

ETAS ASCMO-MOCA V5.16



ユーザーガイド

著作権について

本書のデータをETAS GmbHからの通知なしに変更しないでください。ETAS GmbHは、本書に関してこれ以外の一切の責任を負いかねます。本書に記載されているソフトウェアは、お客様が一般ライセンス契約あるいは単一ライセンスをお持ちの場合に限り使用できます。ご利用および複写はその契約で明記されている場合に限り、認められます。

本書のいかなる部分も、ETAS GmbHからの書面による許可を得ずに、複写、転載、伝送、検索システムに格納、あるいは他言語に翻訳することは禁じられています。

© **Copyright 2025** ETAS GmbH, Stuttgart

本書で使用する製品名および名称は、各社の（登録）商標あるいはブランドです。

MATLABおよびSimulinkはThe MathWorks, Inc.の登録商標です。その他の商標の一覧は[mathworks.com/trademarks](https://www.mathworks.com/trademarks)をご覧ください。

ASCMO-MOCA V5.16 | ユーザーガイド R01 JP | 11.2025

目次

1	はじめに	6
1.1	本製品に関する注意事項	6
1.2	製品の正しい使用法	6
1.3	対象ユーザー	6
1.4	安全に関する注意事項の記述書式	6
1.5	安全に関する注意事項	7
1.6	データ保護	8
1.7	データと情報のセキュリティ	8
1.7.1	データとその保管場所	8
1.7.1.1	ライセンス管理	8
1.7.2	技術的／組織的な対策	9
2	ASCMO-MOCAについて	10
2.1	基本説明	10
2.2	より詳しい情報について	11
3	インストール	12
3.1	システム要件	12
3.2	ソフトウェア要件	12
3.3	インストール	13
3.4	ファイルとディレクトリ	14
3.5	Pコードバージョン	15
3.6	ライセンス管理	16
3.7	アンインストール	17
4	ASCMO-MOCAの基礎	18
4.1	ASCMO-MOCAの応用分野	18
4.1.1	ECUのセンサデータの適合	19
4.1.2	研究、機能／システム開発	19
4.1.3	ASCMO-MOCA Runtimeの応用分野	19
4.2	データ	20
4.2.1	入力データの評価	20
4.2.1.1	モデルデータの一覧表示	20
4.2.1.2	入力の関連性の確認	21
4.2.1.3	ファンクションの評価と改良	21
4.2.2	変数RMSEおよびR2	25
4.2.2.1	RMSE（平均二乗誤差）	25

4.2.2.2	決定係数R2	25
4.2.3	RMSEとR2を用いたファンクション評価	26
4.3	パラメータ	26
4.3.1	例	27
4.3.2	パラメータの種類	27
4.3.3	システム定数	31
4.3.4	パラメータセット	32
4.4	可視化	32
4.5	モデル	39
4.5.1	定常状態	41
4.6	ファンクション	43
4.6.1	ファンクションノード用の演算子	43
4.6.2	フィードバックループ	49
4.7	最適化	50
4.7.1	最適化手法について	51
4.7.2	最適化アルゴリズム	51
4.7.3	オブティマイザのオプション	54
4.7.4	粗度の考慮	62
4.7.5	最適化目標	63
4.7.6	シーケンスを使用しない最適化	64
4.7.7	シーケンスを使用する最適化	64
4.7.8	パラメータの相関	64
4.7.9	パラメータの感度	65
4.8	関数同定問題 (Symbolic Regression)	66
4.8.1	操作方法 (関数同定問題- Symbolic Regression)	66
4.8.2	アルゴリズムの詳細 (関数同定問題 - Symbolic Regression)	69
5	ASCMO-MOCAの操作	74
5.1	ASCMO-MOCAのユーザーインターフェース	74
5.2	ASCMO-MOCAユーザーインターフェースの要素	75
5.2.1	ASCMO-MOCAのメインメニュー	75
5.2.2	ツールバー	75
5.2.3	ASCMO-MOCAのナビゲーションペイン	77
5.2.4	ログウィンドウ	79
6	チュートリアル : ASCMO-MOCAの操作	80
6.1	チュートリアルについて	80
6.1.1	チュートリアルの課題	80
6.1.2	チュートリアルの構成	80

6.1.3	測定データの要件	81
6.1.4	モデリング用データ	81
6.2	ASCMO-MOCAの起動	81
6.3	ステップ1 : データのインポート	83
6.3.1	測定データの妥当性チェックを行う	84
6.3.2	コンフィギュレーションの保存とロードを行う	88
6.3.3	測定データをインポートする	88
6.3.4	測定チャンネルを変数に割り当てる	89
6.3.5	ASCMO-MOCAのデータステップでの作業	90
6.3.5.1	データポイントの重み付けを行う	91
6.3.5.2	データセット内のデータの管理	93
6.4	ステップ2 : データの分析	95
6.5	ステップ3 : パラメータ	99
6.6	ステップ4 : モデル	100
6.6.1	Simulink®モデルとスクリプトを追加する	100
6.6.2	Simulink®パラメータをマッピングする	102
6.6.3	Simulink®入力をマッピングする	104
6.6.4	Simulink®出力をマッピングする	105
6.6.5	Simulink®モデルの検証と使用	106
6.6.5.1	Simulink®モデルを検証する	106
6.7	ステップ5 : ファンクションの構築	110
6.7.1	ファンクションのモデリング	110
6.8	ステップ6 : 最適化	117
6.9	ステップ7 : エクスポート	119
7	お問い合わせ先	122
	用語集	123
	図	125
	式	126
	索引	127

1 はじめに

本章には、使用目的、対象となるユーザー、安全性や個人情報に関する情報が記載されています。

本製品を使用する際には、ユーザーの負傷やデバイスの損壊などを避けるため、「ETAS Safety Advice - 安全上のご注意」([Help > Safety Advice](#))、および下記の注意事項をよくお読みいただき、その指示に従ってください。

製品の不適切な使用や安全に関するメッセージに従わないことにより生じた一切の損害について、ETAS GmbHは責任を負いません。

1.1 本製品に関する注意事項

本製品を安全に使用するには、一般的な注意事項に加え、以下の要件も守ってください。

- 一 本製品の準備や操作を行う前に、本製品を使用する環境が所定の条件を満たしていることを確認してください。各条件については、使用するPCやハードウェアのドキュメントを参照してください。

1.2 製品の正しい使用法

ETAS ASCMOツールファミリーは、オフラインデータに基づくモデリングや、モデルベース適合、さらには物理値モデルのパラメータの効率的な最適化を目的としています。稼働中のシステムで直接動作することは意図されていません。

ASCMO-STATICとASCMO-DYNAMICを使用すれば、少量の測定データから複雑なシステムの挙動を正確にモデル化することができます。このモデルは、入力パラメータの分析と最適化に使用したり、他のシミュレーション環境においてプラントモデルのブラックボックスとして使用することもできます。対照的にASCMO-MOCAは、定義された構造を持つ既存の物理値モデルを使用して、モデル自体のパラメータを調整して最適化します。その結果は「提案」として扱われるもので、後続の処理を行う前に検証を行う必要があります。

製品の不適切な使用や安全に関する注意事項に従わないことにより生じた一切の損害について、ETAS GmbHは責任を負いません。詳細は [Help > Safety Advice](#) を参照してください。

1.3 対象ユーザー

この製品は、自動車用ECUの開発および適合の部門において訓練を受けた有資格者を対象としています。本書の内容をご理解いただくには、計測や、制御ユニットのエンジニアリングに関する専門知識が必要です。

1.4 安全に関する注意事項の記述書式

以下の「安全に関する注意事項」は、人身事故や物的損害につながる危険性を警告するものです。

**危険**

危険：記載事項を守らないと死亡または重傷のリスクが高い危険性について説明しています。

**警告**

警告：記載事項を守らないと死亡または重傷のリスクを招く可能性のある危険性について説明しています。

**注意**

注意：記載事項を守らないと軽～中程度の負傷のリスクを招く可能性のある危険性について説明しています。

ご注意ください！

ご注意ください！：記載事項を守らないと物的損害を招く可能性のある危険性について説明しています。

注意喚起

注意喚起：記載事項を守らないとデータ損失、データ破損、システムの脆弱性といったデジタル資産への損害を招く可能性のある危険性について説明しています。

1.5 安全に関する注意事項

ご注意ください！

誤った適合データにより物的損傷の生じる恐れがあります。

ASCMO-MOCAモデルから得られた適合データの使い方を誤ると、エンジンやテストベンチの損傷を招く可能性があります。

最適化を行った後は、エクスポートを行う前に、残差解析機能を使って測定データとモデルが作成したデータとを比較してください。この機能を利用するには、Analysis > Residual Analysis > Training and Test Data > Absolute Error Analysis を選択します。

最適化を実行する（ページ118）、パラメータステップまたは最適化ステップのエクスポートオプション、および6.9 ステップ7：エクスポート（ページ119）を参照してください。

ご注意ください！

外部ソースには悪意のあるコードが潜在している可能性があります。

FMU、ASCET、TSiMタイプのモデルの場合、ASCMO-MOCAは、モデル評価中に外部ランナブルを実行します。

モデルの外部ランナブルが信頼できるソースからのものであることを確認してください。

1.6 データ保護

製品に個人データを処理する機能が含まれている場合、データ保護法およびデータプライバシー法の法的要件は、お客様によって遵守されるものとします。データ管理者として、お客様は通常、その機能に続く処理を設計します。したがって、保護措置が十分であるかどうかは、お客様に確認していただく必要があります。

1.7 データと情報のセキュリティ

本製品に関するデータを安全に取り扱うため、データと保管場所、および技術的／組織的な対策について、以降の項に書かれている内容を参照してください。

1.7.1 データとその保管場所

以下の項では、さまざまな使用例におけるデータとその保管場所について説明します。

1.7.1.1 ライセンス管理

ユーザーネットワーク上のFNPライセンスサーバーで管理されるユーザー固定ライセンスをETASライセンスマネージャで扱うと、ライセンス管理の目的で以下のデータが保存されます：

データ

- － 通信データ：IPアドレス
- － ユーザーデータ：WindowsユーザーID

保存場所：

- － FNPライセンスサーバーログファイル：カスタムネットワーク上

FNEのPC固定ライセンスとして提供されたホストベースライセンスをETASライセンスマネージャで扱うと、ライセンス管理の目的で以下のデータが保存されます：

データ

- － アクティベーションデータ：アクティベーションID
 - ライセンスのアクティベーションの目的でのみ使用され、ライセンス使用時に継続的に使用されることはありません。

保存場所

- － FNEトラステッドストレージ

C:\ProgramData\ETAS\FlexNet\fne\license\ts

1.7.2 技術的／組織的な対策

お客様のIT部門による適切な技術的／組織的対策（古典的な盗難防止策やハードウェアおよびソフトウェアへのアクセス保護など）を講じていただくことをお勧めします。

2 ASCMO-MOCAについて

ASCMO-MOCAは、与えられたデータを用いてファンクションのモデリングと適合（**Modeling and Calibration**）を行うツールです。各ファンクションは、変更可能なパラメータについての数学演算（ルックアップテーブル演算など）で構成されます。ツールの目標は、与えられたデータに対するファンクション出力の逸脱を最小化することにあります。その達成のため、最適化がファンクションの各パラメータを調整（適合）します。その際にはカーブ/マップの平滑度や傾斜といった各種制約も考慮されます。

結果はさまざまなビュー（スコープビュー、散布図など）で表示できます。外れ値などの問題点を残差分析によって検出することもできます。

ASCMO-MOCAには、2つのバージョン（フルバージョンとランタイムバージョン）があります。フルバージョンでは、ファンクションや最適化手順、さらには最適化処理自体をモデリングすることができます。ランタイムバージョンでは、フルバージョンで作成したプロジェクトを開き、データをインポートして最適化を実行することができますが、ファンクションや最適化手順の定義は行えません。

ASCMO-MOCAの構成要素は、スカラ値、ルックアップテーブル、RBF（Radial Basis Function：放射基底関数）ネットワークのほか、Simulink[®]などの別ソースのモデルです。

内部ステートとループを含まない時間依存のファンクションは、ASCMO-MOCAで直接モデリングすることができます。より複雑な時間依存ファンクションのモデリングにはSimulink[®]などの外部ツールを使用する必要があります。その場合は、ASCMO-MOCAが最適化を実行する際にも外部ツールが必要となります。

2.1 基本説明

ETAS ASCMO-MOCAは、ECUやシミュレーション環境で使用されるような物理ベースモデルのパラメータ最適化を可能にします。この目的のために、さまざまなプラントモデルやコントローラモデルをロード、接続、モデリングすることができます。また、測定データのロード、モデルパラメータのインポート/エクスポート、最適化タスクの定義も行えます。ASCMO-MOCAは、使用されているモデル以外にも、データを可視化して分析するための幅広い機能とオプションを提供します。強力なアルゴリズムにより、平滑性や単調性などの制約条件を考慮しながら、多数の自由パラメータを同時に最適化することができます。

ASCMO-MOCAの一般的な用途は、トルクや排気ガス温度などのECUモデル（仮想センサ）の予測品質を最適化することであり、エンジンテストベンチや車両内のすべての測定ポイントにおける実測値に対するモデル予測値の偏差を最小化することです。

別の応用例として、動的/過渡的な走行サイクルにおける複雑な内燃エンジンの排出ガスと燃料消費の最適化があります。これには、ASCMOの古典的なデータベースモデルをECUソフトウェアの一部にリンクさせる必要があります。ASCMO-MOCAでは、このような異なるサブコンポーネント同士の連携や共同最適化が簡単に行えます。

ASCMO-MOCAで使用される手法の対象は内燃エンジンに限定されないため、電動モビリティ（充電戦略など）やコンポーネント開発のような分野にも応用できます。

2.2 より詳しい情報について

このユーザーガイドに加え、特にユーザーインターフェースを操作する際にはオンラインヘルプのご使用をお勧めします。**Help > Online Help** でアクセスでき、また F1 を押下して現在開いているウィンドウに応じたコンテキストヘルプを開くことができます。

Pコードバージョンの機能についてのヘルプにアクセスするには、**Help > Interface Help** を使用します。

3 インストール

インストールを行う前に、お使いのPCがシステム要件を満たしていることを確認してください（システム要件はMOCA ASCMOを参照）。必要なユーザー権限とネットワーク接続があることも確認してください。

製品をオフラインで使用するには、ETASライセンスマネージャ（**LiMa** メインウィンドウ > **ライセンス** > **選択されたライセンスの借用／全ライセンスの借用**）でライセンスを借用する必要があります。詳細は[Licensing](#)を参照してください。

3.1 システム要件

システムが以下の最小要件を満たしている必要があります：

必要なハードウェア	1.0 GHz PC 4 GB RAM 解像度1024 x 768以上、32 MB RAM以上のグラフィックス
必要なオペレーティングシステム	Windows® 10、Windows® 11
必要なディスクの空き容量	4 GB（アプリケーションデータを含まない）

推奨されるシステム要件は以下のとおりです：

推奨ハードウェア	4.0 GHz Quad-Core PC、またはそれと同等のもの 32 GB RAM 解像度1680 x 1050、128 MB RAMのグラフィックス
推奨オペレーティングシステム	Windows® 10、Windows® 11
推奨されるディスクの空き容量	4 GB以上

3.2 ソフトウェア要件

ETAS ASCMOにはMATLAB® Compiler Runtime 2022bが必要で、ない場合は自動的にインストールされます。また、Windows® 10/11に含まれる.Net Framework V4.6も必要です。

ETAS ASCMOの基本製品およびアドオンのインストール時に必要なソフトウェア要件はありません。足りないソフトウェアコンポーネントは、インストール実行中にすべてインストールされます。

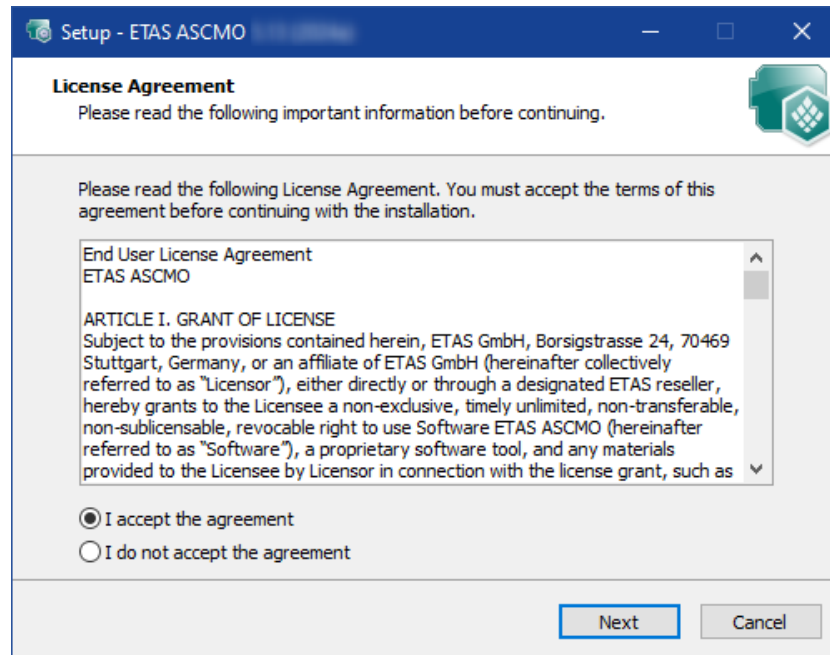
ETAS ASCMOアドオンの*Software Developer Kit (SDK)*を使用するには、MATLAB®バージョンR2021b ~ R2023bと、MATLAB®の*Optimization Toolbox*と*Statistics Toolbox*が必要です。

3.3 インストール

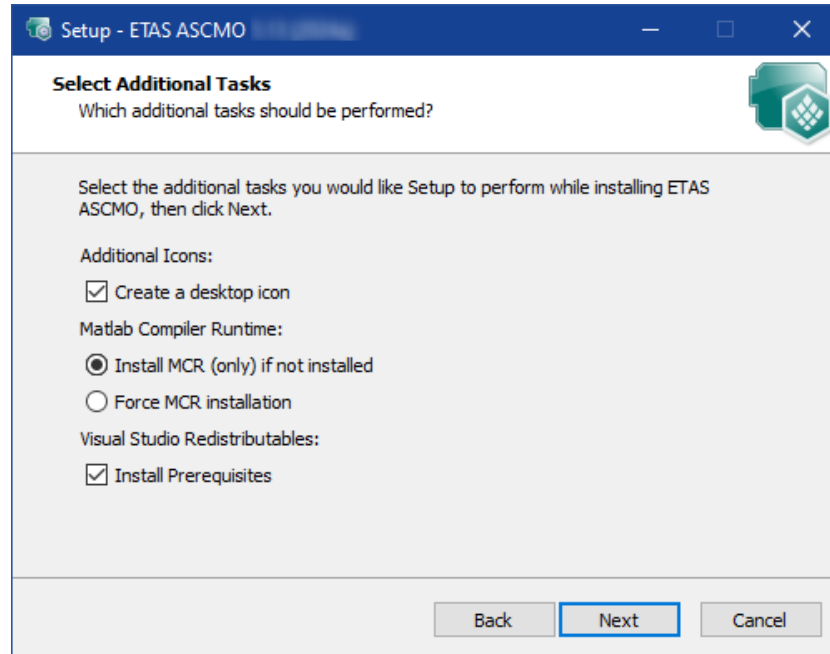
ETAS ASCMOのインストール

1. ASCMOインストールファイルが格納されているディレクトリを開きます。
2. Setup_ETAS-ASCMO_Vx_x_20xxx.exeをダブルクリックします。

License Agreement ウィンドウが開きます。



3. ライセンス許諾書の内容をよく読んで、**I accept the agreement** オプションを選択します。
4. **Next** をクリックします。
すでにASCMOがインストールされている場合は、既存のインストールパス（インストール先とスタートメニューフォルダ）が使用され、ステップ5～8はスキップされます。
5. **Set Destination Location** ウィンドウで、提示されるデフォルトフォルダを承諾するか、または **Browse** をクリックして新しいディレクトリを選択します。
6. **Next** をクリックします。
7. **Select Start Menu Folder** ウィンドウで、提示されるデフォルトフォルダを承諾、または **Browse** をクリックして新しいフォルダを選択します。
8. **Next** をクリックします。



- i. デスクトップアイコンを作成するには **Create a desktop icon** チェックボックスをオンにします。
- ii. MATLAB[®] Compiler Runtimeを強制的にインストールするか、または、まだインストールされていない場合にのみインストールするかを選択します。
- iii. 必要に応じて **Install Prerequisites** をオンにします。

9. **Next** をクリックします。
10. **Ready to Install** ウィンドウで、**Install** をクリックしてインストールを開始します。

または

設定内容を変更するには **Back** をクリックします。

インストールが開始され、進捗インジケータにインストールの処理状況が表示されます。インストール処理がすべて終了すると、**Completing the ETAS ASCMO Setup Wizard** ウィンドウが開きます。

11. **Finish** をクリックします。
- ⇒ インストールが完了します。ASCMOが起動できる状態になります。

3.4 ファイルとディレクトリ

プログラムに関連するすべてのファイルは、インストール時に指定された *<installation>* ディレクトリとそのサブディレクトリに保存されています。

<installation> のデフォルトは `C:\Program Files\ETAS\ASCMO x.x` です。

スタートメニュー

インストールが正常に終了すると、インストーラの **Select Start Menu Folder** ウィンドウで設定した**スタートメニュー**のフォルダに、以下のエントリが追加されます。

- **ASCMO Desk V5.16**
ASCMO-DESKウィンドウが起動し、ここからETAS ASCMOの各コンポーネントを起動することができます。
- **ASCMO Dynamic V5.16**
ASCMO-DYNAMICを起動します。
- **ASCMO ExpeDes Dynamic V5.16**
ASCMO-DYNAMIC ExpeDesを起動します。
- **ASCMO ExpeDes V5.16**
ASCMO-STATIC ExpeDesを起動します。
- **ASCMO MOCA Runtime V5.16**
ASCMO-MOCA Runtime（機能制限付き）を起動します。
- **ASCMO MOCA V5.16**
ASCMO-MOCAを起動します。
- **ASCMO Static V5.16**
ASCMO-STATICを起動します。
- **Manuals and Tutorials**
ASCMOドキュメントフォルダ（<installation>\Manuals）が開きます。ここには以下の情報や文書が保存されています。
 - ASCMOInterfaceDoc - インターフェースに関する文書が保存されたフォルダ
 - Examples - 各種サンプルデータ（ASCMOプロジェクト、MF4/DCM/XLS/FMUファイル、テンプレート、プラグインなど）が含まれるフォルダへのショートカット
 - HTMLフォルダ - インストール済みコンポーネント用のオンラインヘルプファイル（<F1> キーで開きます）
 - ASCMO-DYNAMIC_V5.16_User_Guide_*.pdf - ユーザーガイド - ASCMO-DYNAMICの基本機能の説明やチュートリアルを含む
 - ASCMO-STATIC_V5.16_User-Guide_*.pdf - ユーザーガイドASCMO-STATICの基本機能の説明やチュートリアルを含む
 - ASCMO-MOCA_V5.16_User-Guide_*.pdf - ユーザーガイドASCMO-MOCAの基本機能の説明やチュートリアルを含む

Pコードファイル

MATLAB®/Simulink®用のPコードファイルは、<installation>\pCode\ascmo に保存されています。

詳細はPコードバージョン（下記）を参照してください。

3.5 Pコードバージョン

PコードバージョンのETAS ASCMOは、MATLAB®から直接起動することができます。

前提条件

Pコードバージョンを使用するには、MATLAB® R2021b ~ R2023bがインストールされている必要があります。さらに、以下のMATLAB®ツールボックスも必要です。

- Optimization Toolbox™
- Statistics and Machine Learning Toolbox™

ETAS ASCMOの実行

MATLAB®で、ディレクトリ `<installation>\pCode\ascmo` に切り替えます。続いてコマンドウィンドウに以下の表のコマンドを入力します。

コマンド	機能
<code>AscmoDesk</code>	ASCMO-DESKを起動します。
<code>ascmo static</code>	ASCMO-STATICを起動します。
<code>ascmo expedes</code>	ASCMO-STATIC ExpeDesを起動します。
<code>ascmo dynamic</code>	ASCMO-DYNAMICを起動します。
<code>ascmo expedesdynamic</code>	ASCMO-DYNAMIC ExpeDesを起動します。
<code>ascmo moca</code>	ASCMO-MOCAを起動します。
<code>ascmo mocaruntime</code>	ASCMO-MOCA Runtimeを起動します。
<code>ascmo cyclegenerator</code>	ASCMO Cycle Generator (スタンドアロン走行サイクルジェネレータ) を起動します。
<code>ascmo essentials</code>	ASCMO Essentialsを起動します。

ETAS ASCMOツール内で実行される全ステップは、コマンドを使用して自動化することができます。コマンドについての説明を参照するには、メインメニューから **>Interface Help** を選択してください。

3.6 ライセンス管理

本ソフトウェアを使用するには、有効なライセンスが必要です。ライセンスは、以下のいずれかの方法で入手できます：

- ツール管理者の方から
- ETASウェブサイトwww.etas.com/support/licensingのセルフサービスポータル経由で
- ETASライセンスマネージャ経由で

ライセンスをアクティベートするには、製品のご購入時にETASから発行されたアクティベーションIDが必要です。

ETASライセンス管理の詳細については、[ETAS License Management FAQ](#) またはETASライセンスマネージャのヘルプを参照してください。

ETASライセンスマネージャのヘルプを開く：

ETASライセンスマネージャは、ETASソフトウェアをインストールしたPC上でご利用いただけます。

1. Windowsスタートメニューから **E > ETAS > ETAS License Manager** を選択します。
ETASライセンスマネージャが開きます。
2. ETASライセンスマネージャのウィンドウで、F1 を押します。
ETASライセンスマネージャのヘルプが開きます。

3.7 アンインストール

**注記**

各コンポーネントを個別にアンインストールすることはできません。この処理によって **すべての** ETAS ASCMOコンポーネントがアンインストールされます。

ETAS ASCMOのアンインストール

1. ASCMOインストールファイルが格納されているディレクトリを開きます。
アンインストールを開始します。
確認のメッセージが開きます。
2. `unins000.exe` をダブルクリックします。
確認のメッセージが開きます。
3. ETAS ASCMOとそのコンポーネントを完全に削除するには、**Yes** をクリックします。
アンインストールが開始されます。アンインストールが完了すると、メッセージウィンドウが開きます。
4. **OK** をクリックしてアンインストールを完了します。
⇒ ETAS ASCMOとそのすべてのコンポーネントが正しくアンインストールされます。

4 ASCMO-MOCAの基礎

ASCMO-MOCAは、モデルパラメータを最適化し、モデルの予測結果と実センサの出力値との差異を最小化することを可能にするツールです。

最新のECUには、物理学に基づくモデルが組み込まれ、実センサの役割を果たしたり監視を行ったりしているものがあります。このような物理モデルは汎用的なものなので、実際のエンジンに合わせた調整が必要です。そのためには、テストベンチや車両などを用いて実際に計測した値を用いてモデルパラメータ（マップ、カーブ、スカラー）を最適化する必要があります。

ASCMO-MOCAは、ユーザーが入力した一連の式によりモデルを表現します。あるいは、Simulink[®]などで作成されたモデルを用いることもできます。

本章ではASCMO-MOCAの機能と一般的な操作方法について説明します。

具体的な内容は以下のとおりです。

- **ASCMO-MOCAの応用分野（下記）**

本項では、ASCMO-MOCAの幅広い応用分野について、概要を紹介します。

- **ASCMO-MOCAユーザーインターフェースの要素（ページ75）**

本項では、ASCMO-MOCAのユーザーインターフェースの構成と機能について簡単に説明します。

- **データ（ページ20）**

本項では、測定データのインポート、分析、処理について説明します。

- **入力データの評価（ページ20）**

本項では、ASCMO-MOCAがパラメータ最適化の目的で使用する入力データの品質にアクセスする方法について説明します。

- **モデル（ページ39）**

本項では、外部モデルをASCMO-MOCAにインポートして使用方法について説明します。

- **ファンクション（ページ43）**

本項では、一連の式で構成されるファンクションを定義することによってモデルを作成する方法について概説します。

- **パラメータ（ページ26）**

本項ではASCMO-MOCAで行われるパラメータの最適化処理について概説します。

- **パラメータの種類（ページ27）**

本項では、最適化（ステップ6：最適化（ページ117）参照）のためのファンクションで利用できるパラメータのタイプ（ステップ5：ファンクションの構築（ページ110）参照）についての概要を説明します。

- **最適化（ページ50）**

本項では、パラメータの最適化に使用できる各種最適化メソッドと最適化目標について説明します。

4.1 ASCMO-MOCAの応用分野

本項では、ASCMO-MOCAの幅広い応用分野について、概要を紹介します。

4.1.1 ECUのセンサデータの適合

- － パラメータの最適化
- － 時間に依存する動的ファンクションの最適化
- － ECUモデルのパラメータ設定（シリンダ充填、トルク、...）

ASCMO-MOCAを使用して適合を行うと、以下の一連の効果が得られます。

- － 計測と分析の工数低減による効率向上
- － 複雑性への対応
- － データ品質の向上
- － モデルの再利用化

4.1.2 研究、機能／システム開発

- － 実験用エンジンの迅速な適合と評価
- － エンジンモデルを使用した新しい機能（新しいコントローラストラテジなど）の開発と試験
- － 未知のシステムの分析と最適化

研究・開発の領域においては、さまざまなインパクト解析が行え、システムをより短時間でより深く把握することが可能になります。

4.1.3 ASCMO-MOCA Runtimeの応用分野

ASCMO-MOCAの「ランタイムバージョン」は、一部の機能へのアクセスが制限されています。制限をかける理由としては、IP保護や、重要な情報が不用意に変更されてしまうことを防ぐ、といったことがあります。

このランタイムバージョンはPCに単独でインストールできますが、メインバージョン（開発者用バージョン）と同じPCにインストールして同時に運用することもできます。



注記

ランタイムバージョンでは、ファンクションの作成や修正が行えません。

ASCMO-MOCA Runtimeでは以下の作業が行えます。

- － 定常データや過渡データのインポートと、名前のマッピング
- － 変換規則（変換パラメータ／変換式）の定義
- － パラメータとシステム定数のインポート、エクスポート、作成、削除、編集
- － パラメータの反復的な最適化と適合

特に、最適化条件を含むプロジェクトを作成した担当者とは異なる別の担当者が最適化を行うようなケースにおいては、ASCMO-MOCA Runtimeのインストールをお勧めします。

知的財産の保護や安全性の確保に役立ちます。

- － ファンクションや最適化ロジックに関する特殊なノウハウを他の担当者と共有する必要がありません。
- － 重要なパラメータや設定は最適化を行うユーザーによって変更されません。予期しないモデル挙動の発生を防止することができます。

4.2 データ

ASCMO-MOCAでは、まず最初に測定データをインポートし、その分析と前処理を行います。これらの操作はデータステップで行います。

詳しい操作方法は、以降の各項、またはオンラインヘルプを参照してください。

- 入力データの評価（下記）
- ステップ1：データのインポート（ページ83）（チュートリアル）

4.2.1 入力データの評価

本項では、ASCMO-MOCAがパラメータの最適化に使用する入力データの品質を評価する方法について説明します。

- モデルデータの一覧表示（下記）
- 入力に関連性の確認（次ページ）
- ファンクションの評価と改良（次ページ）
 - データ/ファンクションノードのグラフィカル分析（次ページ）
 - 残差分析（ページ22）
 - モデル品質の改良（ページ25）
- 変数RMSEおよびR2（ページ25）
- RMSEとR2を用いたファンクション評価（ページ26）

4.2.1.1 モデルデータの一覧表示

メニューコマンド **Analysis > Data Table > Training Data / Test Data / Training and Test Data** を選択すると、インポートされたデータの列、変換されたデータの列、さらにファンクションで算出されたノードを表示するテーブルが開きます。最適化目標が定義されている場合は、残差も表示されます。

注記

"* Data"ウィンドウ内のデータは編集できません。

このテーブルには、以下の値が表示されます。

- インポート済みデータ
- 変換規則による変換済みデータ
- 各ファンクションのノード
- 最適化目標に基づく残差

File	Dataset No.	Weight	Speed	Rel_Airmass [-]	Ignition [-]	Torque_Meas [-]	trqOpt [-]	ignOpt [-]	deltaSpark [-]	etaSpark [-]	product [-]	dragTorque [-]
1	1	1	597	47.8089	-26.8200	7.0220	283.9725	24.9511	51.7711	0.1799	51.0782	40.2839
2	1	1	597	51.6611	-26.8200	19.5251	312.5512	24.5889	51.4089	0.1832	57.2704	35.9652
3	1	1	597	40.1047	-22.2111	-3.6778	225.5456	25.5795	47.7905	0.2169	48.9110	49.0773
4	1	1	597	43.9568	-22.2111	8.1453	254.7590	25.2653	47.4763	0.2198	55.9899	44.6806
5	1	1	597	47.8089	-22.2111	21.1618	283.9725	24.9511	47.1621	0.2227	63.2394	40.2839
6	1	1	881.2105	47.8089	-22.2111	1.0806	276.6741	26.2306	48.4416	0.2108	58.3248	43.6713
7	1	1	597	51.6611	-22.2111	34.1432	312.5512	24.5889	46.8000	0.2261	70.6555	35.9652
8	1	1	881.2105	51.6611	-22.2111	12.9679	304.4300	26.2328	48.4439	0.2108	64.1695	39.5691
9	1	1	597	55.5132	-22.2111	47.6720	326.5844	23.1275	45.3385	0.2396	78.2627	33.4335
10	1	1	597	59.3653	-22.2111	62.1931	340.6176	21.6660	43.8771	0.2542	86.5820	30.9017
11	1	1	597	32.4005	-17.6021	-14.1937	167.1187	26.2079	43.8100	0.2549	42.6035	57.8708
12	1	1	597	36.2526	-17.6021	-1.9017	196.3321	25.8937	43.4958	0.2584	50.7297	53.4740
13	1	1	881.2105	36.2526	-17.6021	-18.1929	191.5710	26.0289	43.6310	0.2569	49.2145	56.1435

図4-1: "* Data" ウィンドウ

4.2.1.2 入力の関連性の確認

データをインポートする際に、入力の関連性（出力に対する影響度）を確認することができます（チュートリアルの入力の関連性を確認する（ページ86）を参照）。

これを調べるためには、入力と出力に対して多項式の段階的回帰が行われます。重要度が5%未満の入力が無視されるので、依存性のある入力を見つけることができます。

たとえば、トレーニングデータに3つの入力（*speed*、*load*、*speed + load*）が含まれていて、そのうちのひとつの重要度が低い、といった例が挙げられます。

各入力の順番が重要になります。段階的回帰によって入力は列ごとに並べ替えられ、擬似RMSEが入力ごとに計算されて、ヒューリスティックな入力の関連性が得られます。調査結果は"Relevance of Inputs" ウィンドウにプロットされます。

4.2.1.3 ファンクションの評価と改良

Analysis メニューには、モデルから出力される予測値をファンクション出力の測定データと比較するための多くの機能が含まれています。主な機能は、以下のとおりです。

- － 測定データとファンクションノードのグラフィカル分析
詳細はデータ/ファンクションノードのグラフィカル分析（下記）を参照してください。
- － 残差分析
詳細は残差分析（次ページ）を参照してください。

データ/ファンクションノードのグラフィカル分析

Analysis メニューで以下の散布図ウィンドウを開くと、測定データとファンクションをグラフィカルに評価することができます。

- － "Data - Training Data/Test Data/Training and Test Data"
- － "FunctionNode - Training Data/Test Data/Training and Test Data"
- － "Data and Nodes - Training Data/Test Data/Training and Test Data"

測定データを評価する際には、特に以下の点を考慮してください。

- － すべてのパラメータが実験計画（DoE : Design of Experiment）に従って変化し、計測対象のシステムが目的に合った動作モードを示しているか
- － 出力変数の値は物理的に意味のある範囲内になっているか
- － 除外する必要のある外れ値が含まれていないか

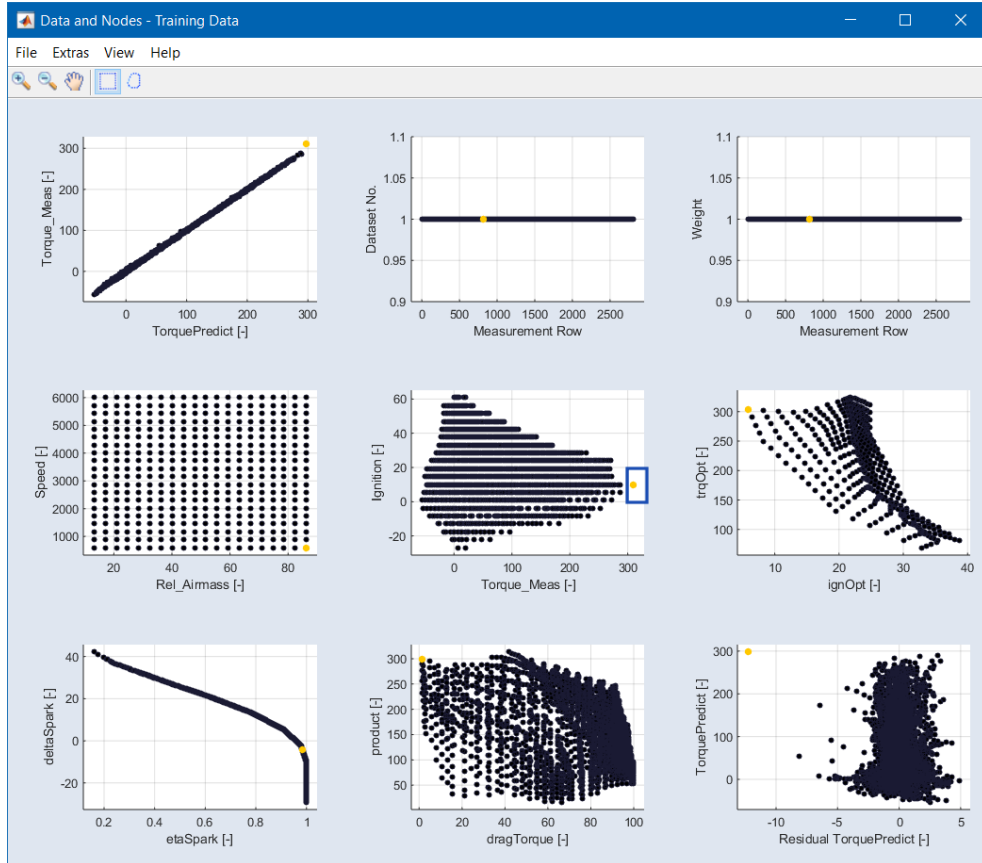


図4-2: "Data and Nodes" ウィンドウ

残差分析

残差とは、最適化目標に基づいて計算されたデータと測定データとの差です。

残差分析には3つのタイプがあります。

— Absolute Error Analysis (絶対誤差分析)

絶対誤差分析では、残差の値がすべて表示されます。

$$Y_{measured} - Y_{predicted}$$

— Relative Error Analysis (相対誤差分析)

相対誤差分析では、残差を測定値で割った商に100を掛けた値が表示されます。

$$100 \cdot \left(\frac{Y_{measured} - Y_{predicted}}{Y_{measured}} \right)$$

つまり、相対誤差がパーセント表示されます。

— Studentized Error Analysis (スチューデント化誤差分析)

スチューデント化誤差分析を実行すると、残差をRMSE4.2.2.1 RMSE (平均二乗誤差) (ページ25) で割った商が表示されます。

$$\frac{Y_{measured} - Y_{predicted}}{RMSE}$$

つまり、RMSEに対する残差の割合が表示されます。

残差分析を行うには、メインメニューから **Analysis > Residual Analysis > *** を選択します。これにより、以下の4つのプロットウィンドウが表示されます。

"Histogram" ウィンドウ

"Histogram" ウィンドウには、ファンクション出力の予測値に対する現在の残差分布が青色の棒グラフでヒストグラム表示されます。さらに正規分布も赤色の曲線で表示されます。この機能により、現在の残差分布が正規分布に当てはまっているかどうかを検証することができます。

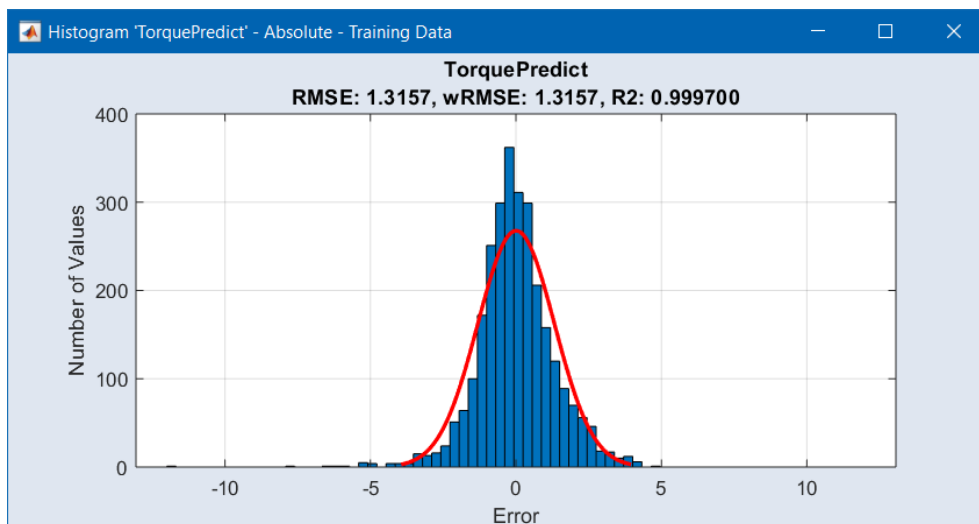


図4-3: "Histogram" ウィンドウ

"Residuals over Inputs" ウィンドウ

このウィンドウには複数の散布図が含まれ、データセット番号、各測定番号に対するアクティブフラグと重み、測定データに対する算出データの誤差（絶対誤差／相対誤差／スチューデント化誤差）が表示されます。詳細は[モデル品質の改良（ページ25）](#)を参照してください。

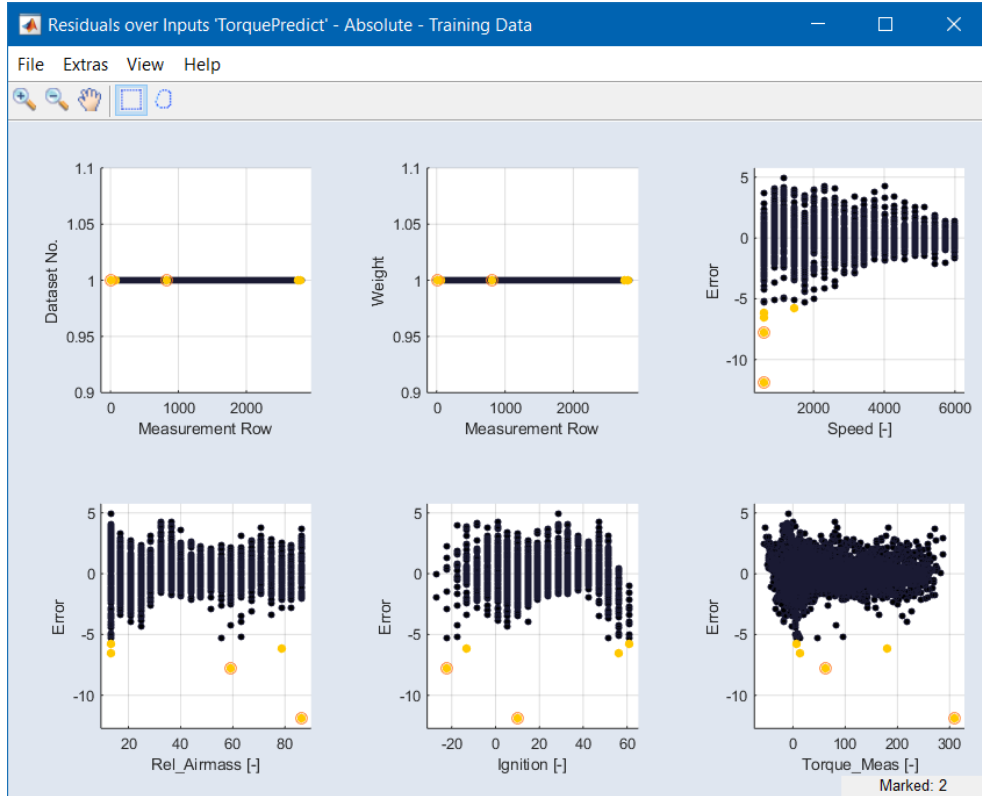


図4-4: "Residuals over Inputs" ウィンドウ

"Residuals over Outputs" ウィンドウ

このウィンドウには複数の散布図が含まれ、ファンクションノードに対する算出データの誤差（絶対誤差／相対誤差／スチューデント化誤差）が表示されます。

"Measured vs. Predicted" ウィンドウ

このウィンドウには、モデル出力をX値、測定値をY値とする散布図が表示されます。出力値と測定値が完全に一致すると、「パールのネックレス」($y = x$)が描かれます。この $y = x$ ラインから外れるポイントが多いほど測定値とモデル出力の差が大きいことになります。

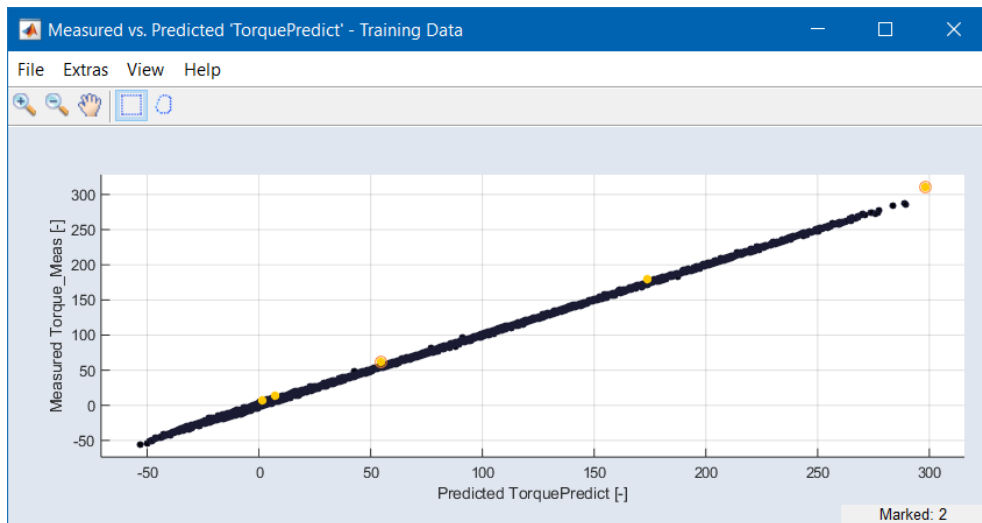


図4-5: "Measured vs. Predicted" ウィンドウ

"Residuals over *" ウィンドウと "Measured vs. Predicted" ウィンドウについての詳細は、オンラインヘルプを参照してください。

モデル品質の改良

外れ値が発生する原因としては、測定誤差や不十分なファンクション品質が考えられます。データ/ファンクションノードのグラフィカル分析（ページ21）と残差分析（ページ22）の各セクションに示された散布図を利用して、モデル品質を視覚的に判定し、改良することができます。外れ値を囲み線（四角形または投げ縄）で囲んでマークし、削除したり、非アクティブ状態にしたり重みを低くしたりすることができ、基準値を設定して自動的に外れ値を検出することもできます。

4.2.2 変数RMSEおよび R^2

ファンクション品質の定量化には、一連の変数が使用されます。ここではこれらの変数について説明します。

4.2.2.1 RMSE（平均二乗誤差）

RMSE（Root Mean Square Error：平均二乗誤差）は、モデルについて予想される「ばらつき」（標準偏差）を表します。第2の測定値は、68%の確率でモデル予測値から1 RMSE未滿（95.5%で2 RMSE未滿、99.7%で3 RMSE、など）になります。

RMSEは以下のように定義されます。

$$RMSE = \sqrt{\frac{SSR}{N}}$$

式4-1: 平均二乗誤差（RMSE）

ここで、Nは測定データの数です。また、SSRは以下の式で表されます。

$$SSR = \sum_{i=1}^N (X_{i,predicted} - X_{i,measured})^2$$

式4-2: 残差二乗和（SSR）

このように、SSRは残差の二乗の総和（SSR = **S**um of **S**quared **R**esiduals）になります。

4.2.2.2 決定係数 R^2

決定係数 R^2 は、モデルトレーニング後に残るばらつき（SSR）と、すべての測定データの平均値のばらつき（SST）との比較から、以下のように導き出されます。

$$R^2 = 1 - \frac{SSR}{SST}$$

式4-3: 決定係数 R^2 - ここでSSTは以下の式で表されます。

$$SST = \sum_{i=1}^N (X_{i,measured} - \bar{X}_{i,mean})^2$$

式4-4: 二乗の総和（SST）

R^2 はファンクションの出力誤差を評価するための相対的な指標です。分散した測定データのどの部分がファンクションにより表されているかを示します。

4.2.3 RMSEと R^2 を用いたファンクション評価

R^2 による評価

最も重要な変数は決定係数 R^2 （決定係数 R^2 （前ページ））です。この指標によりファンクションは以下のように評価されます。

- － 決定係数 R^2 の値は最大で1になります。この場合、ファンクションの予測値は各測定値に正確に合致します。
- － ファンクションが、すべての入力データについて、測定される出力の平均を単純に予測する場合は、 R^2 の値は0になります。 R^2 が負の値の場合は、予測結果が単純な予測よりも悪いことを示します。
- － R^2 が1であることは完全な一致を意味し、ファンクションの予測がすべて測定データと一致します。ただし一般的な測定データにはノイズが含まれています。この場合、 R^2 が1であることはオーバーフィッティングであることを意味します。ノイズを考慮したうえでより高い R^2 を目指すようにしてください。
- － 各信号は、それぞれ異なる精度で測定されたものである可能性があります。信号によっては R^2 が0.6であっても十分に優れた品質であるといえます。その反面、他の信号では、 R^2 が0.99を超えないと好ましい品質とはいえない場合もあります。

RMSEによる評価

絶対誤差RMSE（RMSE（平均二乗誤差）（前ページ）を参照）は、個別に評価する必要があります。

- － 最良の場合、RMSEは実験再現性と同程度の望ましさになる可能性があります。
- － R^2 が望ましい値であっても、モデリングされた変数の変動範囲が非常に大きい場合などは、RMSEが小さくなり過ぎてしまう可能性があります。
- － R^2 の値が小さくても、ファンクションの入力パラメータ全域にわたり、モデリングされた変数の変動範囲が僅かである場合などは、RMSEは十分に望ましい値になっている可能性があります。

4.3 パラメータ

今日では多くのECUに以下のようなさまざまな物理モデル¹⁾が搭載され、実センサをエミュレートしたり監視したりする機能が実装されています。

- － エンジントルク
- － 給気量
- － 排ガス温度
- － 燃料補正

¹⁾ これは、一般的なHiLシステムなどで使用されるモデルと同様のものです。

各モデルには、予測値を算出するためのマップ（マップ（Map）（次ページ）を参照）やカーブ（カーブ（Curve）（ページ29）を参照）といった制御パラメータが含まれ、テストベンチや車両などで計測されたデータを用いてこれらの「適合作業」を行う必要があります。

しかし最近のエンジンはアクチュエータの数が多く、それらに関する機能の複雑さやパラメータ数が増加し続けています。

手動での適合作業は非常に時間がかかり、場合によっては不可能な場合もあります。

ASCMO-MOCAは、このような煩雑な適合と最適化の作業を効率的に行うためのツールです。

4.3.1 例

チュートリアル：ASCMO-MOCAの操作（ページ80）では、センサ用パラメータの最適化の例を紹介しています。さまざまなマップやカーブを最適化し、測定値とモデル予測値との差異を小さくしていきます。

4.3.2 パラメータの種類

本項では、最適化（ステップ6：最適化（ページ117）参照）の対象となる各ファンクション（ステップ5：ファンクションの構築（ページ110）参照）内で使用される各種パラメータについての概要を説明します。

パラメータには以下の種類があります。

- － マップ（Map）（次ページ）
- － カーブ（Curve）（ページ29）
- － スカラ（Scalar）（ページ30）
- － 3D／4Dキューブ（Cube-3D / Cube-4D）（ページ30）
- － 圧縮モデル（Compressed Model）（ページ30）
- － 行列（Matrix）（ページ30）
- － グループ軸（Group Axis）（ページ31）
- － テキストのスカラ軸／行列軸／カーブ軸／グループ軸（ページ31）

スカラ／3Dキューブ／4Dキューブとカーブ／マップとの違いは、軸の数（スカラ：なし、3Dキューブ：X/Y/Z1、4Dキューブ：X/Y/Z1/Z2）のみです。このタイプのパラメータの作成方法は、オンラインヘルプを参照してください。

Specify Parameter Information ?

Parameter Name: myParameter

Parameter Type: Cube 4D

Value Bounds: Lower Bound: 0, Upper Bound: 1

Unit: -, Channel: -, Use Range and Unit

Input 1: -, -, Use Range and Unit

Input 2: -, -, Use Range and Unit

Input 3: -, -, Use Range and Unit

Input 4: -, -, Use Range and Unit

Output: -

	Begin	End	Count
Breakpoints X	0	1	10
Breakpoints Y	0	1	10
Breakpoints Z1	0	1	4
Breakpoints Z2	0	1	3

Extrapolation: Clip

Depend on Formula

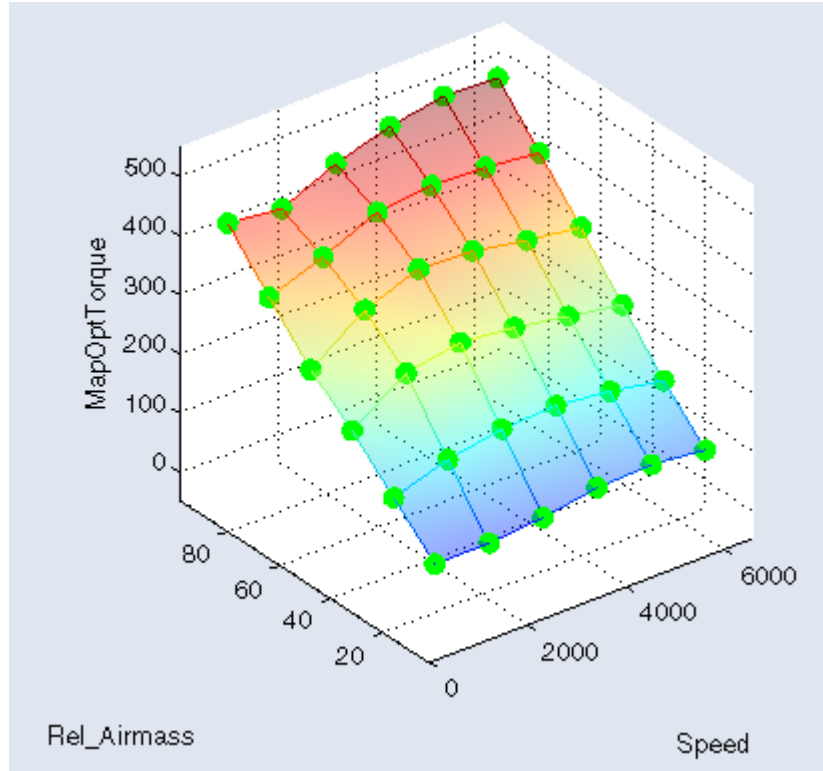
OK Cancel Edit...

マップ (Map)

マップは、X軸とY軸で示される2次元のグリッドと、各グリッドに対応するZ値（出力値）が定義されたものです。

各グリッドの間においては双線形補間によりZ値が算出されます。したがって、関数依存性は $z = z(x, y)$ となります。マップは、2次元のルックアップテーブルとして保存されます。

グリッドの外側においては、クリッピング（最後の値を使用）または線形補間が適用されます。



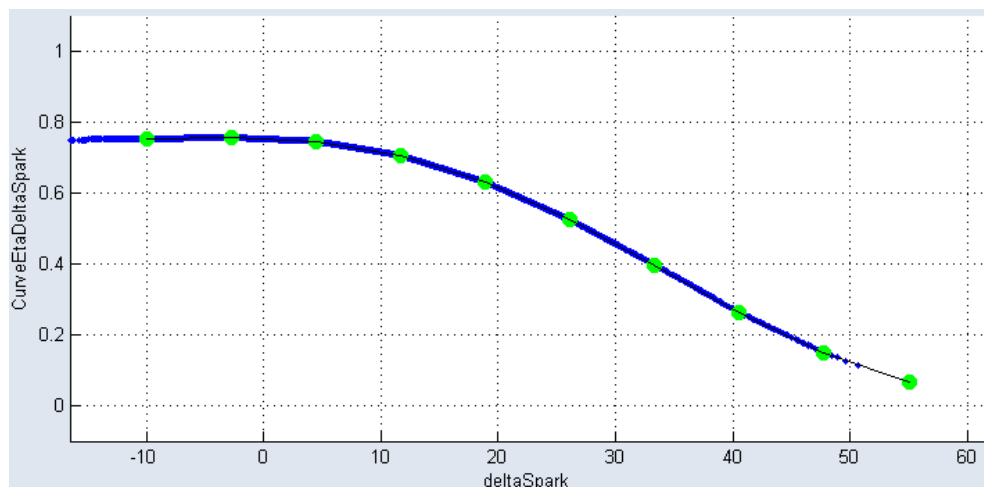
マップパラメータについては、入力に応じた上下限值を設定することができます。これらは **Parameter <parameter_name>** ウィンドウ（詳細はオンラインヘルプを参照してください）で編集できます。

カーブ (Curve)

カーブは、X軸で示される1次元のグリッドと、各グリッドに対するY値（出力値）が定義されたものです。

各グリッドの間においては線形補間によりZ値が算出されます。したがって、関数従属性は $y = y(x)$ です。カーブは、1次元のルックアップテーブルとして保存されます。

グリッドの外側においては、クリッピング（最後の値を使用）または線形補間が適用されます（下図参照）。



カーブパラメータについては、入力に応じた上下限值を設定することができます。これらは **Parameter <parameter_name>** ウィンドウ（詳細はオンラインヘルプを参照してください）で編集できます。

スカラ (Scalar)

スカラは0次元の適合パラメータです。

3D/4Dキューブ (Cube-3D / Cube-4D)

カーブ（入力数：1）とマップ（入力数：2）に加え、ASCMO-MOCAは、入力数が3または4の以下のルックアップテーブルをサポートしています。3Dキューブ、4Dキューブ

圧縮モデル (Compressed Model)

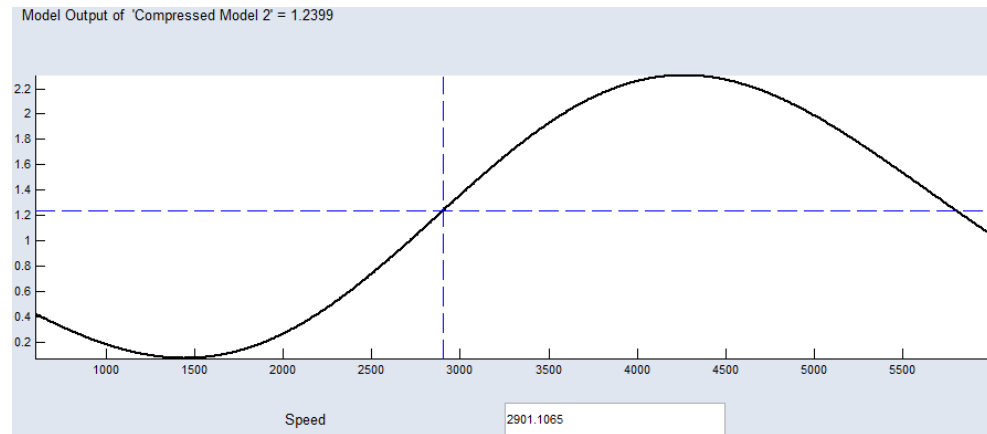
ASCMO-MOCAは、ルックアップテーブル（カーブ、マップ、キューブ）に加え、squared exponential kernel をパラメータとして使用する放射基底関数ネットワーク（RBF Net-SE）をサポートしています。

パラメータの入力数はユーザーが指定できます。基底関数（カーネル）の数もユーザーが選択します。入力や基底関数の数が多くなると、パラメータの最適化や評価のための演算の複雑度が增大します。

パラメータの評価関数は、ガウス関数の重ね合わせです。関数の複雑度の概算は、入力数に基底関数の数を掛けた数のe関数の評価となります。

これはブラックボックスデータに基づくモデルとしてみなすことができ、ASCMOでも圧縮モデル ("Compressed Model") として扱われています。これは、複数のルックアップテーブルとそれらの関連性からなる関数全体の代用となるものです。

基底関数（カーネル）の数が多くなると、モデルの忠実度は向上しますが、オーバーフィッティングを招く可能性があり、テストデータを用いた試験が必要になります。



行列 (Matrix)

ASCMO-MOCAは行列パラメータをサポートしています。行列は2次元で、インデックスされる一連の要素が含まれます。行列内の各スカラ値の位置は、インデックス値（非負の整数値）によって決定されます。

グループ軸 (Group Axis)

ASCMO-MOCAは、複数のパラメータ (カーブやマップ) が共有する「グループ軸」(「共有軸」などとも呼ばれます) をサポートしています。グループ軸は独立したパラメータタイプとして扱われません。これを使用することによりデータの一貫性を保つことができます。グループ軸は、Simulink® またはFMUパラメータのマッピングにおいて特に役立ちます。たとえば、Simulink®の複数の変数マッピングがSimulink®モデル内の同じ変数を指している場合や、計算によって求められる複数のパラメータがFMUモデル内の同じ値を参照しているような場合です。グループ軸は、DCMまたはCDFXファイルとしてエクスポート/インポートでき、モデルステップのスキャン機能や検証機能を使用して自動的に検出され作成されます。

グループ軸はファンクション内では使用できません。グループ軸は最適化することができます。グループ軸を使用することによって外部モデルとの通信が高速化されます。

テキストのスカラ軸/行列軸/カーブ軸/グループ軸

注記

テキストパラメータは、DCMとA2Lファイルからインポートすることができますが、作成や編集は行えません。これらをインポートするには、**Import** をクリックしてDCMファイルを選択し、**Open** をクリックします。続いて、**Load A2L File** をクリックしてA2Lファイルを選択します。

テキストスカラ : テキストスカラの場合、**Enumeration** 列のドロップダウンでラベルを選択することができます。このラベルは、テキスト部分と、それに対応する制御ユニット内の実際の値から構成されます。

テキスト行列 : テキスト行列はテキストスカラと似ていますが、ドロップダウンを持つ複数のセルが含まれ、ラベルを選択することができます。これらのラベルは、テキストと、それに対応する制御ユニット内の実際の値から構成されます。

テキストカーブ : テキストカーブの場合は、**Label** 列がX軸を表し、テキスト部分とそれに対応する制御ユニット内の実際の値を含んでいます。**Value** 列で、対応するY値を見つけることができます。値を編集するには、そのセルをダブルクリックします。

テキストグループ軸 : 複数のカーブを含むグループ軸で、X軸がテキストとして **Label** 列に表示されます。対応する制御ユニット内の実際の値は、**Index** 列に表示されます。**#** 列は連続カウントを表します。

参照

4.3.3 システム定数

システム定数は、パラメータにデフォルト値を提供ために用いられます。タイプの異なる複数のパラメータを1つのシステム定数に割り当てることができ、各パラメータに共通のデフォルト値を与えることができます。システム定数に非スカラ値を割り当てると、各ポイントに同じ定数値が使用されません。

システム定数をアクティブにすると、そのシステム定数に割り当てられたすべてのパラメータに同じデフォルト値が与えられます。

システム定数の作成と管理は、パラメータステップの "System Constant" タブで行います。

システム定数の作成方法については、オンラインヘルプを参照してください。

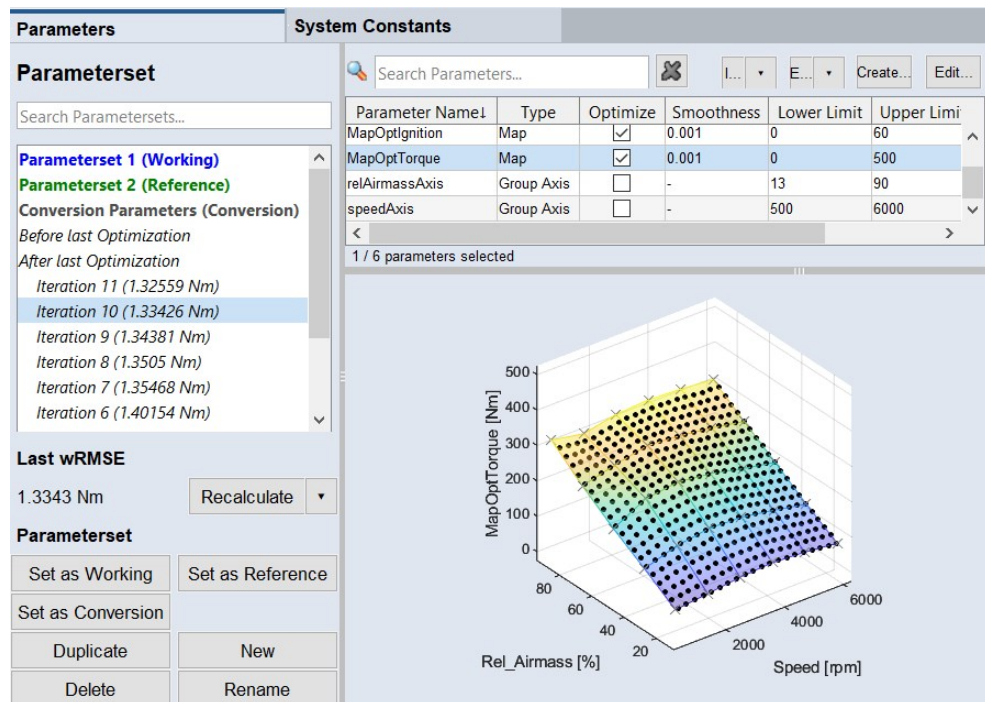
4.3.4 パラメータセット

ASCMO-MOCAでは、複数のパラメータセットを管理して使用することができます。1組のパラメータセットは必ずワーキングパラメータセットとリファレンスパラメータセットとして定義されます。Optimization Stepにおいてはワーキングパラメータセットに対して最適化が行われます。パラメータセットごとに異なるパラメータを持つことができ、同じ名前のパラメータが異なる軸ポイントベクトル (support vector) を持つことも可能です。

最適化処理においては、ワーキング/リファレンスパラメータセットに加え、以下のパラメータセットが作成されます。

- － Before last Optimization (一般的には空の状態、最適化の前にデフォルトのリファレンスデータセットとして作成されます)
- － After Last Optimization (最適化後のパラメータセット)
- － Iteration 1, Iteration 2, ... (反復のたびに作成されるパラメータセット)

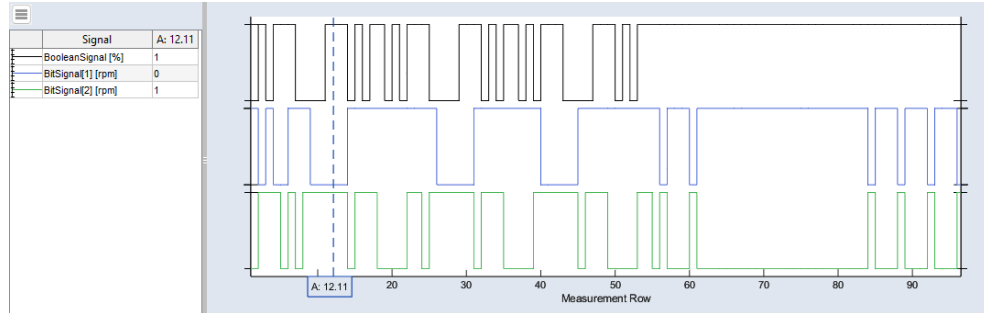
最適化の1回の反復は、自動的に1つのパラメータセットとして保存されます。これにより、最適化の進捗状況を分析したり、前のセットに戻ったりすることができます。



4.4 可視化

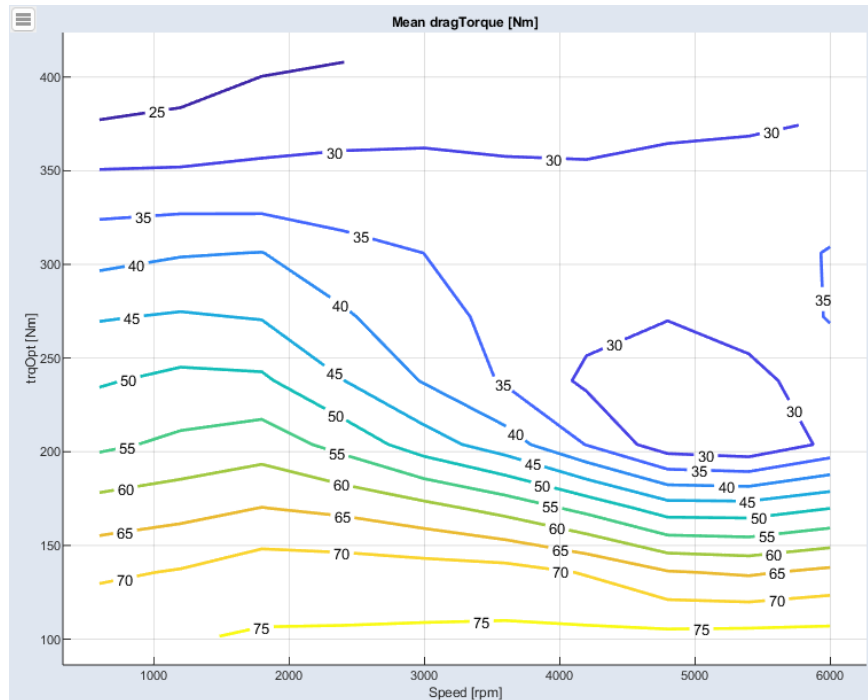
ASCMO-MOCAには、データステップから最適化ステップまでの実行ステップに加え、結果を一元的に可視化できる可視化ステップが用意されています。ここでは、以下のようなタイプのエレメントを組み合わせることで信号やパラメータ (マップ、カーブなど) を自由に表示することができます。可視化ステップは、データの比較やパラメータの適合に利用することができます。

Bit Plot (ビットプロット) : ビットプロットが表示され、複数のビット信号を上下に並べてスコープ表示することができます。



Data Contour Plot（データ等高線プロット）：選択されているデータの平均値、最小値、最大値、中央値、データ密度が、等しい値の点を結んだ線で表示されます。

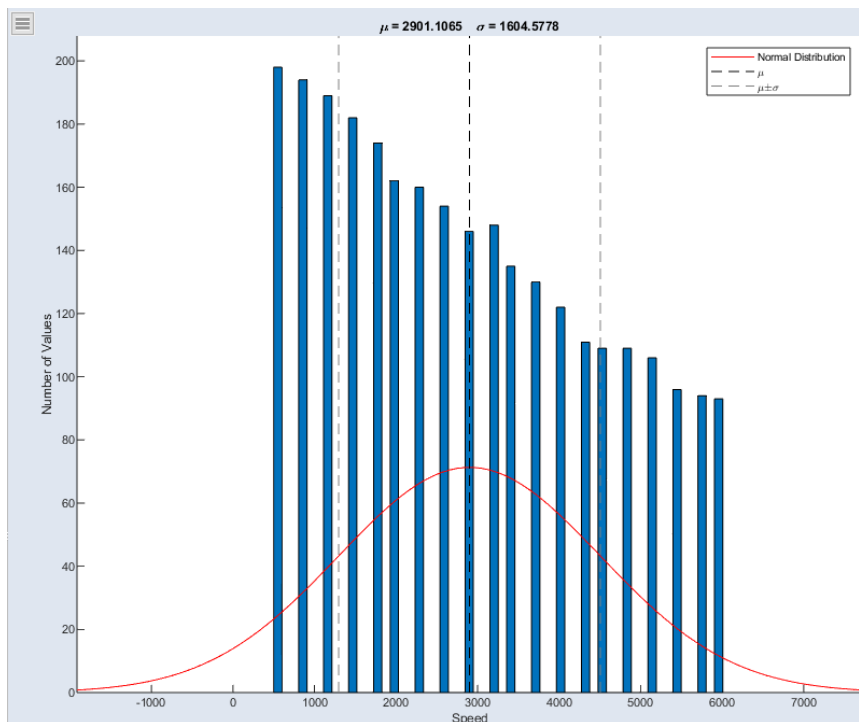
各データポイントは重複するビン／区間に割り当てられ、区間ごとのデータポイントについて指標（平均、最小値、最大値）が計算されます。計算された値は特定のグリッド値に割り当てられます。二次元のグリッド値から内部的にマップが計算され、等高線で可視化されます。



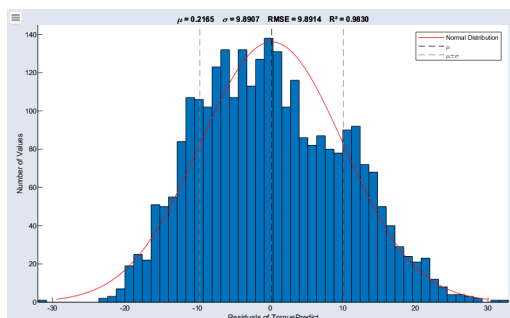
Data Table（データテーブル）：各プロット上で選択された値が、対応する色でハイライト表示されます。

	Speed	Rel_Airmass	Ignition	Torque_Meas	TorquePredict
1	597.000	47.8089	-26.8200	7.02201	10.7943
2	597.000	51.6611	-26.8200	19.5251	21.3053
3	597.000	40.1047	-22.2111	-3.67776	-0.166347
4	597.000	43.9568	-22.2111	8.14535	11.3093
5	597.000	47.8089	-22.2111	21.1618	22.9555
6	881.211	47.8089	-22.2111	1.08058	14.6535
7	597.000	51.6611	-22.2111	34.1432	34.6904
8	881.211	51.6611	-22.2111	12.9679	24.6004
9	597.000	55.5132	-22.2111	47.6720	44.8292
10	597.000	59.3653	-22.2111	62.1931	55.6803
11	597.000	32.4005	-17.6021	-14.1937	-15.2673
12	597.000	36.2526	-17.6021	-1.90174	-2.74430
13	881.211	36.2526	-17.6021	-18.1929	-6.92899
14	597.000	40.1047	-17.6021	10.9310	9.98072
15	881.211	40.1047	-17.6021	-5.53035	4.35335
16	597.000	43.9568	-17.6021	23.8206	22.9078
17	881.211	43.9568	-17.6021	6.26536	15.5937
18	597.000	47.8089	-17.6021	38.2851	36.0368
19	881.211	47.8089	-17.6021	19.5687	26.7921
20	1165.42	47.8089	-17.6021	0.470531	18.0611

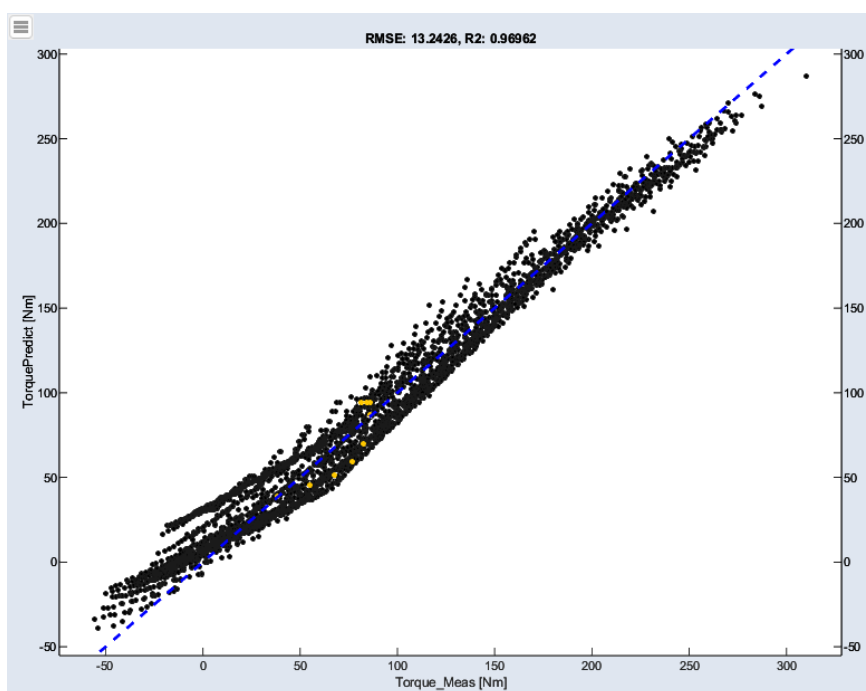
Histogram (ヒストグラム) : 複数のビンに割り当てられた値が、バーとして表示されます。



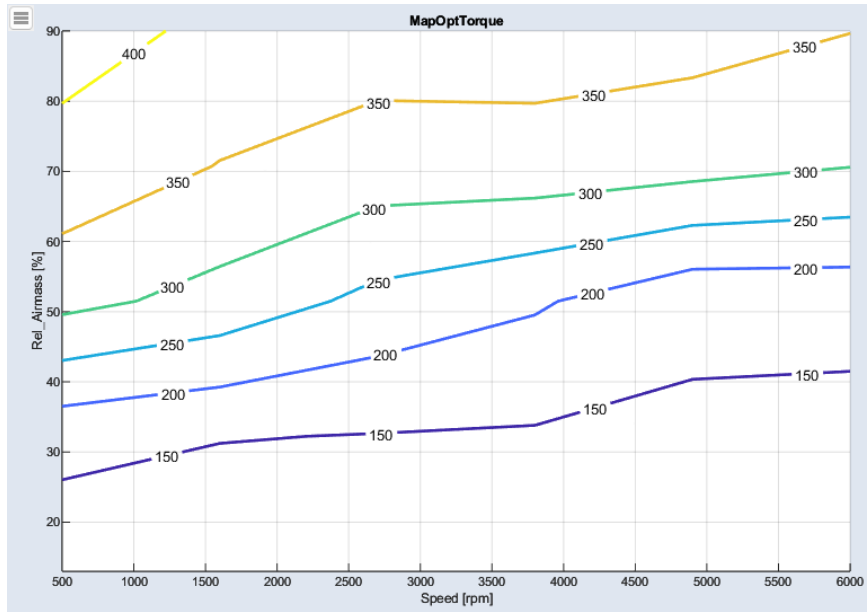
Residual Histogram (残差ヒストグラム) : 残差 (観測値とフィッティングされた値との差) の分布が、複数のビンにまとめて表示されます。



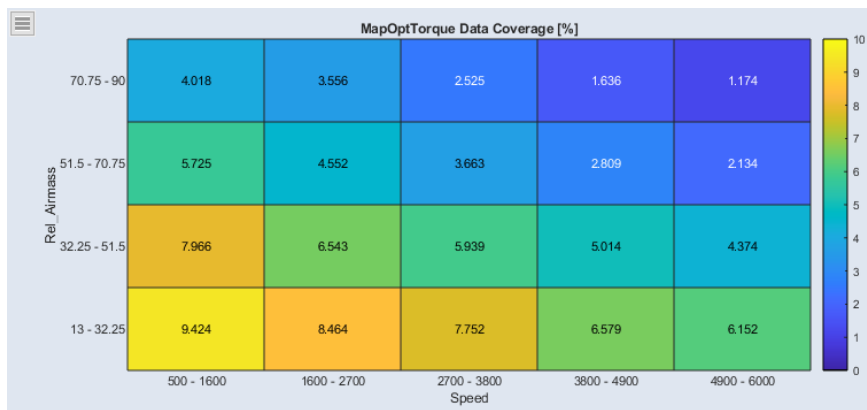
Measured vs. Predicted : 測定された信号値と予測された信号値が比較表示されます。プロットには、二等分線、RMSE、 R^2 が表示されます。



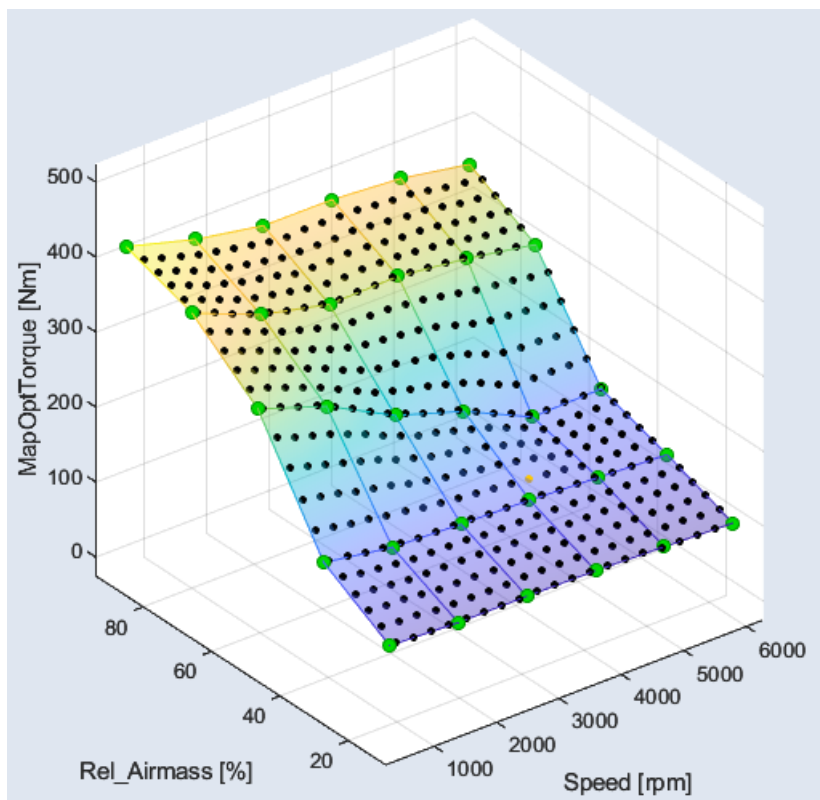
Parameter Contour Plot : マップとカーブが、2次元平面において等しい値の点を結んだ線が表示されます。



Parameter Heatmap : パラメータのデータカバレッジ



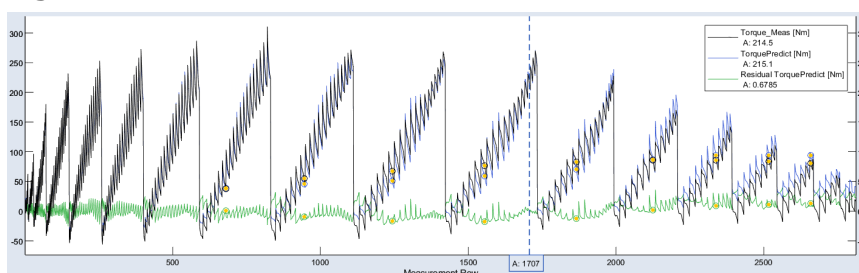
Parameter Plot : パラメータのプロット表示



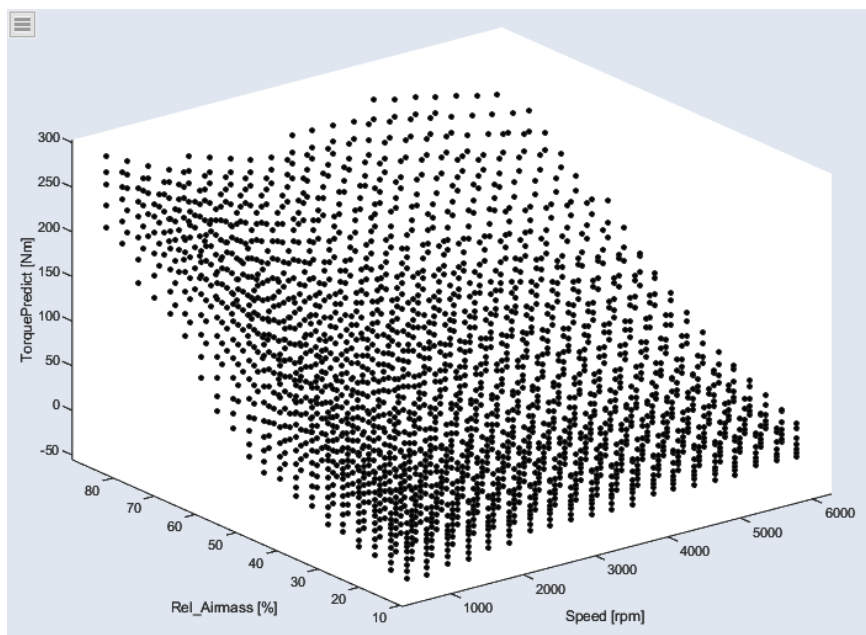
Parameter Table (パラメータテーブル) : パラメータのテーブル表示

MapDragTorque (Calibration)						
Y \ X	500	1600	2700	3800	4900	6000
13	77.0804	78.6306	78.7216	78.9457	79.6010	80.1531
32.25	57.2164	66.5855	68.2530	69.1787	71.0912	72.4950
51.5	34.8366	48.8347	39.5829	30.8083	15.6306	32.8890
70.75	22.7061	30.7942	35.3824	33.8130	35.0545	37.8163
90	16.5684	21.8265	25.3030	25.1882	26.5461	29.5359

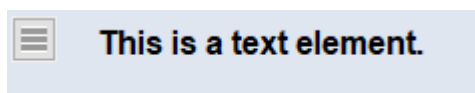
Signal Plot : 信号値の散布図とスコープ図



Signal Plot 3D : パラメータを3Dプロット表示



Text : 任意に入力されたテキスト

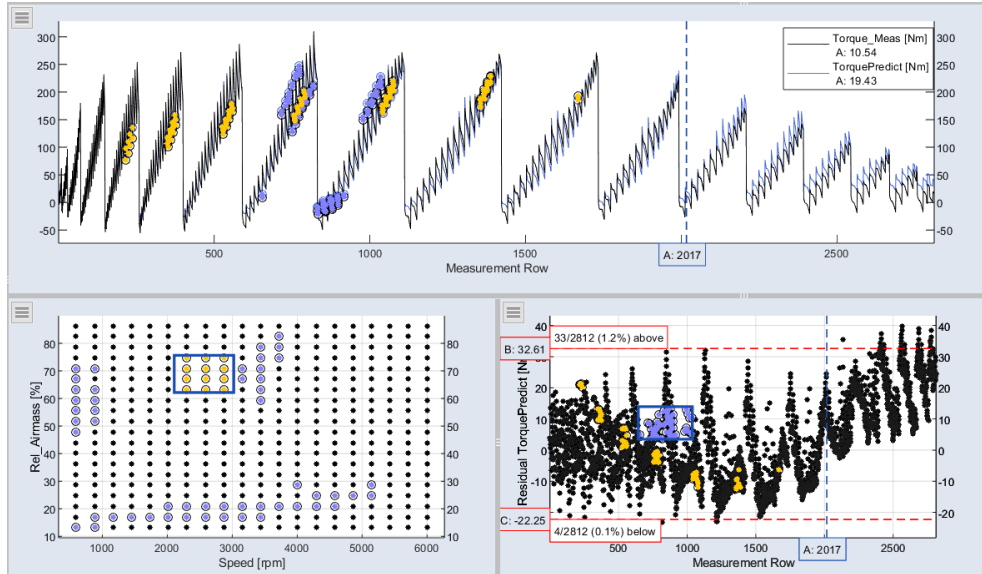


信号値（インポート値と計算値）を表示するエレメントにおいては、1つまたは複数のデータセットを選択して可視化することができます。さらに、データポイントのみを表示したり、ポイント間の線を表示したり、グリッドを背景に表示したりすることもできます。

パラメータを表示するタイプでは、パラメータ自体のほかに、基準値や境界値などを追加して選択することができます。パラメータプロットは、現在の適合値が表示されるだけでなく、その値を変更することもできます。これらの機能により、可視化ステップを実験環境のように利用することができます。変更された値の影響は、他のすべてのプロットに直接反映され、可視化されます。



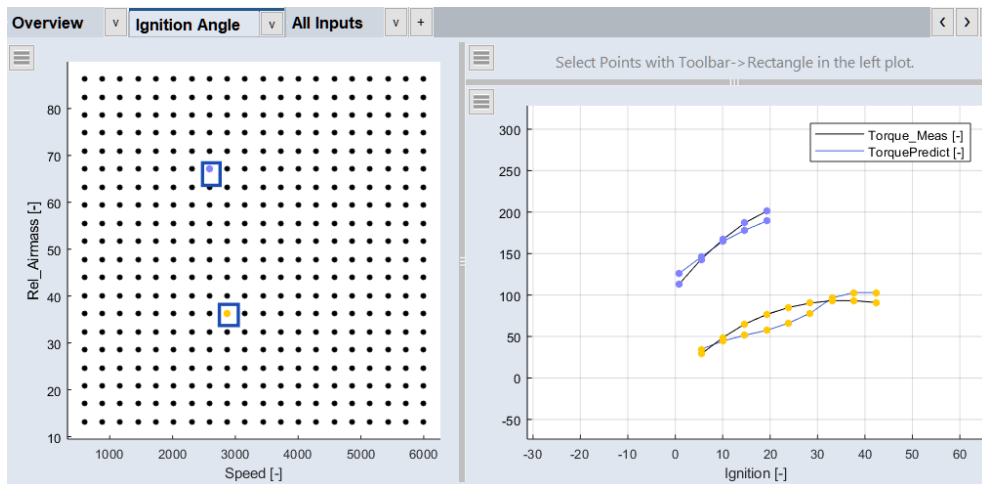
複数のビューを含むレイアウトを定義することができます。信号プロット内では囲み線（四角形と投げ縄）で範囲を選択し、そのデータをすべてのビューでハイライト表示することができます。カーソルには、カーソルが示す時間を表示することができます。カーソルを制限値として設定し、制限値外の値の数を表示することができます。選択範囲とカーソルは、プロジェクトに保存されます。



可視化ステップでは各タブごとに1つのビューを設定でき、ビューを分割して任意の要素を配置することができます。最下部のボタン ("View" と "Configuration") で、2つのモードを切り替えることができます。Viewモードでは、印刷や、画像のエクスポートやクリップボードへのコピーも可能です。プロット表示のほか、要素にテキストを挿入することもできます。各タブは、ドッキングを解除して画面上のどこにでも配置することができます。タブをダブルクリックすると、名前を変更することができます。

可視化ステップでは、データの比較や、データの概要が一目で把握できるような可視化を行うことができます。

Only Marked Data オプション (☰ > **Configure Single Element** > **Only Marked Data**) を使用すると、動作ポイントごとの予測値と点火角度を比較する際に、データの概要が把握しやすくなります。



4.5 モデル

ASCMO-MOCAでは、一連の式として定義されたモデルや、インポートされたモデル (ASCET、FMU、Simulink、ASCMO-STATIC、ASCMO-DYNAMICで作成されたもの) を使用することができます。これらのモデルはASCMO-MOCAプロジェクト内のファンクション

ノードとして使用できます。

外部モデルのインポートと接続は、モデルステップで行います。

－ ASCETモデル

ASCMO-MOCAでASCETモデルを使用するには、ASCETとASCET-PSLで *.dll ファイルを作成する必要があります。この *.dll ファイルをASCMO-MOCAプロジェクトに追加します。詳細はオンラインヘルプを参照してください。

ASCETモデルは、ASCMO-MOCAでは「ブラックボックス」として扱われます。モデルの内容は変更できず、インポート時にはASCETモデルやASCETへのリンクは作成されません。

－ FMUモデル

ASCMO-MOCAでFMUモデルを使用するには、FMIで *.fmu ファイルを作成する必要があります。この *.fmu ファイルをASCMO-MOCAプロジェクトに追加します。オンラインヘルプを参照してください。



注記

ASCMO-MOCAは、FMI 2.xまたはFM 3.xを使用したFMUモデルをサポートしています。FMI 1のFMUモデルはサポートしていません。

この操作では、FMUファイルの名前のみがASCMO-MOCAプロジェクトに追加されます。モデル自身を開くことはできません。最適化の際には、FMUモデルはブラックボックスとして扱われます。つまり、ASCMO-MOCAはモデルに対して入力を渡し、出力を受け取るのみです。モデルが出力を算出する方法はASCMO-MOCAからはわかりません。

Linux FMU（FMUにWin32またはWin64バイナリが含まれていないもの）の実行はサポートされていますが、それには適切なLinuxイメージが存在する必要があります。WSL2（Windows Subsystem Linux）がインストールされていれば、Linuxバイナリを含むFMUを実行することができます。仮想マシンには、ZeroC Iceとlibgompが必要です。

たとえば、Debianには以下のようにしてインストールできます。

```
sudo apt install libgomp1
```

```
sudo apt install libzeroc-ice3.7 libzeroc-ice zeroc-ice-compilers zeroc-ice-slice
```

インストールするIceのバージョンは3.7.6以降である必要があります。

－ Simulink®モデル

ASCMO-MOCAでSimulink®モデルを使用する方法は、チュートリアル[のステップ4：モデル（ページ100）](#)を参照してください。

－ ASCMO-STATIC／ASCMO-DYNAMICモデル

これらのモデルはブラックボックスとして使用されます。インポートされたモデルからASCMOプロジェクトへのリンクは失われ、モデルはASCMO-MOCAプロジェクトの一部になります。インポート時には、インポートする出力（一部またはすべて）を選択することができます。各出力は、それぞれ個別のモデルとして追加されます。オンラインヘルプのImporting ASCMO-STATIC/ASCMO-DYNAMIC Modelsの項を参照してください。

ー TSiMプラグイン

TSiMプラグインをASCMO-MOCAで使用するには、*.mexw64 ファイルが必要です。これはFMUに似たボッシュ独自のファイルフォーマットで、一般的に、コンパイルされた形式（DLL）でコントロールユニットの機能を表現します。

このフォーマットは、ASCMO-MOCAでのシミュレーションとパラメータ最適化に使用することができます。

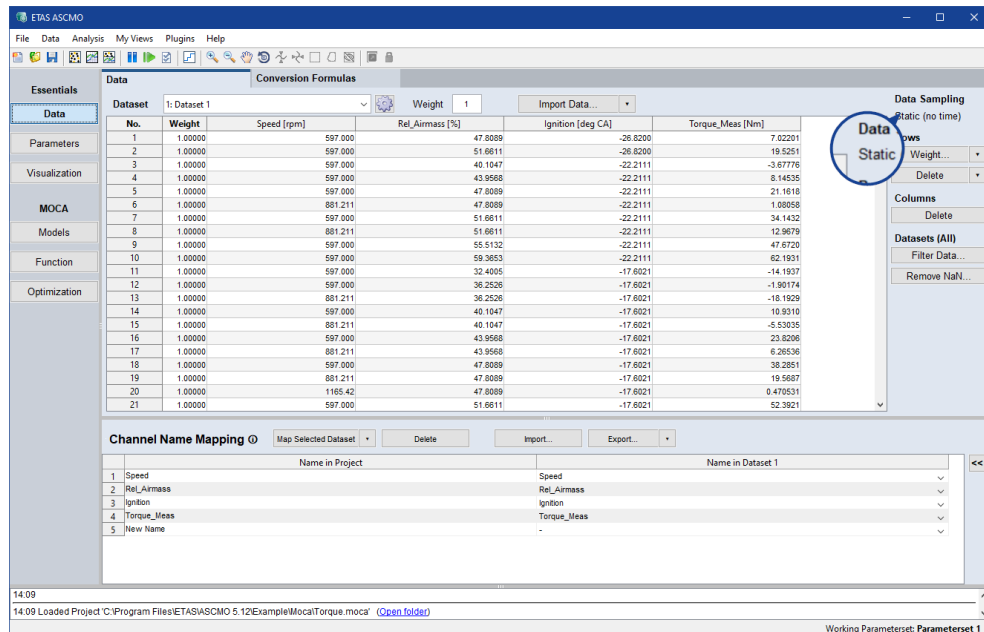
詳しい操作方法は、オンラインヘルプ（F1）を参照してください。

4.5.1 定常状態

定常状態（Steady State）は、ASCMO-MOCAのModelsステップにおいて、以下のモデルタイプを対象に使用されるコンセプトです。

- ー FMUモデル
- ー Simulinkモデル
- ー TSiMプラグイン

定常状態は、インポートされたデータが静的データであった場合にのみ適用されます。これは、データステップに **Data Sampling: Static (no time)** と表示されているので確認できます。



定常状態とは、関連するすべての変数が時間の経過に沿って相対的に一定であるか、または同じ割合で成長する状態（定常発展）であることを示すものです。つまり「均衡のとれた状態」を意味します。ここでは3つのサイズを指定することができます。

- ー **Simulation Step Size** : シミュレーションのステップサイズ（ベースサンプルタイム）を定義します。これはモデル内で指定されているものと一致させる必要があります。
- ー **Time until Steady State** : これは、定常状態に到達するまでのタイムスパン（ワーストケースシナリオ）です。

- **Average Last : Time until Steady State** フェーズの終了時における平均継続時間を定義します。平均値の算出に使用する直前の時間幅を秒単位で入力します。

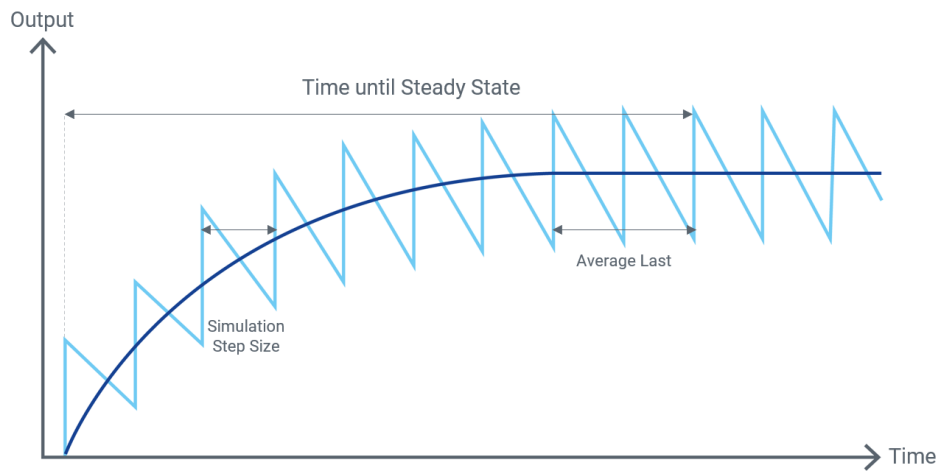


図4-6: 定常状態の可視化：暗い青の線が明るい青の線の平均値です

4.6 ファンクション

ASCMO-MOCAでは、一連の式として定義されたモデルやインポートされた各種外部モデル（ASCET、Simulink、ASCMO-STATIC、ASCMO-DYNAMICで生成されたモデル）をASCMO-MOCAプロジェクトに接続することができます。

一連の式で構成されるファンクションの定義は、"Function" ペインで行います。

ファンクションノードの式の定義には、データチャンネル、パラメータ、他のファンクションノード、インポートされたモデルを使用することができます。各種演算子が利用できます（[ファンクションノード用の演算子（下記）](#)を参照）。

ファンクションはテキストファイルにエクスポートしたりインポートしたりすることができます。テキストファイルの内容は、以下の例のような式の構文になります。

```
trqOpt[-] = %MapOptTorque%(%Speed%,%Rel_Airmass%)
ignOpt[-] = %MapOptIgnition%(%Speed%,%Rel_Airmass%)
deltaSpark[-] = %ignOpt% - %Ignition%
etaSpark[-] = %CurveEtaDeltaSpark%(%deltaSpark%)
product[-] = %SubFunction%(%deltaSpark%, %trqOpt%,
%CurveEtaDeltaSpark%)
dragTorque[-] = %MapDragTorque%(%Speed%,%Rel_Airmass%)
TorquePredict[-] = %product% - %dragTorque%

function SubFunction(InDeltaSpark : Data, InTrqOpt : Data,
myCurve : Curve)
curveOut[-] = %myCurve%(%InDeltaSpark%)
functionOut[-] = %curveOut% .*%InTrqOpt%
```

詳しい操作方法は、以降の各項、またはオンラインヘルプを参照してください。

- [ファンクションノード用の演算子（下記）](#)
- [ステップ5：ファンクションの構築（ページ110）](#)（チュートリアル）

4.6.1 ファンクションノード用の演算子

ノードの作成と編集は、Insert/Edit Node ウィンドウで行います。このウィンドウの右側には、一般的な演算記号を挿入するためのボタンが表示されます。



x	演算子 (*、/、+、-) を式に挿入します。 これにより、要素ごとの加減乗除演算が行われます。
/	
+	
-	
√	平方根演算子 <code>sqrt</code> (を式に挿入します。式の末尾には) が必要です。
x ^a	演算子 ^ を式に挿入します。 <code>x^y</code> は、行ごとに <code>x</code> を <code>y</code> 乗するものです。
abs	絶対値演算子 <code>abs</code> (を式に挿入します。式の末尾には) が必要です。 例 ¹⁾
bswitch	演算子 <code>bswitch</code> (を式に挿入します。式の末尾には) が必要です。 要素ごとのバイナリスイッチ： <code>y = bswitch (x, a, b)</code> <code>y = a for x <= 0</code> <code>y = b for x > 0</code> 例 ²⁾
multi switch	演算子 <code>multiswitch</code> (を式に挿入します。式の末尾には) が必要です。 複数の入力要素から選択するセクタです。 セクタは1ベースのインデックスを使用します。 <code>y = multiswitch (selector, x1, x2, x3,...)</code> <code>y[n] = x1[n] for selector[n] = 1</code> <code>y[n] = x2[n] for selector[n] = 2</code> <code>y[n] = x3[n] for selector[n] = 3</code> 例 ³⁾
()	開き括弧／閉じ括弧を式に挿入します。

1) `abs(-3) => 3`

2) `bswitch (%speed% > 2000, %Y_1%, %Y_2%)`

3)

```
y = multiswitch (selector, x1, x2)
```

```
Selector = [2, 1, 2]
```

```
x1=[10, 20, 30]
```

```
x2 = [100, 200, 300]
```

```
y = [100, 20, 300]
```

,	カンマを式に挿入します。
min	演算子 min (/ max (を式に挿入します。式の末尾には) が必要です。
max	min = 2つの入力の最小値 max = 2つの入力の最大値 例 ¹⁾
&	要素ごとのAND（論理積）を求める演算子 & を式に挿入します。 例 ²⁾
	要素ごとのOR（論理和）を求める演算子 を式に挿入します。 例 ³⁾
~	NOT（論理否定）演算子 ~ を式に挿入します。 例 ~(%x1% & %x2%)
cumsum	数値列の累積和を求める演算子 cumsum (を式に挿入します。式の末尾には) が必要です。 例 ⁴⁾
<	演算子 < , <= , == , >= , > を式に挿入します。
<=	
>	

- 1) `min(%in1%, %in2%), max(%in1%, %in2%)`
- 2) `%Speed% > 2000 & %Load% > 6`
- 3) `%Speed% > 2000 | %Load% > 6`
- 4) `cumsum(%y%): [1 2 4] ≥ [1 3 7]`

warnIf

演算子 **warnIf** (を式に挿入します。式の末尾には) が必要です。
最適化実行後に自動的に行われるチェックを定義することができます。定義された条件が満たされていると、指定の警告メッセージがログウィンドウに出力されます。

warnIf演算子を使用するには、ボタンをクリックした後、式のフィールドに条件を挿入します。warnIf演算子の構文は以下のとおりです：

```
y = warnIf(condition, 'warningText')
```

例

```
warnIf(%MapDragTorque%( %Speed%, %Rel_Airmass%) > 0, 'Warning')
```

上記の例では、最適化実行後にマップパラメータMapDragTorqueのいずれかの値がゼロより大きくなっていると、"Warning" というテキストがログウィンドウに出力されます。

```
warnIf(%speed% < 0, 'speed less than zero')
```

この場合は、最適化実行後に速度の値がゼロ未満になっていると、"speed less than zero" という警告が出力されます。

timeDelay

演算子 **timeDelay** (を式に挿入します。式の末尾には) が必要です。
これにより、信号を1タイムステップ分だけ遅延させることができます。ボタンをクリックした後、式のフィールドに以下の構文で条件を挿入します：

```
timeDelay(x, initialValue)
```

この式では、initialValueが最初のステップで返され、これがファンクション内の次のノードにアクセスするための唯一の方法となります。

例

```
timeDelay(%transferFcn%, %setpoint%(1))
```

この例では、transferFcnからの信号が1タイムステップ分遅延され、入力setpointの1番目の要素に初期値が設定されます。

```
y = timeDelay(%y%, 0.0)
```

ここでは **y** の値が1タイムステップ分遅延され、初期値が0.0になります。

dT

デルタT演算子 (サンプルングタイム、**dT**) を式に挿入します。

これはデータのサンプルングタイムを表し、実行時に実際の値に置き換わりません。

例

```
dT ./%filterConstant%
```

この例では、**dT** は **Data** ステップの **time [s]** 列からの値で、**%filterConstant%** は計算で使用されるパラメータです。

```
y = timeDelay(%y%, 0.0) * dT
```

この場合は、時間遅延を適用した後にyの値がサンプルングタイムdTで乗算されます。

roundTo

ノードを特定の離散値に丸める演算子 **roundToDiscreteValues** (を式に挿入します。式の末尾には) が必要です。

演算子を挿入した後、離散パラメータを選択して値を入力します。構文は以下のとおりです：

```
roundToDiscreteValues(node, [value1, value2, ...])
```

例

```
roundToDiscreteValues(%calMap_SCV(%speed%, %load%), [0, 1])
```

この例では、適合マップ **calMap_SCV** が入力 **speed** および **load** に基づいて離散値 0 または 1 に丸められます。

```
roundToDiscreteValues(%myTernary%, [0, 1, 2])
```

ここでは、**myTernary** の値が離散値 0、1、2 に丸められます。

steady States

ウィンドウの長さと高さに基づいて信号の定常状態を計算する演算子 **steadyState_abs** (を式に挿入します。式の末尾には) が必要です。構文は以下のとおりです：

```
steadyState_abs(x, windowLength, windowHeight, sampleRate)
```


例

```
steadyState_abs(%signal%, 10, 200, dT)
```

この例では、**signal** の定常状態が、ウィンドウの長さ 10、高さ 200、およびサンプリングレート **dT** を用いて計算されます。



式のフィールド内の最後のエントリを削除します (backspace) 。

以下の演算子がサポートされています。バーガーメニュー () から選択するか、または手入力します：

log (x) 自然対数 (底が e) 。
例¹⁾

log10 (x) 常用対数 (底が 10) 。
例²⁾

exp (x) オイラー数のべき乗 e^x 。
例³⁾

1) **log (exp (2))** => 2

2) **log10 (10^2)** => 2

3) **exp (1)** => 2.718

sin(x), 三角関数 – 入力はラジアン単位です。
cos(x), 例¹⁾
tan(x),
tanh(x),
atan(x)

atan2(x, y) 四象限逆正接関数 – 入力はラジアン単位です。
 例²⁾

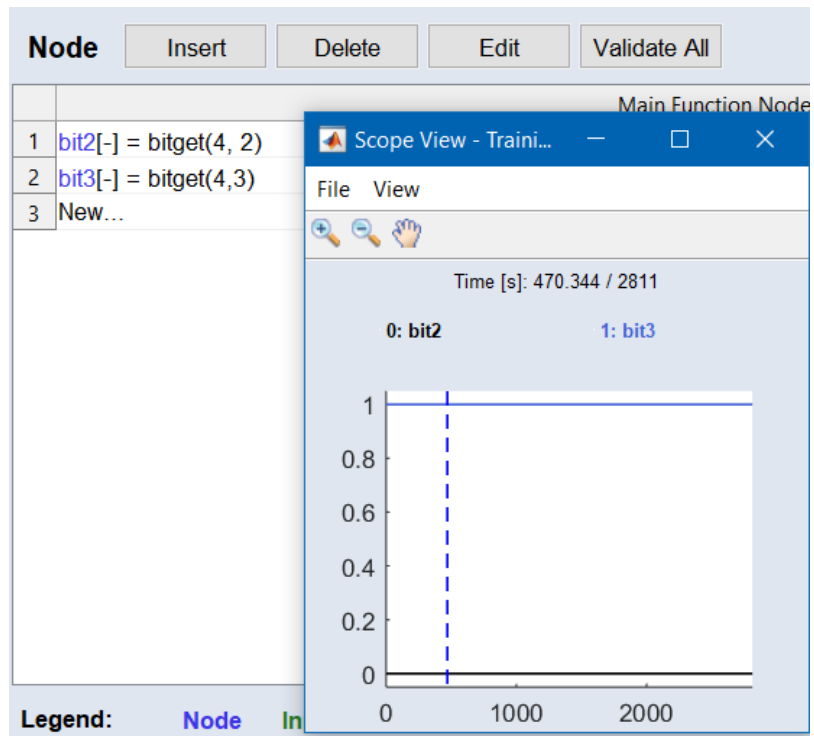
delayseq(data, n) 信号をnタイムステップ分遅らせます。信号の先頭部分にはゼロが充填されます。
 例³⁾

bitget(x, bitPos) MATLAB[®]関数 **bitget** は指定位置 (**bitPos**) のビット値を返します。
 例

bitget(4, 3) は1を返します。

bitget(4, 2) は0を返します。

この例では、整数4は2進数で100と表されます。関数**bitget(4, 3)**は位置3のビット、1を取得し、**bitget(4, 2)**は位置2のビット、0を取得します。



以下のような表記もサポートされています：

- 1) `sin(3.1416) ~> 0`
- 2) `atan2(%id%, %iq%)`
- 3) `delayseq([1; 2; 3; 4;], 2) => [0; 0; 1; 2]`

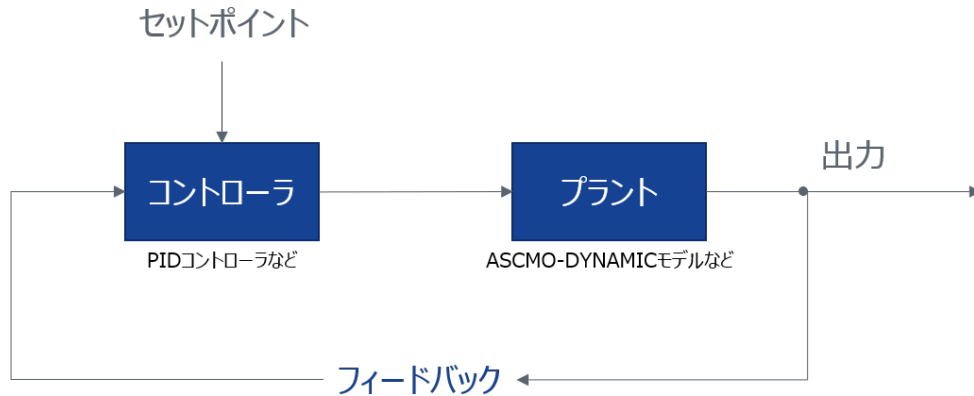
$\mathbf{z} = [\mathbf{x}, \mathbf{y}]$ 多次元ベクトルを作成します。1つのベクトルは、 $\mathbf{y} = \mathbf{z}(:, 2)$ で抽出できます。これは、たとえばサブ関数が複数のノードやベクトルを返す必要がある場合に役立ちます。

$\mathbf{y} = \mathbf{zeros}(\text{size}(\mathbf{x}))$ ゼロのベクトルを作成します。

$\mathbf{y} = \mathbf{ones}(\text{size}(\mathbf{x}))$ 1のベクトルを作成します。

4.6.2 フィードバックループ

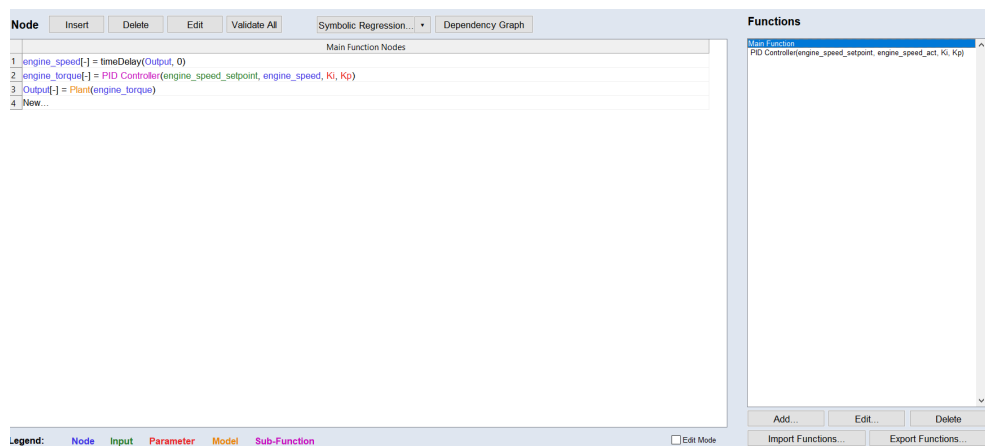
ファンクションノード内では、時間遅延を伴うフィードバックループを使用して、未来のノードにアクセスすることができます。これは、ダイナミックモデルでも使用できます。以下に、概念図とASCMO-MOCAでの設定例を示します。



メインのファンクションノード

```

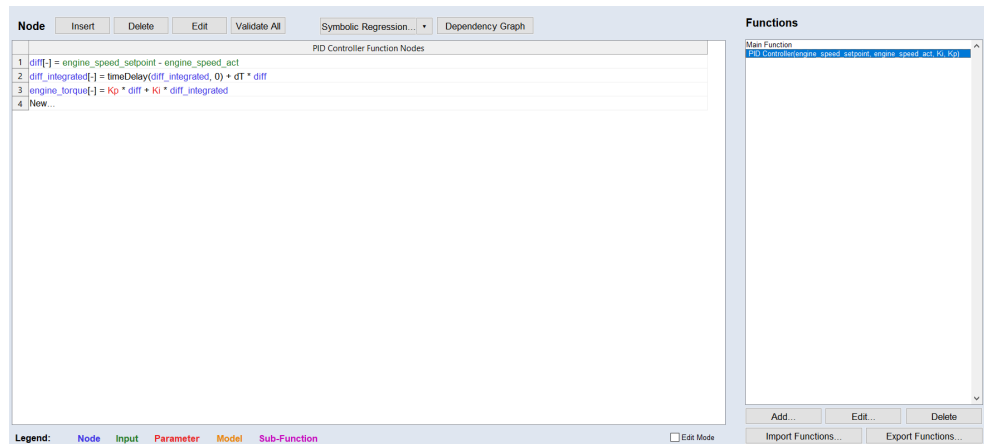
engine_speed[-] = timeDelay(%Output%, 0)
engine_torque[-] = %PID Controller%(engine_speed_setpoint%,
    %engine_speed%, %Ki%, %Kp%)
Output[-] = %Plant%(engine_torque%)
  
```



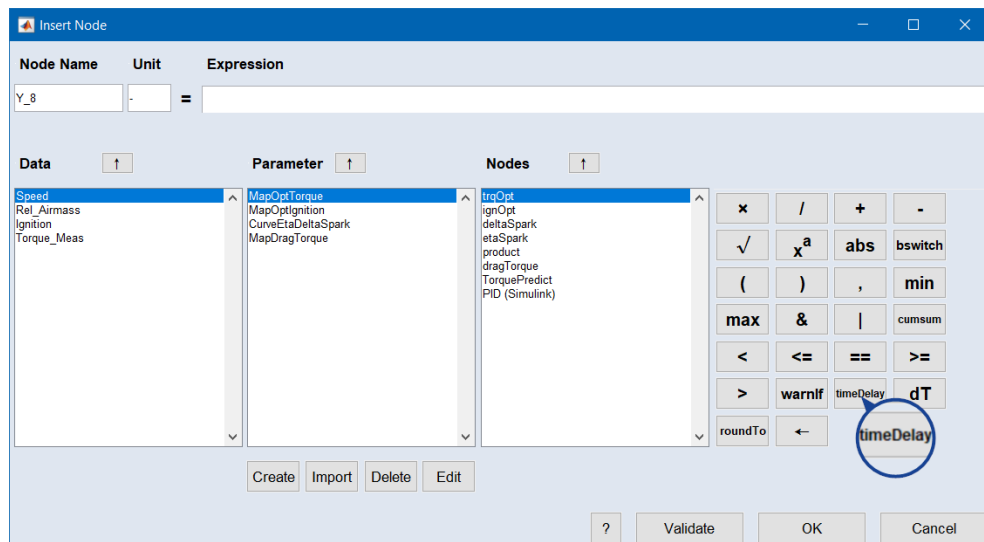
PIDコントローラのファンクションノード

```
diff[-] = %engine_speed_setpoint% - %engine_speed_act%
```

```
diff_integrated[-] = timeDelay(%diff_integrated%, 0) + dT
.*%diff%
engine_torque[-] = %Kp% .*%diff% + %Ki% .*%diff_integrated%
```



時間遅延エレメント (**timeDelay**) は、"Insert/Edit Node" ウィンドウで使用できる演算子のひとつです。



4.7 最適化

本項では、パラメータの最適化に使用できる各種最適化メソッドと最適化目標について説明します。

本項は以下の項で構成されています。

- 最適化手法について (次ページ)
- 粗度の考慮 (ページ62)
- 最適化目標 (ページ63)
- シーケンスを使用しない最適化 (ページ64)
- シーケンスを使用する最適化 (ページ64)
- パラメータの相関 (ページ64)
- パラメータの感度 (ページ65)

4.7.1 最適化手法について

最適化は、測定値と予測値（それぞれ n 個）の偏差を最小化することを目的として、マップ/カーブの適合値 p を適合します。

$$\operatorname{argmin}_p \sum_{i=1}^n \left(W_o * (Y_{i,predicted}(p) - Y_{i,measured}(p))^2 + \dots \right)$$

式4-5: 最適化手法

p	適合値
n	測定ポイントの数
$Y_{predicted}$	ASCMO-MOCA/ASCMO-MOCA Runtimeの予測値
$Y_{measured}$	インポート済みデータ
W_o	最適化の重み
W_c	制限の重み
W_g	勾配の重み (1~D次元)
W_k	平滑度の係数 (1~D次元)

絶対偏差は二乗することにより最小化され、偏差が大きいくほどペナルティが大きくなります。基本の式に、平滑度、ローカルな制限、勾配の限界を加えることができます。これは以下のような式で表されます。

平滑度

$$\operatorname{argmin}_p \sum_{i=1}^n \left(W_o * (Y_{i,predicted}(p) - Y_{i,measured}(p))^2 + \sum_{k=1}^D W_k * \text{Roughness} + \dots \right)$$

4.7.5 最適化目標 (ページ63) を参照してください。

勾配の限界

$$\operatorname{argmin}_p \sum_{i=1}^n \left(W_o * (Y_{i,predicted}(p) - Y_{i,measured}(p))^2 + \sum_{g=1}^D W_g * \text{Gradients} + \dots \right)$$

最適化目標 (ページ63) を参照してください。

ローカルな制限

$$\operatorname{argmin}_p \sum_{i=1}^n \left(W_o * (Y_{i,predicted}(p) - Y_{i,measured}(p))^2 + W_c * \text{Constraint} + \dots \right)$$

4.7.2 最適化アルゴリズム

Optimization (最適化) ステップでは、以下の最適化アルゴリズムを選択することができます:

- [Default \(Gradient Descent\) \(下記\)](#)
- [Respect Constraints \(Gradient Descent\) \(次ページ\)](#)
- [Gradient-free Optimizer \(次ページ\)](#)
- [Surrogate Optimizer \(Global Optimization\) \(次ページ\)](#)
- [Genetic Algorithm \(Global Optimization\) \(ページ54\)](#)
- [Simulated Annealing \(Global Optimization\) \(ページ54\)](#)
- [Particle Swarm \(Global Optimization\) \(ページ54\)](#)

Gradient Descent (勾配降下法) とGlobal Optimization (グローバル最適化)

勾配降下最適化は、初期値として設定されたワーキングパラメータセットで開始し、コスト関数の勾配を使用して、コスト関数の最小値を目指して最急降下方向へ反復的に向かいます。勾配降下最適化は局所最小値しか探索しないため、最適化の開始位置が重要になります。

ASCMO-MOCAは、関数の勾配を解析的に計算します。外部モデル (Simulink、FMUモデルなど) からの勾配は、有限差分法を用いて計算します。これにより、最適化は最小回数で関数評価によって素早く局所最小値を求めることができます。メモリ消費量は少なくとも、データポイント数にパラメータ数を乗じて、最適化アルゴリズム自体に2倍を乗じたものとなります。

グローバル最適化は勾配なし最適化で、グローバル最適解 (大域的最適解) の探索を試みるものです。最適化問題に対するいくつかのランダムな解の候補が探索空間全体に広がっている状態から開始します。その後探索空間は、より良い正確な解に絞り込まれていきます。最適化が完璧に最適解にヒットするとは限らないので、グローバル最適化から始めてグローバル最適解を見つけ、さらに続けて勾配降下最適化を行って結果を洗練させることができます。

ASCMO-MOCAにおける典型的な最適化問題は、マップとカーブの最適化です。このような最適化問題には通常多くのパラメータが含まれ (たとえば、20x20のマップには400のパラメータがあります)、グローバル最適化は良い解を見つけるために多くの反復を必要とすることがあります。

勾配なしのグローバル最適化は、関数/モデルの勾配が連続的でなくても解を見つけることができます。これは、モデルが数値やパラメータの固定小数点表現で実装され、入力や出力が離散的である場合に起こります。また、モデルが64ビット浮動小数点数ではなく32ビット浮動小数点数で実装されている場合にも起こります。

Default (Gradient Descent)

これは勾配降下最小二乗法による最適化です。これがデフォルトの最適化として選ばれたのは、最適化タスクに多くのパラメータが含まれる場合に優れた性能を発揮するため、マップとカーブを使用するような場合に適しています。

残差は最適化アルゴリズムにベクトルとして入力されるため、最適化は100点のデータセットに対して100個の残差を取得します。これは計算コストが高くなりますが、良い最適化結果につながります。

残差は最適化によって暗黙のうちに2乗されるので、参照値との差は常に最小化されます。最小化または最大化を行うには、最適化する低値と高値を明示的に指定する必要があります。

例

出力は0から1000の範囲です。最大化を行うには、最適化目標を**minarg(y(x)-1000)**と定義します。

ローカルな制限は、和の式の一部 ($W * Constraint$) として最適化に含まれます。ASCMOにおいては、これをソフト制限と呼びます。このようなローカルな制限の重みは、制限が満足されるまで10回の反復ごとに増加します。

$$\operatorname{argmin}_p \sum_{i=1}^n \left((Y_{i,predicted} - Y_{i,measured})^2 + W * Constraint \right)$$

最適化実行中に制限違反が生じると、制限の重みが増加します。これでは不十分であり、最適化後も制限違反が存在する可能性があります。

用途：勾配降下最適化は、MOCAが扱う典型的な問題に対する理想的なアルゴリズムです。大量のメモリを必要としますが、高速で良好な結果をもたらします。他の最適化アルゴリズムは、以下に示すように特定の状況でのみ検討するようにしてください。

Respect Constraints (Gradient Descent)

これも勾配降下最小二乗法による最適化ですが、制限が考慮されます。

最適化は、制限をハード限界値として扱います。制限違反しない限りは、デフォルトの最適化の方が好ましいでしょう。デフォルト最適化後に制限違反が認められた場合は、このアルゴリズムを使用します。

用途：この最適化アルゴリズムは、最適化実行後にローカル制限違反が発生した場合に役立ちます。まずデフォルトの最適化アルゴリズムを試行し、ローカル制限の重みを、たとえば、10、100、1000、...というように段階的に増加させていくことを推奨します。このアプローチに失敗した場合、本アルゴリズムが制限内で解を見つける可能性があります。

Gradient-free Optimizer

勾配を含まない最適化は、最適化にシプレックスアルゴリズムを使用します。このアルゴリズムは勾配に依存しないため、勾配降下アルゴリズムよりも多くの反復を必要とします。一般的に、勾配降下最適化ほど良い解は得られません。

この勾配なし最適化は、関数/モデルの勾配が連続的でない場合に使用します。これは、モデルが数値やパラメータの固定小数点表現で実装され、信号が離散的である場合に起こります。また、モデルが64ビット浮動小数点数ではなく32ビット浮動小数点数で実装されている場合にも起こります。

用途：この最適化アルゴリズムは、内部で固定小数点表現を使用するなど、パラメータの小さな変化に敏感でない外部モデルに適しています。外部モデルが倍精度ではなく単精度（32ビット）を使用している場合は、有限差分係数を20,000に増やし、デフォルトの最適化を使用することを検討してください。

Surrogate Optimizer (Global Optimization)

サロゲート最適化は、グローバル最適解の探索を試みます。最初にサロゲートモデルを構築し、元の関数/モデルの代わりにそれを最適化します。これは、関数/モデルの評価に時間がかかるような場合に便利です。

用途：この最適化アルゴリズムは、時間がかかる外部モデルの評価に役立ちます。最適化のために、迅速に評価可能な代替モデルが構築されます。

Genetic Algorithm (Global Optimization)

遺伝的アルゴリズムは、グローバル最適解の探索を試みます。これは、自然淘汰とゲノム交換にヒントを得たものです。ランダムな解の候補（ここでは「集団」と呼びます）から開始します。詳細は[Wikipedia:Genetic Algorithm](#)を参照してください。母集団のサイズがメモリ消費量に大きく影響します。多くの信号やデータを含む最適化タスクは、メモリ不足の問題を引き起こす可能性があります。母集団のサイズを小さくすることで、メモリを解放することができます。このアルゴリズムはすべての候補解をベクトル化するので、FMUモデルやTSimモデルのモデル評価を並行して実行することができます。

用途：この最適化アルゴリズムは、感度が低いモデルに適した勾配フリー最適化アルゴリズムです。主な目的は、特にデフォルトの最適化アルゴリズムがローカル最小値に陥る可能性がある場合に、グローバル最小値を見つけることです。このアルゴリズムはデフォルトの最適化アルゴリズムよりもメモリ使用量が少ない一方で、同様の結果を得るまでに時間がかかる場合があります。

Simulated Annealing (Global Optimization)

焼きなまし法は、グローバル最適解の探索を試みます。最適化はランダムな候補解から始めます。高い初期温度を使用するため、大きなパラメータ変更が可能です。何回かの反復により温度が下がるにつれてパラメータの変化が制限されるようになり、より正確な解を見つけることが可能になります。詳細は[Wikipedia:Simulated Annealing](#)を参照してください。粒子の数がメモリ消費量に大きく影響します。多くの信号やデータを含む最適化タスクは、メモリ不足の問題を引き起こす可能性があります。粒子の数を減らすことで、メモリを解放することができます。

用途：この最適化アルゴリズムは、感度が低いモデルに適した勾配フリー最適化アルゴリズムです。主な目的は、特にデフォルトの最適化アルゴリズムがローカル最小値に陥る可能性がある場合に、グローバル最小値を見つけることです。このアルゴリズムはデフォルトの最適化アルゴリズムよりもメモリ使用量が少ない一方で、同様の結果を得るまでに時間がかかる場合があります。

Particle Swarm (Global Optimization)

量子群最適化は、グローバル最適解の探索を試みます。最適化はランダムな候補解（ここでは「粒子」と呼びます）から開始します。各粒子は位置と速度を持ちます。最適化は高速で始まり、何度か繰り返すうちに速度が低下して、より正確な解を見つけることができます。詳細は[Wikipedia:Particle Swarm Optimization](#)を参照してください。多くの信号やデータを含む最適化タスクは、メモリ不足の問題を引き起こす可能性があります。粒子の数を減らすことで、メモリを解放することができます。このアルゴリズムはすべての候補解をベクトル化するので、FMUモデルやTSimモデルのモデル評価を並行して実行することができます。

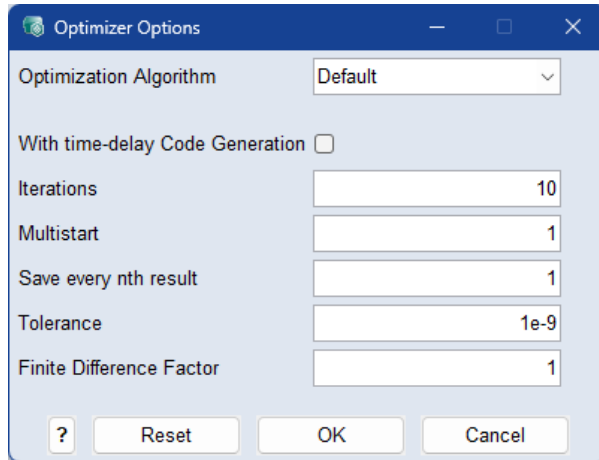
用途：この最適化アルゴリズムは、感度が低いモデルに適した勾配フリー最適化アルゴリズムです。主な目的は、特にデフォルトの最適化アルゴリズムがローカル最小値に陥る可能性がある場合に、グローバル最小値を見つけることです。このアルゴリズムはデフォルトの最適化アルゴリズムよりもメモリ使用量が少ない一方で、同様の結果を得るまでに時間がかかる場合があります。

4.7.3 オプティマイザのオプション

Optimization ステップ > Configure

Optimization Algorithm ドロップダウンリストで、設定をカスタマイズしたい最適化アルゴリズムを選択します：

Default



With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値で最適化を実行する回数を入力します。

1より大きい値を設定すると、最適化は複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

Tolerance

最適化の終了条件です。一般的な値の範囲は1e-9～1e-16です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- － 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- － コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Finite Difference Factor

外部モデル（FMUなど）の勾配が有限勾配によって計算されます。

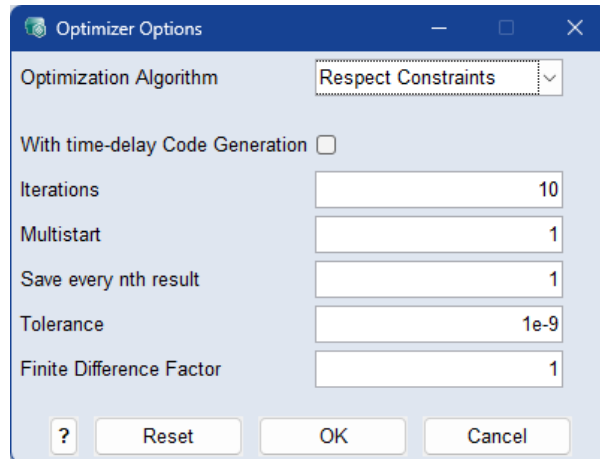
$$\text{Gradient} = (\mathbf{f}(\mathbf{x}+\mathbf{e}) - \mathbf{f}(\mathbf{x})) / \mathbf{e}$$

($e = \text{Finite Difference Factor} * \text{Normalized Parameter Range} * 1.5e-8$)

モデルの出力が上記のような小さな値で変化しない場合、最適化は最初の反復で停止します。その場合は、**Finite Difference Factor** を大きくする必要があります。典型的な値は 10, 100, 1000, 10,000,... です。

1.5e-8 は、倍精度浮動小数点数で表示できる最小数の平方根です。モデルが単精度浮動小数点数を使用している場合は、精度の低下を考慮して係数を10,000に設定してください。

Respect Constraints



With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値で最適化を実行する回数を入力します。

1より大きい値を設定すると、最適化は複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

Tolerance

最適化の終了条件です。一般的な値の範囲は1e-9~1e-16です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- － 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- － コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Finite Difference Factor

外部モデル（FMUなど）の勾配が有限勾配によって計算されます。

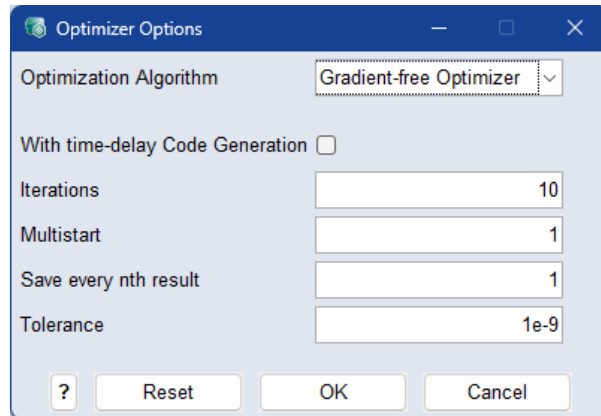
$$\text{Gradient} = (\mathbf{f}(\mathbf{x}+\mathbf{e}) - \mathbf{f}(\mathbf{x})) / \mathbf{e}$$

($\mathbf{e} = \text{Finite Difference Factor} * \text{Normalized Parameter Range} * 1.5\text{e-}8$)

モデルの出力が上記のような小さな値で変化しない場合、オプティマイザは最初の反復で停止します。その場合は、**Finite Difference Factor** を大きくする必要があります。典型的な値は 10, 100, 1000, 10,000,... です。

1.5e-8 は、倍精度浮動小数点数で表示できる最小数の平方根です。モデルが単精度浮動小数点数を使用している場合は、精度の低下を考慮して係数を10,000に設定してください。

Gradient-free Optimizer



With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値でオプティマイザを実行する回数を入力します。

1より大きい値を設定すると、オプティマイザは複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

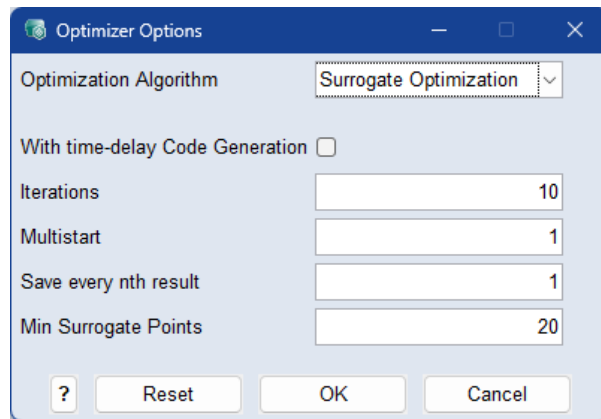
Tolerance

最適化の終了条件です。一般的な値の範囲は $1e-9$ ～ $1e-16$ です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- － 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- － コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Surrogate Optimization



With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値で最適化を実行する回数を入力します。

1より大きい値を設定すると、最適化は複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

Tolerance

最適化の終了条件です。一般的な値の範囲は $1e-9$ ～ $1e-16$ です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Min. Surrogate Points

サロゲートモデルは放射状基底関数モデルであり、モデルで使用される基底関数の最小数がここで定義されます。初期モデルは、これらの基底関数の位置において評価されます。

Genetic Algorithm

Parameter	Value
Optimization Algorithm	Genetic Algorithm
With time-delay Code Generation	<input type="checkbox"/>
Use Vectorized	<input checked="" type="checkbox"/>
Iterations	10
Multistart	1
Save every nth result	1
Tolerance	1e-9
Population Size	200
Elite Count	10
Crossover Fraction	0.8

With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Use Vectorized

オンにすると、MOCAファンクションは、1回のベクトル化された呼び出しにおいてすべての候補解を使用して評価されます。このファンクションのメモリ使用量は、評価処理中に母集団の個体数に比例して増加します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値でオプティマイザを実行する回数を入力します。

1より大きい値を設定すると、オプティマイザは複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Tolerance

最適化の終了条件です。一般的な値の範囲は1e-9～1e-16です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- － 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- － コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Population Size

最適化処理中に利用できる候補解の数です。母集団サイズを大きくすると、より良い解が得られますが、処理時間が長くなり、メモリ使用量も増えます。この値の一般的な範囲は100～1000です。

Elite Count

変更せずに次の反復に持ち越される良い候補解の数。この数を小さくすると、アルゴリズムの収束速度が遅ります。一般的な値は、全体の母集団サイズの5～20%です。

Crossover Fraction

交叉候補解は、2つの候補を組み合わせることによって生成されます。候補をランダムに変更することで突然変異候補解が作成されますこの交叉率（範囲は0～1）が交叉と突然変異の量を制御します。値を0.8にすると、候補の20%が突然変異を起こし、80%が交叉によって生成されます。一般的な値の範囲は0.5～0.9です。

Simulated Annealing

Parameter	Value
Optimization Algorithm	Simulated Annealing
With time-delay Code Generation	<input type="checkbox"/>
Iterations	10
Multistart	1
Save every nth result	1
Tolerance	1e-9
Initial Temperature	100
Reannealing Interval	100

With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値でオプティマイザを実行する回数を入力します。

1より大きい値を設定すると、最適化は複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

Tolerance

最適化の終了条件です。一般的な値の範囲は $1e-9$ ～ $1e-16$ です。

この値は、以下のようなケースで最適化を終了させる際に使用されます：

- － 最適化アルゴリズムによるパラメータの変化がこの値よりも小さくなる。
- － コスト関数の変化がこの値よりも小さくなる。最適化が収束した。

Initial Temperature

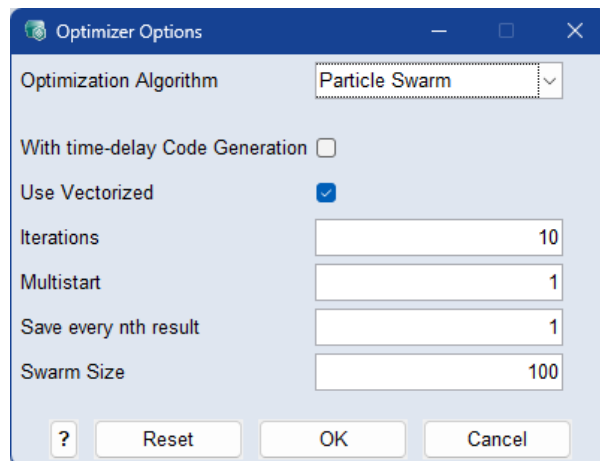
この値は、最適化処理において、より悪い解を受け入れる確率を制御します。一般的な値の範囲は50～200です。

初期温度は反復ごとに **Initial Temperature * $0.95^{\text{Iterations}}$** だけ減少します。温度が下がるにつれて、より悪い解を受け入れる確率が下がります。

Reannealing Interval

この回数の反復を行った後、温度が上昇します。一般的な値の範囲は50～200です。

Particle Swarm



With time-delay Code Generation

オンにすると、ファンクションからCコードが生成され、コンパイルされたCコードに対して最適化が行われます。これはファンクションが外部モデルを使用しない場合にのみ機能します。ファンクション内で **timeDelay** メソッドが使用されている場合は最適化が高速化され、それ以外の場合は一般的に低速化します。

Iterations

最適化実行中に実施する最大の反復回数を入力します。

Multistart

異なる初期値で最適化を実行する回数を入力します。

1より大きい値を設定すると、最適化は複数回実行されます。1回目の最適化は現在のワーキングパラメータセットで開始され、それ以降の最適化はランダムなパラメータ値で開始されます。これはグローバル最適解の探索に使用されます。

Save every nth result

n回目ごとの反復のみを一時パラメータセットとして保存します。一般的な値の範囲は[1, 100]で、デフォルトは1です。

Swarm Size

最適化処理中に利用できる候補解の数です。群サイズを大きくすると最適化時間が長くなり、メモリ使用量も増えますが、より良い解が得られる可能性があります。一般的な値の範囲は50~200です。

各最適化アルゴリズムについての説明は、[最適化アルゴリズム \(ページ51\)](#) を参照してください。

4.7.4 粗度の考慮**カーブの粗度 (Roughness)**

粗度 (Roughness) r は、カーブ c の各ブレイクポイントにおける勾配の変化を表します。カーブが式 $c(x)$ で与えられる場合、その粗度は各ブレイクポイント $x_i, i=1..k$ における二次導関数を合計することにより得られます。

つまり、カーブでは以下ようになります。

$$r_{curve} = \sum_{i=1}^k \left(\frac{d^2c}{dx^2} \Big|_{x_i} \right)^2$$

式4-6: カーブの粗度 r

マップの粗度 (Roughness)

マップ $m = m(x_1, x_2)$ の粗度については、以下のように第2の入力変数を考慮する必要があります。

$$r_{map} = \sum_{i=1}^k \left(\frac{d^2m}{dx_1^2} \Big|_{x_{1i}} + \frac{d^2m}{dx_2^2} \Big|_{x_{2i}} \right)^2$$

式4-7: マップの粗度

ここで、 K はマップ内のブレイクポイント $(x_{11}, x_{21}), \dots, (x_{1K}, x_{2K})$ の数です。

粗度は、"Parameter Optimization Properties" ウィンドウ (**Optimization Step > Optimization Criteria** ボタン) に表示されます。

4.7.5 最適化目標

1つまたは複数の出力を最適化するには、平滑度（**Smoothness**）という目標を使用します。これは、マップやカーブの粗度（**粗度の考慮（前ページ）**を参照）の変化を制限するものです。この係数は重み付きのペナルティ項です。

$$\operatorname{argmin}_p \left(\sum_{i=1}^N (Y_{i,predicted} - Y_{i,measured})^2 + \sum_{i=1}^M S_i * r_{curve/map} \right)$$

式4-8: 平滑化係数 S_i

ここで、 S は平滑化係数、 M はマップ／カーブ内のブレイクポイントの数です。

各軸の平滑化係数

マップ、3Dキューブ、4Dキューブにおいては、各入力方向に平滑化係数 S が使用されます。1つの値のみ与えられた場合は、係数は全方向に作用します。ベクトルが設定されている場合は、その各要素がそれぞれ1つの入力方向に適用されます。たとえば、マップのX方向の平滑度を0.1、Y方向を0.001にするには、配列 [0.1 0.001] を指定します。

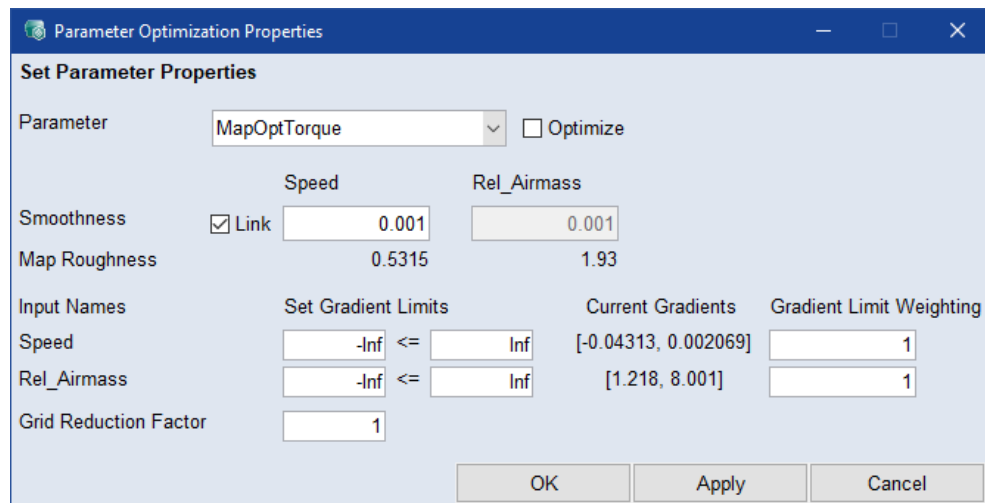
平滑化係数の値は0以上の実数です。この値を設定するには、最適化ステップで、

Optimization Criteria をクリックするか、パラメータテーブル内の "Smoothness" 列を直接編集します。

最適化目標の選択

カーブ、マップ、3Dキューブ、4Dキューブの各入力方向の勾配は、その上限と下限を定義することにより制限することができます。勾配制限を定義するには、最適化ステップの **Optimization Criteria** をクリックして以下のダイアログボックスを開きます。勾配制限は、オプティマイザにおいては「弱い」制限 ("weak constraint") として扱われます。

ここでは、勾配制限の各セットについて重みを定義することもできます。この重みは、1番目の最適化目標に対する各勾配制限の優先度を設定するものです。重みが大きいほど勾配制限の優先度が上がります。



最適化目標の設定手順は、オンラインヘルプに記載されています。

4.7.6 シーケンスを使用しない最適化

特別な要件が割り当てられたプロジェクトでない限り、最適化ステップはすべて "Optimization" ペインの "Optimize" タブで操作できます。

- 最適化の準備を行う
 - 最適化オプションを指定する
 - 最適化目標を指定する
 - ローカルな制限を指定する
- 最適化を実行する
- オプション処理を実行する
 - データを表示する
 - 参照パラメータを扱う

これらの操作手順については、オンラインヘルプを参照してください。

4.7.7 シーケンスを使用する最適化

最適化の手順について特殊な要件がある場合は、**Optimization** ステップの **Sequence** タブで、最適化ステップのシーケンスを定義することができます。

例：

- 最初に、使用可能なデータのサブセットを使用して1つのマップを適合する。
- その後、この最初のマップの結果を保持したまま、残りのパラメータを適合する。

このような場合は、シーケンスを作成して定義し、そのシーケンスを使用して最適化を実行します。

最適化シーケンスを作成して実行するための詳しい操作方法は、オンラインヘルプを参照してください。

4.7.8 パラメータの相関

メニューコマンド **Analysis > Parameter Correlation** で、パラメータ間の相関の有無を評価することができます。2つのパラメータに強い相関 (+1 または -1) がある場合、これは、2つのパラメータが互いに独立的にファンクションノードに影響することがない、ということの意味します。相関の判定は、以下のように行われます。

ASCMO-MOCAは、すべてのパラメータについて勾配行列 G を算出します。

$$G = \frac{\partial F(x, p)}{\partial p}$$

- F - 最小化する最適化関数
- x - トレーニングデータ
- p - パラメータ

続いてASCMO-MOCAは、共分散行列 C を算出します。

$$C = ((G^T * G) + I)^{-1}$$

- G^T - G の転置行列
- I - 単位行列

さらに、パラメータ a および b の相関係数 c を算出します。 C_{ab} 、 C_{aa} 、 C_{bb} が共分散行列の要素となります。

$$c_{ab} = \frac{C_{ab}}{\sqrt{C_{aa}} * \sqrt{C_{bb}}}$$

この結果は、"Parameter Correlation" ウィンドウに表示されます。

4.7.9 パラメータの感度

メニューコマンド **Analysis > Parameter Sensitivity** で、ファンクションノードに対するパラメータの影響度を評価することができます。

ASCMO-MOCAは、すべてのトレーニングデータポイント x_i のすべてのパラメータに対する、パラメータ p_j のノード勾配 G を算出します。

$$G = \frac{\partial F(x, p)}{\partial p}$$

- F - 最小化する最適化関数
- x - トレーニングデータ
- p - パラメータ

勾配は、パラメータの範囲に正規化されます。

$$g_{ij} = \frac{\partial F(x_i, p_j)}{\partial p_j} * (u_j - l_j)^{-1}$$

- u_j - パラメータ p_j の上限値
- l_j - パラメータ p_j の下限値

この結果は、"Parameter Sensitivity (Normalized)" ウィンドウに以下のように表示されます。

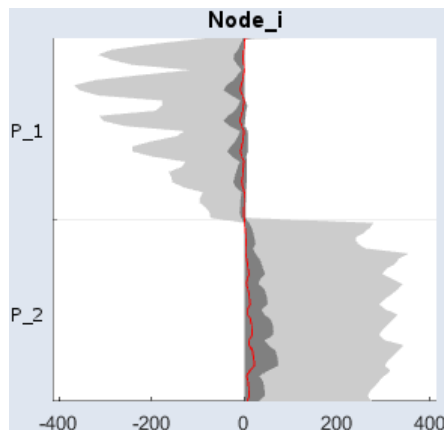


図4-7: 正規化されたパラメータ感度

- 濃灰色の領域：すべてのトレーニングデータポイントに対するパラメータ p_j の最大勾配
- 赤い線：すべてのトレーニングデータポイントに対するパラメータ p_j の平均勾配
- 淡灰色の領域：すべてのトレーニングデータポイントに対するパラメータ p_j の平均勾配 $\pm 1\sigma$

注記

感度の絶対値が小さいほど、パラメータがノードに与える影響が少なくなります。

4.8 関数同定問題（Symbolic Regression）

「関数同定問題」（symbolic regression - 「記号回帰」とも呼ばれます）は、記号レベルでの回帰分析の一種です。ASCMO-MOCAのコンテキストにおいてアプリケーションに転送されますが、これは、方程式ベースまたはハイブリッド（方程式ベースとデータベースのアプローチの混合）のモデルを自動的に見つけ出すことを意味します。このモデルには以下のような特徴があります。

- － データセットに対して統計的な意味でふさわしいものである
- － 要件に応じたコンパクトなサイズ
- － 人間が解釈できる

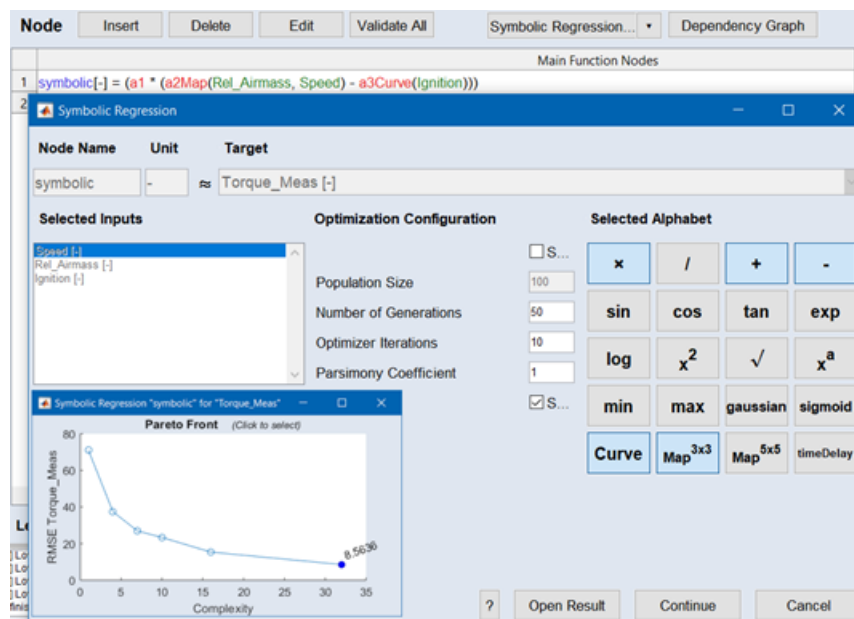
ASCMO-MOCAの関数同定問題プラグインは、方程式の構造レベルでの最適化とローカル最適化を実行することによって解を提供し、それによって特定のモデルをデータに適合させます。この2つのステップは、組み込みソフトウェアの機能エンジニアリングにおける「機能エンジニアリング」と「適合」に相当します。

ASCMO-MOCAは、人工知能を使用してエンジニアがこの手順をより効率的かつ効果的に実行できるようにサポートします。別の観点からこの手法は、システム同定のための自動化された手段としてとらえることもできます。

関数同定問題の機能はFunction Stepに含まれています。

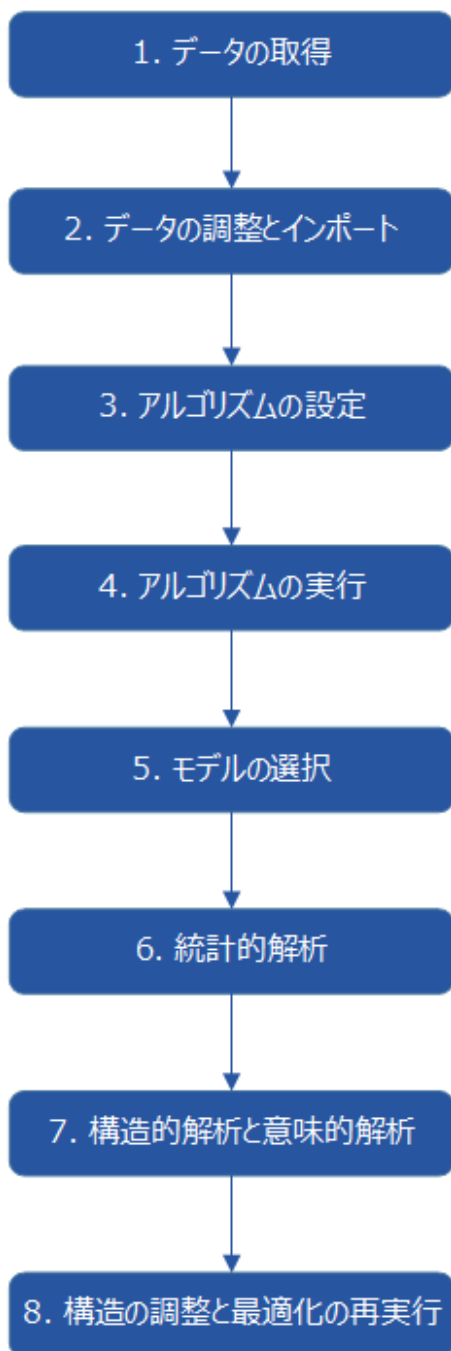
4.8.1 操作方法（関数同定問題- Symbolic Regression）

関数同定問題は、遺伝的アルゴリズムを使用してデータから数式を自動的に導出する作業を支援します。また、時間遅延のある時間依存の動的関数に対して関数同定問題を用いることもできます。



The screenshot displays the Symbolic Regression dialog box. At the top, there are tabs for 'Node', 'Insert', 'Delete', 'Edit', 'Validate All', 'Symbolic Regression...', and 'Dependency Graph'. The main function node is defined as: $\text{symbolic}[-] = (a1 * (\text{a2Map}(\text{Rel_Airmass}, \text{Speed}) - \text{a3Curve}(\text{Ignition})))$. The target is 'Torque_Meas [-]'. The 'Selected Inputs' list includes 'Speed [1]', 'Rel_Airmass [-]', and 'Ignition [-]'. The 'Optimization Configuration' section has 'Population Size' set to 100, 'Number of Generations' to 50, 'Optimizer Iterations' to 10, and 'Parsimony Coefficient' to 1. The 'Selected Alphabet' includes operators like \times , $/$, $+$, $-$, \sin , \cos , \tan , \exp , \log , x^2 , $\sqrt{\quad}$, x^a , \min , \max , gaussian , sigmoid , Curve , $\text{Map}^{3 \times 3}$, $\text{Map}^{5 \times 5}$, and timeDelay . A 'Pareto Front' plot is visible, showing 'RMSE Torque_Meas' on the y-axis (0 to 80) and 'Complexity' on the x-axis (0 to 35). A blue dot on the curve indicates the selected solution at complexity 32 and RMSE 8.5636.

関数同定問題（symbolic regression）を実行するワークフローは、以下のようないくつかのステップに分かれます。



ー **データの取得**：

システムを的確に表すデータを取得します。データセットのサイズとしては、トレーニングデータ、検証データ、テストデータに適切に分割できる量が必要です。

ー データの調整とインポート :

データには以下の条件が満たされている必要があります。

- 正しさ : すべての値が有意のものであり、NaN (非数) や、計測機器の不具合などに起因する不正な値が含まれないこと
 - ラベル : すべての値にラベル (測定変数名) と単位が定義されていること
 - 分割可能 : データセットをトレーニング用、検証用、テスト用に分割できること。すべてのデータが統計的な意味で代表的なものであること。
- 上記の要件に従ってデータを調整し、**インポート**を実行します。

ー アルゴリズムの設定 :

- Function Stepの **Symbolic Regression** ボタンをクリックして、アルゴリズム設定を開始します。
- **Target** フィールドに目標量を設定し、使用する入力量を **Available Inputs** フィールドから **Selected Inputs** リストに移動して、回帰問題を定義します。
- **Optimization Configuration** の各フィールドで、アルゴリズムの詳細を設定します。「[アルゴリズムの詳細](#)」を参照してください。
- **Selected Alphabet** 領域のボタンで、使用したい関数や演算子のタイプを選択します。

ー アルゴリズムの実行 :

"Symbolic Regression"ウィンドウの **OK** をクリックしてアルゴリズムの実行を開始します。ログウィンドウの下ステータスバー内の **Stop** をクリックすると、実行を停止することができます。ログウィンドウ内のコマンド行には、現在の反復ステップで見つかった最適なモデル用に選択されている **Fitness Method** の値が表示されます。

ー モデルの選択 :

アルゴリズムが完了すると、ASCMO-MOCAのウィンドウが開いてパレートフロントが表示されます。パレートフロントはパレート集合に含まれるモデルで構成されます。パレート集合はFitness Method (Y軸) とComplexity (X軸) の空間で定義されます。丸いシンボルをクリックしてモデルを選択します。選択されているフィットネスメソッドの値がその近くに表示されます。さらにASCMO-MOCAは、ファンクションステップで選択したモデルを提供します。

ー 統計的解析 : **Analysis > Residual Analysis** で、選択されているモデルのパフォーマンスを統計的に評価します。[残差分析](#)を参照してください。

ー 構造的解析と意味的解析 :

ファンクションステップとパラメータステップでは、それぞれ構造レベルと意味レベルでモデルを分析して解釈できます。ファンクションステップでは、モデル内部を確認することができます。パラメータステップでは、選択されているモデルで使用されるすべてのパラメータとマップを詳細に検査することが可能です。

ー 構造の調整と最適化の再実行 :

ASCMO-MOCAでは、[ファンクション](#)に説明されている内容に従い、モデル構造をシームレスに調整することができます。終了後は、この構造の再最適化を実行することができます。[最適化 \(ページ50\)](#) を参照してください。

4.8.2 アルゴリズムの詳細（関数同定問題 - Symbolic Regression）

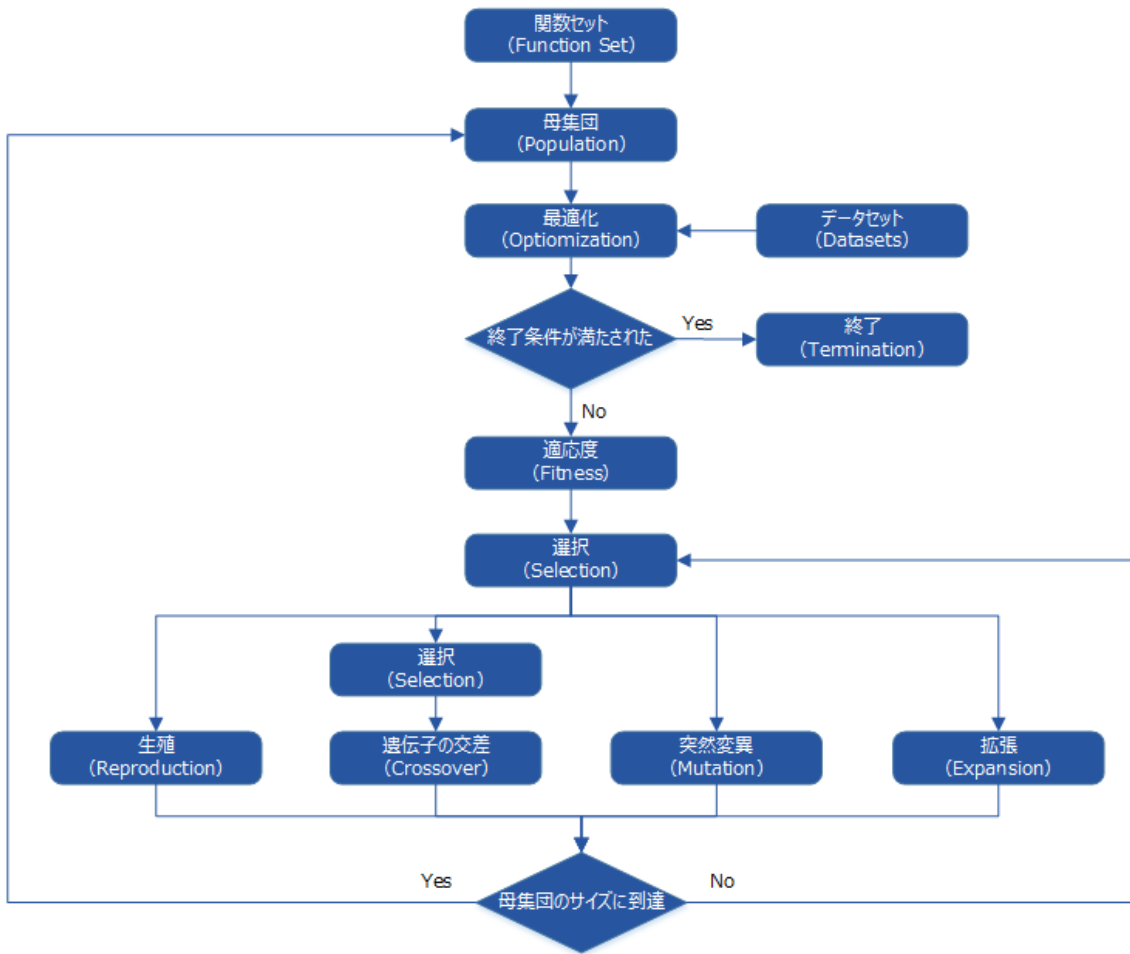


図4-8: アルゴリズムの詳細（関数同定問題 - Symbolic Regression）

関数同定問題を実行するにあたり、ASCMO-MOCAは遺伝子プログラミングの修正版を採用しています。このアルゴリズムは、要求されたタスクの実行に特に適しています。上図に示されるアルゴリズムの主要なステップについて、以下に解説します。

関数セット（Function Set）

アルゴリズムの過程で使用される関数式のセットを定義します。ASCMO-MOCAのこのステップでは、Selected Alphabet 領域のボタンで使用したい関数や演算子のタイプを選択します。

母集団（Population）

関数セットが定義された後に、モデルのセット（母集団）が確率論的に作成されます。このステップから、モデルはグラフで表されます。作成工程は、指定された母集団のサイズとグラフの規模によって制限されます。アルゴリズムの過程で、母集団は進化操作により進化します。アルゴリズムのメインループの1回の反復ステップが一世代に相当します。

"Symbolic Regression" ウィンドウ内の関連するコンフィギュレーションパラメータ

（**Optimization Configuration** / **Initial Configuration**）は、以下のとおりです。

- **Random Seed** : 乱数生成のシードとして使用される整数。アルゴリズム内のすべての確率論的ステップに影響します。この値を変更して、重要なケースのパフォーマンスを改善します。
- **Population Size** : 各母集団に含まれるモデルの数。難解な問題については、一般的に「数千」の規模が適切に機能します。
- **Population Creation Method (Initial Configuration)** : 初期の母集団は、3種類のメソッドのうちのいずれかを用いて作成されます。未知の問題については一般的に "half and half" が適しています。
 - **half and half** : 母集団の半分は "full" メソッドで作成され、残りの半分は "grow" メソッドで作成されます。採用される最大深度値は、各グラフごとに、**Population Init Depth Min** と **Population Init Depth Max** の範囲内の均一分布から決定されます。
 - **full** : 各グラフは、すべてのブランチが **Population Init Depth Max** の深度を持つように作成されます。
 - **grow** : 各グラフは、1つ以上のブランチが **Population Init Depth Max** の深度を持つように作成されます。
- **Population Init Depth Min** : 最小の初期深度。
- **Population Init Depth Max** : 最大の初期深度。深度10のグラフは非常に巨大で複雑です。4前後の数値が適切です。
- **Max Program Complexity (Optimization Configuration)** : 初期時と進化中において、この「最大複雑度」よりも複雑なグラフは作成されません。グラフの複雑度は、グラフを構成するすべてのノードの複雑度の合計である整数です。ノードの複雑度は、ASCMO-MOCA内部で経験的に定義されます。関数同定問題の複雑性について明確な要求と経験があれば、このオプションを使用してアルゴリズムの実行を高速化できます。
- **Max Program Depth** : 初期時と進化中において、この最大深度よりも深いグラフは作成されません。
- **Max Program Size** : 初期時と進化中において、この最大サイズ（最大ノード数）より大きなグラフは作成されません。

データセット (Datasets)

関数同定問題を実行するには1つ以上のデータセットが必要です。

最適化 (Optimization)

各グラフごとにローカル最適化が実行されます。この最適化の目的は、"Fitness Method" で指定されたメソッドを用いて得られる値を最小化することです。

関連するコンフィギュレーションパラメータ (Optimization Configuration)

- **Optimizer Method**
 - **Levenberg-Marquardt** メソッド (LM)
 - **Trust-region-reflective** メソッド
 - **Dogbox** アルゴリズム
- **Optimizer Iterations** : ローカル最適化の最大反復回数

終了条件 (Termination criteria fulfilled Request)

以下の条件が1つ以上満たされると、アルゴリズムは終了します。

- － 母集団のモデルの1つについて、"Fitness Method" で指定されたメソッドにより得られた値が終了しきい値よりも小さい。
- － 世代の最大数に達した。

関連するコンフィギュレーションパラメータ (**Optimization Configuration**)

- － **Termination Threshold** : 終了条件として使用されるしきい値。しきい値は正のダブル (倍精度) です。
- － **Number of Generations** : 世代の最大数

適応度 (Fitness)

母集団内のモデルごとに適応度が計算されます。ここでは、以下の2つの部分の合計として、未加工の適合度が計算されます。

1. "Fitness Method" (費用関数) として指定されたメソッドにより得られた値
2. グラフの複雑さにペナルティを科すスカラ値

実際の適応度は、未加工の適応度を $[0, 1]$ の範囲の値に正規化することにより得られます。

関連するコンフィギュレーションパラメータ (**Optimization Configuration**)

- － **Fitness Method** : 以下の費用関数から選択します。
 - **rmse** 2乗平均平方根誤差
 - **mse** 2乗平均誤差
 - **abs** 正規化絶対誤差 (L^1 -normの平均)
- － **Parsimony Coefficient** : 複雑性ペナルティの影響を重み付けするための係数。未知の問題については値1が妥当な出発点です。値を小さくするほど式は複雑になります。

プログラムの選択 (Program Selection)

グラフの構造的最適化は、母集団からグラフを選択することから始まります。選択は、前のステップで得られた適応性に基づき、以下のメソッドで行われます。

関連するコンフィギュレーションパラメータ (**Optimization Configuration**)

- － **Program Selection Method** :
 - **tournament** : 指定されたトーナメントサイズのグラフが母集団からランダムに描画され、最大の適応度を持つグラフが選択されます。
 - > **Tournament Size** : トーナメントにより選択するプログラムの数。2つ以上の個体を選択すると、アルゴリズムの進行が非常に不安定になります。数が **Population Size** (母集団のサイズ) に収束する場合は、最も適応したプログラムの優位性が強制されます。出発点としては "Population Size" の約10%を選択するのが好ましいといえます。
 - **fitness-based** : 現在の母集団のすべてのグラフの適応度から確率分布が導出されます。この分布をサンプリングしてグラフを選択します。グラフの適応度が高いほど選択される可能性が高くなります。
 - **greedy overselection** : 母集団が適応度の高いグループと低いグループに分割されます。両グループに対して適応度ベースの選択が行われます。適応度の高いグループを採用する確率は **Probability Top** で決定されます。それ以外の場合は適応度の低いグループが選択されます。

- > **Fraction Top** : 適応度の高いグループのサイズを決定するための、母集団に対する割合です。適応度が最も高いグラフがグループに含まれることとなります。
- > **Probability Top** : 適応度ベースの選択において適応度の高いグループが使用される確率です。
- **multi tournament** : tournamentと似ています。確率を選択できますが、適応度の代わりに適応度／複雑度が基準として使用されます。このオプションは、遺伝的プログラミングのような勾配のない手法にのみ与えられる、複数の基準による最適化の可能性を示しています。
 - > **Tournament Size** : トーナメントにより選択するプログラムの数。2つ以上の個体を選択すると、アルゴリズムの進行が非常に不安定になります。数が**Population Size** (母集団のサイズ) に収束する場合は、最も適応したプログラムの優位性が強制されます。出発点としては "Population Size" の約10%を選択するのが好ましいといえます。
 - > **Probability Fitness** : 適応度の代わりに適応度／複雑度を使用する確率

進化 (Evolution)

進化ステップは、生殖、拡張、突然変異、遺伝子交叉、といったグラフ変更操作で構成されます。これは実際の構造的最適化ステップであり、前のステップで選択したグラフに依存します。各手法の確率に応じて、それらがこのグラフにどのように適用されるかが決定されます。

関連するコンフィギュレーションパラメータ (Evolution Probabilities)

以下のオプションで、4つの操作を実行する確率を定義します。確率の合計が100になるようにしてください。

- **Crossover** : 80～90の範囲が適しています。
- **Reproduction** : この操作は、適応度の高いグラフがそのまま変更されずに次世代に引き継がれるようにするものです。
- **Expansion**
- **Mutation**

Reproduction (生殖)

「グラフの生殖」は、次世代の母集団にグラフをそのまま引き継ぐことを意味します。

Expansion (拡張)

「グラフの拡張」は以下のように行われます。

- グラフの終端ノードをランダムに選択します。
- 深度2の新しいランダムグラフを作成します。
- 終端ノードを新しいグラフに置き換えます。

拡張されたグラフの方が元のグラフより適応度が高い場合は、拡張されたグラフが次の世代に引き継がれます。そうでない場合は元のグラフがそのまま引き継がれます。

Mutation (突然変異)

「グラフの突然変異」は、グラフ内でランダムに選択されたノードに適用される3つの操作により実行されます。

- － **New Tree** : 選択されているノードとそのすべてのサブノードを、新しく作成されたグラフ (最大深度3) に置き換えます。
- － **Hoist** : グラフの正当性は保持したまま、選択されているノードを削除します。この操作は、グラフの膨張を防ぐのに役立ちます。
- － **Point** : 選択されているノードを、同じ入力数のランダムノードに置き換えます。

どのメソッドをどのように適用するかは、個々の確率で決まります。3つの確率の合計が100になるようにしてください。

関連するコンフィギュレーションパラメータ (**Mutation Probabilities**)

- － **New Tree** : New Tree処理を実行する確率
- － **Hoist** : Hoist Mutation処理を実行する確率
- － **Point** : Point Mutation処理を実行する確率

Crossover (遺伝子の交叉)

遺伝子交叉の操作は基本的に、2つのグラフを再結合してより適応度の高いグラフを見つけることです。操作は、次の手順で実行されます。

1. 前のステップで選択したグラフを「ターゲットグラフ」とします。
2. 第2のグラフを選択します。このグラフは、"Program Selection" で指定され、ソースグラフとして定義されたものです。
3. 両グラフにおいて、1つのノードとそのすべてのサブノードを、交換するブランチとしてそれぞれランダムに選択します。
4. ターゲットグラフ内のブランチをソースグラフのブランチに置き換えます。

この操作の結果であるグラフが、次世代の母集団に引き継がれます。

要求された母集団サイズに到達 (**Reached Population Size Request**)

進化の各操作は、次世代の母集団が **Population Size** で定義された数のメンバーを持つまで適用されます。

5 ASCMO-MOCAの操作

5.1 ASCMO-MOCAのユーザーインターフェース

本項では、ASCMO-MOCAのユーザーインターフェースについて概説します。

メインメニューの機能と、それに関連する各種ダイアログウィンドウについての詳細な説明は、コンテキスト依存のオンラインヘルプ (<F1>、または **Help > Online Help**) に記載されています。

5.2 ASCMO-MOCAユーザーインターフェースの要素

本項では、ユーザーインターフェースの重要な要素についての概要をご紹介します。下図は、ASCMO-MOCAのメインユーザーインターフェースです。

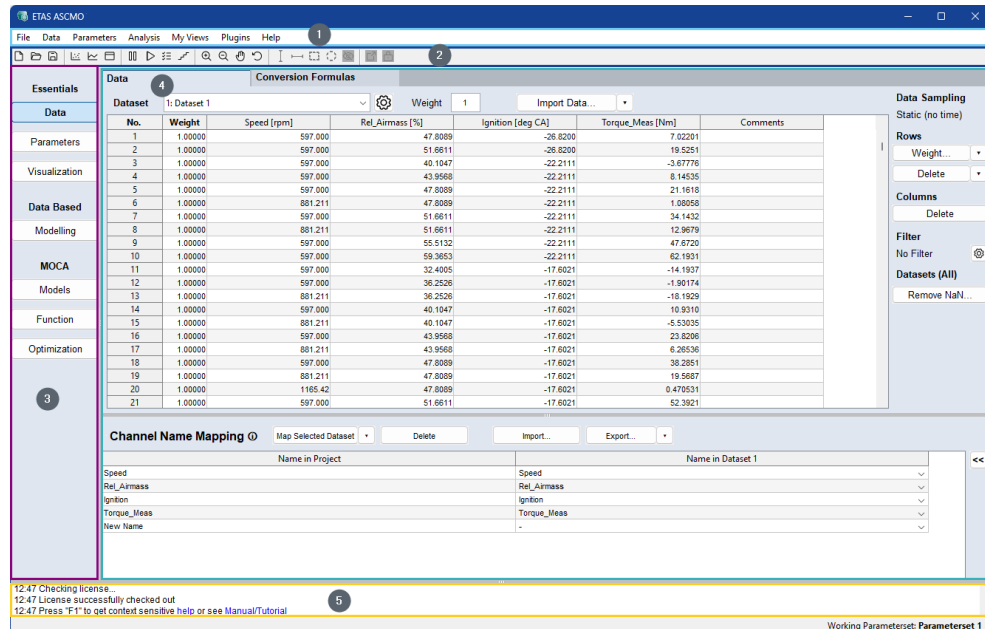


図5-1: ASCMO-MOCAのメインウィンドウの構成

主要な操作は、すべてこのメイン作業ウィンドウから行います。ナビゲーションペインとメインメニューバーからはさまざまな対話型編集機能が利用でき、これらについてはそれぞれの項で詳しく説明します。

- ① メインメニュー
- ② ツールバー（ツールバー（下記）を参照）
- ③ ナビゲーションペイン（77を参照）
- ④ メイン作業ウィンドウ
- ⑤ ログウィンドウ（ログウィンドウ（ページ79）を参照）















ウィンドウ最下部のステータスバー：現在の処理内容についての情報が表示されます。

5.2.1 ASCMO-MOCAのメインメニュー


メインメニューの機能の詳細については、オンラインヘルプ（**Help > Online Help**）、またはその状況依存トピック（<F1>）を参照してください。






5.2.2 ツールバー

ツールバーには以下のような操作ボタンが含まれています。

	New project	ASCMO-MOCAの新しいインスタンスが開きます。
	Open project	データ選択ダイアログが開き、保存されているプロジェクト (*.moca) を開くことができます。
	Save	現在のプロジェクトを保存します。
	Scatter plot for training data	Data and Nodes - Training Data ウィンドウが開きます。 データ/ファンクションノードのグラフィカル分析 (ページ21) を参照してください。
	Scope view of residuals, function evaluation and training data	Scope View - Training Data ウィンドウが開きます。
	Open Visualization in separate window (可視化ステップを別ウィンドウで開く)	
	Pause evaluation	Optimize をクリックするまでファンクションと外部モデルの評価を一時停止します。 オンにすると、NaNのみが返ります。
	Recalculate once, even if pause is active, update RMSE display (一時停止中であっても再計算を1回行ってRMSEを表示)	
	Select active datasets	Active Datasets ウィンドウが開きます。
	Automatically quantize all parameters and limits of all parametersets on parameter change	パラメータを変更すると、自動量子化によって自動的にA2L変換式が適用されます。量子化は、最適化実行中には行われず、最適化終了時に適用されます。
	Zoom In	プロットをクリックすると表示が拡大されます。
	Zoom Out	プロットをクリックすると表示が縮小されます。
	Pan	プロットをウィンドウ内で移動させることができます。
	Rotate 3D	プロットを任意の方向に回転させることができます。

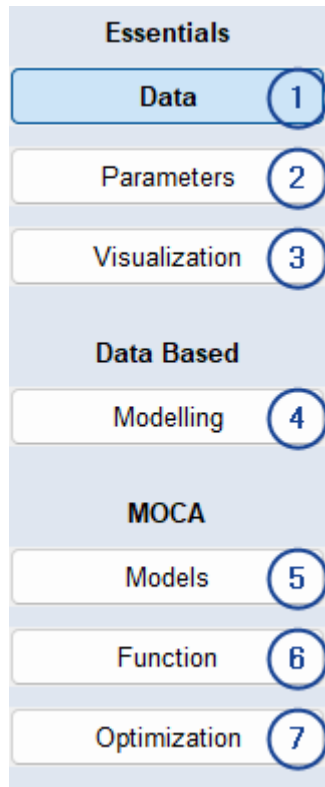
Visualization (可視化) ステップでのみ利用可能

-  **Add X Axis Cursor in Plot (プロットにX軸カーソルを追加)**
-  **Add Y Axis Cursor in Plot (プロットにY軸カーソルを追加)**

-  **Mouse selection in Plot with Rectangle** (マウスを使用してプロットの一部を四角形で選択)
-  **Mouse selection in Plot with Lasso** (マウスを使用してプロットの一部を投げ縄で選択)
-  **Hide all selections** (選択を非表示にする)
-  **Undock current Visualization in separate window** 現在の可視化タブをドッキング解除して、個別のウィンドウに表示します。
-  **Lock visualization against changes** **Optimize** をクリックしない限り、変更に対してすべてのタブの可視化をロックします。

5.2.3 ASCMO-MOCAのナビゲーションペイン

ウィンドウ左部にあるナビゲーションペインに含まれるボタンは、測定データのインポートから最適化済みパラメータのエクスポートまでの一連の処理ステップの流れをナビゲートするものです。



1 - Data : Data ステップでは、測定ファイルをインポートして編集し、さらにそのデータを測定ファイルにエクスポートすることができます。測定ファイルから読み取ったデータチャンネルをファンクション変数にマッピングすることもできます (Channel Name Mapping)。

詳細は、[データ \(ページ20\)](#) と [チュートリアル \(ステップ1 : データのインポート \(ページ83\)\)](#)、[オンラインヘルプ](#)を参照してください。

2 - Parameters : Parameters ステップでは、マップ、カーブ、スカラ、3Dキューブ、4Dキューブ、圧縮モデル、行列を管理して、**Function** ステップでの使用に合わせて変更することができます。

詳細は、[パラメータ \(ページ26\)](#) とチュートリアル ([ステップ3 : パラメータ \(ページ99\)](#))、オンラインヘルプを参照してください。

3 - Visualization : Visualization ステップでは、データを可視化することができます。

詳細は、[可視化 \(ページ32\)](#) とオンラインヘルプを参照してください。

4 - Modelling : Modelling ステップでは、データベースモデル (機械学習モデル、ブラックボックスモデルなど) を作成してトレーニングすることができます

 **注記**

このステップを使用するには、ASCMO-DYNAMICのライセンスが必要です。

5 - Models : Models ステップでは、モデルをインポートし、そのパラメータや入出力をASCMO-MOCAのパラメータとリンクさせることができます。

 **注記**

ASCMO-MOCAにおいてSimulink[®]モデルを使用するには、有効なライセンスでインストールされたSimulink[®]が必要です。

詳細は、[モデル \(ページ39\)](#) とチュートリアル ([ステップ4 : モデル \(ページ100\)](#))、オンラインヘルプを参照してください。

6 - Function : Function ステップでは、ノードを段階的に作成してファンクションを構築することができます。ここではパラメータとデータとのリンクが重要になります。

詳細は、[ファンクション \(ページ43\)](#) とチュートリアル ([ステップ5 : ファンクションの構築 \(ページ110\)](#))、オンラインヘルプを参照してください。

7 - Optimization : Optimization (最適化) ステップでは、以下のメインタスクを実行することができます :

- パラメータベースの最適化目標 (平滑度、勾配の制限など) を定義します。
- 適合結果と比較する参照値となるパラメータを定義します。
- 最適化を開始します。
- 最適化済みパラメータをエクスポートします。

詳細は、[最適化 \(ページ50\)](#) とチュートリアル ([ステップ6 : 最適化 \(ページ117\)](#))、オンラインヘルプを参照してください。

5.2.4 ログウィンドウ

メインウィンドウの下部は「ログウィンドウ」と呼ばれ、実行中の処理のステータス（最適化に関する情報など）が表示されます。

ログウィンドウ内で青い下線のついた部分は、オンラインヘルプやユーザーガイド（図中a、b）、サンプルプロジェクト（図中c～e）などを開くためのリンクになっています。また、ログファイルを保存して解析やエラー処理などに利用することができます。

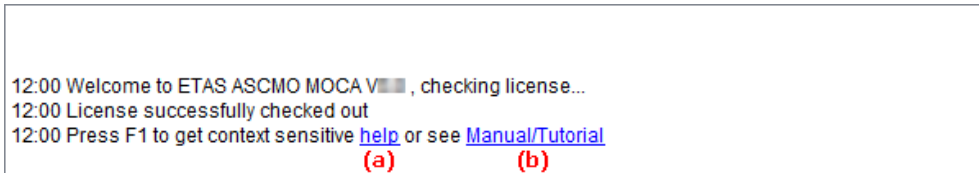


図5-2: ログウィンドウに表示される情報の例 ((a) : オンラインヘルプへのリンク、(b) : ユーザーガイドへのリンク)

ログファイルを保存する

1. ログウィンドウ内でマウスを右クリックし、ショートカットメニューから **Save Log to File** を選択します。
"Save Log file As" ウィンドウが開きます。
2. ファイル名を入力して **保存** をクリックします。
ログファイルが保存されます。

6 チュートリアル：ASCMO-MOCAの操作

このチュートリアルでは、エンジントルクセンサの例を用いてASCMO-MOCAの基本機能と操作手順を紹介します。

6.1 チュートリアルについて

本項では、このチュートリアルの構成と、パラメータの最適化に使用される測定データの要件について説明します。

6.1.1 チュートリアルの課題

一般的なECUには、計算によりセンサ信号を算出するセンサモデルが数多く実装されています。実際のセンサを使用したデータ収集処理は、技術的に困難であったりコストが高くなり過ぎてしまうためです。このチュートリアルで扱うエンジントルクモデルはその代表例のひとつです。ASCMO-MOCAでは、このようなモデルファンクションのセットアップと適合を行い、ファンクションのパラメータをセンサの実測データに基づいて最適化することができます。最適化の目的は、ファンクションのパラメータの平均二乗誤差RMSE（平均二乗誤差）（ページ25）を最小化することです。これによってファンクション予測とセンサ測定データとの差異が最小化されます。

このチュートリアルで段階を追ってモデリングされるトルク関連ファンクションの構造は、[ステップ5：ファンクションの構築](#)（ページ110）に示されています。

6.1.2 チュートリアルの構成

チュートリアルは以下の作業ステップで構成されています。

- － [ASCMO-MOCAの起動](#)（次ページ）
ユーザーPC上でASCMO-MOCAを起動する方法を説明します。
- － [ステップ1：データのインポート](#)（ページ83）
最初のステップでは、測定データをロードし、データチャンネルをファンクションノードに割り当てます。
- － [ステップ2：データの分析](#)（ページ95）
インポートした測定データは、いつでもグラフィカルに可視化して、データを整理したり評価したりすることができます。
- － [ステップ4：モデル](#)（ページ100）
このステップでは、既存のSimulinkモデルをリンクして、パラメータ、入力、出力のマッピングを準備することができます。
- － [ステップ5：ファンクションの構築](#)（ページ110）
測定データを読み取って妥当性をチェックしたら、このチュートリアルでモデリングされるトルクセンサファンクションのセットアップを開始します。
- － [ステップ3：パラメータ](#)（ページ99）
このステップでは、パラメータをチェックして、必要に応じて適合を行います。ここでは、最適化後に参照値として定義したパラメータのみが可視化されます。[ステップ6：最適化](#)（ページ117）を参照してください。
- － [ステップ6：最適化](#)（ページ117）

最適化を開始する前に、最適化に影響を与えるさまざまな条件を設定します。これらの設定が終了すると、最適化を開始できます。

－ ステップ7：エクスポート（ページ119）

このステップでは、作成して最適化したパラメータをエクスポートします。パラメータはDCMファイル（*.dcm）としてエクスポートでき、プロジェクトは機能制限付きのランタイム環境用に保存することができます。

6.1.3 測定データの要件

一般的にASCMO-MOCAでパラメータの最適化を成功させるには、「ファンクションのパラメータ最適化の品質は常に測定データの品質に依存する」というシンプルなルールが重要になります。つまり、計測点分布が偏っているデータや不適切なデータに基づいてパラメータを適合すると、ファンクションの予測はほとんど役に立たないものになってしまいます。

ASCMO-MOCAにインポートする測定ファイルは、以下の条件を満たしている必要があります。

- － データフォーマット
 - Microsoft Excel (*.xls、*.xlsx)
 - MDAでエクスポートされたASCIIフォーマット (*.ascii)
 - カンマ区切りのテキストファイル (*.csv、*.txt)
 - 測定データフォーマット (*.dat、*.mf4、*.mdf、*.mdf3)
- － 各列に含まれる出力値
- － 名前（または名前と単位）は、1行目（または1行目と2行目）に含まれている必要があります。

注記

パラメータ化に使用されるデータは、必ずしもテストベンチなどの実機で計測されたものである必要はありません。コンピュータシミュレーションの結果などのデータも使用できます。

6.1.4 モデリング用データ

このチュートリアルでパラメータの最適化に使用するデータは、**Torque_Data.xlsx**というExcelシート（<installation>\Example ディレクトリ）に保存されています。

<installation> はインストールディレクトリです。デフォルトにおいて<installation> = C:\Program Files\ETAS\ASCMO x.xです。

このファイルに保存された測定データはASCMO-MOCAでパラメータの最適化を成功させるための要件を満たしています。

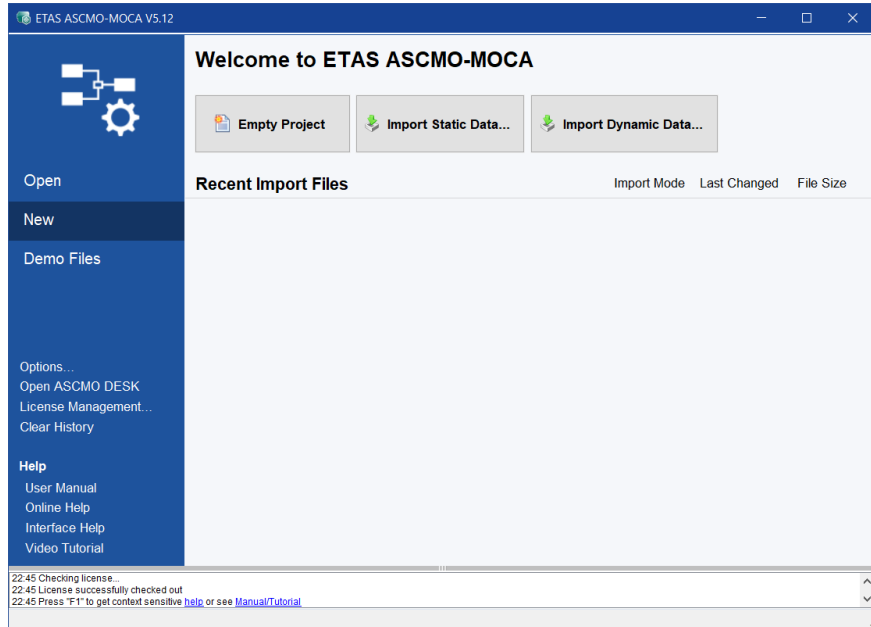
- － センサデータの収集に使用された実験の構成は、DoEメソッドに相当するものです。つまり、測定値は独立的に変化し、均一に分布しています。
- － 測定ファイルに保存された測定データには、不合理な値（トルク値 ≤ 0など）が含まれていません。

6.2 ASCMO-MOCAの起動

ここでは、ユーザーPC上でASCMO-MOCAを起動する方法を説明します。

ASCMO-MOCAを起動する

1. 以下のいずれかを行います。
 - i. **ASCMO-DESK** ウィンドウで、**Model Calibration** タイルをクリックします。
ASCMO-MOCAスタートウィンドウが開きます。



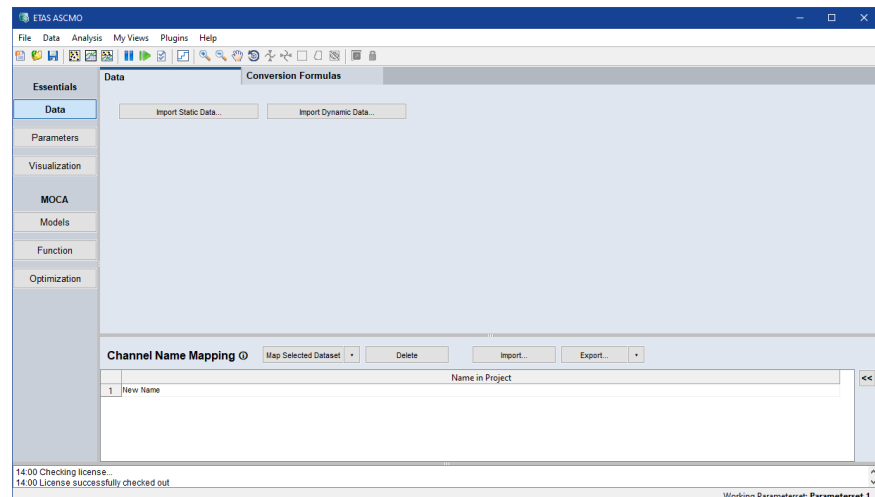
2. 左側のメニューパネル内の **New** をクリックします。



- 3.

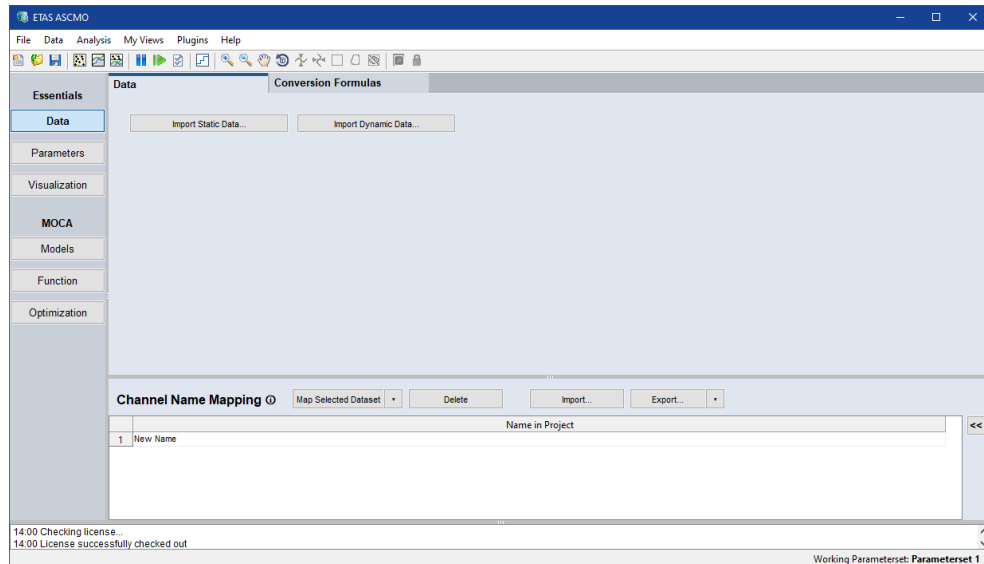
のいずれかをクリックして、空のプロジェクトを開くか、静的データまたは動的データを直接インポートします。

Empty Project をクリックすると、ASCMO-MOCAの空のメインウィンドウが開きます。ここで測定データのインポートを開始することができます。[ステップ1：データのインポート（次ページ）](#)を参照してください。



6.3 ステップ1 : データのインポート

最初のステップでは、測定データをロードします。複数のファイルをインポートしてチャンネル名の不一致などが生じた場合は、プロジェクト固有の名称を割り当てることができます。

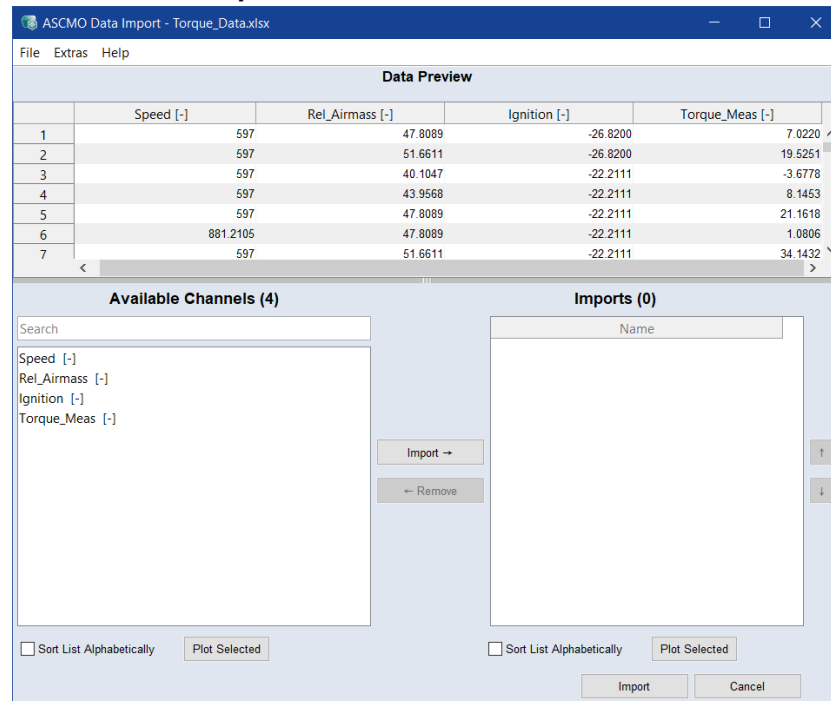


測定ファイルをインポートする

新しいプロジェクトを開始するには、まず、パラメータ化と最適化のために必要なデータを測定ファイルからインポートする必要があります。

1. メイン作業領域で **Import Static Data** ボタンをクリックします。
2. ファイル選択ウィンドウで、`<installation>\Example\Moca` ディレクトリ内のファイル `Torque_Data.xlsx` を選択します。
`<installation>` のデフォルトは `C:\Program Files\ETAS\ASCMO x.x` です。
3. **開く** をクリックします。
 ファイルに複数のワークシートが含まれている場合は **Sheets** ウィンドウが開きます。
4. **Sheets** ウィンドウで、インポートしたいワークシート（このチュートリアルでは `Torque_Data`）を選択し、**OK** をクリックします。

ASCMO Data Import ウィンドウが開きます。



Data Preview フィールドには、ワークシートに含まれるすべてのデータがテーブル形式で表示されます。**Available Channels** フィールドで、インポートするデータチャンネルを選択します。

6.3.1 測定データの妥当性チェックを行う

インポートする前に、データの内容を確認します。ここでは測定データを表示でき、さらに入力の関連性（出力に対する影響度）を確認することができます。

インポートする測定データをプロット表示する

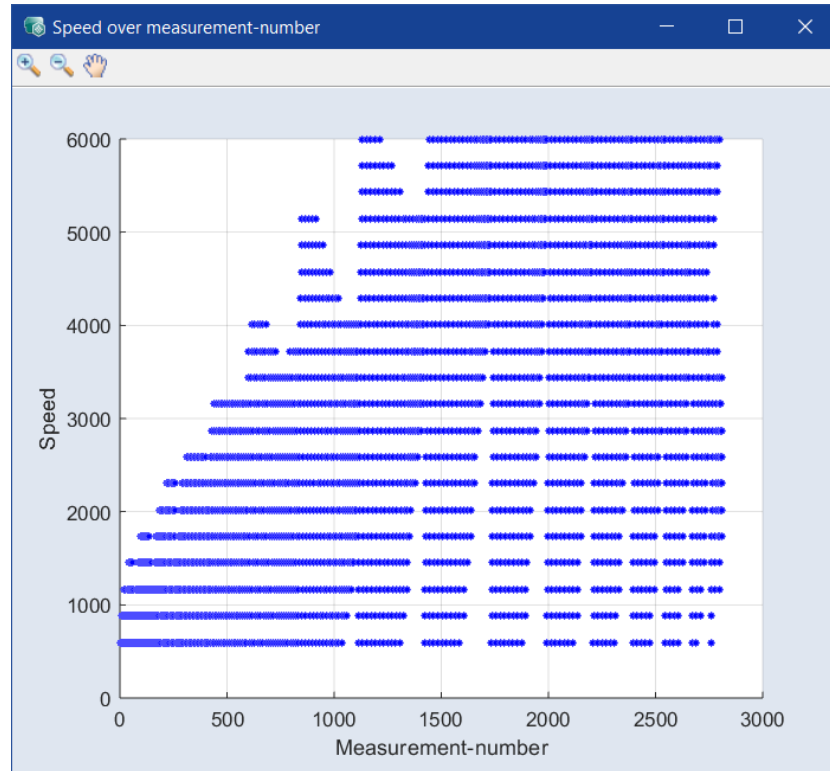
Available Channels フィールドで、測定データチャンネル（複数可）を選択します。

1. テーブル内では、CTRL/SHIFT による標準的な選択機能を使用でき、また LMB を押し下げたままカーソルをドラッグして複数のセルや行を選択することもできます。
2. 以下のいずれかを行います。
 - **Plot Selected** をクリックします。
 - **Extras > Plot Selected** を選択します。

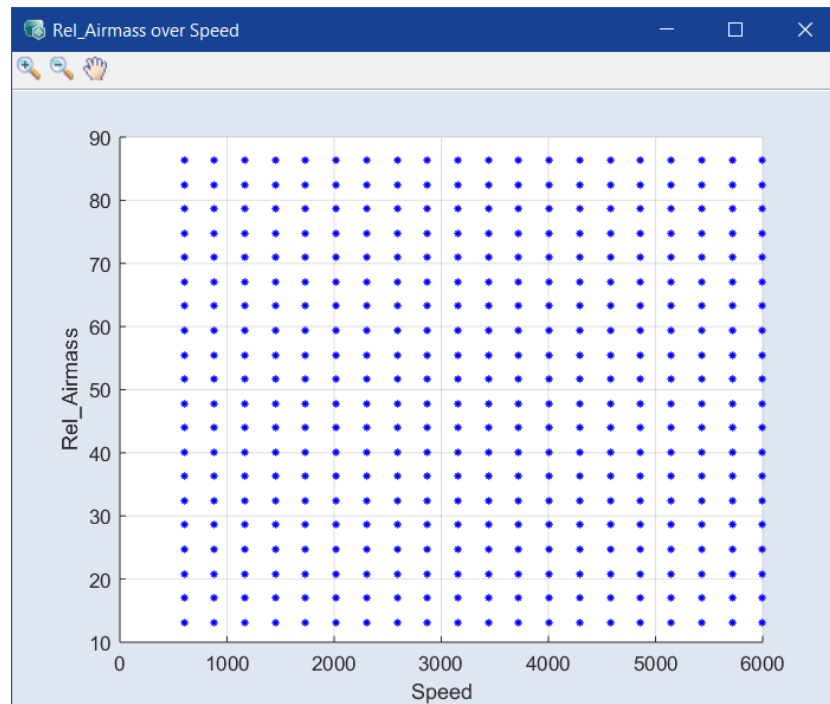
ウィンドウが開き、選択されているチャンネルの数に応じて、いずれかのプロットが表示されます。

3. プロットをチェックして、外れ値や、その他の不適切な値がないかを確認します。

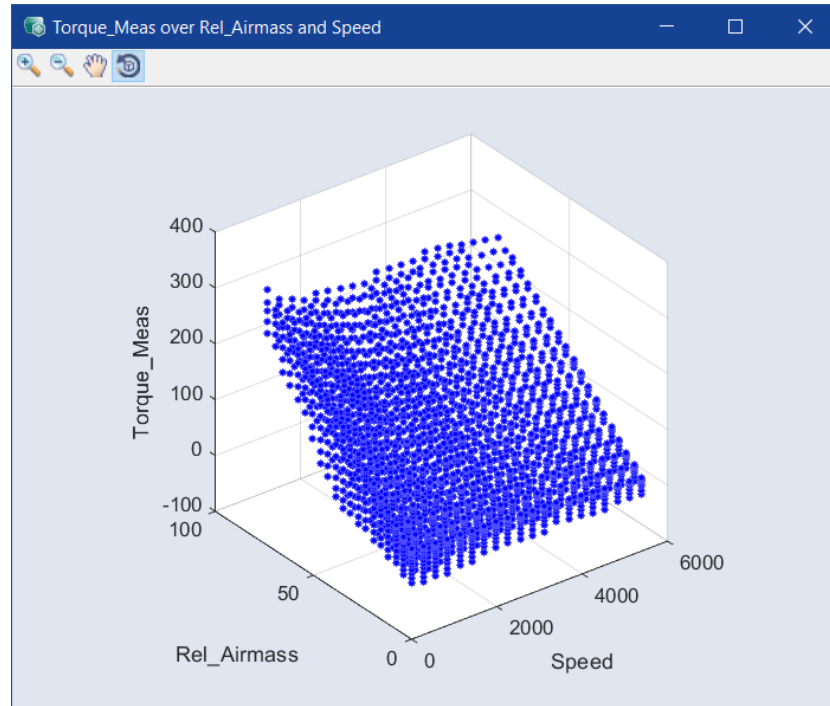
1チャンネル : 測定番号に対する測定データ（例 : 測定番号に対する Speed）



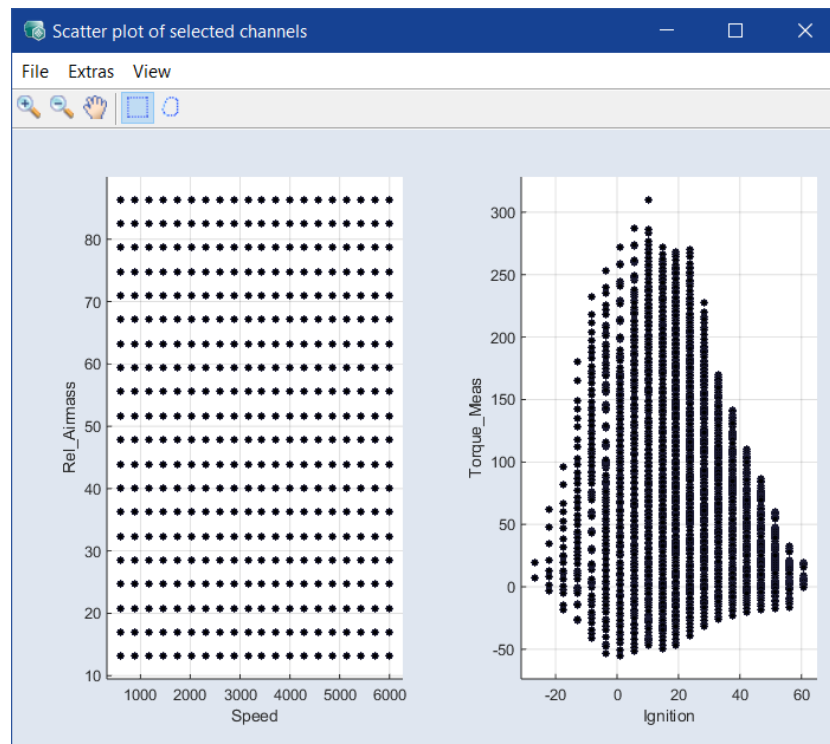
2チャンネル : 1つの列に対する他の列のデータ (例 : Speed に対する Rel_airmass)



3チャンネル : 2つの列のデータで表される平面に対する第3列のデータ (マウスで自由に回転可能) (例 : Speed-Rel_airmass 面に対する Torque_Meas)



4チャンネル以上 : 一連の散布図



入力の関連性を確認する

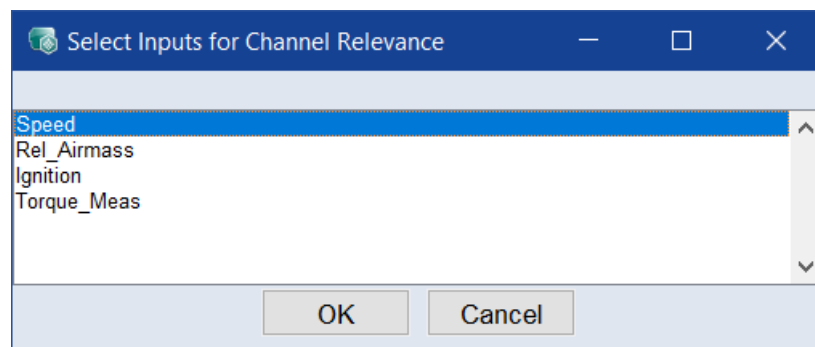
入力の関連性の判定についての詳細は、[入力の関連性の確認 \(ページ21\)](#) を参照してください。

1. **ASCMO Data Import** ウィンドウの **Available Channels** フィールドで、入力測定データチャンネル（複数可）を選択します。

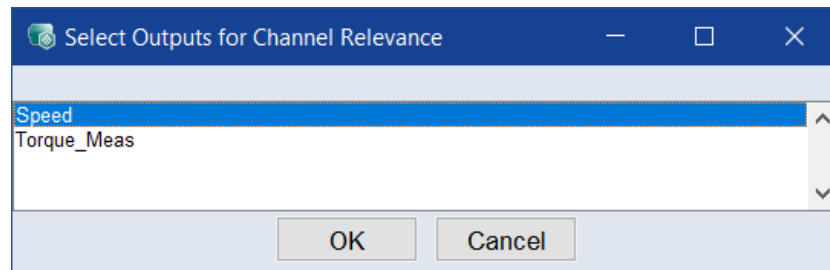
注記

ここでは必ず1つ以上のチャンネルを未選択の状態にしておいてください。すべてのチャンネルを選択すると、関連性のチェックが行えません。

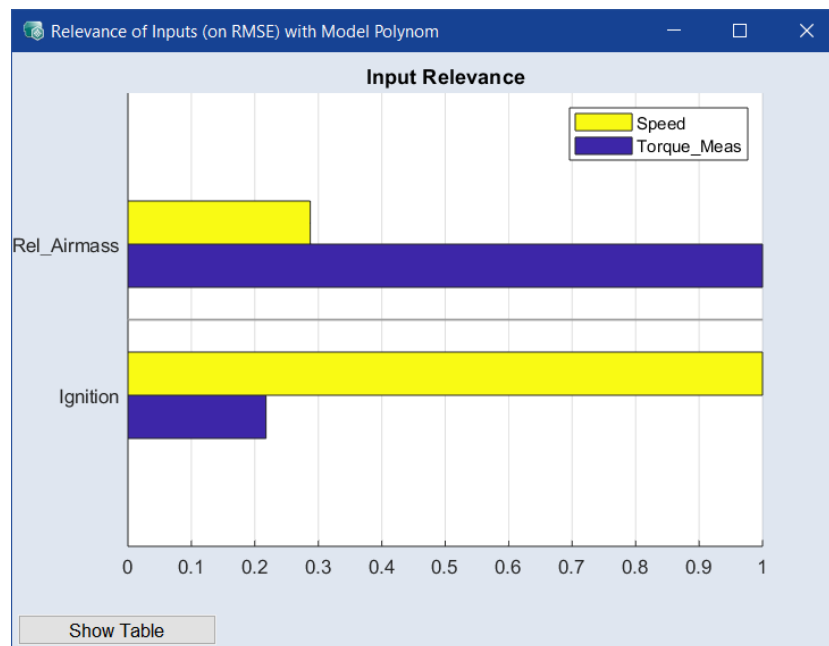
2. **Import** をクリック、またはチャンネルをダブルクリックします。
選択されているチャンネルが **Import** リストに追加されます。
3. **Extras > Find Relevant Channels** を選択します。
Select Inputs for Channel Relevance ウィンドウが開きます。



4. このウィンドウで、2つ以上の入力チャンネルを選択して **OK** をクリックします。
5. **Select Outputs for Relevance** ウィンドウが開きます。



6. このウィンドウで、1つ以上の出力チャンネルを選択して **OK** をクリックします。
"Relevance of Inputs" ウィンドウが開きます。出力に対する入力の影響度が表示されます。
7. **Show Table** をクリックすると、結果が **Relevance of Inputs Table** ウィンドウにテーブル表示されます。
8. 必要に応じて、入力と出力の選択を変更して結果を確認します。



6.3.2 コンフィギュレーションの保存とロードを行う

測定データチャンネルからファンクション変数への割り当ては、コンフィギュレーションファイル (*.ini) に保存しておくことができます。

コンフィギュレーションの保存とロードを行う

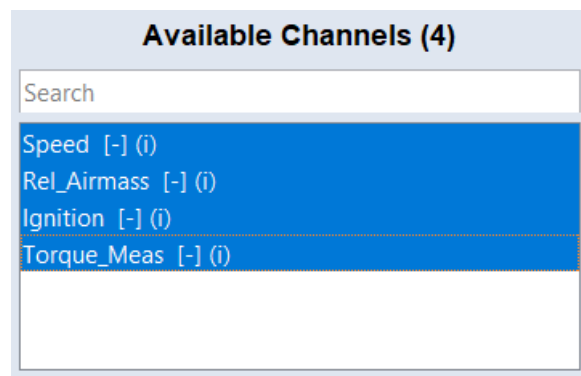
1. **ASCMO Data Import** ウィンドウで **File > Save Channel Config (*.ini)** を選択します。
2. ファイル選択ウィンドウに、現在のコンフィギュレーションを保存するファイルの名前を入力します。
3. すでに保存されているコンフィギュレーションファイルをロードするには、**File > Load Channel Config (*.ini)** を選択します。

6.3.3 測定データをインポートする

データをインポートできる状態になりました。

測定データをインポートする

1. **ASCMO Data Import** ウィンドウで、**Available Channels** フィールド内の全チャンネルを選択します。

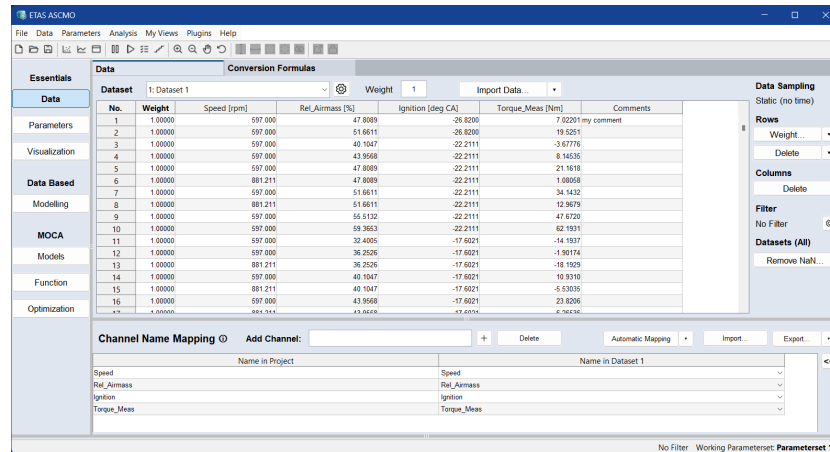


2. **Import** → をクリックします。

各チャンネルが **Import** リストに追加されます。

3. **Import** をクリックします。

データがASCMO-MOCAの現在のプロジェクトにインポートされます。インポートされたファイルの内容が **Data** タブに表示されます。



6.3.4 測定チャンネルを変数に割り当てる

データをインポートしたら、インポートされた測定ファイルのチャンネルを、ファンクション内で使用される変数（ノード）に割り当てます。

手動でマッピングを追加することもできます。**Channel Name Mapping** テーブルの上の入力フィールドに新しい変数名を入力して、**+** をクリックします。新しい行が挿入され、その左側には入力した名前、右側にはデータセット内のチャンネルを選択するためのドロップダウンが表示されます。

長いリストを見やすくするには、テーブルの列ヘッダをクリックして昇順または降順にソートします。

注記

測定ファイルの構造がデータインポート用の要件（[測定データの要件（ページ81）](#)を参照）に適合している場合は、各チャンネルが適切な変数に自動的に割り当てられます。

ここではASCMO-MOCAにより自動的に割り当てが行われるので、[ステップ2：データの分析（ページ95）](#)に進んでインポートされた測定データの分析を行うことができます。

変数名を変更する

ファンクションで使用される変数の名前を変更するには、**Channel Name Mapping** テーブルを使用します。

1. **Name in Project** 列で、変更したい変数を特定します。
2. セルをダブルクリックして新しい名前を入力し、**Enter** を押します。

⇒ 新しい変数名が確定されます。

ヒント : Name in Project 列のヘッダをクリックすると、テーブルがソートされ、目的のエントリを見つけやすくなります。

割り当てを解除する

パラメータとして使用しないチャンネルが測定データに含まれている場合は、"Data" ペインの **Channel Name Mapping** テーブルでその割り当てを削除することができます。

1. **Name in Project** 列で、目的の変数を選択します。
 2. **Delete** をクリックします。
- ⇒ 変数が **Channel Name Mapping** テーブルから削除されます。

6.3.5 ASCMO-MOCAのデータステップでの作業

The screenshot displays the ETAS ASCMO software interface. The main window is titled "Data" and shows a table of data points. Below this, the "Channel Name Mapping" panel is visible, which maps variables from the "Name in Project" column to the "Name in Torque_Data (Torque_Data)" column. The mapping table is as follows:

No.	Name in Project	Name in Torque_Data (Torque_Data)
1	Speed	Speed
2	Rel_Airmass	Rel_Airmass
3	Ignition	Ignition
4	Torque_Meas	Torque_Meas
5	New Name	-

The interface also shows a "Data" table with columns for No., Weight, Speed [-], Rel_Airmass [-], Ignition [-], and Torque_Meas [-]. The "Channel Name Mapping" panel includes buttons for "Map Selected Dataset", "Delete", "Import...", and "Export...".

ASCMO-MOCA では、トレーニングデータとテストデータとして複数のデータセットを使用することができ、これらはデータペインの個別のタブに表示されます。トレーニングデータはすべて最適化に用いられ、テストデータは評価と予測に用いられます。

各データセットには重みを設定することができ、最適化における各データセットの影響の度合いを調整できます。

各データセットごとに列の名前が異なっていても、名前マッピングにより、複数のデータセットが正しく関連付けられます。

初めてインポートされたデータは、必ず第1のトレーニングデータセットとして使用されます。その後、別のファイルをインポートする際には、そのデータセットの用途（トレーニングデータセット/テストデータセットの追加、既存のデータセットを置換）を選択することができます。テストデータセットのRMSEを計算するには、**Analysis > Residual Analysis > Test Data > *** を選択します。

複数のデータセットをインポートする

複数のデータセットをインポートするには、以下のように操作します。

1. **Import Data** をクリックします。
2. ファイル選択ウィンドウで、インポートしたいファイルのパスと名前を指定します。
3. **開く** をクリックします。
ASCMO Data Import - <file name> ウィンドウが開きます。
4. このウィンドウで、入力チャンネルを選択（[測定ファイルをインポートする（ページ83）](#)を参照）します。
5. **Import** をクリックします。
ファイルがインポートされます。

データセットの名前を変更する

1. 名前を変更したいデータセットのタブを選択します。
2. タブを右クリックして、ショートカットメニューから **Rename data set** を選択します。
3. **Rename** ウィンドウで、新しい名前を入力して **OK** をクリックします。

データセットを削除する

1. 削除したいデータセットのタブを選択します。
2. **Delete Dataset** をクリックします。
確認のウィンドウが開きます。
3. 確認のウィンドウで **Delete** をクリックします。
プロジェクトからデータセットが削除されます。

6.3.5.1 データポイントの重み付けを行う

Weight 列では、各データポイントの最適化用の重みを設定することができます。この値を0にすることによりその行のポイントを非アクティブにすることができます。非アクティブな行は、RMSE演算と最適化において無視されます。デフォルトは1で、値を大きくするとそのポイントの重要度が高くなります。つまり、最適化においてそのデータポイントについて最適化目標が達成されることが重要となります。重みは最適化に影響しますが、表示されるRMSEには反映されません。

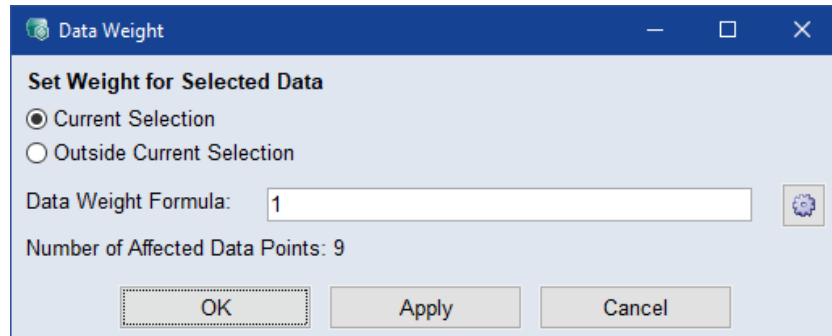
複数行を選択するには、<SHIFT> / <CTRL> キーを押したままマウスで行を選択します。その後、**Weight** ボタンをクリックして複数行の重みを設定します。

データポイントの重みの設定は、散布図ウィンドウ（**Analysis > Scatter Plot > ***）でも行うことができます。散布図で目的のデータポイントをマークした後、**Extras > Set Marked Points Weight** メニューオプションを使用して、マークされたデータの重みを設定することができます。

選択されているデータポイントの重みを設定する

1. **Weight** ボタンを使用してデータポイントの重みを設定するには、以下のように操作します。

- i. 1つまたは複数の行を選択します。
- ii. **Weight** をクリックします。



iii. **Data Weight Formula** ウィンドウで、重みを入力し、現在の選択範囲の内側／外側のいずれのデータポイントに対してそれを設定するかどうかを選択します。

iv. **OK** をクリックします。
重みが割り当てられます。

2. 個々のデータポイントの重みを設定するには、以下のように操作します。

- i. 目的のデータポイントの **Weight** 列をクリックして入力にモードにします。
- ii. 割り当てたい値を入力します。

データセット全体の重みを設定する

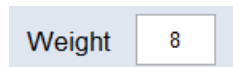
データセット全体の重みを調整するには、以下のように操作します。

注記

データセットが1つしかインポートされていない場合は、データセットの重みは意味を持ちません。

1. **Weight** フィールドに値を入力します。

0に設定すると、データセットが無効になります。このチュートリアルでは 8 を入力します。



2. データセットの変更後の重みの値を表示するには、**Analysis > Data Table > Training Data / Test Data / Training and Test Data** を選択します。

次のウィンドウが開きます。

Training Data						
File						
	Dataset No.	Weight	Rel_Airmass [-]	Ignition [-]	Torque_Meas [-]	
1	1	8	47.8089	-26.8200	7.0220	^
2	1	8	51.6611	-26.8200	19.5251	
3	1	8	40.1047	-22.2111	-3.6778	
4	1	8	43.9568	-22.2111	8.1453	
5	1	8	47.8089	-22.2111	21.1618	
6	1	8	47.8089	-22.2111	1.0806	
7	1	8	51.6611	-22.2111	34.1432	
8	1	8	51.6611	-22.2111	12.9679	
9	1	8	55.5132	-22.2111	47.6720	
10	1	8	59.3653	-22.2111	62.1931	
11	1	8	32.4005	-17.6021	-14.1937	
12	1	8	36.2526	-17.6021	-1.9017	
13	1	8	36.2526	-17.6021	-18.1929	v

Weight 列には、各データの重みの値に **Weight of <Dataset_name>** フィールドに入力された値を掛けたものが表示されます。たとえば行の重みが5でデータセット全体の重みが8の場合は、40という値が表示されます。

6.3.5.2 データセット内のデータの管理

ASCMO-MOCAには、データセット内の各データポイントを管理するための各種機能（編集、フィルタリング、ソートなど）が用意されています。

データポイントを編集する

データセット内の各ポイントのデータを編集するには、以下のように操作します。

1. 編集したいフィールドをクリックして入力モードに切り替えます。
2. 割り当てたい値を入力します。

非数を削除する

インポートしたデータに数値以外のデータが含まれている場合は、すべてのデータセットにおいて該当する行を自動的に削除することができます。以下のように操作します。

1. 非数のデータを含む行を削除するには、**Remove NaN** をクリックします。

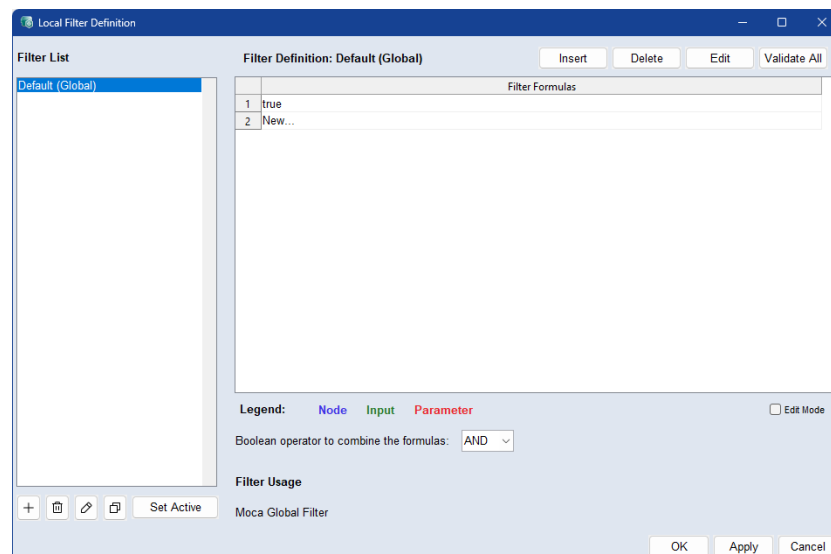
データのフィルタリングを行う

注記

フィルタは、すべてのデータセットの表示に対してのみ影響し、最適化には影響しません。最適化においてデータを無視するには、データの重みを0にセットします。詳しい情報は、MOCAヘルプを参照してください。

1. **Analysis > Filter Data** を選択します。

Filter Data ウィンドウが開きます。



2. 新しいフィルタを追加するには、フィルタリストの下の **+** をクリックします。
3. ポップアップウィンドウでフィルタの名前をキー入力し、**OK** で確定します。
4. フィルタに式を追加するには、**New** と表示された行をクリックします。
5. **Add Filter** ウィンドウが開き、各リスト（Data、Parameters、Nodes）と計算ボタンを使用して式を作成することができます。
データポイントは、式がそのデータポイントに対して真となる場合のみ表示されます。たとえば `%speed% > 2000` の場合は、2000より大きいポイントのみが表示されます。
6. 式を検証するには、**Validate** をクリックします。
7. 式が有効であれば、**OK** をクリックします。
8. 式と式をリンクする論理演算子として、ドロップダウンリストから **AND** または **OR** を選択します。
9. フィルタをグローバルフィルタまたはローカルフィルタとしてアクティブにするには、リストからフィルタを選択して **Set Active** をクリックします。
10. **OK** をクリックします。

データポイントを削除する

1. 行（複数可）を選択します。
2. **Rows** 領域の **Delete** ボタンをクリックします。
確認のウィンドウが開きます。
3. 確認のウィンドウで **Delete** をクリックします。
選択されている行が削除されます。

入力列を削除する

1. 列（複数可）を選択します。
2. **Columns** 領域の **Delete** ボタンをクリックします。
確認のウィンドウが開きます。

3. 確認のウィンドウで **Delete** をクリックします。
選択されている列が削除されます。

6.4 ステップ2：データの分析

インポートした測定データは、グラフィカルに可視化して、データを整理したり評価したりすることができます。

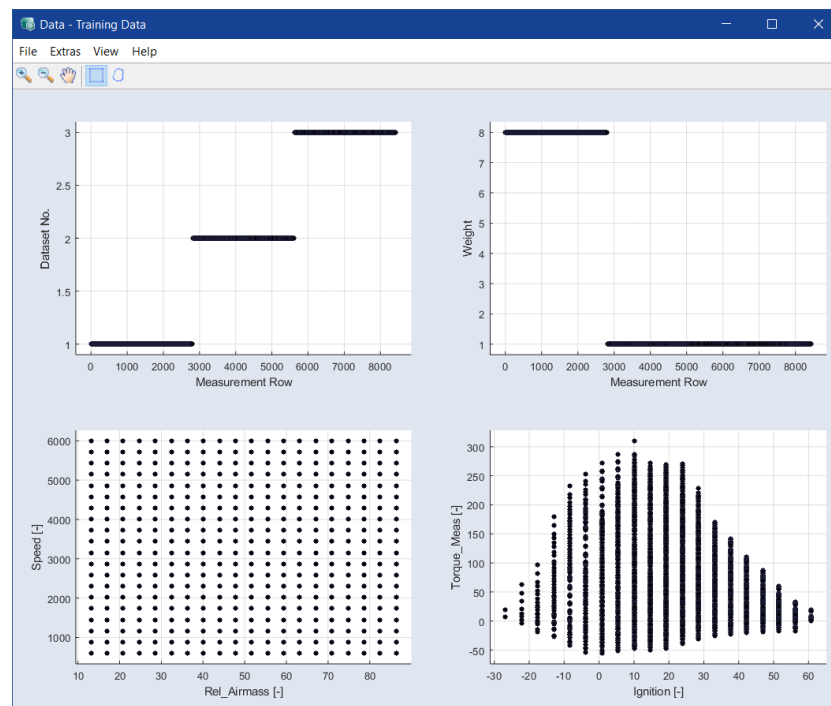
分析を行う際には、特に以下の点を考慮してください。

- － すべてのパラメータが実験計画に従って変化し、計測対象のシステムが目的に合った動作モードを示しているか
- － 出力変数の値は物理的に意味のある範囲内になっているか
- － 除外する必要のある外れ値が含まれているか

測定データの散布図を表示する

1. **Analysis > Scatter Plot > Training Data/Test Data/Training and Test Data > Data/Function Nodes** を選択します。

Data ウィンドウが開き、プロジェクトにファンクションノードが含まれる場合はさらに **Function Nodes** ウィンドウが開きます。このチュートリアルでは **Data** ウィンドウのみ使用します。

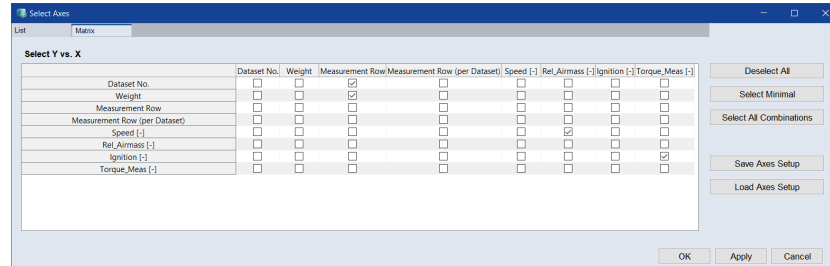


左側のプロット (Rel_Airmass に対する Speed) には、実験計画内のデータの分布状態が表示されます。

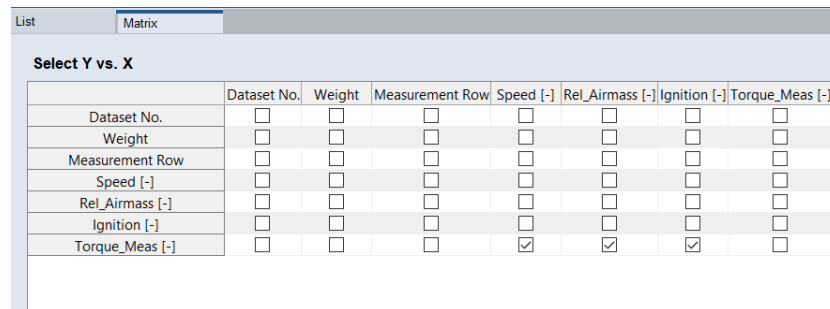
座標軸のペアを変更する

このプロットには関連する測定データ間の依存関係が示されていないので、座標軸のペアを調整する必要があります。表示される座標軸の選択は、プロットウィンドウで直接行うことができます。

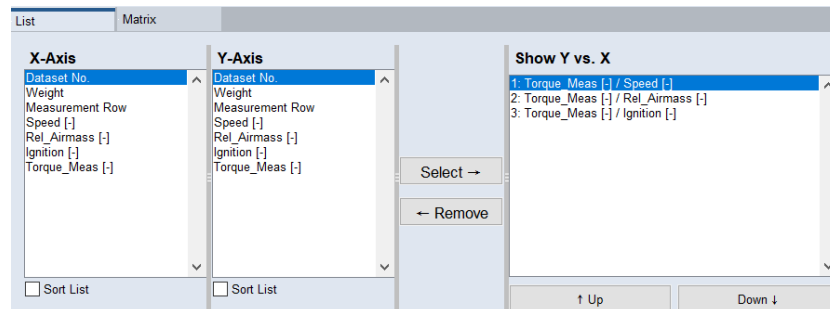
1. 散布図が表示された **Data** ウィンドウで、**View > Select Axes** を選択します。
"Select Axes" ウィンドウが開きます。
2. **Matrix** タブを開きます。



3. **Matrix** タブで、以下の図のように軸のペアを選択します。

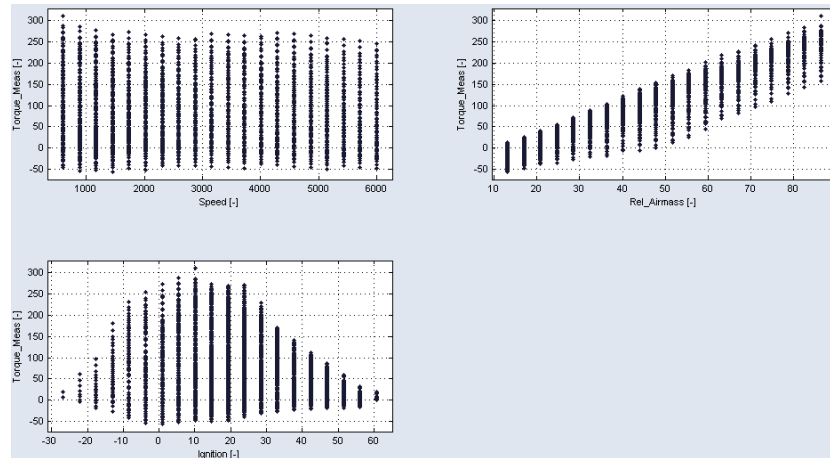


List タブでも、表示したい軸のペアを選択することができます。



4. 以下のいずれかを行います。
 - **Apply** をクリックします。
 - **OK** をクリックします。

⇒ 散布図に表示される軸が変更されます。**Select Axes** ウィンドウは閉じません。



外れ値を削除する

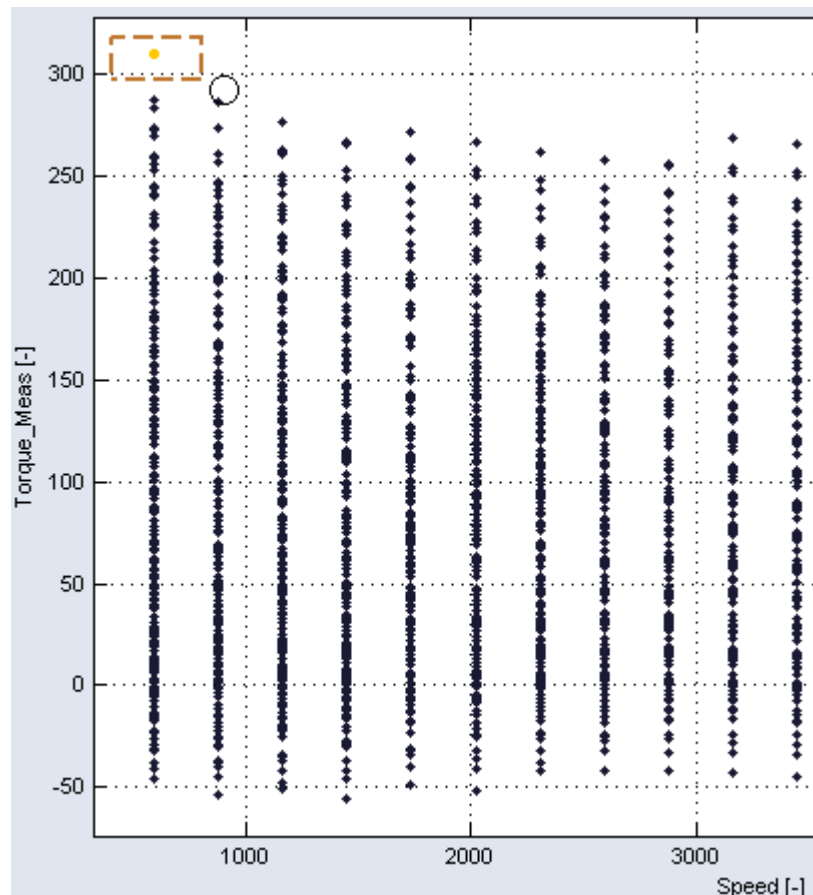
散布図では、外れ値を削除することができます。

1. 散布図内で外れ値を特定します。

このチュートリアルでは、**Torque_Meas over Speed** プロット内で、Torque_Meas > 300 となる左上の点を外れ値とします。

2. ツールバーで、 **Mouse selection in Plot with Rectangle** ボタンをクリックします。

3. プロット内をクリックして、外れ値を囲む線（四角形または投げ縄）を描画します。
線に囲まれた点が、すべての散布図において別色でマークされます。



4. 囲み線内でマウスを右クリックしてショートカットメニューを開き、**Mark Point** を選択します。

マークされたポイントが赤い丸で囲まれます。

5. **Extras > Delete Marked Points** を選択します。

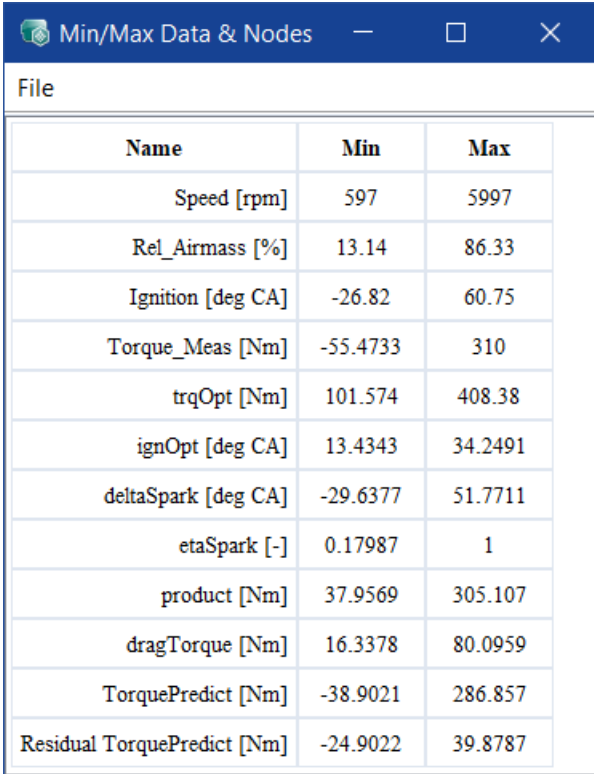
⇒ 外れ値が測定データから削除されます。

測定データの範囲を表示する

測定データから外れ値を削除した後は、測定ファイルの値が妥当な範囲内に収まっているかどうかをチェックします。

1. メインウィンドウで **Analysis > Show Data Min/Max** を選択します。

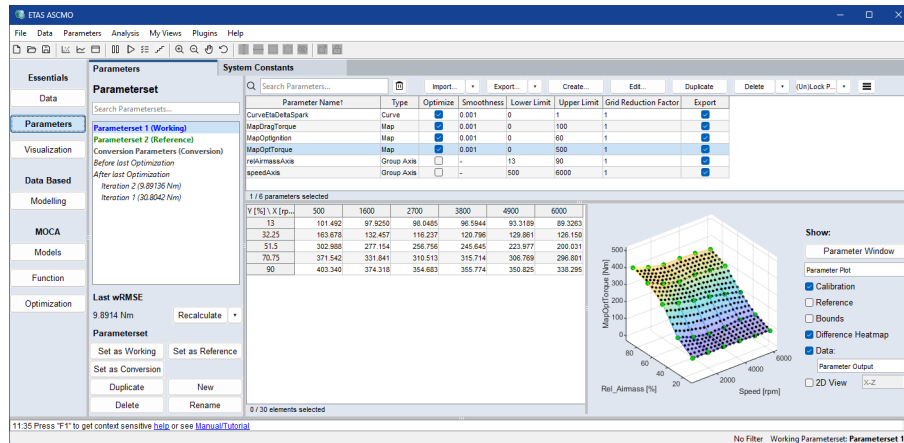
Min/Max Data & Nodes ウィンドウが開きます。



Name	Min	Max
Speed [rpm]	597	5997
Rel_Airmass [%]	13.14	86.33
Ignition [deg CA]	-26.82	60.75
Torque_Meas [Nm]	-55.4733	310
trqOpt [Nm]	101.574	408.38
ignOpt [deg CA]	13.4343	34.2491
deltaSpark [deg CA]	-29.6377	51.7711
etaSpark [-]	0.17987	1
product [Nm]	37.9569	305.107
dragTorque [Nm]	16.3378	80.0959
TorquePredict [Nm]	-38.9021	286.857
Residual TorquePredict [Nm]	-24.9022	39.8787

測定データをインポートして内容を確認したら、ノードにファンクションを追加します。そのためには、ナビゲーションペインの **Function** をクリックしてファンクション設定のステップに進みます（[ASCMO-MOCAユーザーインターフェースの要素（ページ75）](#)を参照）。関連する要素がメイン作業ウィンドウに表示され、ここでファンクションの保存、削除、編集が行えます（[ステップ5：ファンクションの構築（ページ110）](#)を参照）。

6.5 ステップ3 : パラメータ



このステップでは、パラメータをチェックして、必要に応じて適合を行います。ここに表示されるのは参照値のみで、最適化済みパラメータは表示されません。最適化ステップにおいて、最適化したパラメータを新しい参照値として定義することができます。

最適化実行時には、現在のマップとともに参照マップを表示したり、マップとカーブにデータポイントを表示したりすることもできます。

個々のグリッドポイント（格子点）を固定して最適化から除外することも可能です。

パラメータの作成や編集については、[ステップ5 : ファンクションの構築（ページ110）](#)を参照してください。

注記

このチュートリアルではパラメータは完全に定義されているので、このステップをスキップしてこのまま[ステップ6 : 最適化（ページ117）](#)に進むことができます。

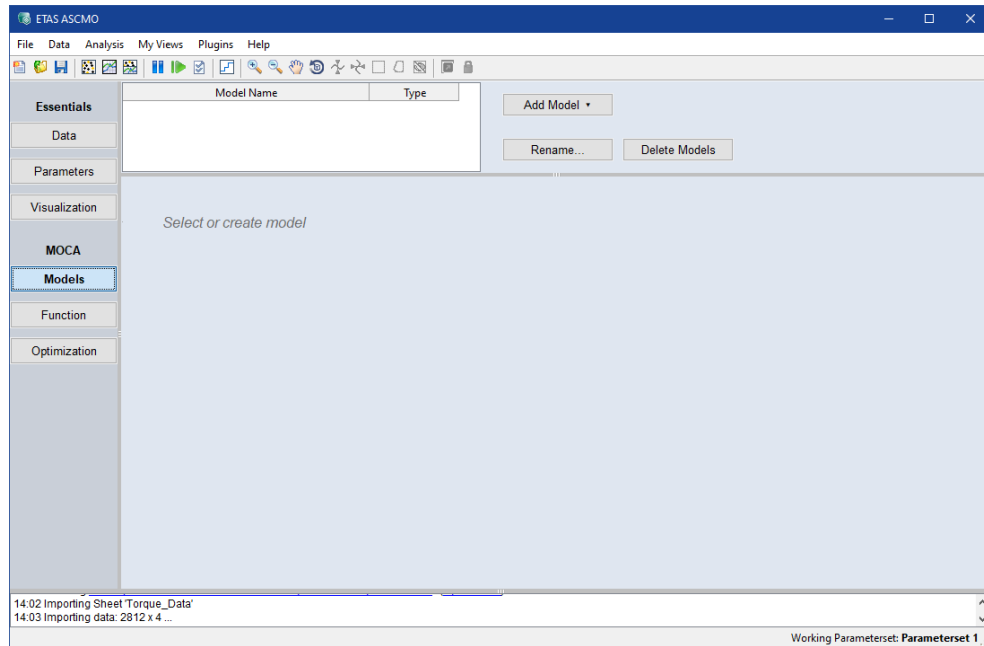
ご注意ください！

ファンクションパラメータのデータ範囲と制限値を実システム（テストベンチなど）の有効範囲まで拡大すると、システムが過負荷状態となり、損傷する可能性があります。

パラメータをエクスポートする際には、必ず、ASCMO-MOCAで設定されたデータ範囲と制限値が実システムに適していることを確認してください。

ある特定の適合と最適化の作業を実行する場合は、これらの値が前提条件と見なされます。

6.6 ステップ4：モデル



このステップでは、既存のSimulink®モデルをASCMO-MOCAにリンクさせて、パラメータ、入力、出力のマッピングを行うことができます。

注記

このチュートリアルでは、モデルのASCMO-MOCAへの組み込みは必要ありません。したがって、このステップを省略して[ステップ5：ファンクションの構築（ページ110）](#)に進むことができます。

このステップの作業を行うには、同じPCにSimulink®がインストールされている必要があります。デフォルトにおいてASCMO-MOCAは、最後にインストールされたMATLAB®/Simulink®バージョンを使用します。

Simulink®バージョンを選択する

1. メインウィンドウで **File > Options** を選択します。
Options ウィンドウが開きます。
2. **Simulink Version** ドロップダウンで、使用したいバージョンを選択します。
3. **OK** をクリックして、設定を確定します。


6.6.1 Simulink®モデルとスクリプトを追加する

ここでSimulinkモデルを追加し、[ステップ5：ファンクションの構築（ページ110）](#)においてそのモデルを「ファンクションノード」として選択することができます。Simulinkモデルを追加するには以下のように操作します。

Simulink®モデルとスクリプトを追加する

1. メイン作業領域で、**Add Model** をクリックして **Connect to Simulink Model** を選択します。

モデルリストに新しい行が追加され、モデルステップの下部に追加のオプションが表示されます。


2. **Simulink Model** フィールドで、最適化するSimulinkモデルのパスと名前を入力（または  ボタンで選択）します。

指定できるSimulinkモデルは、*.mdl（R2012aより前のバージョン）または*.slx（R2012a以降のバージョン）です。


3. <ENTER> を押すか、または別のフィールドをクリックします。

指定されたSimulinkモデルが存在しない場合は、警告が表示されます。以下のように操作します。

- i. 警告を確認して **OK** をクリックします。
- ii. Simulinkモデルの正しいパスと名前を指定します。

4. 必要に応じて、**Pre Load Script** フィールドに実行形式のMスクリプトのパスと名前を入力、または  ボタンで選択します。

このInitスクリプトはオプションで、Simulinkモデルの事前適合値を設定しておくことができます。その値は、**Parameter Mapping** テーブルに割り当てられるMATLABワークスペース変数の値として使用されます。

5. 必要に応じて、**Post Load Script** フィールドに別の実行形式のMスクリプトのパスと名前を入力、または  ボタンで選択します。

このスクリプトはオプションです。

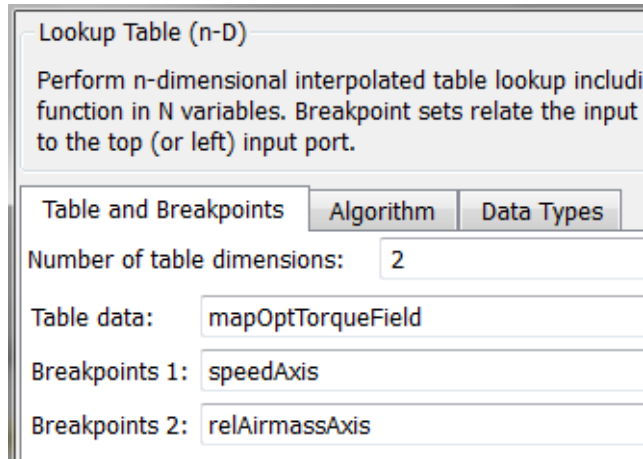
注記

パスとして %ProjectPath% を指定できます。すると、自動的にASCMO-MOCAプロジェクトのパスに置き換わります。

6.6.2 Simulink®パラメータをマッピングする

Parameters Mapping 領域では、プロジェクトのパラメータ（マップ、カーブ、スカラ）を Simulinkモデルのパラメータにマッピングすることができます。

ASCMO-MOCAでは、パラメータはMATLAB®ワークスペース変数として適合されます。これは以下の例のようなSimulinkマップになります。

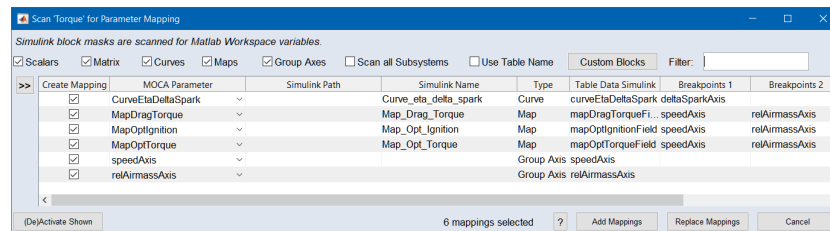


プロジェクトパラメータとSimulinkパラメータのマッピングは、自動的に（Simulink®モデルの検索とパラメータのマッピングを行う（下記）を参照）、または手動で（手動でパラメータをマッピングする（次ページ）を参照）行えます。

Simulink®モデルの検索とパラメータのマッピングを行う

Simulinkモデル内のパラメータを自動検索するには、以下のように操作します。

1. メインウィンドウの **Parameters Mapping** 領域で、**Scan Model** をクリックします。各変数名に対するルックアップテーブルやその他のブロックのブロックマスクが検索されます。結果が **Scan Model <model_name> for Parameter Mapping** ウィンドウに表示されます。に表示されます。



2. **Scalars**、**Matrix**、**Curves**、**Maps**、**Group Axes** チェックボックスをオン/オフして、各タイプのパラメータの表示/非表示を切り替えます。
3. リストのフィルタリング用検索文字列を **Filter** フィールドに入力し、<ENTER> を押します。

注記

フィルタ文字列では大文字と小文字は区別されません。ワイルドカードは使用できません。

Simulinkの名前またはパスに検索文字列が含まれるパラメータのみが表示されます。

4. **Create Mapping** 列で、マッピングする行をすべて選択します。
5. **Add Mappings** をクリックして、選択されているマッピングを **Parameters Mapping** リストに追加します。

Add Mappings では、**Parameters Mapping** リスト内の既存の内容はそのまま保持されます。既存の行が再度選択されると、その選択は無視され、ログウィンドウに警告メッセージが発行されます。

Replace Mappings では、**Parameters Mapping** リスト内の既存の内容が削除されます。

手動でパラメータをマッピングする

1. **Parameters Mapping** 領域の **MOCA Parameter** 列で、コンボボックスからパラメータを選択します。

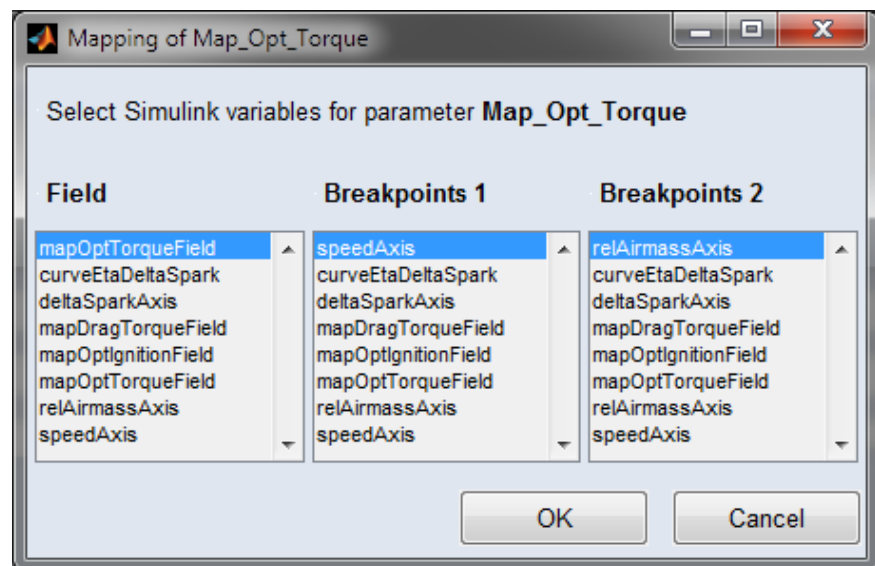
Create Parameter で新しいパラメータを作成するか、**Import Parameter** で DCMファイルをインポートすることができます。

2. **Table Data Simulink** 列と **Breakpoints <n> Simulink** 列で、ブロックマスクから変数名を入力します。

または

3. 以下のように操作します。
 - i. **Edit Mapping** をクリックします。

Mapping of <Parameter> ウィンドウが開きます。ここで Initスクリプトが実行されてモデルがロードされます。既存のMATLABワークスペース変数が表示されます。ASCMO-MOCAは自動的に名前の検索を行います。



- ii. **Field** 列で、パラメータ値またはテーブルデータを記述したMATLABワークスペース変数を選択します。
- iii. **Breakpoint <n>** 列で、テーブルの軸に使用するMATLABワークスペース変数を選択します。
- iv. **OK** をクリックして選択を確定し、**Mapping of <parameter>** ウィンドウを閉じ

ます。

選択した内容に従って、**Parameters Mapping** 領域内の **Table Data Simulink** 列と **Breakpoint <n> Simulink** 列が更新されます。

6.6.3 Simulink®入力をマッピングする

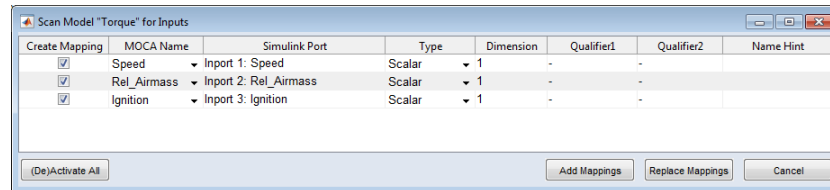
Simulink Inputs Mapping 領域で、インポート済みデータの列（ASCMO-MOCAプロジェクトのノード）をSimulinkモデル入力にマッピングすることができます。

Simulink®モデルの検索や入力のマッピングを行う

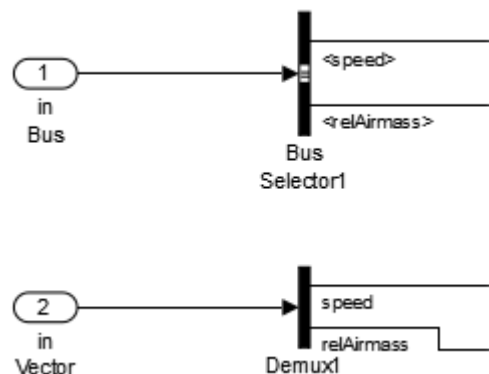
Simulinkモデル内の入力を自動検索するには、以下のように操作します。

1. **Simulink Inputs Mapping** 領域で、**Scan Model** をクリックします。

モデル内のInportブロックとFrom Workspaceブロックが検索されます。その結果が **Scan Model <model_name> for Inputs** ウィンドウに表示されます。



2. 必要に応じて、マニュアル操作で "Dimension" と "Qualifier" を入力します。
ここでは、バスとベクトルを入力とする以下のような2つの入力ポートについて説明します。



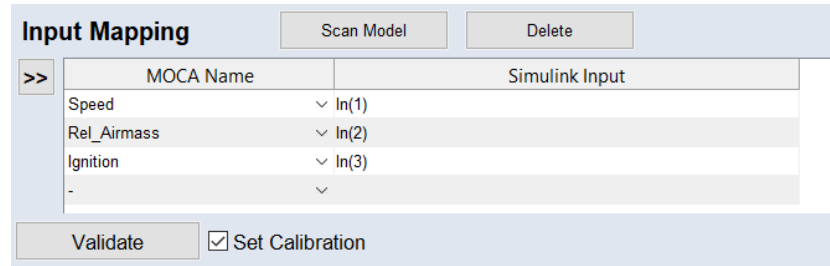
ASCMO-MOCAは、Simulinkの信号フローをたどってこのようなバス信号やベクトル信号を探します。見つからなかった場合は、手動操作で "Type"、"Dimension"、"Qualifier" を選択する必要があります。

3. **Create Mapping** 列で、マッピングする行をすべて選択します。
4. **Add Mappings** をクリックして、選択されているマッピングを **Simulink Inputs Mapping** リストに追加します。

Add Mappings では、**Simulink Inputs Mapping** リスト内の既存の内容はそのまま保持されます。既存の行が再度選択されると、その選択は無視され、ログウィンドウに警告メッセージが発行されます。

Replace Mappings では、**Simulink Inputs Mapping** リスト内の既存の内容が削除されます。

Add/Replace Mappings をクリックすると、**Simulink Inputs Mapping** テーブルにマッピング情報が表示されます。



Speed | In(1) という表記は、1番目のSimulink入力ポートデータに Speed が渡されることを意味しています。

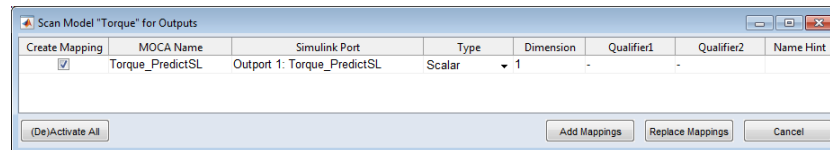
この表記規則については、オンラインヘルプを参照してください。

6.6.4 Simulink® 出力をマッピングする

Simulink Outputs Mapping 領域では、シミュレーション出力をASCMO-MOCAに割り当てることができます。サポートされているブロックタイプはOutportとToWorkspaceです。

Simulink®モデルの検索と出力のマッピングを行う

1. **Simulink Outputs Mapping** 領域で、**Scan Model** をクリックします。
モデル内のOutportブロックとToWorkspaceブロックが検索されます。その結果が **Scan Model <model_name> for Outputs** ウィンドウに表示されます。



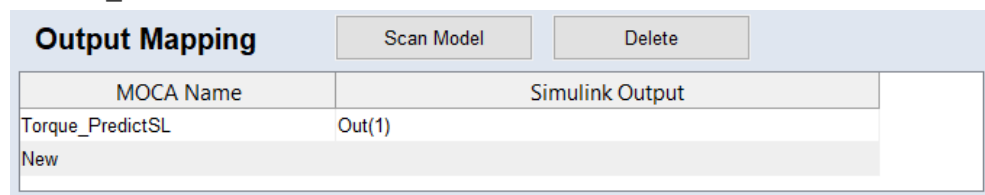
2. 必要に応じて、マニュアル操作で "Dimension" と "Qualifier" を入力します。
3. **Create Mapping** 列で、マッピングする行をすべて選択します。
4. **Add Mappings** をクリックして、選択されているマッピングを **Simulink Outputs Mapping** リストに追加します。

Add Mappings では、**Simulink Outputs Mapping** リスト内の既存の内容はそのまま保持されます。既存の行が再度選択されると、その選択は無視され、ログウィンドウに警告メッセージが発行されます。

Replace Mappings では、**Simulink Outputs Mapping** リスト内の既存の内容が削除されます。

例

Torque_PredictSL | Out(1) と表記すると、最適化の際に、Simulink出力ポートを Torque_PredictSL という名前で使用することができます。



この表記規則については、オンラインヘルプを参照してください。

6.6.5 Simulink®モデルの検証と使用

注記

Simulink®モデルを実行するには、有効なライセンスとともに所定のバージョンのSimulink®がPCにインストールされている必要があります。

パラメータ、入力、出力のマッピングを行った後は、以下のステップを実行します。

6.6.5.1 Simulink®モデルを検証する

Simulink®モデルを検証する

1. メイン作業領域の最下部の **Validate** をクリックします。
検証が実行されます。

検証においては、以下の処理が行われます。

- － Initスクリプトを開始する（任意）
- － MATLAB検索パスにモデルパスを追加する
- － Simulinkモデルを開く
- － ポストロードスクリプトを開始する
- － すべてのパラメータがワークスペース変数として存在しているかをチェックする
- － マッピング情報に従って入力ポートと出力ポートを置き換える
- － サブセット規模のデータを使用してシミュレーションを開始する
- － 出力値を読み取る

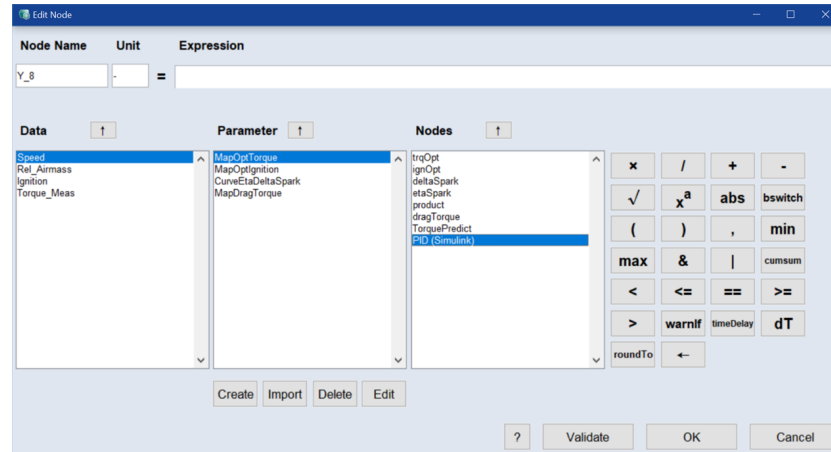
エラーが発生すると、エラーレポートが発行されます。テストが成功すると、成功メッセージが表示されます。最適化処理でSimulinkモデルを使用できるようになります。

最適化処理にモデルを使用するには、あらかじめファンクション内にモデルを追加しておく必要があります。

Simulink®モデルを使用する

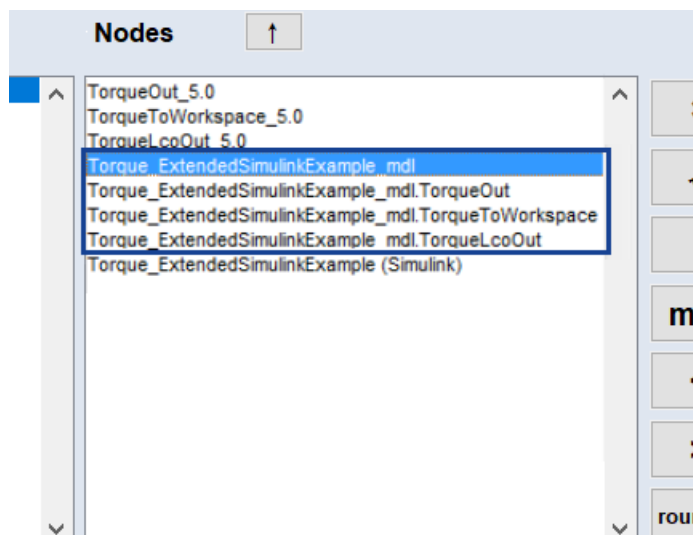
最適化処理にモデルを使用するには、あらかじめファンクション内にモデルを追加しておく必要があります。以下のように操作します。

1. ナビゲーションペインで **Function** をクリックします。
"Function" ペインが開きます。
2. 新しいノードを追加します。操作方法はオンラインヘルプを参照してください。
Simulinkモデルが **Edit Node** ウィンドウの **Nodes** 領域に表示されます。



3. 式 (Expression フィールド) にモデルを挿入します。
 式を %Torque%(Speed, Rel_Airmass, Ignition) のように編集します。さらに Torque_md1 という名前を **Node Name** フィールドに入力します。
4. 必要に応じて、ノードを検証します。
5. **OK** をクリックします。
 モデル用ノードが **Main Function Nodes** テーブルに追加されます。
 さらにモデル出力用の Torque_md1.torque_Simulink_Model という名前のノードが作成されます。
6. 必要に応じて、さらにファンクションノードを作成します。
 詳細は[ステップ5 : ファンクションの構築 \(ページ110\)](#) を参照してください。

複数の出力を持つ Simulink モデルを使用する場合は、各モデル出力に対してファンクションノードが1つずつ作成されます。各ノードは、最適化目標と式に使用することができます。



Main Function Nodes テーブルで使用されるデフォルト表記は以下の図の枠で囲んだ部分のようになります。ASCMO-MOCAの旧バージョンで使用されていた表記 (以下の図の Torque*_5.0 ...の行) も有効です。

Node	Insert	Delete	Edit	Validate All	Symbolic Regression...	De
Main Function Nodes						
1	Torque_ExtendedSimulinkExample_md[-] = Torque_ExtendedSimulinkExample(Speed, Rel_Airmass, Speed, Rel_Airmass, Spee...					
2	TorqueOut[-] = Torque_ExtendedSimulinkExample_md.TorqueOut					
3	TorqueToWorkspace[-] = Torque_ExtendedSimulinkExample_md.TorqueToWorkspace					
4	TorqueLcoOut[-] = Torque_ExtendedSimulinkExample_md.TorqueLcoOut					
5	TorqueOut_5.0[-] = Torque_ExtendedSimulinkExample_md(:, 1)					
6	TorqueToWorkspace_5.0[-] = Torque_ExtendedSimulinkExample_md(:, 2)					
7	TorqueLcoOut_5.0[-] = Torque_ExtendedSimulinkExample_md(:, 3)					
8	New...					

図6-1: 複数の出力を持つSimulinkモデルの表記

注記

複数の出力を持つSimulink®モデルをASCMO-MOCAに実装したサンプルプロジェクト
 Torque_ExtendedSimulinkExample.moca が以下のディレクトリに用意されています。
 <installation>\Example\Moca (<installation> のデフォルト:
 †: <installation> = C:\Program Files\ETAS\ASCMO x.x)

Simulink®モデルを最適化する

1. ナビゲーションペインの **Optimization** をクリックします。
2. **Optimization Criteria** 領域で、最適化が必要な出力（1つまたは複数）を選択します。

Optimization Options

Optimization Algorithm: Default

Number of Iterations: 20

Tolerance/Accuracy: 1e-09

Multistart: 1

Use Sequence

Optimization Criteria

min (((TorquePredict - Torque_Meas)² * 1) -> RMSE: 1.6806

+ (Torque_md1.to... - Torque_Meas)²

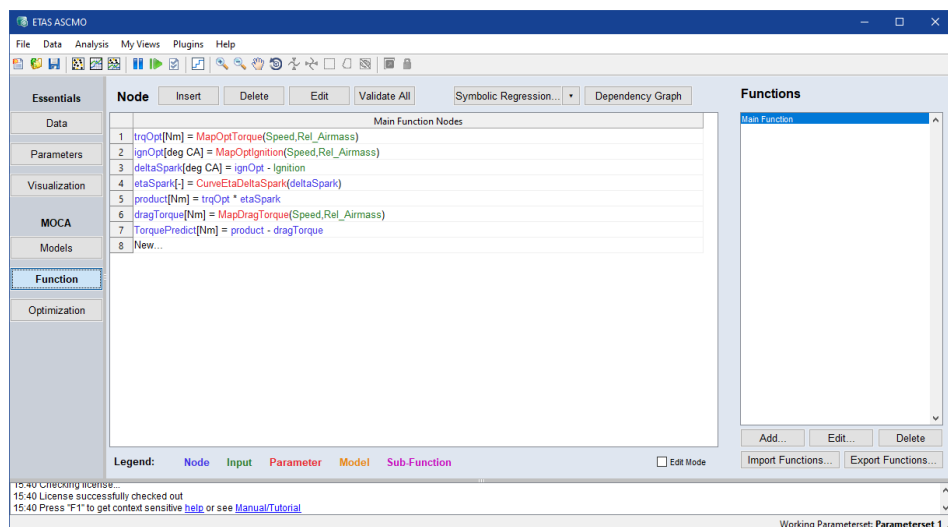
- trqOpt
- ignOpt
- deltaSpark
- etaSpark
- product
- dragTorque
- TorquePredict
- Torque_md1
- Torque_md1.torque_Simulink_Model**
- Remove

Local

3. **Optimize** ボタンをクリックします。

Simulinkモデルの最適化が開始されます。最適化処理の状態（反復回数やRMSEなど）はログウィンドウに表示されます。算出されたRMSEは各最適化目標の下に表示されます。

6.7 ステップ5：ファンクションの構築



測定データを読み取って妥当性をチェックした後は、このチュートリアルでモデリングされるトルクセンサファンクションのセットアップを開始します。使用できる演算子については [ファンクションノード用の演算子（ページ43）](#) を参照してください。

ご注意ください！

ファンクションパラメータのデータ範囲と制限値を実システム（テストベンチなど）の有効範囲まで拡大すると、システムが過負荷状態となり、損傷する可能性があります。

パラメータをエクスポートする際には、必ず、ASCMO-MOCAで設定されたデータ範囲と制限値が実システムに適していることを確認してください。

ある特定の適合と最適化の作業を実行する場合は、これらの値が前提条件と見なされます。

6.7.1 ファンクションのモデリング

このチュートリアルでは、以下のようなエンジントルクモデルのファンクションを構築します。

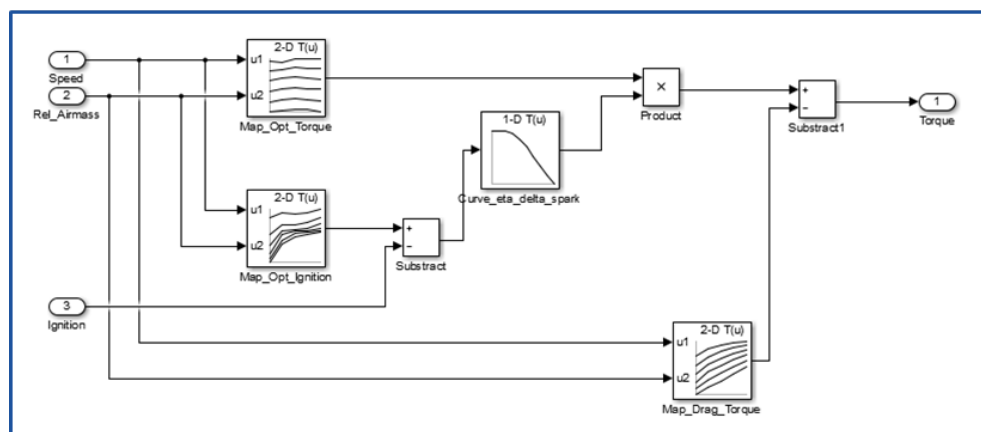


図6-2: モデリングされるファンクションの構造

注記

ファンクションは必ず左から右へとセットアップします。

モデリングされる関数の構造（前ページ）に示すモデル関数には以下の入力が含まれています。

- 1 - Speed
- 2 - Rel_Airmass
- 3 - Ignition

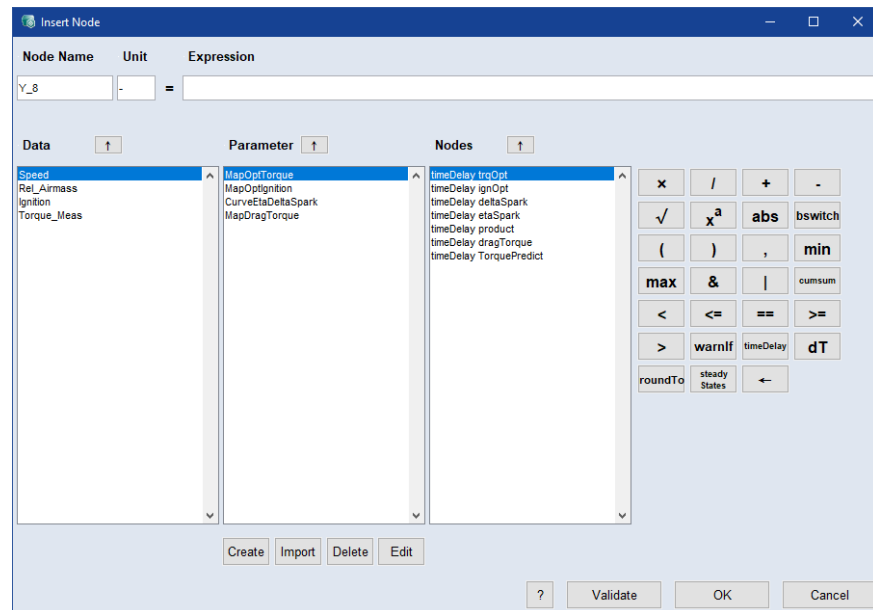
これらの入力に加えて、ステップ1では計測されたモデル出力 Torque_Meas もインポートしました（ステップ1：データのインポート（ページ83）を参照）。これらの値は、最適化の際に、測定値と関数予測 TorquePredict の偏差を最小化するための参照値として使用されます。

最初のノードを作成して編集する

以下のように操作して、最初のノード trqOpt を作成して編集します。

1. 以下のいずれかを行います。
 - **Main Function Nodes** テーブル内の **New** をクリックします。
 - **Insert** をクリックします。

Insert Node ウィンドウが開きます。**Data** 領域には、インポートされたすべてのチャンネルが表示されます。



2. **Node Name** フィールドに trqOpt という名前を入力します。
3. 必要に応じて単位を入力します。
単位は表示用にものみ使用され、パラメータの適合には影響しません。
4. パラメータ MapOptTorque を作成します。作成方法はパラメータを作成して編集する（次ページ）を参照してください。
5. 関数ノードの式を定義します。
 - i. **Parameter** 領域で、パラメータ MapOptTorque を選択します。
 - ii. **↑** をクリックします。
パラメータが **Expression** フィールドに挿入されます。

Node Name	Unit	Expression
trqOpt	-	= %Map_Opt_Torque%(%Speed%, %Rel_Airmass%)

- iii. **Validate** をクリックして、新しいノードの妥当性をチェックします。
6. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。
- ⇒ このノードが **Function Nodes** テーブルの1行目に表示されます。

Function Nodes	
1	trqOpt[-] = MapOptTorque(Speed, Rel_Airmass)

パラメータを作成して編集する

パラメータ MapOptTorque を作成して編集するには、以下のように操作します。

1. **Edit Node** ウィンドウで、**Parameter** フィールドの下にある **Create** ボタンをクリックします。

Create Parameter ウィンドウが開きます。

2. パラメータの情報を入力します。

MapOptTorque の場合は以下の値を使用します。

Parameter MapOptTorque

Name :

Parameter Map

Type :

Value Bounds: 0 - 500

Input 1: Speed

Input 2: Rel_Airmass

Breakpoints X : Begin/End; Begin = 500; End = 6000; Count = 6

Breakpoints Y : Begin/End; Begin = 10; End = 90; Count = 6

Extrapolation : Clip

 **注記**

Use Range をクリックすると、X座標とY座標の値の範囲がチャンネルの最小値と最大値に合わせて自動的に調整されます。

3. **OK** をクリックします。

パラメータが作成されます。作成されたパラメータは **Parameter** 領域に表示されます。

次に、[ノード ignOpt を作成して設定します](#)。

ノード ignOpt を作成して編集する

以下のように操作して、2番目のノード ignOpt を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 ignOpt を入力します。
このノードには、パラメータ MapOptIgnition が必要です。
3. パラメータ MapOptIgnition を作成し ([パラメータを作成して編集する \(前ページ\)](#) を参照)、以下の値を設定します。

Parameter	MapOptIgnition
Name :	
Parameter	Map
Type :	
Value Bounds:	0 - 60
Input 1:	Speed
Input 2:	Rel_Airmass
Breakpoints X :	Begin/End; Begin = 500; End = 6000; Count = 6
Breakpoints Y :	Begin/End; Begin = 10; End = 90; Count = 6
Extrapolation :	Clip

4. ファンクションノードの式を以下のように定義します。

Node	Unit	Expression
ignOpt	-	= %MapOptIgnition%(%Speed%,%Rel_Airmass%)


5. ノードの妥当性をチェックします。
6. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

次に、[ノード deltaSpark を作成して設定します](#)。


ノード deltaSpark を作成して編集する


以下のように操作して、3番目のノード deltaSpark を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 deltaSpark を入力します。
このノードには、ノード ignOpt と入力 Ignition が必要です。

3. **Node** 領域で ignOpt ノードを選択して、 をクリックします。

ignOpt ノードが **Expression** フィールドに挿入されます。

4.  をクリックして減算記号を追加します。

5. **Data** 領域で Ignition を選択して、 をクリックします。

Ignition チャンネルが **Expression** フィールドに追加されます。

6. 式が以下のようになったことを確認してください。

Node	Unit	Expression
deltaSpark	-	= %ignOpt% - %Ignition%

7. ノードの妥当性をチェックします。
8. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

次に、ノード etaSpark を作成して設定します。

ノード etaSpark を作成して編集する

以下のように操作して、4番目のノード etaSpark を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 etaSpark を入力します。
このノードには、パラメータ CurveEtaDeltaSpark が必要です。
3. パラメータ CurveEtaDeltaSpark を作成し（[パラメータを作成して編集する（ページ112）](#)を参照）、以下の値を設定します。

Parameter	CurveEtaDeltaSpark
Name :	
Parameter	Curve
Type :	
Value Bounds:	0 - 1
Input 1:	deltaSpark
Breakpoints X :	Begin/End; Begin = -10; End = 55; Count = 10
Extrapolation :	Clip

4. ファンクションノードの式を以下のように定義します。

Node	Unit	Expression
etaSpark	-	= %CurveEtaDeltaSpark%(%deltaSpark%)

5. ノードの妥当性をチェックします。
6. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

次に、ノード product を作成して設定します。

ノード product を作成して編集する

以下のように操作して、5番目のノード product を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 product を入力します。
このノードには、ノード trqOpt と etaSpark が必要です。
3. **Node** 領域で trqOpt ノードを選択して、 をクリックします。
trqOpt ノードが **Expression** フィールドに挿入されます。
4.  をクリックして除算記号を追加します。
5. etaSpark ノードを式に追加します。
6. 式が以下のようなようになったことを確認してください。

Node	Unit	Expression
product	-	= %trqOpt% .* %etaSpark%

7. ノードの妥当性をチェックします。
8. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

次に、ノード dragTorque を作成して設定します。

ノード dragTorque を作成して編集する

以下のように操作して、6番目のノード dragTorque を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 dragTorque を入力します。
このノードには、パラメータ MapDragTorque が必要です。
3. パラメータ MapDragTorque を作成し（[パラメータを作成して編集する（ページ 112）](#)を参照）、以下の値を設定します。

Parameter Name :	MapDragTorque
Parameter Type :	Map
Value Bounds:	0 - 100
Input 1:	Speed
Input 2:	Rel_Airmass
Breakpoints X :	Begin/End; Begin = 500; End = 6000; Count = 6
Breakpoints Y :	Begin/End; Begin = 10; End = 90; Count = 6
Extrapolation :	Clip

4. ファンクションノードの式を以下のように定義します。


Node	Unit	Expression
dragTorque	-	= %MapDragTorque%(%Speed%,%Rel_Airmass%)

5. ノードの妥当性をチェックします。
6. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

次に、ノード `TorquePredict` を作成して設定します。

ノード `TorquePredict` を作成して編集する

以下のように操作して、最後のノード `TorquePredict` を追加して編集します。

1. **Edit Node** ウィンドウを開きます。
2. ノード名 `TorquePredict` を入力します。
このノードには、ノード `product` と `dragTorque` が必要です。
3. **Node** 領域で `product` ノードを選択して、 をクリックします。
`product` ノードが **Expression** フィールドに挿入されます。
4. 減算記号を追加します。
5. `dragTorque` ノードを式に追加します。
6. 式が以下ようになったことを確認してください。

Node	Unit	Expression
TorquePredicted	-	= %product% - %dragTorque%

7. ノードの妥当性をチェックします。
8. **OK** をクリックしてノードを追加し、**Edit Node** ウィンドウを閉じます。

`TorquePredict` ノードが作成されると、最適化の対象となる物理的トルクモデルのファンクションがすべて作成されたこととなります。**Function Nodes** フィールドは以下ようになります。

Function Nodes	
1	<code>trqOpt[-] = Map_Opt_Torque(Speed, Rel_Airmass)</code>
2	<code>ignOpt[-] = MapOptIgnition(Speed, Rel_Airmass)</code>
3	<code>deltaSpark[-] = ignOpt - Ignition</code>
4	<code>etaSpark[-] = CurveEtaDeltaSpark(deltaSpark)</code>
5	<code>product[-] = trqOpt * etaSpark</code>
6	<code>dragTorque[-] = MapDragTorque(Speed, Rel_Airmass)</code>
7	<code>TorquePredict[-] = product - dragTorque</code>
8	New...

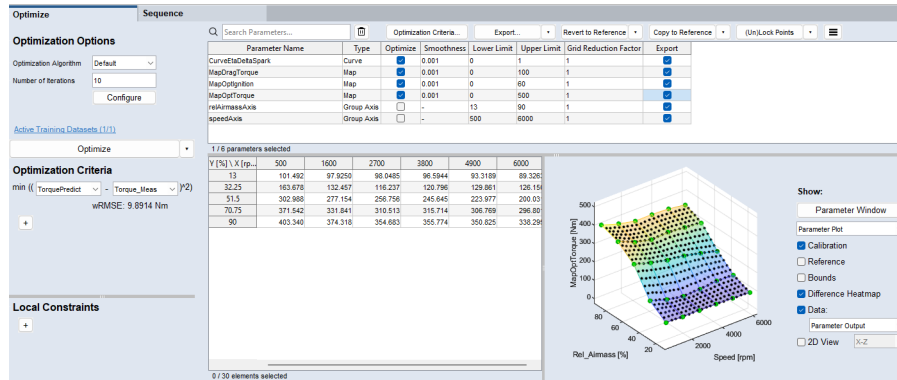
① 注記

メイン作業ウィンドウの **Edit Mode** オプションをオンにすると、**Function Nodes** テーブル内のノードの定義を直接変更することができます。データ、パラメータ、ノードの名前は%で囲みます。

Function Nodes	
1	trqOpt[-] = %Map_Opt_Torque%(%Speed%, %Rel_AirMass%)

次のステップ（ステップ3：パラメータ（ページ99））では、作成されたパラメータをチェックして調整することができます。

6.8 ステップ6：最適化



最適化を始める前に、最適化アルゴリズムを選択する必要があります。最適化アルゴリズム（ページ51）を参照して、目的に応じて最適なアルゴリズムを選択するようにしてください。

アルゴリズムを選択したら、**Optimizer Options** **Configure** をクリックしてオプションをカスタマイズします。各アルゴリズム用のオプションについては、**オプティマイザのオプション**（ページ54）を参照してください。

① 注記

測定値と予測値との差異が、満足できる範囲まで最小化されるまで（変数RMSEおよびR2（ページ25）を参照）、上記の目標や限界を状況に合わせて適宜調整する必要があります。

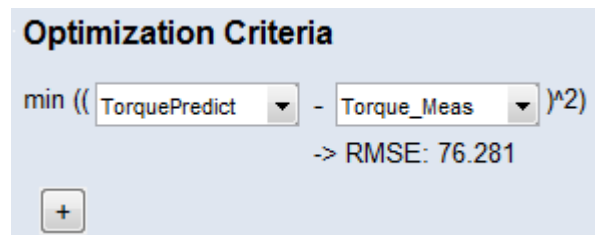
Optimization Options 領域で、いくつかのパラメータを設定します。詳細は6.8 ステップ6：最適化（上記）を参照してください。

Optimization Criteria 領域では最適化目標を定義します。1番目のドロップダウンでファンクションノードを選択し、2番目のドロップダウンでデータチャンネル（または "0"）を選択します。オプティマイザは、これら2つの値の二次偏差を最小にするパラメータセットを見つけるを試みます。

チュートリアルで使用する例をシンプルにしておくため、ここでは **Local Constraints** 領域は使用しません。最適化問題によっては、これらの制限を用いることにより、最適化を特定の方向に導くことができます。

最適化の準備を行う

1. **Optimization Options** 領域で、以下を行います。
 - i. **Number of Iterations** フィールドにステップ数 40 を入力します。
 - ii. **Multistart** フィールド (**Optimizer Options** ▾ **Configure**) の値を大きくして、オプティマイザがローカル最小値においてスタックしてしまうのを防ぎます。
2. **Optimization Criteria** 領域で、以下を行います。



- i. 1番目のドロップダウンリストから、ファンクション出力 **TorquePredict** を選択します。
- ii. 2番目のドロップダウンリストから、インポートされたデータチャンネル **Torque_Meas** を選択します。

RMSEは、パラメータがフラットの状態では大きくなります ([変数RMSEおよびR2 \(ページ25\)](#) を参照)。1回目の最適化の後にRMSEは大幅に小さくなります。

- Add a new Optimization criterion** ボタンをクリックすると、複数の最適化目標を定義することができます。

最適化を実行する

ご注意ください！

誤った適合データにより物的損傷の生じる恐れがあります。

ASCMO-MOCAモデルから得られた適合データの使い方を誤ると、エンジンやテストベンチの損傷を招く可能性があります。

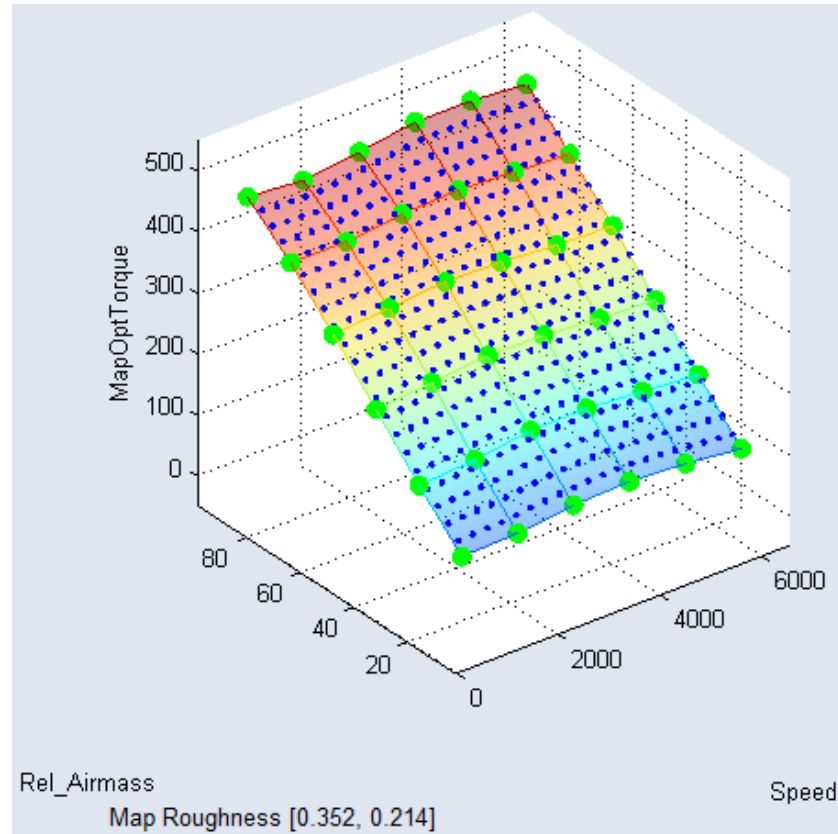
最適化を行った後は、エクスポートを行う前に、残差解析機能を使って測定データとモデルが作成したデータとを比較してください。この機能を利用するには、Analysis > Residual Analysis > Training and Test Data > Absolute Error Analysis を選択します。

[最適化を実行する \(上記\)](#)、[パラメータステップ](#)または[最適化ステップ](#)のエクスポートオプション、および[6.9 ステップ7：エクスポート \(次ページ\)](#) を参照してください。

最適化の準備を行った後は、最適化を実行します。

1. **Optimize** をクリックします。

オプティマイザがパラメータの最適化を開始します。最適化処理の状態 (反復回数やRMSEなど) はログウィンドウに表示されます。最適化終了後、RMSEは最適化目標の下に表示されます。マップの表示は結果に応じて調整されます。



これで、このチュートリアルでの最適化は終了です。**Optimization** ステップにおけるその他のオプション操作については、オンラインヘルプInstructions (Optimization Step) を参照してください。

次のステップ（ステップ7：エクスポート（下記））では、最適化されたパラメータを以降の処理のためにエクスポートします。

6.9 ステップ7：エクスポート

ご注意ください！

誤った適合データにより物的損傷の生じる恐れがあります。

ASCMO-MOCAモデルから得られた適合データの使い方を誤ると、エンジンやテストベンチの損傷を招く可能性があります。

最適化を行った後は、エクスポートを行う前に、残差解析機能を使って測定データとモデルが作成したデータとを比較してください。この機能を利用するには、Analysis > Residual Analysis > Training and Test Data > Absolute Error Analysis を選択します。

最適化を実行する（前ページ）、パラメータステップまたは最適化ステップのエクスポートオプション、および6.9 ステップ7：エクスポート（上記）を参照してください。

このステップでは、作成して最適化したパラメータをエクスポートします。パラメータは各種フォーマットでエクスポートでき、機能制限付きのランタイム環境ASCMO-MOCA Runtime用にプロジェクトを保存することができます。

パラメータをエクスポートする

1. **Optimization** ステップで、**Export** ボタンをクリックします。
Export Parameters ウィンドウが開きます。
2. このウィンドウで、エクスポートファイルのパスと名前を入力、または選択します。
3. **Save as type** ドロップダウンリストからエクスポートフォーマットを選択します。
4. **保存** をクリックします。
⇒ 指定のフォーマットでパラメータがエクスポートされます。

選択したパラメータをエクスポートする

1. **Optimization** または **Parameters** ステップで、エクスポートしたいパラメータの **Export** 列のチェックボックスをオンにします。
複数のパラメータを選択し、ショートカットメニューを使用してエクスポートフラグをセット／クリアすることもできます。
選択したパラメータに対応するグループ軸は、自動的にエクスポートされます。
2. **Export** をクリックします。
Export Parameters ウィンドウが開きます。
3. このウィンドウで、エクスポートファイルのパスと名前を入力、または選択します。
4. **Save as type** ドロップダウンでエクスポートフォーマットを選択します。
5. **保存** をクリックします。
⇒ 指定のフォーマットで選択されたパラメータがエクスポートされます。

プロジェクトをASCMO-MOCA Runtime用にエクスポートする

ASCMO-MOCA Runtime用にエクスポートされたプロジェクトは暗号化され、プロジェクト内のファンクションの表示や編集は行えません。

1. メインメニューから **File > Export to MOCA-Runtime** を選択します。
Export to MOCA-Runtime ウィンドウが開きます。
2. エクスポートされたプロジェクトから最適化シーケンスを表示できるようにするには、**Show Sequence** をオンにします。
このオプションをオンにすると、ASCMO-MOCA Runtimeにおいてシーケンスを表示して編集することができます。**Show Sequence** をオフにするとシーケンスは表示できません。
3. エクスポートされたプロジェクトをASCMO-MOCA RuntimeのPコードバージョンから開けるようにするには、**Allow opening in MATLAB (MOCA-Runtime p-Code)** をオンにします。
Allow opening in MATLAB (MOCA Runtime p-Code) をオンにすると、エクスポートされるプロジェクトは別の鍵を用いて暗号化されます。オフにすると、プロジェクトはASCMO-MOCA RuntimeのPコードバージョンで開くことができません。そのようなプロジェクトは、スタンドアロンバージョンのASCMO-MOCA Runtimeがないと、開くことができません。
4. **Export** をクリックして続行します。

Export MOCA project to MOCA-Runtime ウィンドウが開きます。ファイルフォーマットとして `*.moca_runtime` が選択されている状態となっており、これを変更することはできません。

5. ファイルの保存先と名前を指定します。

6. **保存** をクリックします。

プロジェクトがASCMO-MOCA Runtime用プロジェクトとして保存されます。

7 お問い合わせ先

テクニカルサポート

各国支社の営業やテクニカルサポートについての情報は、以下のETASのウェブサイトをご覧ください。

www.etas.com/hotlines

ASCMOに関する個別のお問い合わせは、以下の専任サポートチームまでご連絡ください。

ascmo.support.de@etas.com

ETASは各製品のトレーニングを提供しています：

www.etas.com/academy



ETAS本社

ETAS GmbH

Borsigstraße 24	Phone :	+49 711 3423-0
70469 Stuttgart	Fax :	+49 711 3423-2106
Germany	Internet :	www.etas.com

用語集

さ

残差 (Residuum)

測定データとファンクション出力との差を指します。ASCMO-MOCAでは、相対残差、絶対残差、スチューデント化残差に分類されます。

残差 (Residual)

測定データとファンクション出力との差を指します。ASCMO-MOCAでは、相対残差、絶対残差、スチューデント化残差に分類されます。

し

システム定数

値の変更ができないコンテナです。これに対するものが「変数」です。

す

スカラ

0次元の適合パラメータです。

そ

粗度 (Roughness)

カーブ/マップ/キューブのグリッドポイント間の勾配の変化を表すものです。

と

トレランス

一般的には、その値を超えるとソルバの繰り返しが停止する「しきい値」です。

ふ

ファンクション

物理モデルの記述に必要なエレメントを定義したものです。

プロジェクト

一連のファンクションとパラメータの作成と最適化は「プロジェクト」の単位で行われます。プロジェクトは、保存してロードすることができます。ASCMO-MOCAの1つのインスタンスで、1つのプロジェクトを開くことができます。

へ

RMSE

RMSE（平均二乗誤差）は、モデル予測値と、モデリングされたオブジェクトの実測値との差異です。各差異は、「残差」と呼ばれます。

平均二乗誤差 (Root Mean Square Error)

RMSE (平均二乗誤差) は、モデル予測値と、モデリングされたオブジェクトの実測値との差異です。各差異は、「残差」と呼ばれます。



図4-1: "* Data" ウィンドウ	21
図4-2: "Data and Nodes" ウィンドウ	22
図4-3: "Histogram" ウィンドウ	23
図4-4: "Residuals over Inputs" ウィンドウ	24
図4-5: "Measured vs. Predicted" ウィンドウ	24
図4-6: 定常状態の可視化：暗い青の線が明るい青の線の平均値です	42
図4-7: 正規化されたパラメータ感度	65
図4-8: アルゴリズムの詳細（関数同定問題 - Symbolic Regression）	69
図5-1: ASCMO-MOCAのメインウィンドウの構成	75
図5-2: ログウィンドウに表示される情報の例（(a)：オンラインヘルプへのリンク、(b)：ユーザーガイドへのリンク）	79
図6-1: 複数の出力を持つSimulinkモデルの表記	108
図6-2: モデリングされるファンクションの構造	110

式

式4-1: 平均二乗誤差 (RMSE)	25
式4-2: 残差二乗和 (SSR)	25
式4-3: 決定係数R ² - ここでSSTは以下の式で表されます。	25
式4-4: 二乗の総和 (SST)	25
式4-5: 最適化手法	51
式4-6: カーブの粗度 r	62
式4-7: マップの粗度	62
式4-8: 平滑化係数S _i	63

索引

A	
ASCMO	
Pコードバージョン	15
ASCMOモデル	40-41
E	
ETAS	
お問い合わせ先	122
F	
FMUモデル	40
M	
Matlab/Simulink	
バージョンの選択	100
MOCA	
アンインストール	17
起動	81
MOCA Runtime	
～にエクスポート	120
Pコードバージョンへのエクスポート	120
MOCAの起動	81
P	
Pコードバージョン	120
R	
R ²	25
RMSE	25
S	
Simulinkモデル	100
最適マイザ	108
追加	100
あ	
圧縮モデル	30
い	
インストール	
MOCAのアンインストール	17
ディレクトリ	14
ファイル	14
ライセンス許諾書	13
インポート	83
コンフィギュレーションのロード	88
コンフィギュレーションの保存	88
測定データ	83
データ表示	84
実行	88
お	
応用分野	18
ASCMO MOCA	18
ASCMO MOCA Runtime	19
重み	
データセット	92
データポイント	91
お問い合わせ先	122
か	
感度	
パラメータの～	64-65
こ	
誤差の分析	
残差分析	22
スチューデント化誤差	22
絶対誤差	22
相対	22
データ	21
ファンクションノード	21
コンセプト	18
パラメータの最適化	26
入力データの評価	20
コンフィギュレーションファイル	
ロード	88
保存	88
さ	
最適化	50,64,117
Simulinkモデル	108
シーケンス	64
パラメータ	26
最適化手法	
説明	51
最適化目標	63
作業手順	
データのインポート	83
データの分析	95
パラメータのエクスポート	119
パラメータの最適化	117
ファンクションの構築	110
モデル	100
散布図	
軸の選択	95

し		データポイント	
シーケンス		削除	94
MOCA Runtime内の	120	重み	91
最適化	64	重み付け	92
最適化用	120	編集	93
システム定数	31	データ品質	
そ		向上	21
関連		評価	21
パラメータの～	64-65	に	
測定データ		入力	
インポート	88	関連性	21,86
チャンネルの削除	94	入力データ	
チャンネルを変数に割り当てる	89	向上	20
妥当性チェック	84	評価	20
変数への割り当ての解除	90	の	
要件	81	ノード	
測定ファイル		算術演算子	43
インポート	83	追加	111
フォーマット	81	は	
粗度	62	はじめに	6
カーブの～	62	外れ値	
マップの～	62	削除	97
た		パラメータ	26
妥当性チェック	84	3Dキューブ	30
ち		4Dキューブ	30
チュートリアル	80	Simulinkパラメータへのマッピング	102
データのインポート	83	エクスポート	120
データの分析	95	カーブ	29
ファンクションの構築	110	グループ軸	31
モデル	100	スカラ	30
測定データ	81	マップ	28
つ		ルックアップテーブル	28
ツールバー	75	圧縮モデル	30
て		確認	99
データ		感度	64-65
NaN（非数）の削除	93	行列	30
関連性チェック	21,86	作成	112
フィルタ	94	種類	27
妥当性チェック	84	関連	64-65
データセット		編集	99
削除	91	パラメータの最適化	26
重み	92	ひ	
複数	90	品質	
複数をロード	90	評価	26
名前の変更	91		
データの分析	95		

ふ	
ファンクション	
ノードの追加	111
構造	110
構築	110
算術演算子	43
ファンクションの評価	
R ²	26
RMSE	26
ファンクション変数	
割り当ての解除	90
測定チャンネルの割り当て	89
フィルタ	
データ	94
へ	
平均二乗誤差	25
変数	
RMSE	25
め	
メソッド	18
も	
モデル	39
ASCET	40
ASCMO-DYNAMIC	40
ASCMO-STATIC	40
FMU	40
Simulink	100
Simulinkモデルの追加	100
Simulink出力のマッピング	105
Simulink入力のマッピング	104
TSim Plugin	41
パラメータのマッピング	102
ゆ	
ユーザーインターフェース	
ツールバー	75
ナビゲーションペイン	77
ログウインドウ	79
概要	75
よ	
用語集	123
ら	
ライセンス管理	16
ろ	
ログファイル	
保存	79