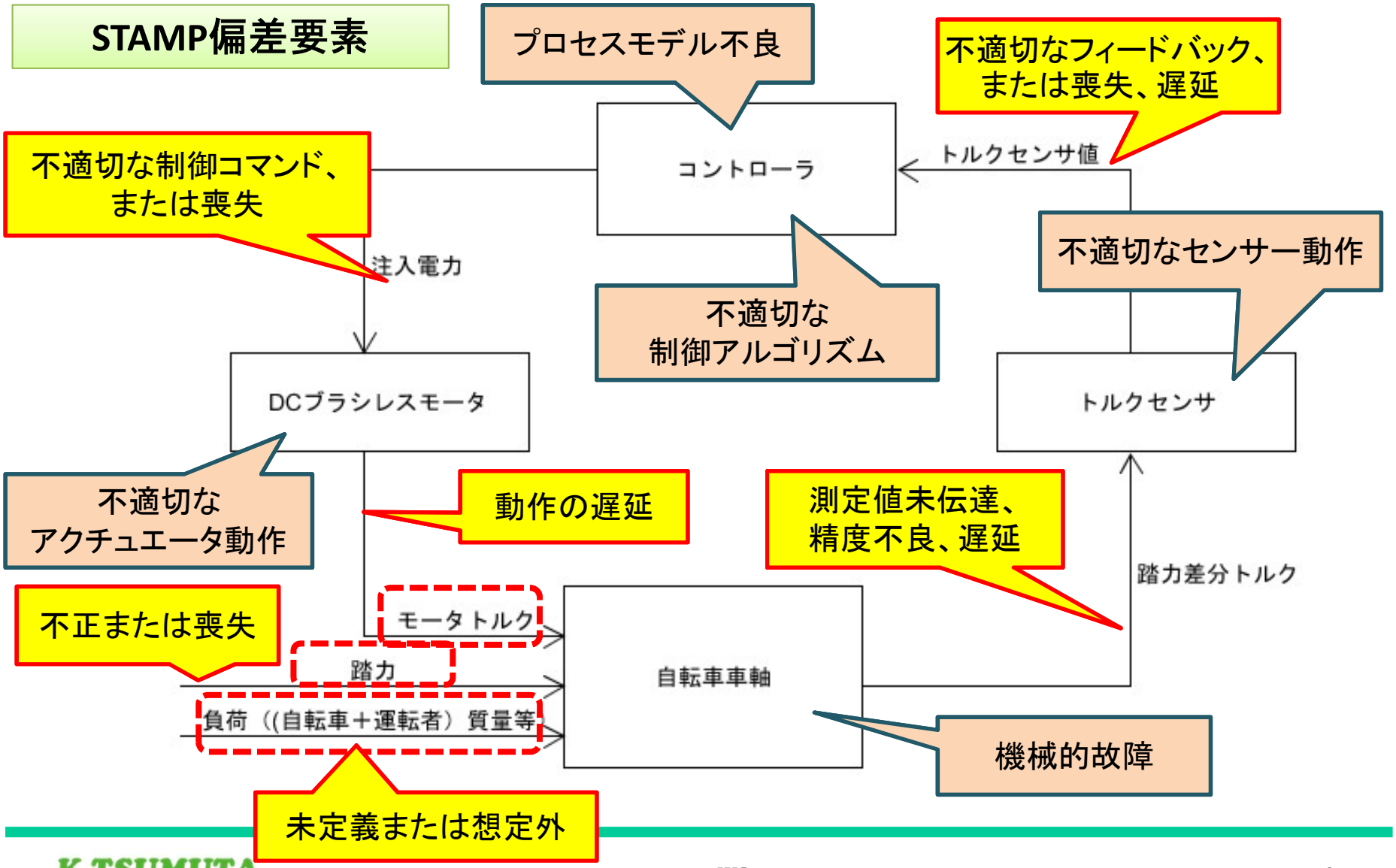
1. STAMP/STPAのVDM++シミュレーション検討（2016/6～8）
2. STAMP/STPAのVDM++シミュレーション実施（2016/9）
3. 専用シミュレータの構想（2016/12）
4. 専用シミュレータの開発開始（2017/1）
5. 偏差注入の検討（2017/3）
6. データフォーマット構成検討（2017/4）
7. モデル作成GUI検討/開発（2017/5）
8. 改善点検討/開発（2017/7）
9. 兼本さん作成モデル適用例による機能改善（2017/9～10）
10. Sim4stampソースコードGitHubで公開（2017/11）
11. ET2017発表（2017/11）
12. 偏差注入の課題検討（2017/12）

シミュレータ手順（電動アシスト自転車モデル） 2016/6～8



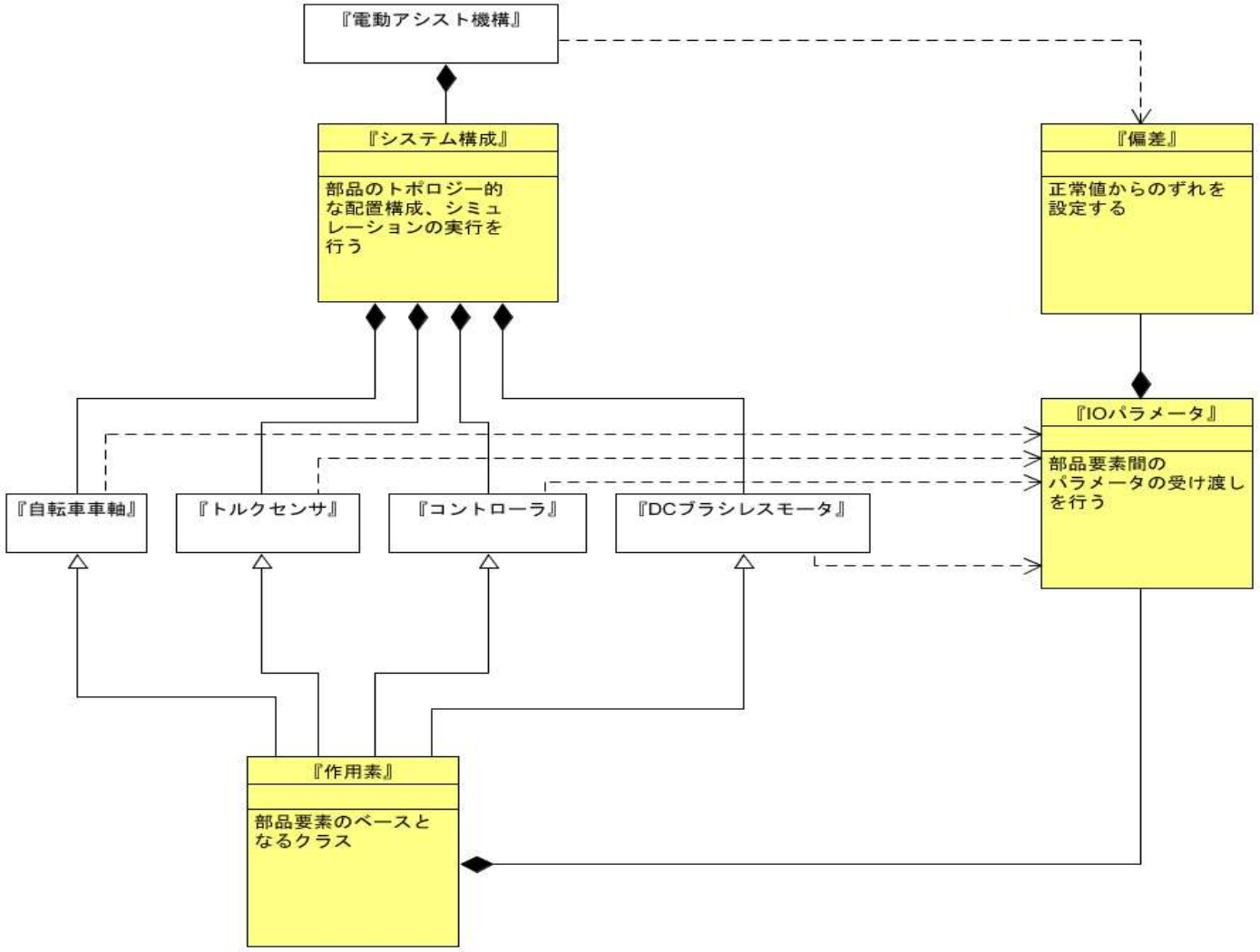
制御ループの変異要素

2016/6～8



クラス図

2016/9

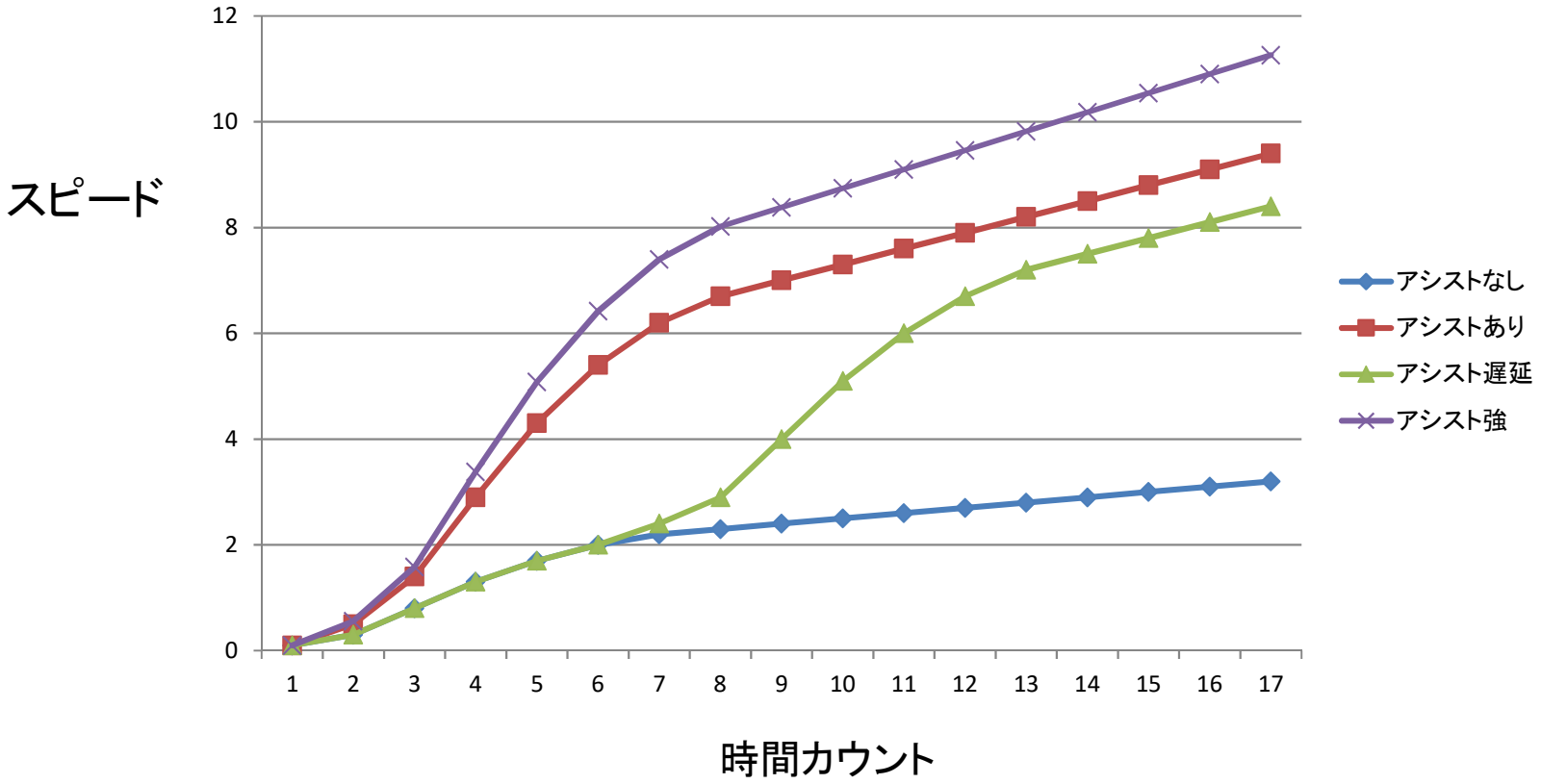

『電動アシスト機構』
『システム構成』
部品のトポロジー的な配置構成、シミュレーションの実行を行う
『偏差』
正常値からのずれを設定する
『IOパラメータ』
部品要素間のパラメータの受け渡しを行う
『自転車車軸』
『トルクセンサ』
『コントローラ』
『DCブラシレスモータ』
『作用素』
部品要素のベースとなるクラス

シミュレーションの実施

2016/9

VDM++シミュレーション例

踏力シーケンス : [1.0, 2.0, 5.0, 5.0, 4.0, 3.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]



VDM++シミュレーションの課題

2016/9

- モデリングが面倒
  やりたいシミュレーションの周辺も作成する必要がある。

- データ入力、結果が直観的でない
  VDM++にはビジュアル化ツールがほとんど無く、数値の列を解釈しなければならない。

- 入力、出力の設定が面倒
  原始的な方法しかない！

シミュレーション方式の提案

2016/12

思考の道具としてのツール

- **GUIベース**
  シミュレーション構成、実行、結果確認はGUIベース。
  シミュレーションシナリオはGUIで作成。
  変更を簡単に！ 目に見えるようにする！

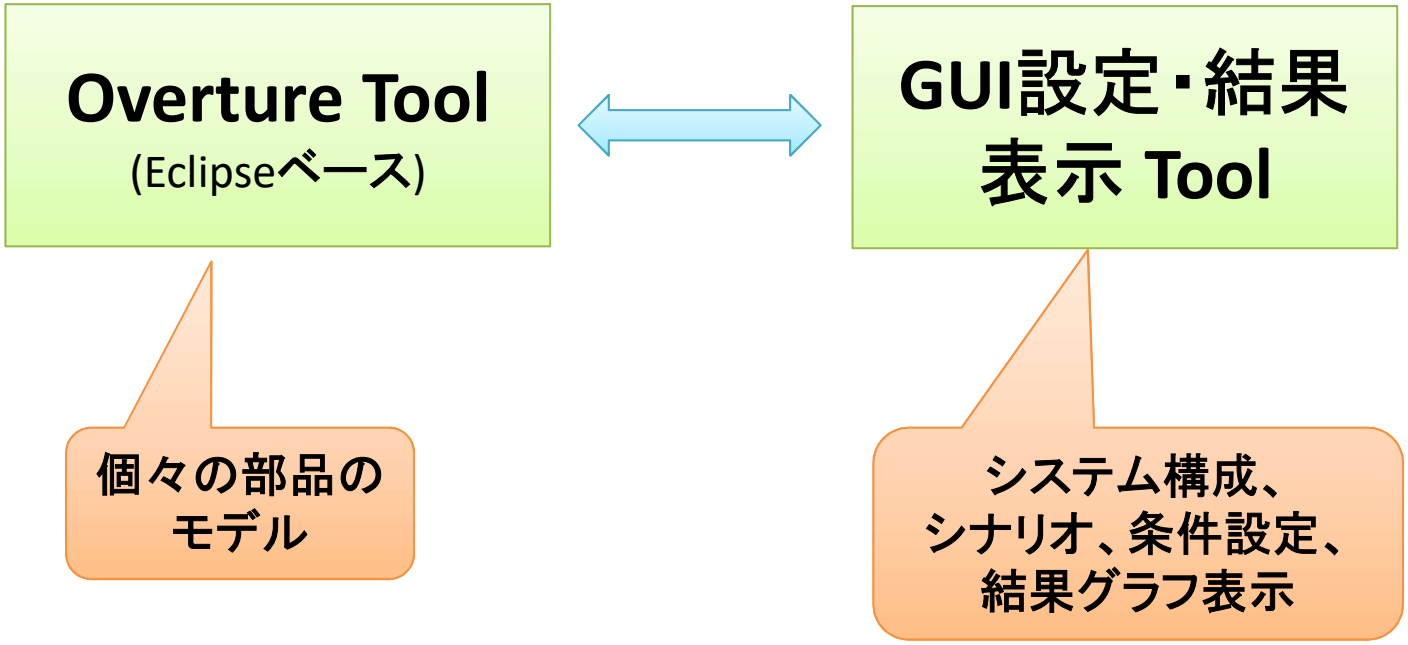
- **個々の装置モデルはVDM++で作成**
  強力なVDM++で、個々の装置の処理アルゴリズム作成。

- **ユーザが作成するVDM++は必要最小限に**
  GUI設定のシミュレーション構成からVDM++のテンプレートを自動生成。
  ユーザの思考を本質的な部分に集中させる。

SimSTAMPの構成

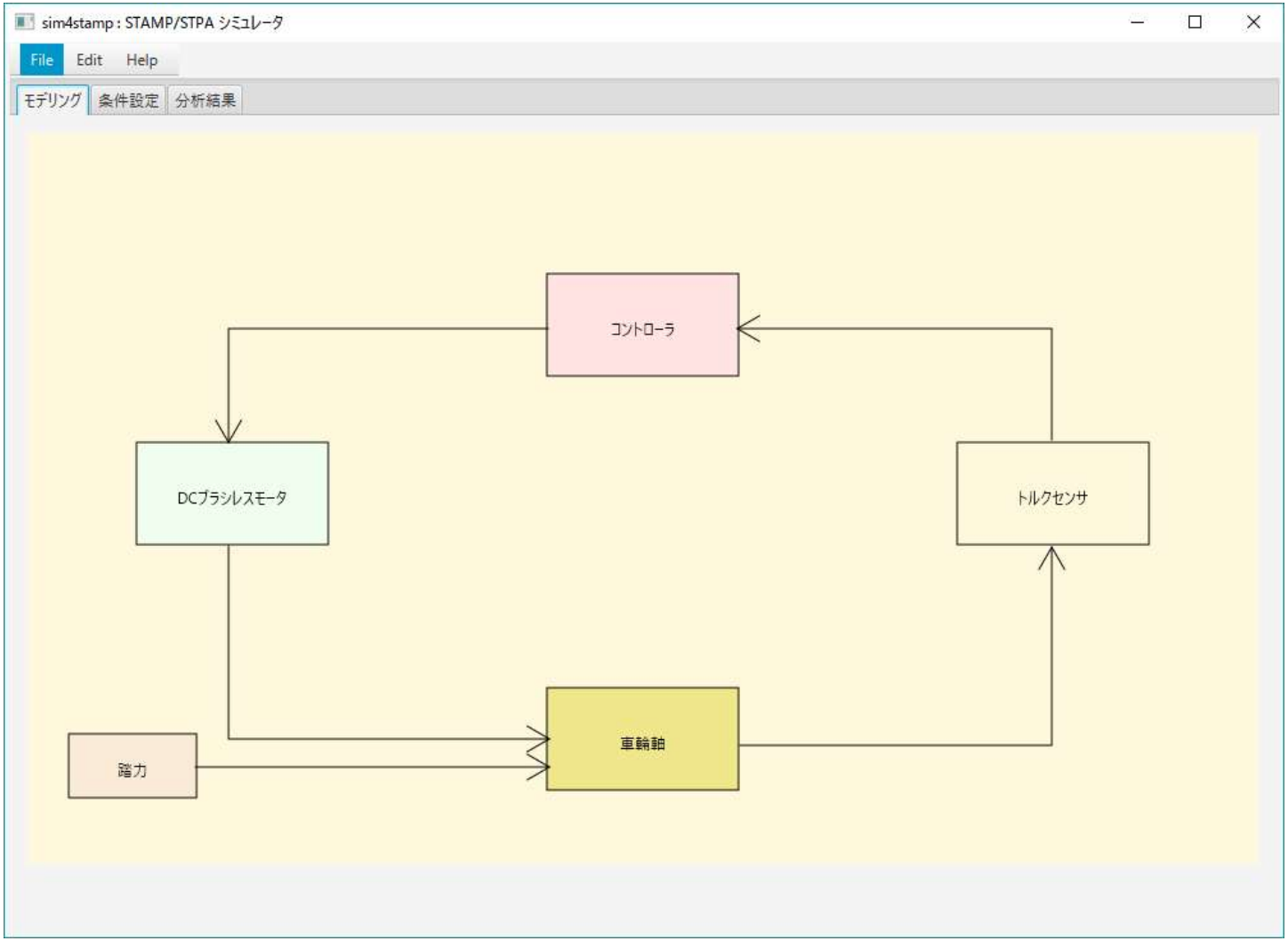
2016/12

VDMを利用したSTAMP/STPAシミュレーションツール



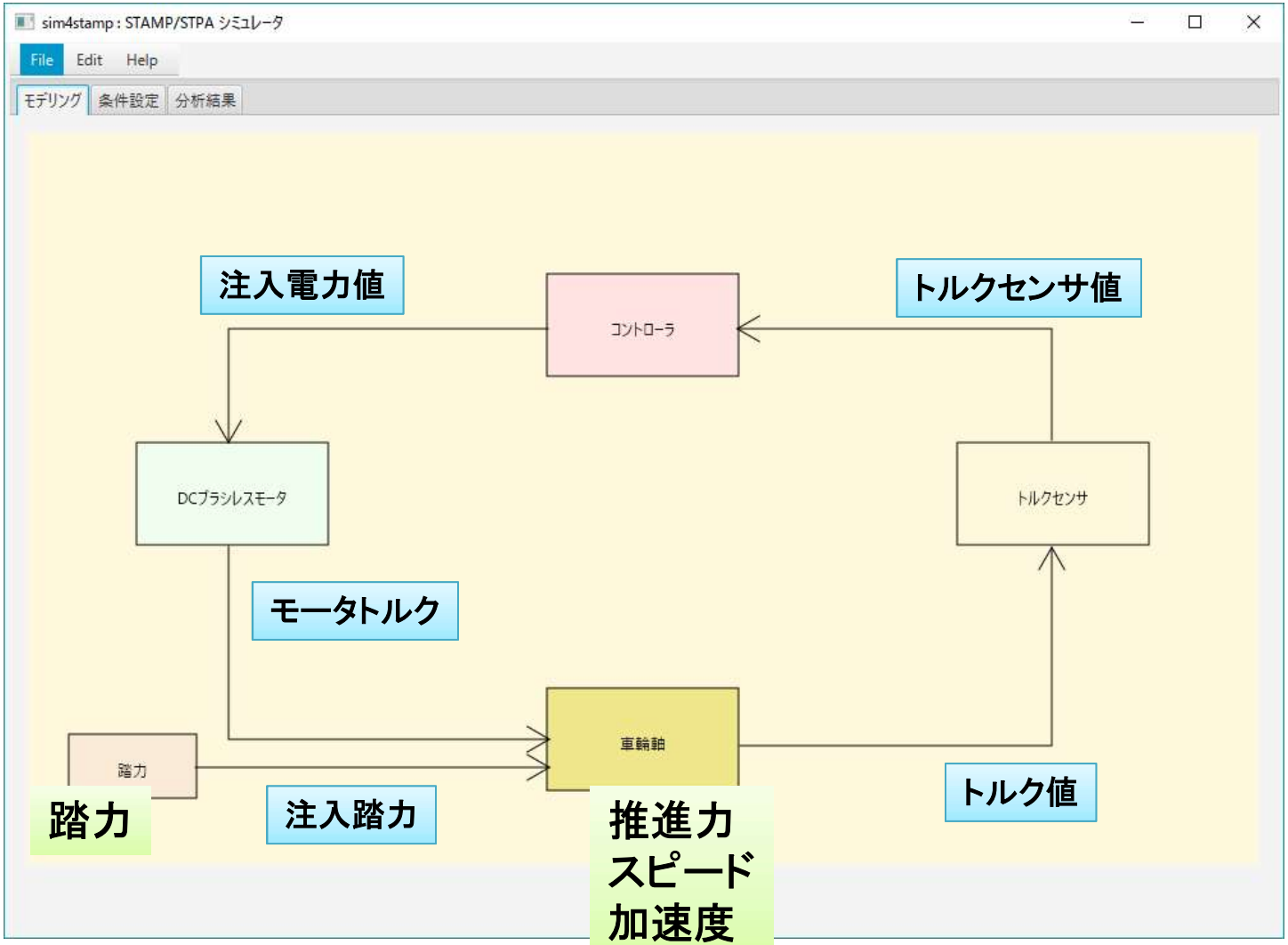
Java側ツールのGUI(モデル定義-1)

2017/1



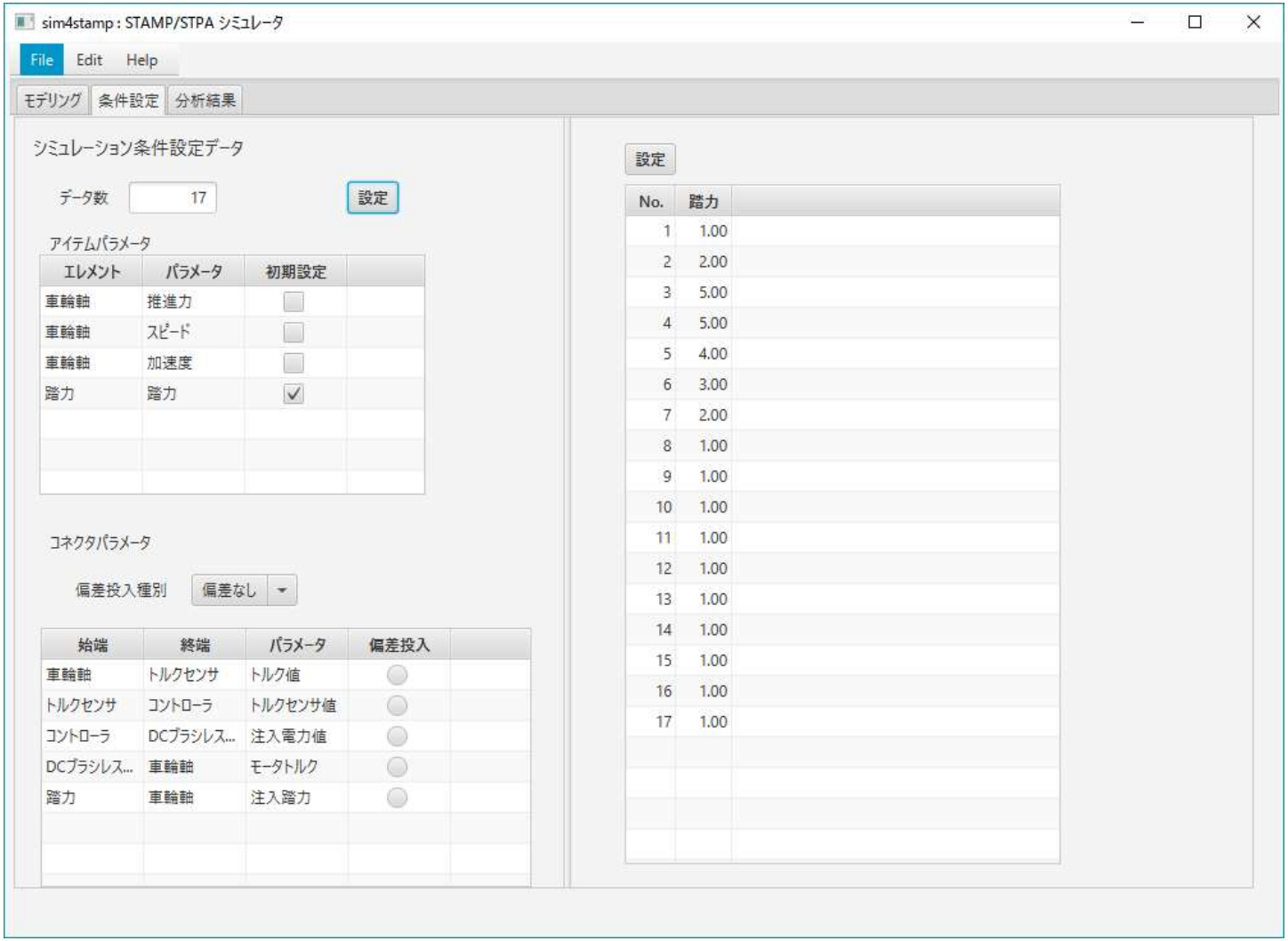
Java側ツールのGUI(モデル定義-2)

2017/1



Java側ツールのGUI（パラメータ設定）

2017/1



Java側ツールのGUI（偏差注入）

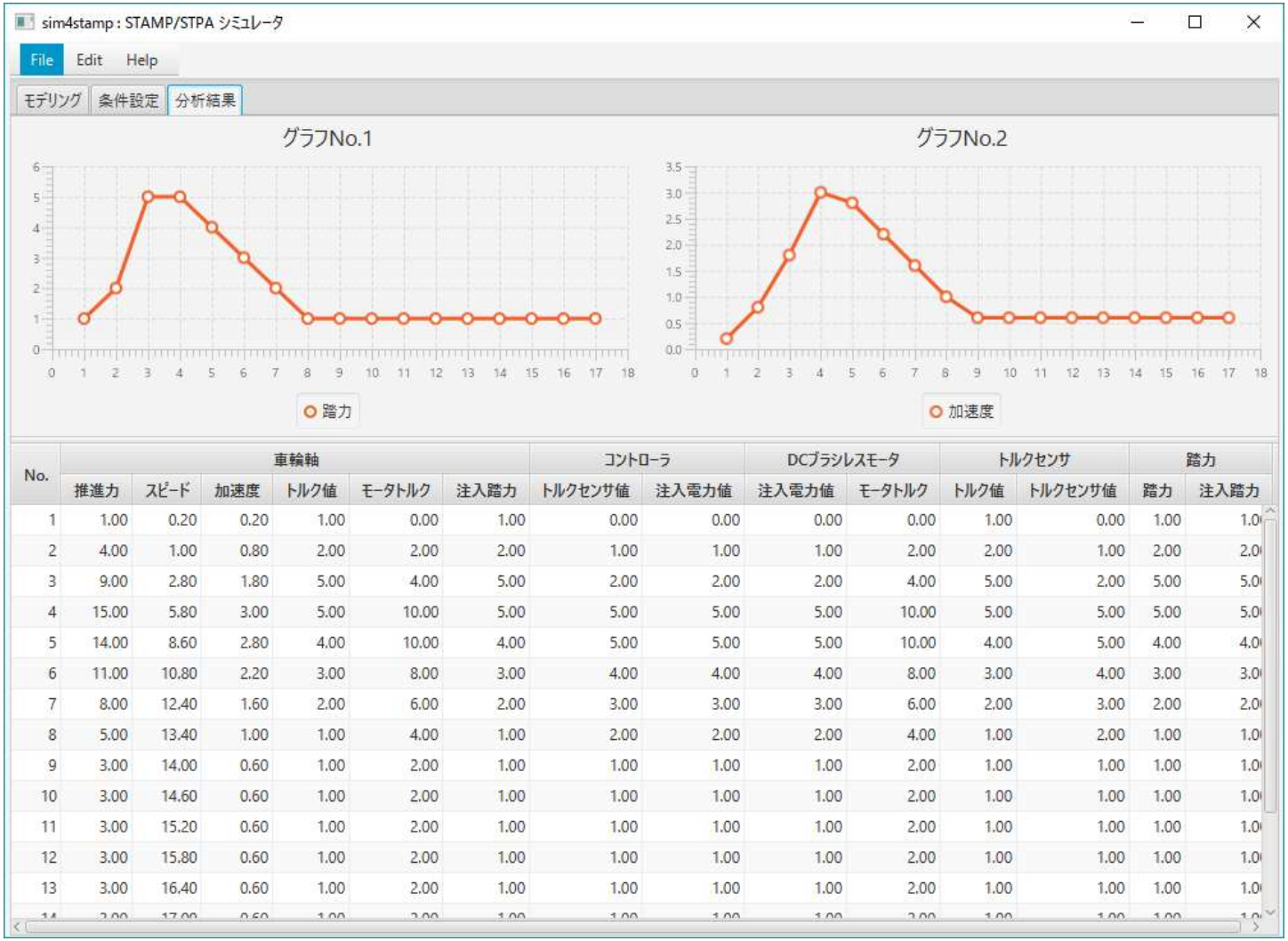
2017/1

コネクタパラメータに偏差注入を行う。

偏差注入されたものとリファレンス（偏差なし）を比較する

Java側ツールのGUI(結果表示)

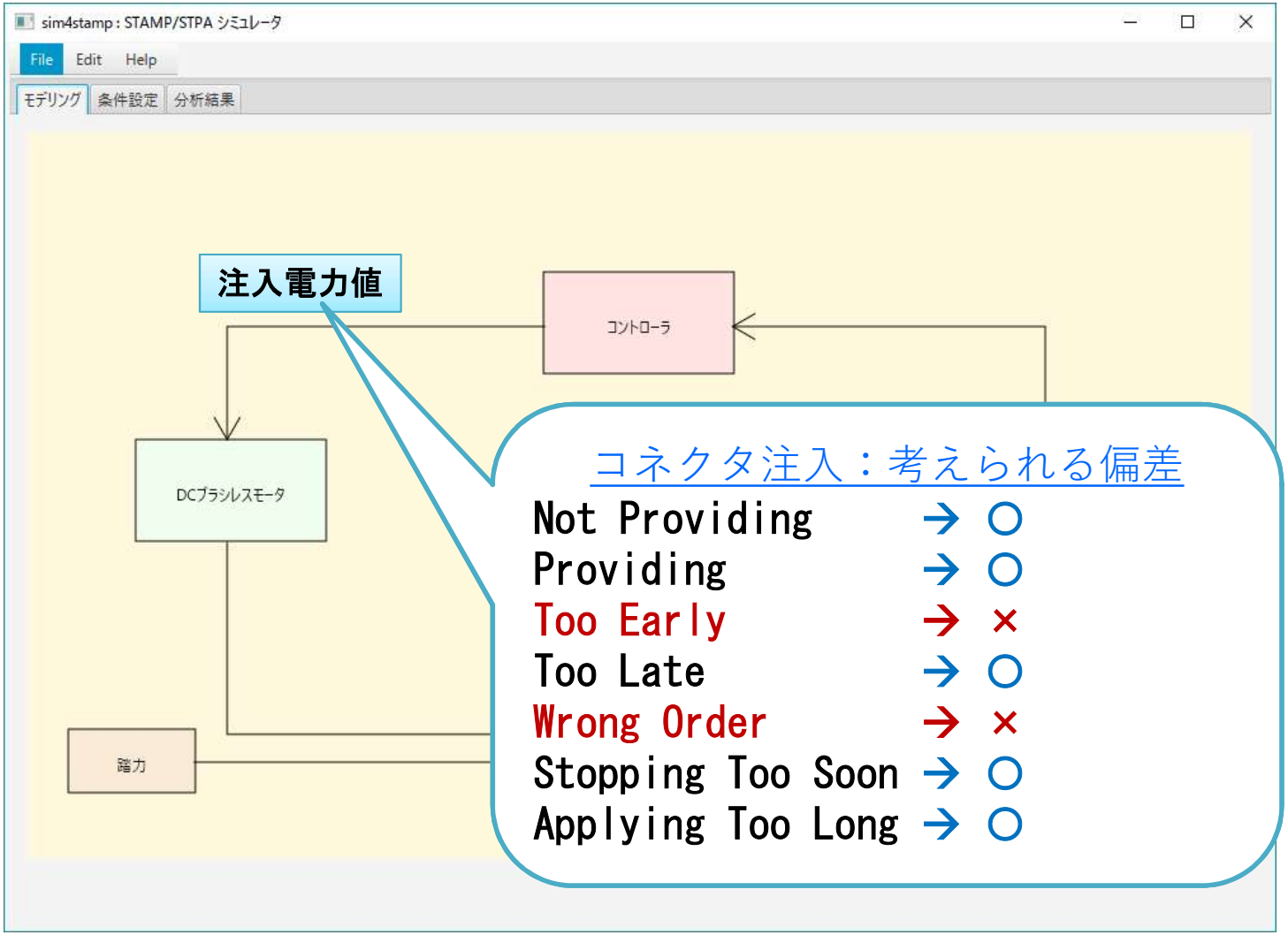
2017/1



No.	車輪軸						コントローラ		DCブラシレスモータ		トルクセンサ		踏力	
	推進力	スピード	加速度	トルク値	モータトルク	注入踏力	トルクセンサ値	注入電力値	注入電力値	モータトルク	トルク値	トルクセンサ値	踏力	注入踏力
1	1.00	0.20	0.20	1.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	1.0
2	4.00	1.00	0.80	2.00	2.00	2.00	1.00	1.00	1.00	2.00	2.00	1.00	2.00	2.0
3	9.00	2.80	1.80	5.00	4.00	5.00	2.00	2.00	2.00	4.00	5.00	2.00	5.00	5.0
4	15.00	5.80	3.00	5.00	10.00	5.00	5.00	5.00	5.00	10.00	5.00	5.00	5.00	5.0
5	14.00	8.60	2.80	4.00	10.00	4.00	5.00	5.00	5.00	10.00	4.00	5.00	4.00	4.0
6	11.00	10.80	2.20	3.00	8.00	3.00	4.00	4.00	4.00	8.00	3.00	4.00	3.00	3.0
7	8.00	12.40	1.60	2.00	6.00	2.00	3.00	3.00	3.00	6.00	2.00	3.00	2.00	2.0
8	5.00	13.40	1.00	1.00	4.00	1.00	2.00	2.00	2.00	4.00	1.00	2.00	1.00	1.0
9	3.00	14.00	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
10	3.00	14.60	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
11	3.00	15.20	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
12	3.00	15.80	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
13	3.00	16.40	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0

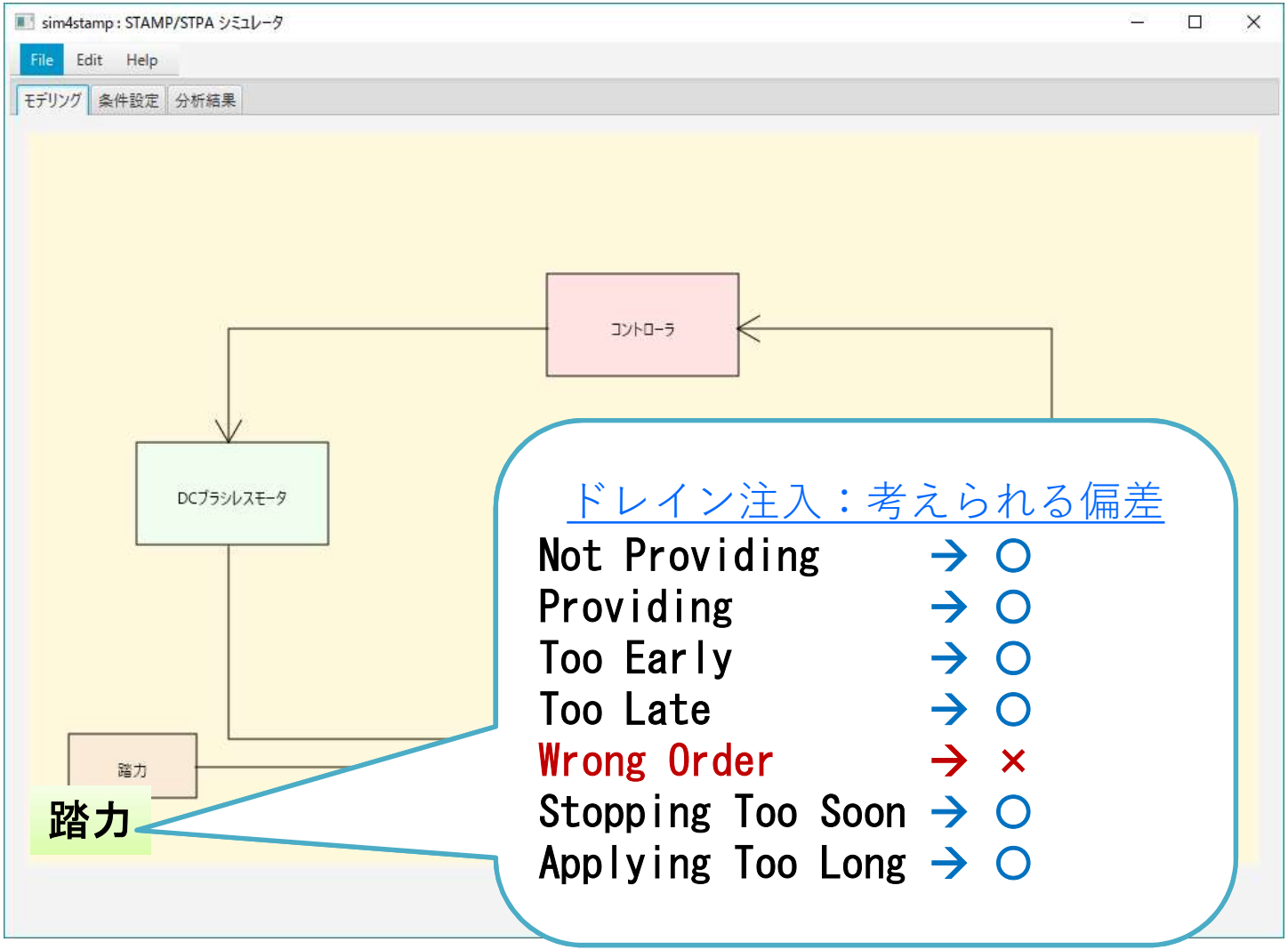
偏差注入の種類

2017/3



偏差注入の種類

2017/3



偏差注入方法(Stopping Too Soon)

2017/3

No.	車輪軸						コントローラ		DCブラシレスモータ		トルクセンサ		踏力	
	推進力	スピード	加速度	トルク値	モータトルク	注入踏力	トルクセンサ値	注入電力値	注入電力値	モータトルク	トルク値	トルクセンサ値	踏力	注入踏力
1	3.00	4.80	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
2	6.00	9.80	1.20	2.00	4.00	2.00	2.00	2.00	2.00	4.00	2.00	2.00	2.00	2.0
3	15.00	24.60	3.00	5.00	10.00	5.00	5.00	5.00	5.00	10.00	5.00	5.00	5.00	5.0
4	21.00	36.80	4.20	7.00	14.00	7.00	7.00	7.00	7.00	14.00	7.00	7.00	7.00	7.0
5	6.00	27.80	1.20	4.00	2.00	4.00	4.00	4.00	1.00	2.00	4.00	4.00	4.00	4.0
6	5.00	27.00	1.00	3.00	2.00	3.00	3.00	3.00	1.00	2.00	3.00	3.00	3.00	3.0
7	4.00	23.60	0.80	2.00	2.00	2.00	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.0
8	5.00	28.40	1.00	3.00	2.00	3.00	3.00	3.00	1.00	2.00	3.00	3.00	3.00	3.0
9	6.00	33.80	1.20	4.00	2.00	4.00	4.00	4.00	1.00	2.00	4.00	4.00	4.00	4.0
10	5.00	32.60	1.00	3.00	2.00	3.00	3.00	3.00	1.00	2.00	3.00	3.00	3.00	3.0
11	3.00	26.80	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
12	3.00	27.00	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0
13	3.00	26.80	0.60	1.00	2.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00	1.0

偏差注入結果の評価

2017/3

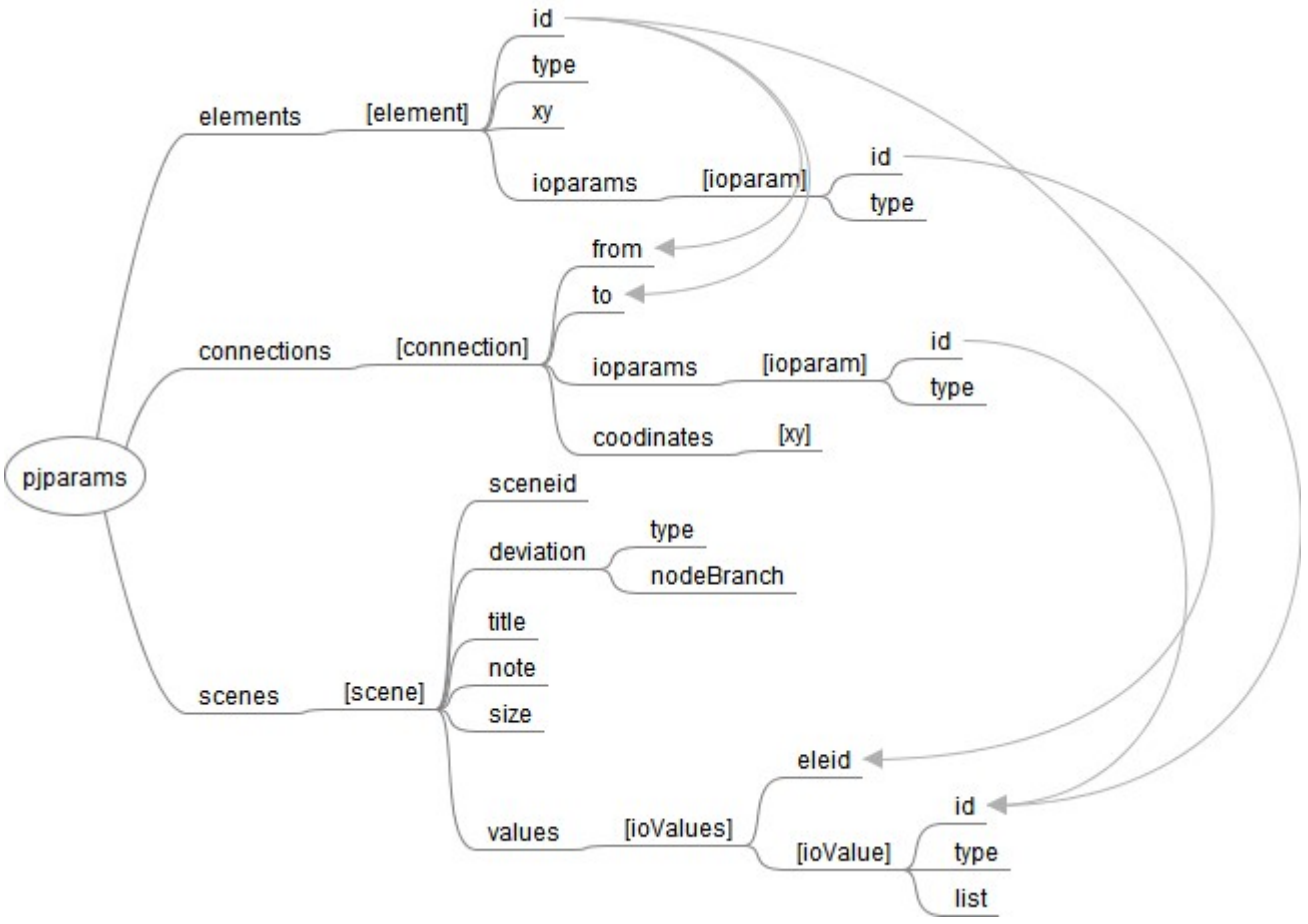
Sim4stamp

- 通常動作と偏差注入の両者の結果を比べる。
- 偏差注入した結果として値の変化が急な部分を調べる。

データフォーマット構造

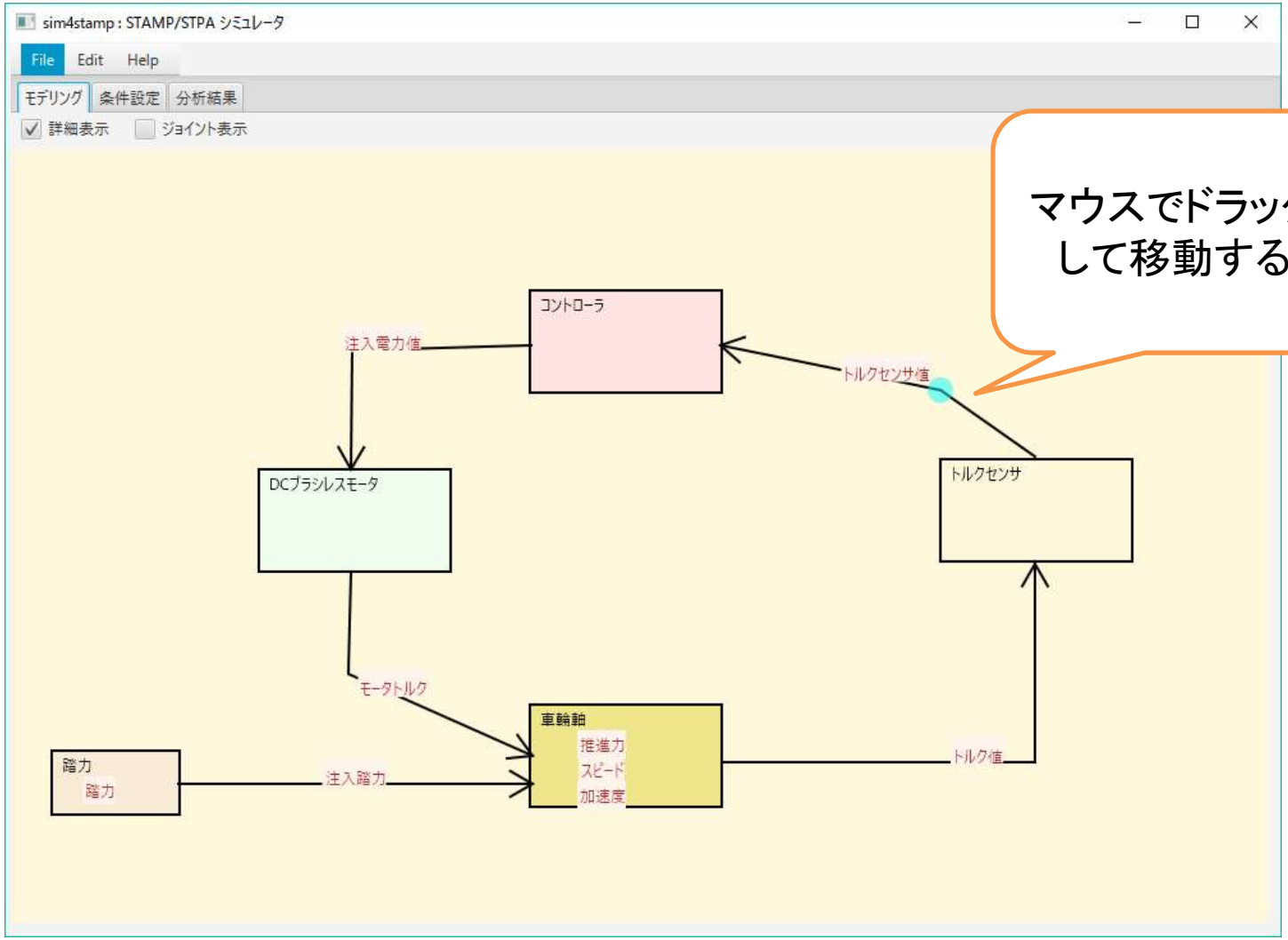
2017/4

マインドマップで表現したデータ構造



モデルの配置調整(1)

2017/5



モデリング内部管理データの生成

2017/5

- GUI上のデータを接続関係に変換する。
- GUI上のパラメータをそれぞれの要素のパラメータに付属させる。

  構成要素　：初期条件設定データ、計算結果データ
  コネクタ　：構成要素のFromからToへの受け渡しデータ

- 計算（VDM++の実行）の際の評価順序を求める。

改善点

2017/7

（1）Overtureコマンドラインツールの使用可

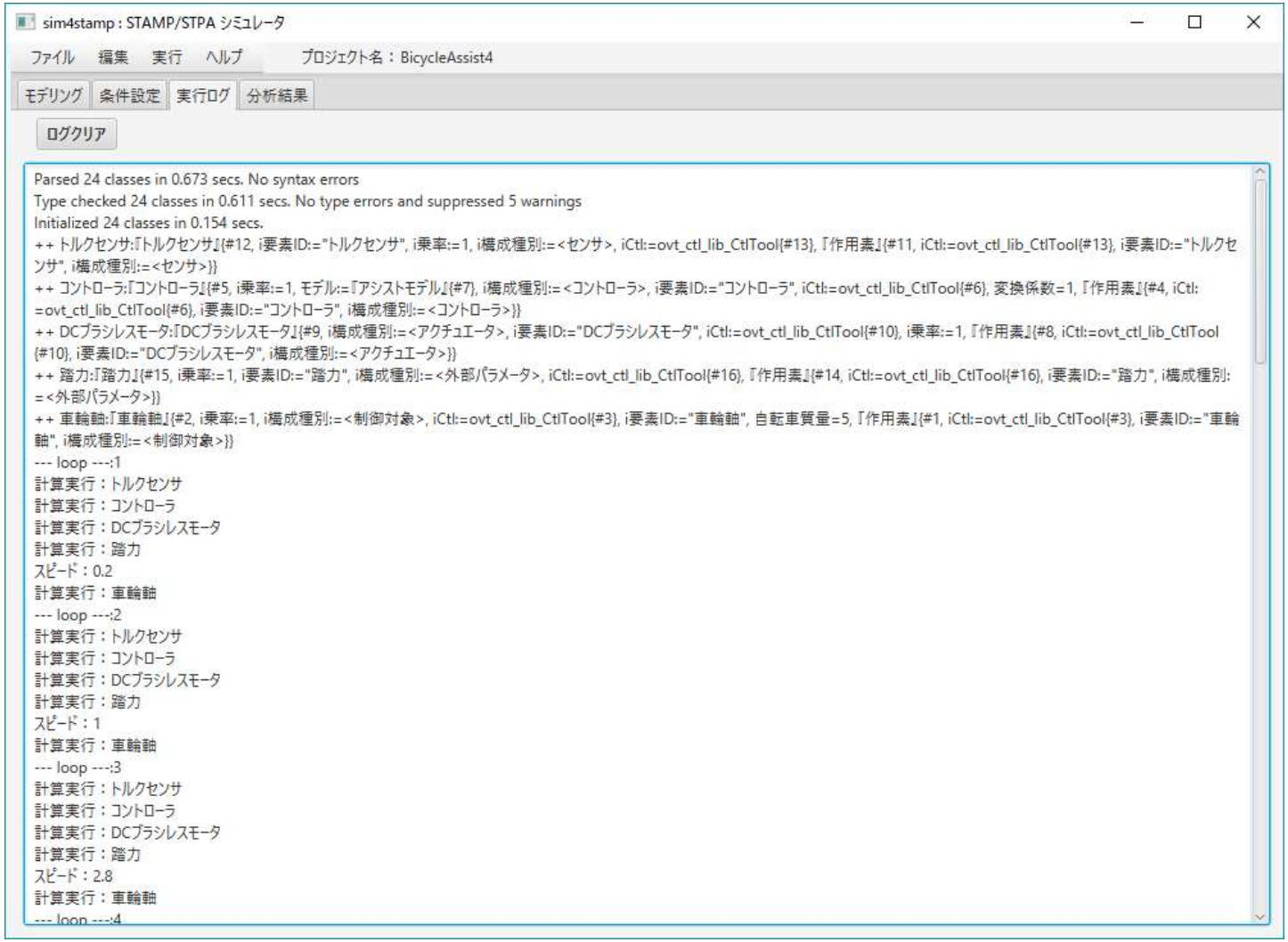
- Overture IDEとコマンドラインの両方から実行可能
- コマンドライン用の実行ログ画面の新設

（2）実行結果グラフのデータ選択

- データ選択機能の追加
- 偏差投入種別間の結果比較

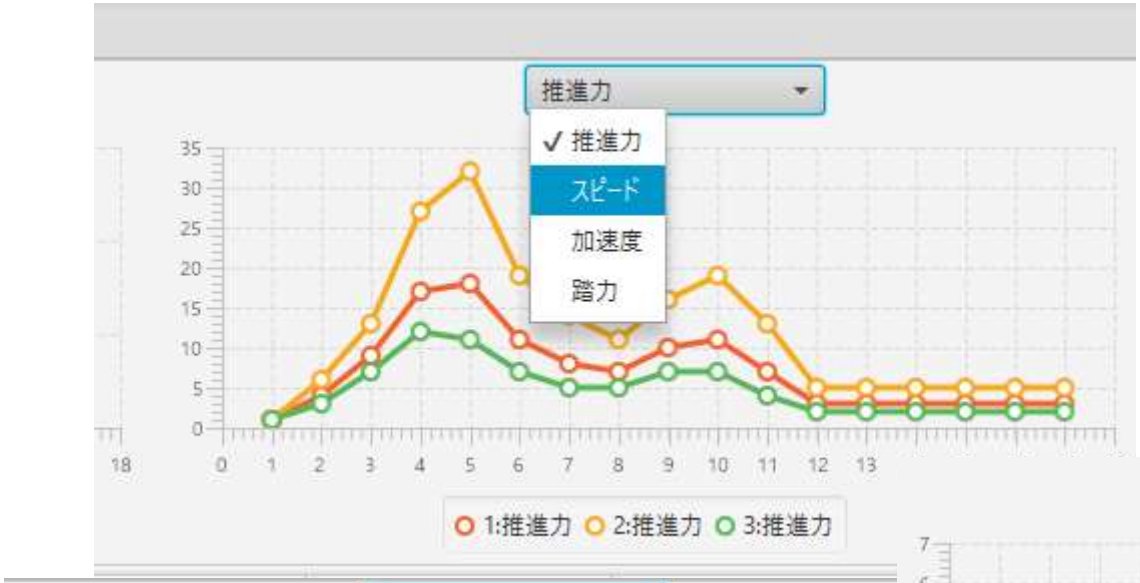
実行ログ表示

2017/7



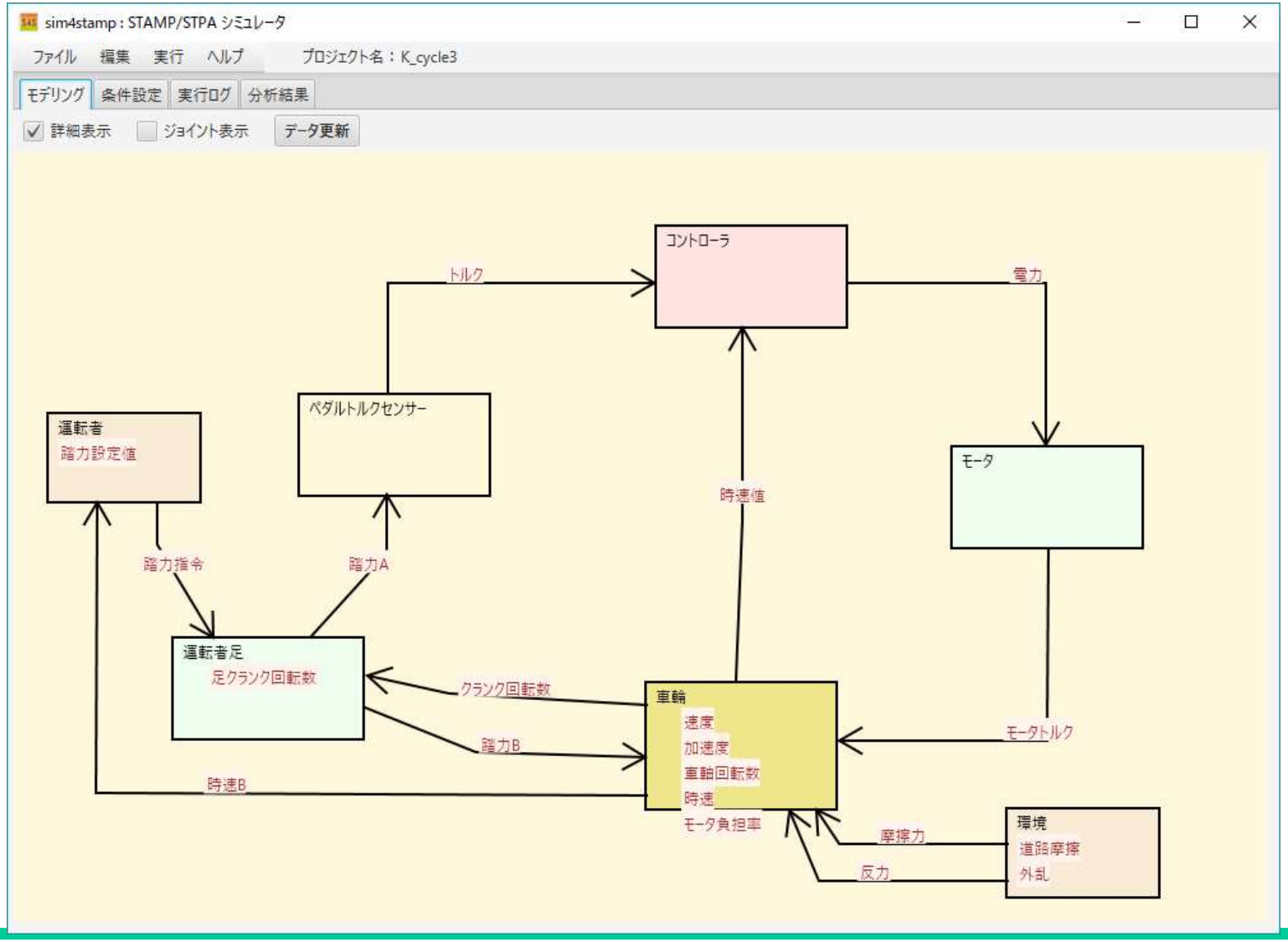
グラフ切り替え

2017/7



Sim4stampによるモデル化

2017/9～10

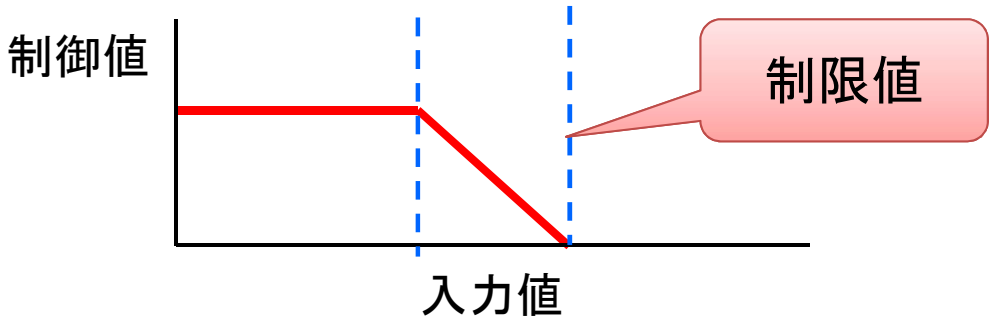


Sim4stampによるモデル化

2017/9～10

モデル化のポイント（改善後）

- 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）。**

ET2017発表

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

一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

GitHubリポジトリ

2017/11

- Sim4stampの入手先、開発への参加

- GitHubリポジトリ：
  https://github.com/KeiTsumuta/Sim4stamp

偏差の課題

2017/12

- 偏差注入を与える箇所が多い
- 偏差注入の種類も多い

偏差注入をシミュレーションですべて試すのは非効率

- 偏差注入のシミュレーションは同時に１ヶ所のみでよいのか？

複数個所の同時偏差注入をシミュレーションですべて試すのはさらに非効率

大きな影響のあるパラメータ

2017/12

- 自転車質量（自転車＋運転者）
- 運転者踏力（踏力シーケンス、応答シーケンス）
- 反力（坂道）
- アシスト係数（踏力とモータアシストの割合）

複数の要因が重なりあって最悪ケースが出現すると考えるのがリーゾナブルか？

Sim4stampの次の課題

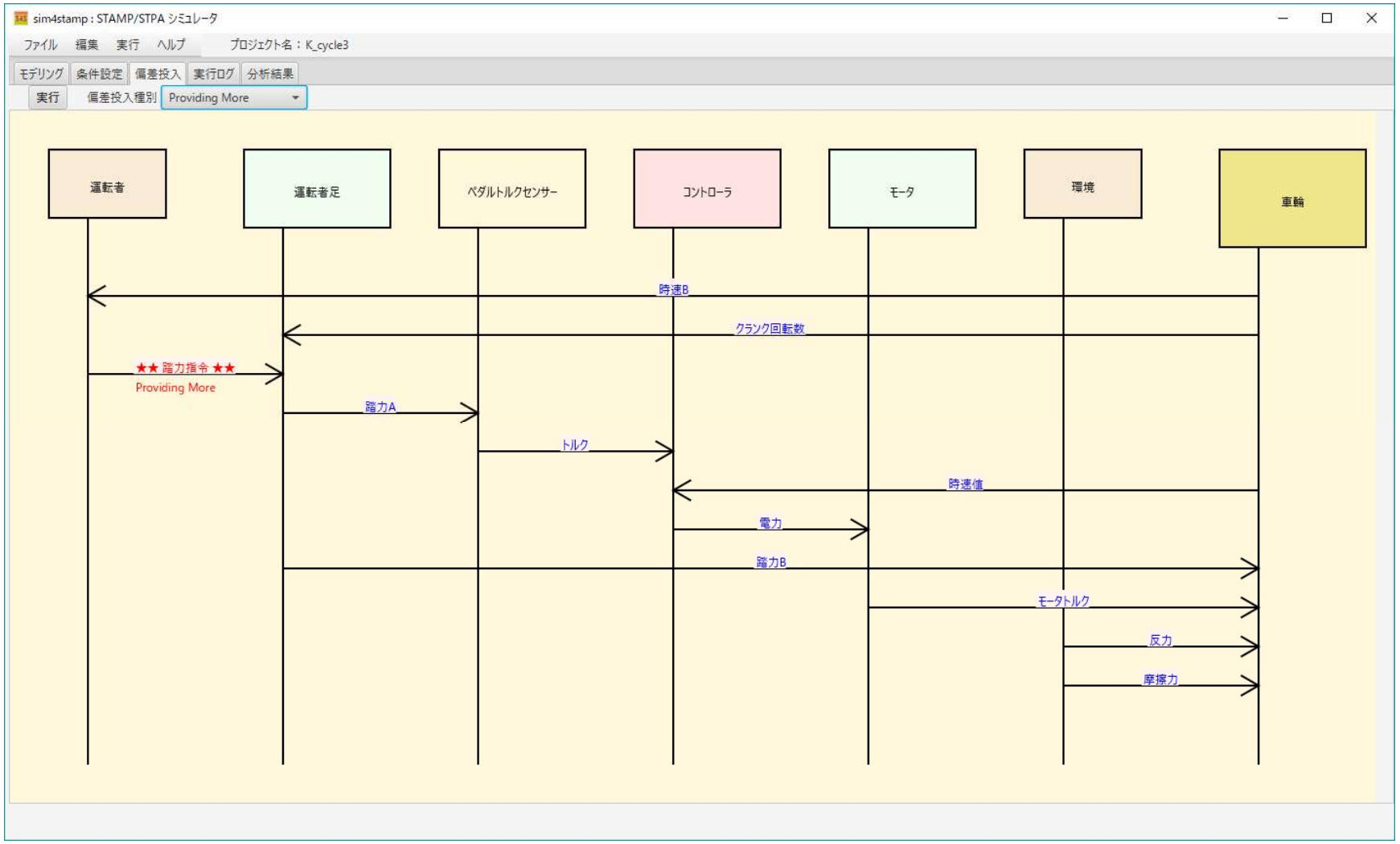
偏差注入

- 効果的な偏差注入とは？
- 画面操作上、偏差注入設定が分かり難い。

IPAツールとの連携

- データのimport, export。

偏差注入画面の検討事項



偏差注入画面の検討事項

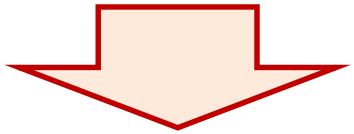
- モデル図からシーケンス図的なものを自動生成する。
  ＝＞ 表示して欲しい絵にするのが難しい。
  エレメントの表示順、コネクタの表示位置

- 制御ループを見易くする。
- 図から偏差投入を行うデータを指定する。
- 複数同時注入を行えるようにする。

まとめ ＜これからの課題＞

第一ステップ

- 機能を実現すること、正しく動作することに重点。
  プログラムの作り易さを重視。



第ニステップ以降

- いろいろなタイプの事例の具体的な試行事例から
  改良点を見つける。
- 使い易さを追求する。