D5.3 – i4Q Prescriptive Analysis Tools

WP5 – BUILD: Rapid Manufacturing Line Qualification and Reconfiguration

i4Q has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958205.

Document Information

GRANT AGREEMENT NUMBER	958205	ACRONYM	i4Q	
FULL TITLE	Industrial Data Services for Quality Control in Smart Manufacturing			
START DATE	01-01-2021	DURATION	36 months	
PROJECT URL	https://www.i4q-project.eu/			
DELIVERABLE	D5.3 – i4Q Prescriptive Analysis Tools			
WORK PACKAGE	WP5 – BUILD: Rapid Manufacturing Line Qualification and Reconfiguration			
DATE OF DELIVERY	CONTRACTUAL	June 2022	ACTUAL	June 2022
NATURE	Report	DISSEMINATION LEVEL	Public	
LEAD BENEFICIARY	IKERLAN			
RESPONSIBLE AUTHOR	Estela Nieto (IKERLAN)			
CONTRIBUTIONS FROM	1 – CERTH, 6 – EXOS			
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community			
DELIVERABLE CONTEXT/ DEPENDENCIES	This document presents a technical overview of the Prescriptive Analysis solution (i4Q^{PA}). A second version will be provided namely "D5.9 i4Q Perspective Analysis Tool v2."			
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None			
READING NOTES	None			
ABSTRACT	This document is a Technical Specification document about the development of the i4Q Prescriptive Analysis Tools (i4Q^{PA}). This document which provides a thorough description and analysis of the functionalities, features, and the current implementation status. It provides an in-depth technical overview of the principal functional sub-components (i.e., features) of the Solution.			

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	12-May-2022	ToC	ToC created and sent for review	IKER
0.2	10-Jun-2022	Working Version	1st input to all sections	IKER
0.3	17-Jun-2022	1st Draft	First draft sent for internal review	IKER
0.4	20-Jun-2022	Internal review	Internal review	AIMPLAS, ITI
0.5	24-Jun-2022	2nd Draft	Addressing the comments from the internal review. Updated draft sent to the coordinator.	IKER
1.0	30-Jun-2022	Final doc	Final quality check and issue of final document	CERTH

Disclaimer

Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

TABLE OF CONTENTS

Executive summary **5**

Document structure **6**

1. General Description **7**

1.1 Overview 7

1.2 Features 7

2. Technical Specifications **8**

2.1 Overview 8

2.2 Architecture Diagram 8

3. Implementation Status **10**

3.1 Current implementation 10

3.1.1 Solution features analysed and mapping with user requirements 11

3.2 Next developments 11

3.3 History 12

4. Conclusions **13**

References **14**

Appendix I **15**

LIST OF FIGURES

Figure 1. i4Q^{PA} Solution Architecture 9

LIST OF TABLES

Table 1. i4Q^{PT} Version history 12

ABBREVIATIONS/ACRONYMS

API	Application Programming Interface
DT	Digital Twin
FMI	Functional Mock-up Interface
FMU	Functional Mock-Up
i4Q	Industrial data services for Quality Control in Smart Manufacturing
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
OF	Optimization Function
PA	Prescriptive Analysis
PDF	Portable Document Format
RestAPI	RESTful Application Programming Interface

Executive summary

This document presents an executive explanation of the **i4Q Prescriptive Analysis Tool** (i4Q^{PA}) Solution providing the general description, the technical specifications and the implementation status. The deliverable **D5.3** is the Source Code of the i4Q^{PA} Solution that is in a private repository of Gitlab: https://gitlab.com/i4q.

The documentation associated to the i4Q^{PA} Solution is deployed on the website http://i4q.upv.es. This website contains the information of all the i4Q Solutions developed in the project "Industrial Data Services for Quality Control in Smart Manufacturing" (i4Q). The direct link to the i4Q^{PA} Solution documentation is http://i4q.upv.es/19_i4Q_PA/index.html.

Such documentation is structured according to:

- General description
- Features
- Images
- Authors
- Licensing
- Pricing
- Installation requirements
- Installation Instructions
- Technical specifications of the solution
- User manual

Document structure

Section 1: Contains a general description of the **i4Q Prescriptive Analysis Tool**, providing an overview and the list of features. It is addressed to final users of the i4Q Solution.

Section 2: Contains the technical specifications of the **i4Q Prescriptive Analysis Tool**, providing an overview and its architecture diagram. It is addressed to software developers.

Section 3: Details the implementation status of the **i4Q Prescriptive Analysis Tool**, explaining the current status, next steps and summarizing the implementation history.

Section 4: Provides the conclusions.

APPENDIX I: Provides the PDF version of the **i4Q Prescriptive Analysis Tool** web documentation, which can be accessed online at: **http://i4q.upv.es/19_i4Q_PA/index.html**.

1. General Description

1.1 Overview

i4Q^{PA} will allow the user to launch a wide range of simulations of a model, even if the model has been generated in the i4Q^{DT} solution or not. These simulations will be evaluated according to some metrics and evaluation criteria that a user can define in the solution. Thus, the i4Q^{PA} will provide a ranking for the different configurations defined for the simulations, providing a prescription based on the evaluation criteria.

1.2 Features

- Providing the capability of defining several scenarios to run simulations of the Digital Twin.
- Providing the capability of defining custom optimization functions for evaluation and prescription of simulations' results.
- Providing the capability of defining the optimization functions and its inputs.
- Providing tables and graph for an easy analysis of evaluation and prescription results.

2. Technical Specifications

2.1 Overview

i4Q^{PA} is mapped to the Digital Twin Services sub-component of the Platform Tier, providing the capability of defining several scenarios to run simulations of the Digital Twin.

It is also related to the Data Analytic and Services subcomponent to evaluate the performance of each simulation run, and the Orchestration management to efficiently handle the execution of the defined simulations, both from the same Tier and the Simulation and Optimisation sub-component of the Enterprise Tier performing exhaustive simulation to identify the best scenario proposed.

The i4Q^{PA} will use single or compound, physic-based or data-driven models, and time series as inputs to perform the simulations. Apart from that, once the simulation results are obtained from the i4Q^{DT}, the i4Q^{PA} will use performance evaluation algorithms in order to rank all the simulations. Thus, different evaluation criteria will be defined based on the KPIs or the time series that the i4Q^{DT} results provide.

2.2 Architecture Diagram

The processes and services that are being included in the i4Q^{PA} software tool are mapped to two tiers in the i4Q Reference Architecture:

- **Enterprise Tier:** The i4Q^{PA} mapping to “Simulation and Optimization” sub-component enables the capability of obtaining prescriptions after the execution of multiple simulations in order to identify best and worse scenarios, as well as a full ranking, and establishing some thresholds in certain variables.
- **Platform Tier:** The Digital Twin service is providing the capability of simulating a model under some specific conditions which is the core of this solution, whereas the Orchestration management will allow the correct handling of all the simulations to be launched, Data Analytics and Services will be providing the capability to evaluate the performance of each model.

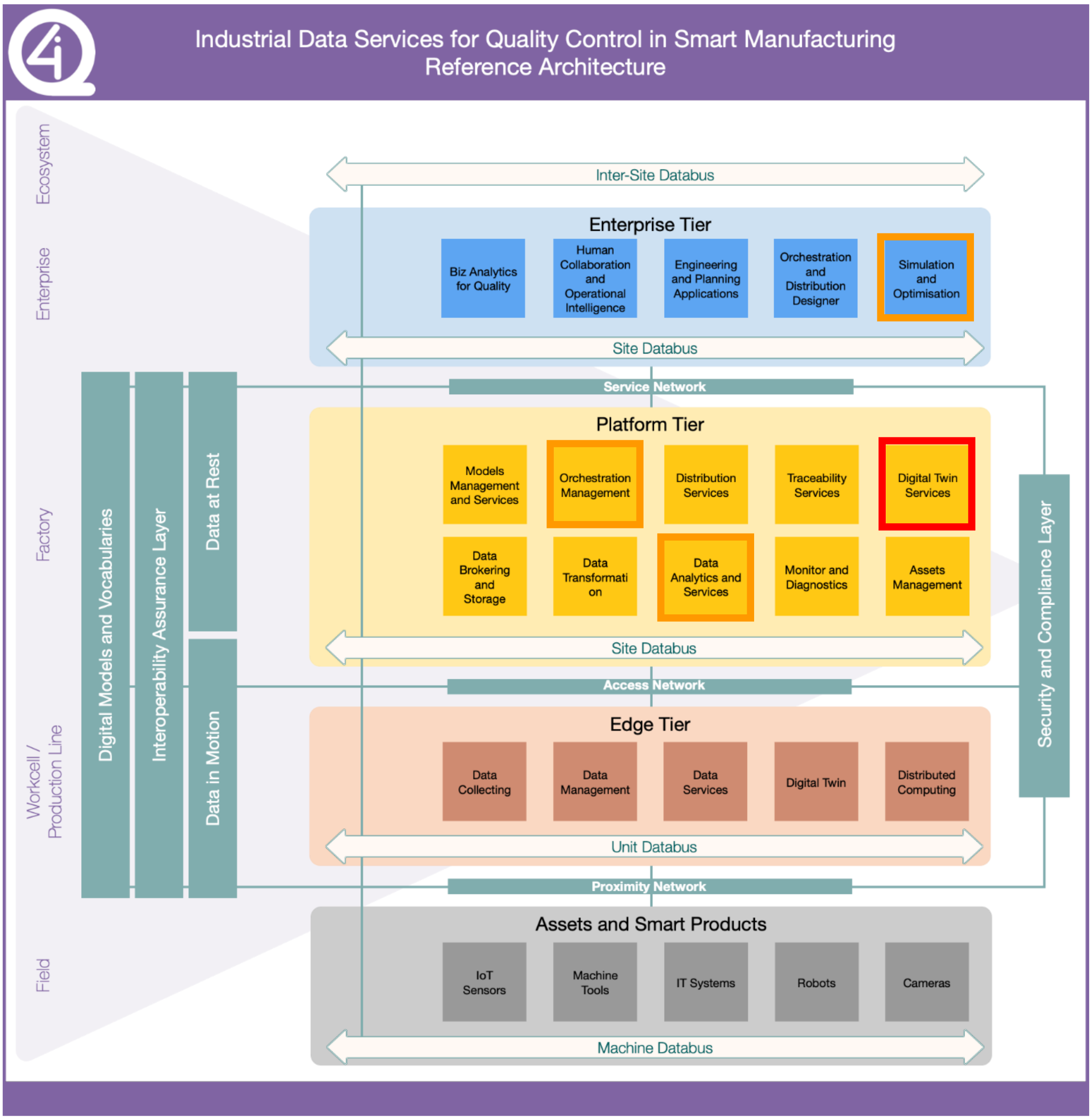


Figure 1. i4Q^{PA} Solution Architecture

3. Implementation Status

3.1 Current implementation

The current implementation of the i4Q^{PA} Solution is:

- Analysis of existing **FMU-related Python libraries**: FMPy, PyFMI, PySimulator and PythonFMU:
- PythonFMU library was first discarded as its main function is the generation of FMU models, which is out of scope of this solution.
- PySimulator is limited when it comes to model's internal parameters reading.
- PyFMI library conflicted with newer versions of Python (3.7 and above).
- FMPy have model reading and simulation functionalities. However, an error raised whenever a model was instantiated more than 1050 times. In order to simulate a model, first it has to be instantiated, which cannot be done more than 1050 times without reinitializing the software. That issue was solved by adding a function to the library which instantiates the model once, but carries out as many simulations as required.
- Development of a **mock-up** with all i4Q^{PA} solution's basic functionalities: model reading, simulation, evaluation and prescription.
- **Mock-up code adaptation** to current application structure, composed of a frontend, a backend and a RestAPI.
- The frontend and backend of the solution are communicated through **RestAPI** Flask. Every frontend widget, or backend table or dictionary has its own isolated request function, therefore, a global class had to be created to communicate all request functions.
- The **frontend** was designed taking the i4Q web template as starting point and customizing it to meet the specific needs of the i4Q^{PA} solution. These are the functionalities or features added to the frontend:
- Template's tabs' structure adaptation to the needs of the i4Q^{PA} backend. The frontend now has 4 main tabs (home, simulation, evaluation/prescription, and help) and some secondary tabs within the main ones.
- Redux addition to all tabs, a predictable state container for JavaScript apps.
- The mock-up code was the starting point of the **backend**, therefore it had to be adapted to the RestAPI. Functionalities or features added to the backend:
- fmisim library development based on FMPy library:
- Physic-based models' internal parameters reading.
- Simple and compound model reading and simulation.
- Both co-simulation and model exchange type models' individual and multiple simulations, meaning that the same model can be simulated more than once without instantiating it.
- Development of an evaluation library composed of metrics and optimization functions (OFs).
- The metrics are Python functions which have to be added to the "metrics.py" file.
- The OFs are Python-like files saved as a txt files. (one file for each function):
- The file must contain a header indicating which parameters are required in order to carry out the calculations, the importation of the metrics, and the OF.

- The OF must have two inputs, the simulations results and a dictionary containing the parameters indicated in the header.
- The OFs can be predefined or custom. If the function is custom, it must be saved in the path indicated in the solution's configuration file.
- **Functionalities** or features added to the i4Q^{PA} solution:
- Model selection, model's internal parameters range configuration and simulation configuration.
- Simulations results filtering through two filtering layers. In the first input the user can define which outputs and simulations they want to plot. And in the second filter individual signals can be selected from a signal table.
- Simulations results plotting.
- Simulations results' evaluation configuration.
- Simulations results' evaluation.
- Prescription of the optimum model configuration according to the selected evaluations.

3.1.1 Solution features analysed and mapping with user requirements

A set of features has already been developed for i4Q^{PA}, based on the set of user requirements referring to i4Q^{PA} (Deliverable 1.9) and in line with the functional viewpoints (Deliverable 2.6). Similar requirements have been assigned into common categories of tasks based on an extensive technical study conducted on user requirements, available datasets, etc., introduced to ensure the generalization abilities of the i4Q^{PA} solution.

- PC4r8.3.1 "Define the evaluation criteria to be performed among variables, e.g. if vibrations are rising, check the roughness of the part" is covered by the feature of providing the capability of defining custom OFs for the evaluation and prescription of the simulations' results. Therefore, any evaluation criteria can be defined.
- PC4r8.3.2 "Establish the limited values where the variables should keep. E.g. if you are machining aluminium in summer, do not go over 100 ºC" is covered by the feature of providing the capability of defining the OF and its inputs. All defined inputs' values are defined through the frontend. Therefore, the user can program the OF so that if the signal goes beyond the introduced input, the result of the evaluation is 0, for instance.
- PC4r8.3.3 "The system shall perform simulations that isolate a single variable in order to detect the independent variables that lead to changes in dependent variables" is covered by the feature of providing the capability of defining several scenarios to run simulations of the DT. The solution allows the user to select which internal parameters of the model to vary and how.
- PC4r8.3.4 "The solution should evaluate and learn from the results of the simulations and establish the thresholds of the variables. E.g. if you machine aluminium over 100ºC, you have quality issues" is covered by the feature of providing tables and graph for an easy analysis of evaluation and prescription results. The information displayed in tables and graphs give the user the required information to establish thresholds of variables.

3.2 Next developments

In the coming months, the following features will be implemented:

- The configuration and simulation of data-based models.
- The configuration and simulation of composed models (models created from individual FMI-based (*Functional Mock-up Interface*) models by the i4Q^{DT}).
- Add to the frontend the option to request the i4Q^{DT} solution to carry out simulations, and implement that functionality in the backend.
- Implement the message broker.
- Evaluation and prescription of a model excited by different inputs.
- Addition of new metrics and optimization functions to the evaluation library.
- Improve the prescription visualization tab.
- Dockerization of the software.
- Update "Help" and "Home" tabs.

3.3 History

Version	Release date	New features
v0.0.1	09/12/2021	Adaptation of fmpy library to carry out multiple simulations.
v0.0.2	14/12/2021	Mock up – Model reading, configuration, simulation and prescription.
v0.0.3	17/12/2021	Mock up – First version of evaluation process.
v0.0.4	11/01/2022	Created i4Q^{PA} solution ' s structure: frontend, backend and web framework.
v0.0.5	27/01/2022	Web framework – Addition of Flask app. Backend – Adaptation of mock up to frontend-backend i4Q^{PA} structure: model reading, configuration and simulation. Frontend – Adaptation of i4Q frontend template to i4Q^{PA} solution ' s needs (tab's structure). Development of model selection, configuration and simulation. Addition of simulations' progress bar. General visual improvement.
v0.0.6	10/02/2022	Bug fix. Addition of simulations' results first filter.
v0.0.7	22/02/2022	Addition of simulations' results second filtering and display.
v0.0.8	08/04/2022	Added evaluation function class and class initialization. fmisim library's first version.
v0.0.10	12/04/2022	Multiple simulation function added to fmisim library. Backend – Adaptation of code to fmisim library. Frontend – Merge of model selection and simulations' results' visualization tabs. Addition of type of simulation (serial/parallel) selector.
v0.0.11	25/04/2022	Backend – Evaluation structure added and tested. Evaluation library is composed by a single metric and evaluation function.
v0.0.12	04/05/2022	Frontend – Added evaluation configuration widgets

Table 1. i4Q^{PT} Version history

4. Conclusions

Deliverable D5.3 Prescriptive Analysis Tools is a technical specification document, providing an in-depth technical overview of the i4Q^{PA} solution. It describes in detail the role, the functionalities, and the conceptual architecture of i4Q^{PA}. It presents a study detailing the main features of the solution to clarify the key functionalities and objectives of the i4Q^{PA} solution, describing its architecture diagram with respect to i4Q Reference Architecture.

The current implementation status of i4Q^{PA} is detailed thoroughly, presenting the significant progress of this overall development. This document presents these approaches which include 1) the pilots requirements analysis and engineering to clarify the technical specifications, 2) the technical studies conducted to define the solution's abstract level architecture 3) the input and output definition of the solution, 4) the objectives of i4Q^{PA}.

References

Functional Mock-up Interface. Available at: https://fmi-standard.org/ (Accessed: 5 May 2022).

Appendix I

APPENDIX I: Provides the PDF version of the i4Q **< Prescriptive Analysis Tools >** web documentation, which can be accessed online at: **http://i4q.upv.es/19_i4Q_PA/index.html**

i4Q Prescriptive Analysis Tools

General Description

i4Q^{PA} (Prescriptive Analysis Tools) is a micro-service consisting of simulation models as a service, taking as input the manufacturing resources, current production planning and process condition, and proposing process configuration parameters, ensuring that non-simulation experts may also exploit the prescriptive analyses.

i4Q^{PA} uses different data analytics platforms, thus allowing the evaluation and comparison of the results that come from the simulation of different industrial scenarios. i4Q^{PA} provides an extensive set of optimisation techniques that help in the development of the mathematical methods that will make the prescriptive analysis. These optimisation techniques have as objective the selection of the best of the digital twin instances that have been simulated, or even the selection of those simulation scenarios that need to be selected or avoided in order to reach faster the optimal solution, based on a scenarios map.

Features

This i4Q Prescriptive Analysis Tools offers following features:

- Provides the capability of defining several scenarios to run simulations of the Digital Twin.

Commercial information

Authors

Partner	Role	Website	Logo
IKERLAN	Leader	www.ikerlan.es/en/	ikerlan MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE
EXOS	Vice-Leader	www.exos-solutions.com/	EXOS Operational Consulting
CERTH	Participant	https://www.certh.gr/root.en.aspx	CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

Technical Specifications

i4Q^{PA} makes use of the i4Q^{DT} to test different possibilities and proposing process configuration parameters. First a set of possible scenarios needs to be defined, they can then be simulated through the Digital Twin and the results evaluated and compared to provide a prescriptive analysis. The software solution can be containerized.

i4Q^{PA} allows to define different simulations to be performed (the scenarios) and its performance tested. For Digital Twins of the FMU type it uses the pyFMI package. In general, the communication with the Digital Twins can be managed through REST APIs provided by the i4Q^{DT} solution. This analysis can be visualized using a GUI generated using tools such as Shiny, Tkinter, Plotly.

i4Q^{PA} depends on i4Q^{DT}, as it supplies the simulation models. It may depend on some of the basic solutions (security, data traceability, ...), it may also depend on the data integration and fusion micro-service, as well as the i4Q Data Repository. It may also give inputs to the i4Q Line Reconfiguration Toolkit.

Technical Development

This i4Q Solution has the following development requirements:

- Development Language: Python
- Container: Docker
- Deployment/Orchestration: -
- User Interface: -
- Application Interfaces: RestAPI
- Database engine: i4QDR
- Python libraries: Numpy, Pandas, SciPy, Pywraplp, Pulp

License

Pricing

Subject	Value
Payment Model	One-off - Monthly - Yearly paument

Associated i4Q Solutions

Required

- The solution can operate without the need for another i4Q solution.

Optional

- i4Q Digital Twin simulation services to carry out part of the simulations defined by i4Q Prescriptive Analysis Tools.
- i4Q Data Repository and/or i4Q Data Integration and Transformation Services to get models' inputs to simulate.
- i4Q Anaisis Dashboard for data and signal visulization.

System Requirements

The component may be run on any platform that support Docker images.

- Docker Desktop for Windows
- Docker Desktop for Mac

i4Q^{PA} configuration

Being the repository files and folders structure as follows:

```
New folder
|
|---- frontend
|     |-- Dockerfile
|     |-- src
|         |-- config.yml
|
|---- backend-python
|     |-- Dockerfile
|
|---- docker-compose.yaml
```

Before launching the solution, the following steps must be carried out:

- Open the "docker-compose.yaml" file and indicate in the "services/backend/volumnes" in which folder the models and evaluation functions are going to be stored.
- Open the "config.yml" file and:

1. Indicate in the "models/location" and in the "eval_fcn/location" the folder in which the models and evaluation functions are going to be stored.
2. If the solution is going to be launched as a Docker image, set as "true" the "docker" option. If not, set it as "false".

```
docker-compose up --build
```

i4Q^{PA} docker installation

- Download the zip file from this link: [Download]
- Create a new folder in your local computer and extract the content of the zip file.
- Open a new terminal in the folder where the docker-compose file is located and run the following command:

```
docker-compose up --build
```

- The application will be running as soon as the process finishes. The url where the user interface is running will be shown in the terminal, which by default is http://localhost:8082
- In order to stop the application the terminal needs to be closed.

i4Q^{PA} front-end installation (optional)

Install project dependencies, open a terminal in frontend folder:

```
cd \subsystems\frontend
npm install
```

User Manual

The following user manual is divided into two sections: the API specification and the description of the front-end.

API specification

To start the server application on a local computer, a tab in the default browser will be opened. Then the following line should be entered in shell.

```
python src/server.py
```

This action starts a server running in port 5000 and allows to use the following API methods. Those methods are accesible through the application front-end or sending the proper request using any other tool like Postman, or directly with Python code:

Resource	POST	GET	PUT	DELETE
/models_information/names		Supported		
/selected_model	Supported			
/models_information/parameters		Supported		
/submitted_parameters	Supported			
/simulation_information/model_configuration		Supported		
/simulation_configuration	Supported			
/simulation/start	Supported			
/simulation/state		Supported		
/simulation_results		Supported		
/evaluation/information	Supported			
/evaluation/configuration	Supported			
/evaluation/results		Supported		

- /models_information/names → Get a 3 level data-tree with the available models.
- The first level indicates the folder in which the models are saved whitin the folder indicated in the configuration. If the models are not saved in an specific folder, the models are grouped with the label “misc”.

- The second level indicates which type of model is: physics-based or data-driven (“data_driven”). Physics-based models can be simple (“simple”) or copmosed of 1 or more simple models (“compound”).
- The third level indicates the models’ name and key. The key is required to indicate which model is wanted to be prescribed.
- /selected_model → Select the model to prescribe by means of the key.
- Request type: {“selected_model”: “2-0-1”}.
- /models_information/parameters → Get the model’s internal parameters information.
- /submitted_parameters → Set the range of the model’s internal parameters. The range can be indicated as a range or a couple of limits. The range will be added and substracted to the original value to calculate the limits.
- Request type: {“0”: {“name”: “mass1.m”, “valueReference”: 25, “startValue”: 50, “rangeValue”: 2, “stepValue”: 1}, “1”: {“name”: “mass2.m”, “valueReference”: 27, “startValue”: 100, “downValue”: 10, “upValue”: 150, “stepValue”: 1}}
- /simulation_information/model_configuration → Get the map of configurations of the model created with the submitted parameters’ range (“submitted_parameters”).
- /simulation_configuration → Set the simulations’ configuration. For serial simulations the “simulation_type” variable must be set as “0”, for parallel simulations to “1”.
- Request type: {“simulation_type”: 0, “simulation_cores”: 2, “start_time”: 0, “stop_time”: 5, “step_size”: 0.01}
- /simulation/start → Start the simulations.
- Request type: {“start_simulation”: 1}
- /simulation/state → Get the state of the simulations: the amount of simulations to be carried out, the current simulation time, … Only to be used with parallel simulations.
- /simulation_results → Get the results of the simulations.
- /evaluation/information → Get a dictionary with all the predefined and custom evaluation functions.
- /evaluation/configuration → Submit the configuration of the evaluation. The structure of this request is dependant of the optimization functions saved in the folder indicated in the configuration of the solution. In the following line there is an example of a configuration:
- Request type : {“functions”: [“fcn_name1”, “fcn_name2”], “params”: [[“T1”, 0, “1”], [“T2”, 1]], “name”: [[“signal”, “objetive”, “bigger_than”], [“signal”, “objective”]], “type”: [[“signal”, “float”, “bool”], [“signal”, “float”]]}
- /evaluation/results → Get a table with the results of the evaluation, one line per simulation. The columns show the results of a specific optimization function.

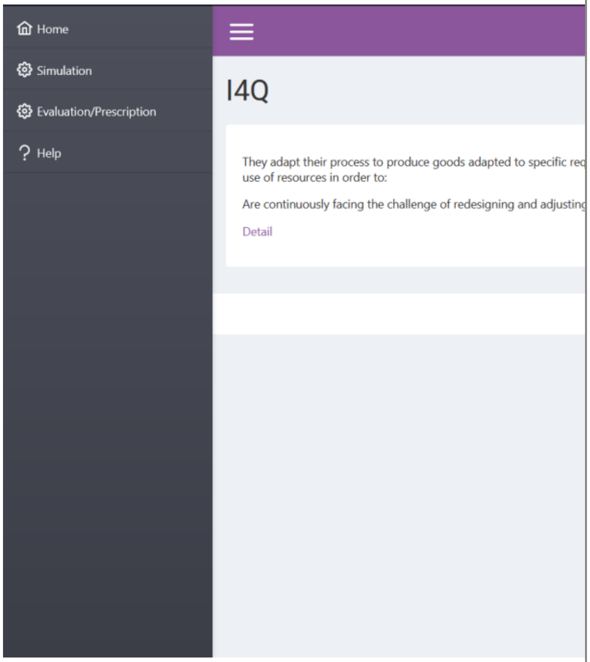
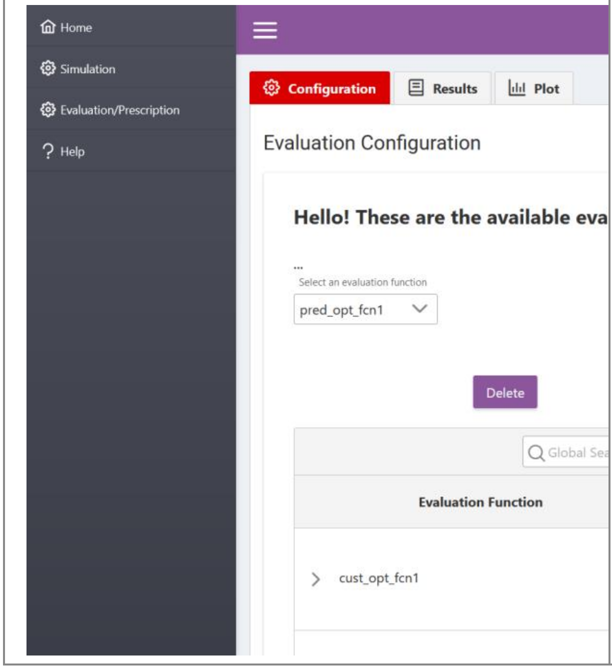
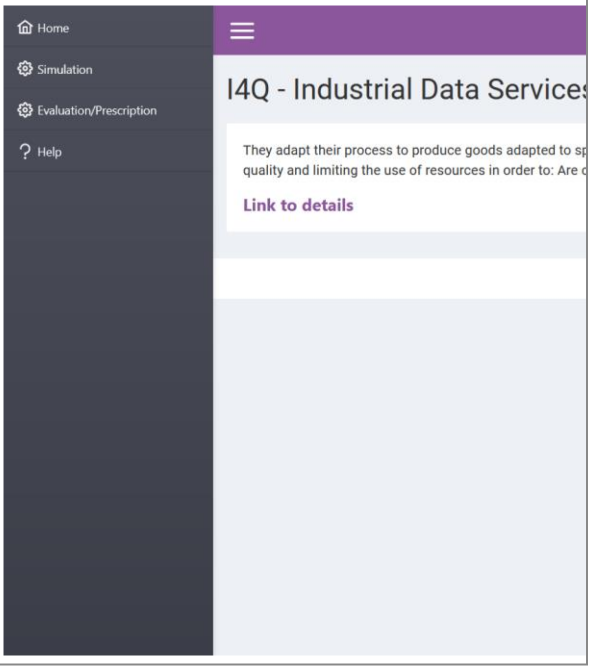
Front-end User Manual

To start application on localhost, a tab in the default browser will be opened:

```
npm start
```

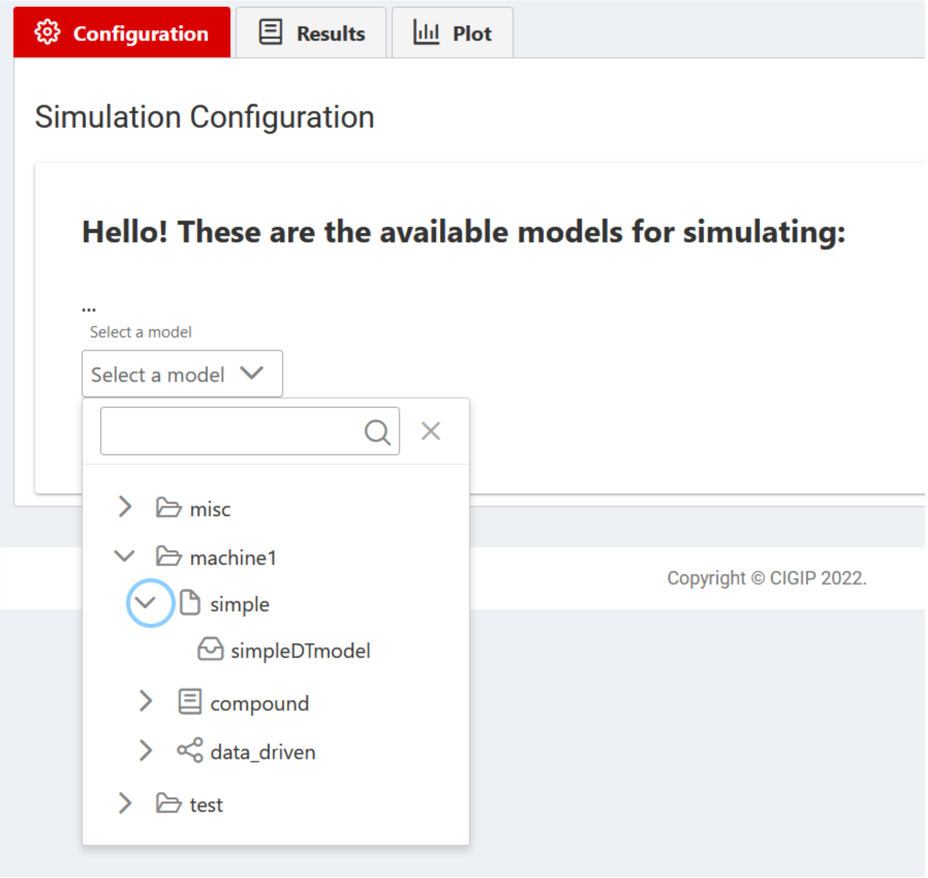
Application will open in one of its 4 tabs, the home tab precisely. The tabs navigation widget is placed in the left, and available tabs are:

- Home → Tab in which an introduction of i4Q^{PA} is displayed.
- Simulation → Tab in which the simulation is configured, from the model selection, to the range of the model's internal parameters.
- Evaluation/Prescription → Tab in which the evaluation and prescription is configured, from the selection of optimization functions, to the parametrization of those functions.
- Help → Tab in which guidelines for the user are displayed.

Home	**Simulation**
 I4Q They adapt their process to produce goods adapted to specific req use of resources in order to: Are continuously facing the challenge of redesigning and adjusting Detail	
Evaluation	**Help**
 Home Simulation Evaluation/Prescription Help Configuration Results Plot Evaluation Configuration Hello! These are the available eva ... Select an evaluation function pred_opt_fcn1 Evaluation Function cust_opt_fcn1	 I4Q - Industrial Data Service They adapt their process to produce goods adapted to sp quality and limiting the use of resources in order to: Are c Link to details

Three type of models can be analized: simple and compound physic-based models, and data-driven models. All three are analized similarly. The procedure is explained in the following points:

- Simulation tab
- Selection and configuration of a model through a 3 level data-tree widget.



Simulation Configuration

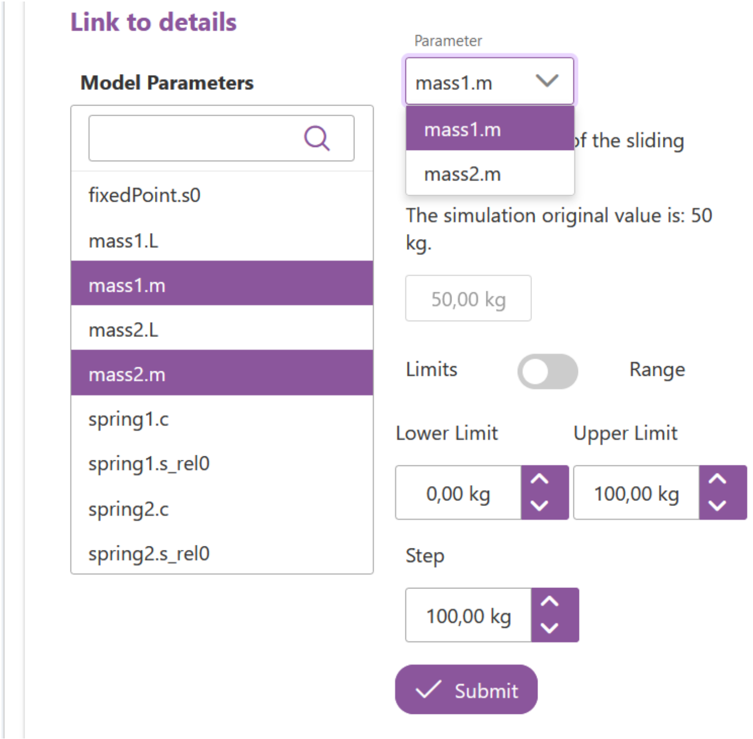
Hello! These are the available models for simulating:

...

Select a model

Select a model

- Selected model's internal parameters' selection and range configuration. First, all the parameters to configure have to be selected from the list to the left. Then, individual parameters have to be selected in the top-right widget. Finally, the selected individual parameter's range has to be introduced in the bottom-right widgets. The range can be introduced as an upper and bottom limit ("Limits") or as a +-range ("Range").



Link to details

Model Parameters

fixedPoint.s0
mass1.L
mass1.m
mass2.L
mass2.m
spring1.c
spring1.s_rel0
spring2.c
spring2.s_rel0

Parameter: mass1.m

mass1.m
mass2.m

of the sliding

The simulation original value is: 50 kg.

50,00 kg

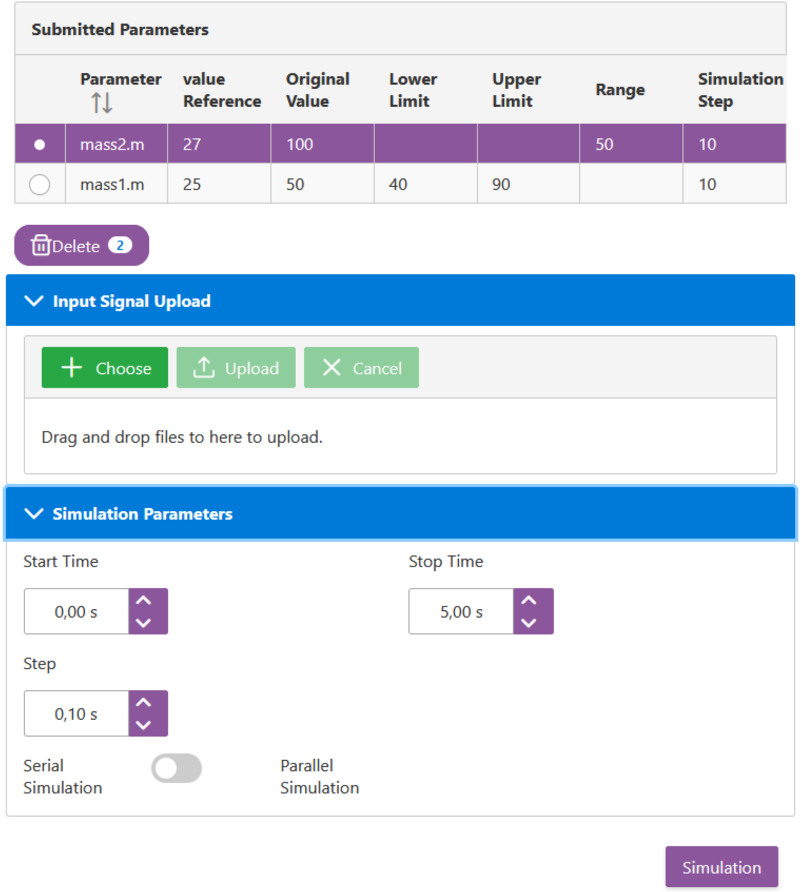
Limits / Range

Lower Limit: 0,00 kg

Upper Limit: 100,00 kg

Step: 100,00 kg

- Input addition widget and simulations’ other configurations’ input. Simulations’ input can be defined through the upper widget. A single or multiple csv files can be selected. The csv’s headers must coincide with the selected model’s inputs names. As for the other configurations, the simulations’ duration can be defined through the bottom widget (independently of the csv’s input time). Serial or parallel simulations can be selected too.



Submitted Parameters

	Parameter ⇅	value Reference	Original Value	Lower Limit	Upper Limit	Range	Simulation Step
●	mass2.m	27	100			50	10
○	mass1.m	25	50	40	90		10

Drag and drop files to here to upload.

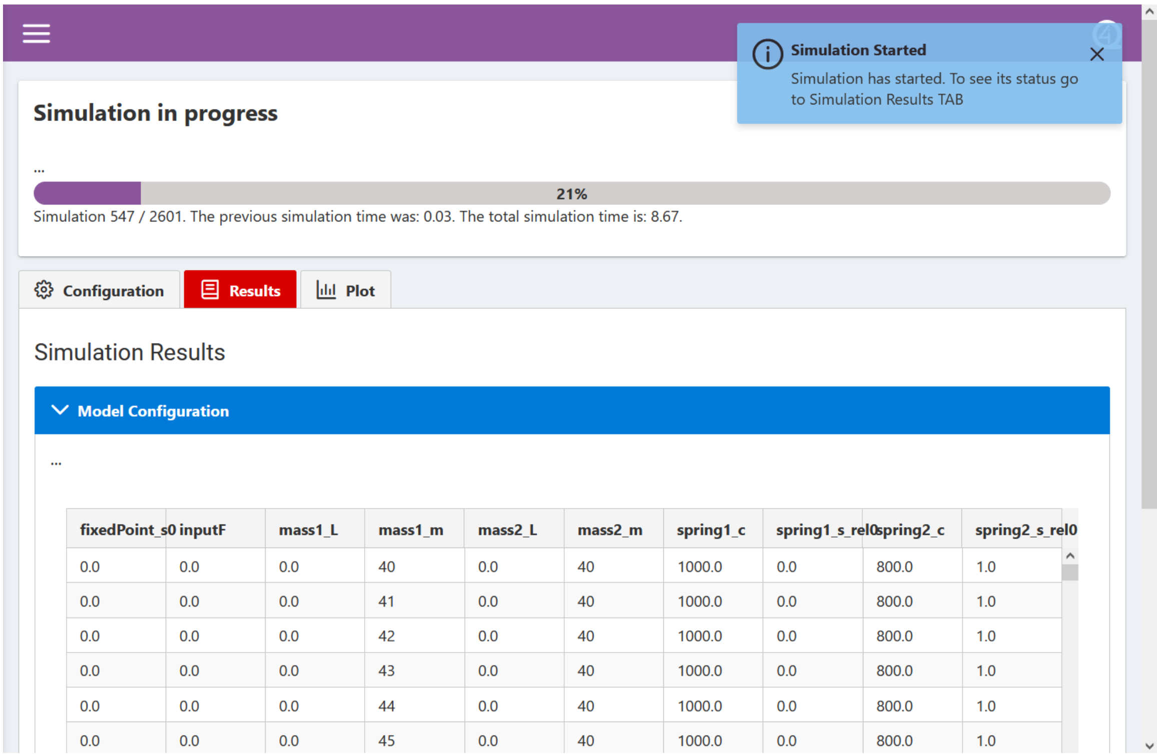
Start Time: 0,00 s

Stop Time: 5,00 s

Step: 0,10 s

Serial Simulation / Parallel Simulation

- Once the simulations are configured, the “Simulation” button must be pressed.
- When the “Simulation” button is pressed, the frontend moves the user from the “Simulation/Configuration” tab, to the “Simulation/Results” tab, and a simulations progress bar is displayed. While the simulations are being carried out, the user can check the map of configurations (selected model’s parameters values for each simulation) in the widget below the progress bar.



Simulation Started

Simulation has started. To see its status go to Simulation Results TAB

Simulation in progress

...

21%

Simulation 547 / 2601. The previous simulation time was: 0.03. The total simulation time is: 8.67.

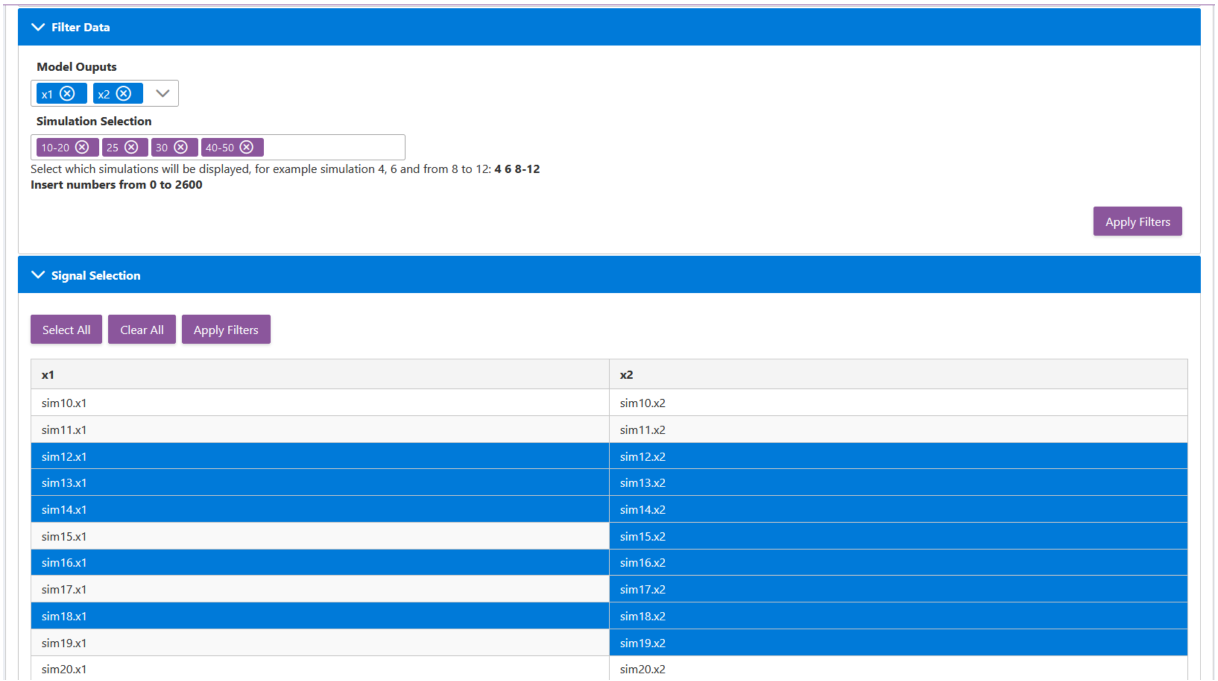
Simulation Results

Model Configuration

...

fixedPoint_s0	inputF	mass1_L	mass1_m	mass2_L	mass2_m	spring1_c	spring1_s_rel0	spring2_c	spring2_s_rel0
0.0	0.0	0.0	40	0.0	40	1000.0	0.0	800.0	1.0
0.0	0.0	0.0	41	0.0	40	1000.0	0.0	800.0	1.0
0.0	0.0	0.0	42	0.0	40	1000.0	0.0	800.0	1.0
0.0	0.0	0.0	43	0.0	40	1000.0	0.0	800.0	1.0
0.0	0.0	0.0	44	0.0	40	1000.0	0.0	800.0	1.0
0.0	0.0	0.0	45	0.0	40	1000.0	0.0	800.0	1.0

- When the simulations are finished, a two steps filter is made available. In the first filter, the signals and simulaitons to be plotted can be selected. In the second filter, in turn, individual signals can be selected. When the "Apply Filters" is selected, the frontend moves the user to the "Simualtion/Plot" tab.



Filter Data

Model Ouputs

x1 ⊗ x2 ⊗

Simulation Selection

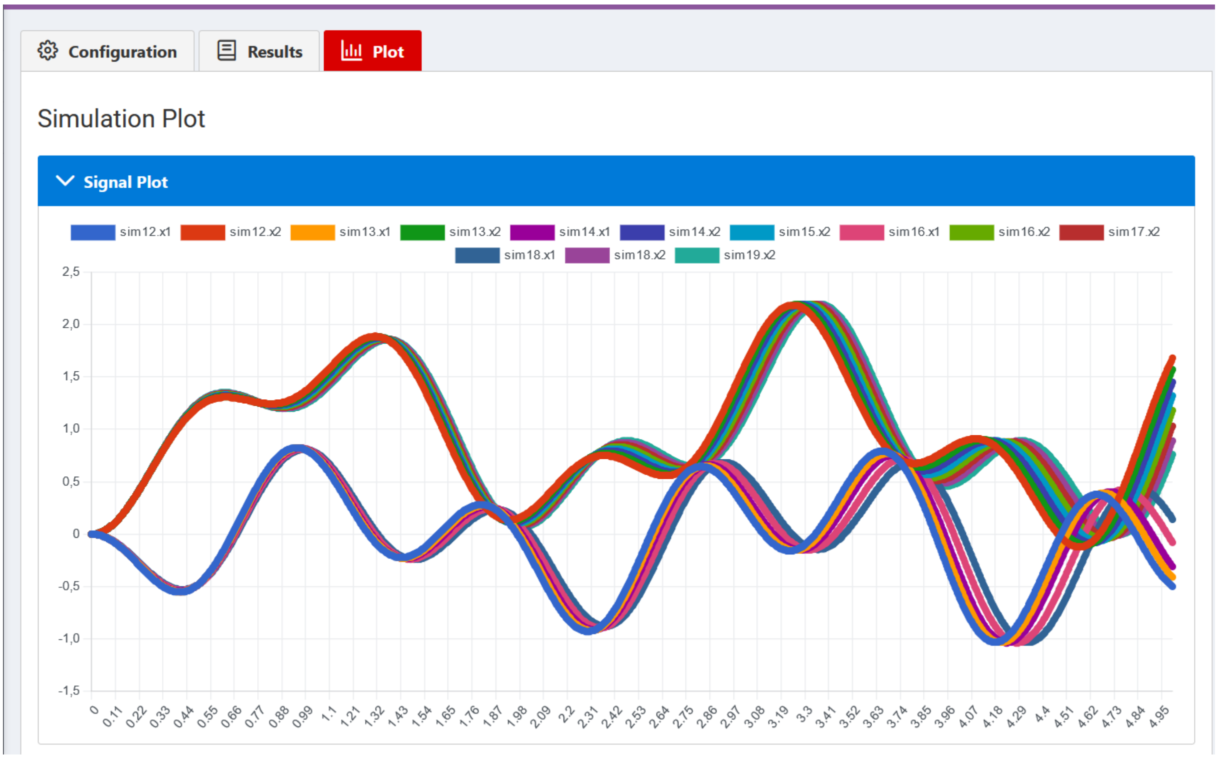
10-20 ⊗ 25 ⊗ 30 ⊗ 40-50 ⊗

Select which simulations will be displayed, for example simulation 4, 6 and from 8 to 12: **4 6 8-12**
Insert numbers from 0 to 2600

Signal Selection

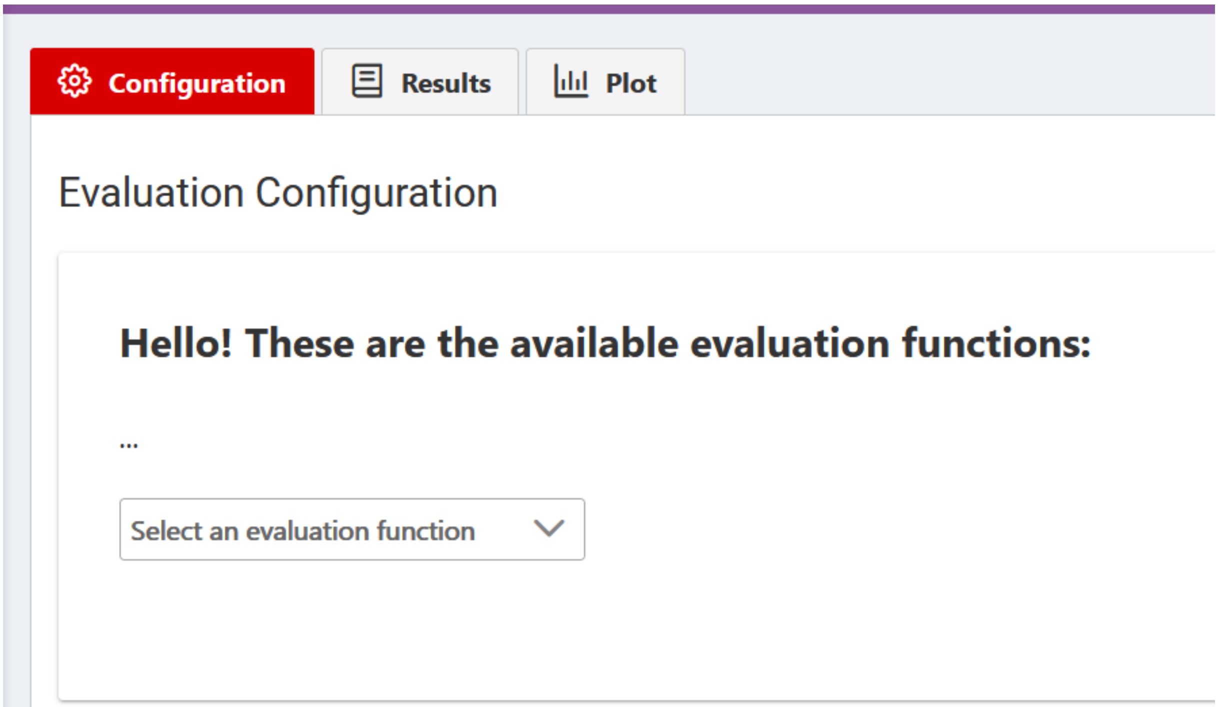
x1	x2
sim10.x1	sim10.x2
sim11.x1	sim11.x2
sim12.x1	sim12.x2
sim13.x1	sim13.x2
sim14.x1	sim14.x2
sim15.x1	sim15.x2
sim16.x1	sim16.x2
sim17.x1	sim17.x2
sim18.x1	sim18.x2
sim19.x1	sim19.x2
sim20.x1	sim20.x2

- In the "Simulation/Plot" tab, the selected signals can be visualized.



Simulation Plot

- Evaluation/Prescription tab * When the simulations are finished, the “Evaluation/Prescription/Configuration” tab’s evaluation function selection widget is made available. An “Add” button appears when a function is selected. Evaluation functions must be added one-by-one. A function can be added more than once.

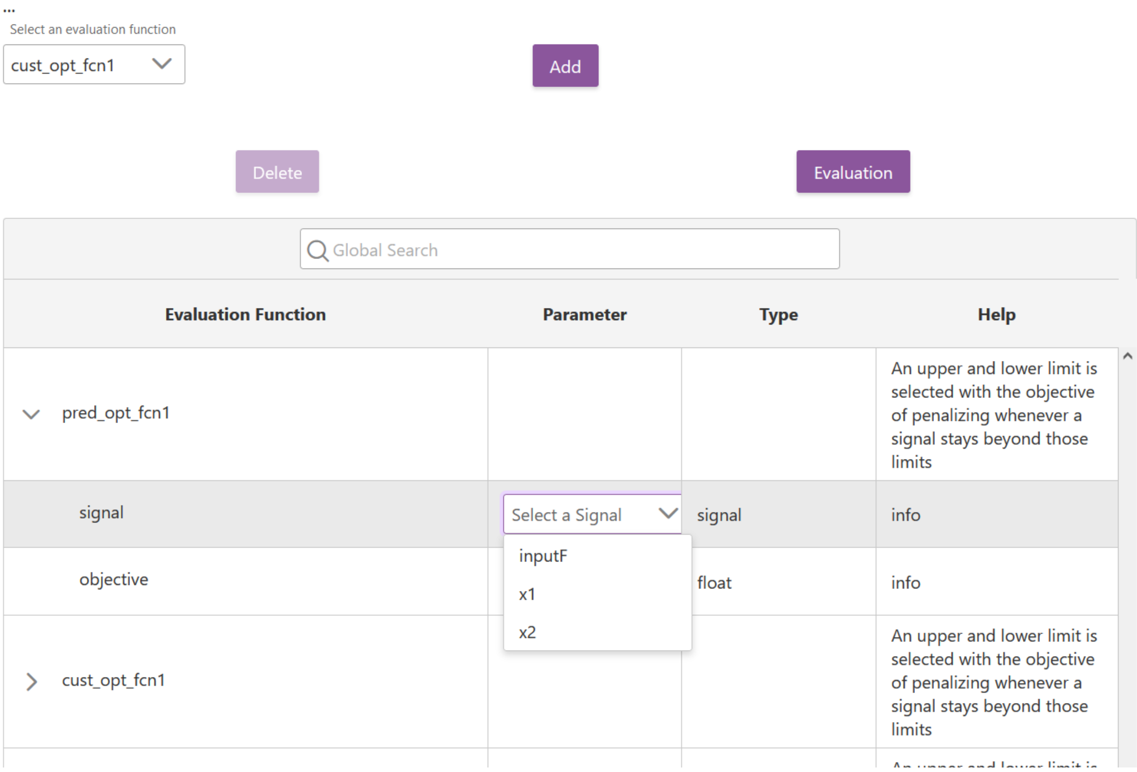


Evaluation Configuration

Hello! These are the available evaluation functions:

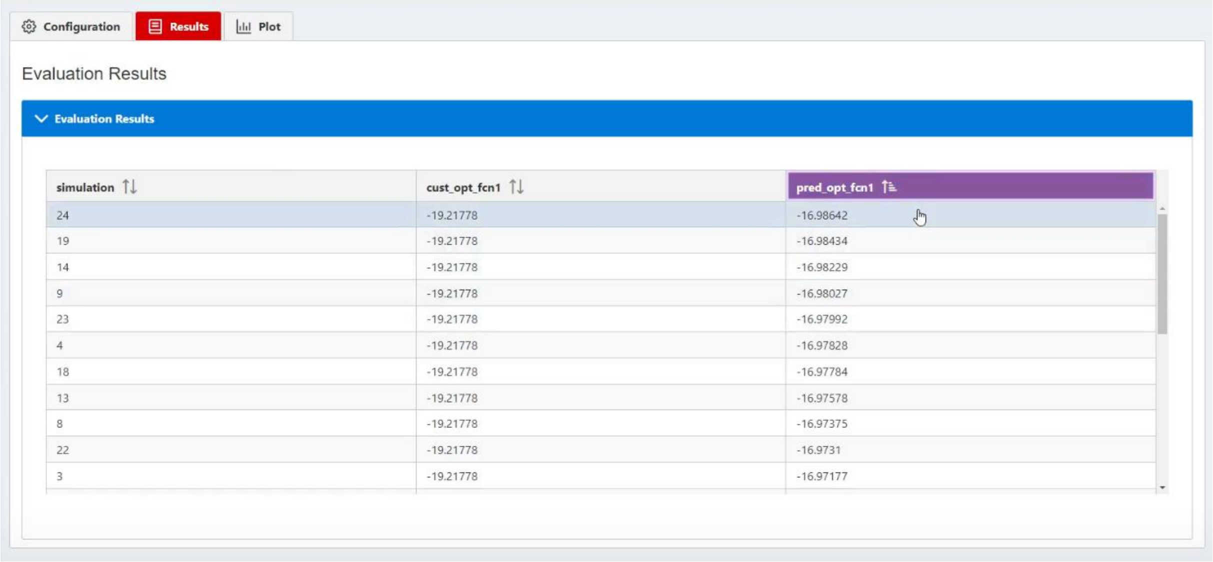
...

- All added functions are displayed in a table, in which the user must introduce some data. For instance, in the image below, the function “pred_opt_fcn1” asks for a signal and a float. The signal must be the an input or output of the model.



Evaluation Function	Parameter	Type	Help
pred_opt_fcn1			An upper and lower limit is selected with the objective of penalizing whenever a signal stays beyond those limits
signal	Select a Signal	signal	info
objective		float	info
cust_opt_fcn1			An upper and lower limit is selected with the objective of penalizing whenever a signal stays beyond those limits
			An upper and lower limit is

- When the evaluation is configured and the button "Evaluation" is pressed. The frontend moves the user to the "Evaluation/Prescription/Results" tab, in which the results of all the evaluations are displayed in a table. The results can be ordered by pressing the columns names.

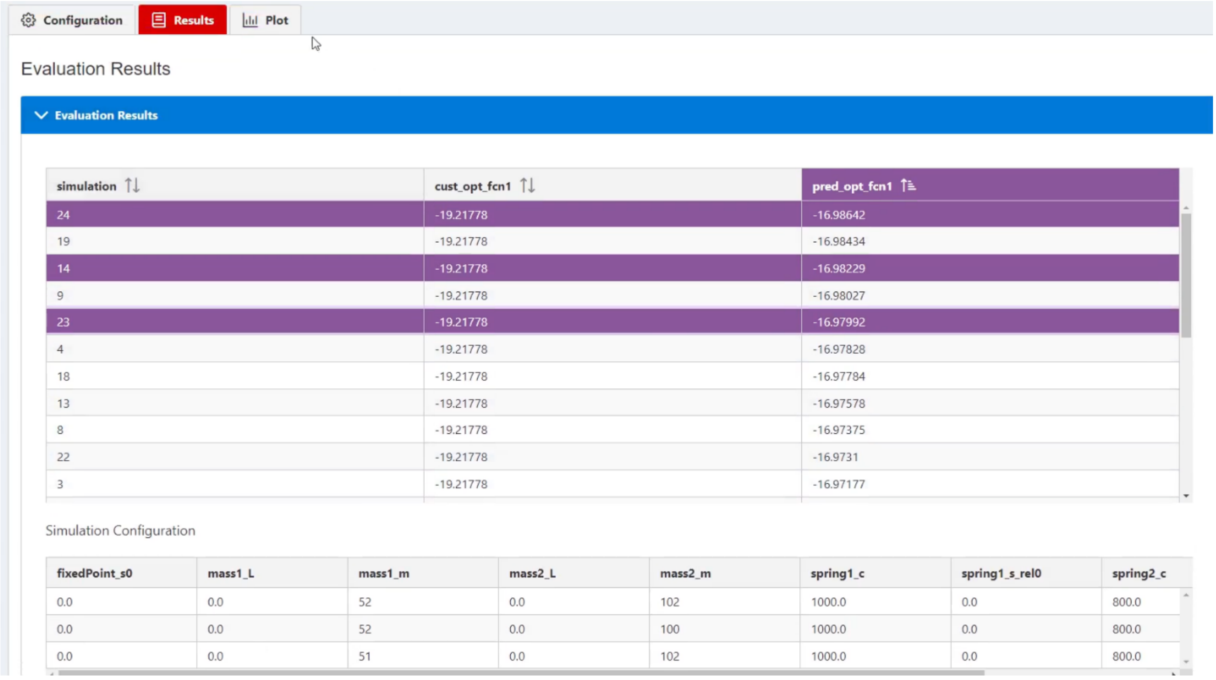


Evaluation Results

Evaluation Results

simulation	cust_opt_fcn1	pred_opt_fcn1
24	-19.21778	-16.98642
19	-19.21778	-16.98434
14	-19.21778	-16.98229
9	-19.21778	-16.98027
23	-19.21778	-16.97992
4	-19.21778	-16.97828
18	-19.21778	-16.97784
13	-19.21778	-16.97578
8	-19.21778	-16.97375
22	-19.21778	-16.9731
3	-19.21778	-16.97177

- To get the prescription of an evaluation function, the user must select a line of the evaluation results table. All the selected lines's prescription are displayed in a table below the evaluation results table.



Evaluation Results

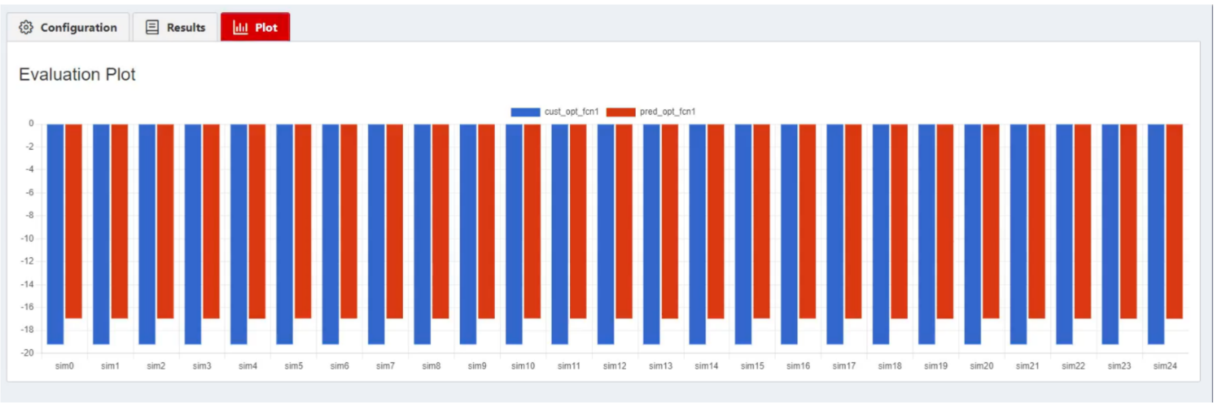
Evaluation Results

simulation	cust_opt_fcn1	pred_opt_fcn1
24	-19.21778	-16.98642
19	-19.21778	-16.98434
14	-19.21778	-16.98229
9	-19.21778	-16.98027
23	-19.21778	-16.97992
4	-19.21778	-16.97828
18	-19.21778	-16.97784
13	-19.21778	-16.97578
8	-19.21778	-16.97375
22	-19.21778	-16.9731
3	-19.21778	-16.97177

Simulation Configuration

fixedPoint_s0	mass1_L	mass1_m	mass2_L	mass2_m	spring1_c	spring1_s_rel0	spring2_c
0.0	0.0	52	0.0	102	1000.0	0.0	800.0
0.0	0.0	52	0.0	100	1000.0	0.0	800.0
0.0	0.0	51	0.0	102	1000.0	0.0	800.0

- Lastly, the user can visualize the evaluations results through a bar plot, if needed, in the “Evaluation/Prescription/Plot” tab.



Misc.

If desired, to ease code debugging, install Google Chrome React Developer Tools and Redux DevTools extensions to debug React code and see Redux state. To open developer tools:

```
bash CTRL + SHIFT + i
```