PEGASE

PAN EUROPEAN GRID ADVANCED SIMULATION AND STATE ESTIMATION

Deliverable D6.2

Specification of the methodology for prototypes validation

Grant Agreement Number: 211407 implemented as Large-scale Integrating Project
Coordinator: Tractebel Engineering S.AA
Project Website: http://www.fp7-pegase.eu

Document Information

Document Name:	D6.2: Specification of the methodology for prototypes validation
ID:	DEL_WP6.2_D6.2_Specification_of_the methodology_for_prototypes_validation_20110622_v1.5
WP:	6
Task:	6.2
Revision:	1.5
Revision Date:	22/06/2011
Author:	V.Sermanson, A.Szekut, S.Lopez, S.Leclerc

Diffusion list

STB, SSB, Coordinator

Approvals

First version to be submitted for review to the European Commission

	Name	Company	Date	Visa
Author	S.Lopez	REE		
Author	V.Sermanson	RTE		
Author	A.Szekut	TE		
Author	S.Leclerc	RTE		
Task Leader	S.Lopez	REE		
WP Leader	V.Sermanson/S.Leclerc	RTE		

Documents history

Revision	Date	Modification	Author
1.1	29/07/2010	Dynamic tests	S.Lopez et all
1.2	28/02/2011	Format,and wording	S.Lopez et all
1.3	07/06/2011	First feedback from STB	S.Leclerc
1.4	15/06/2011	Second feedback from STB	S.Leclerc
1.5	22/06/2011	Executive summary and formatting	S.Leclerc

Executive summary

The present document defines the specifications for the test procedures to be carried out in task 6.4.

Its first goal is to provide methodologies to validate the compliance of prototypes developed in WP2, WP3, and WP4 to requirements defined in WP1.

The second goal is to provide methodologies to assess their limits, in terms of functionalities, quality or performance. Those tests will be referred to as “robustness” tests.

The conceptual framework is first described. For each one of the prototypes, specifications will be provided for:

- test cases, which only consist in the input data (size of the system, types of equipments…)
- test procedures, which include the definition of scenarios developed from the test cases, as well as the definition of methods to assess compliance to the requirements, performance, and quality of the results

A simple organizational framework for testing activities is also defined: partners involved in the validation process, the associated responsibilities, and procedures to ensure correct interactions between testers, developers, and the coordinator. Procedures are in particular defined for anomaly and non-compliance reporting and resolution, for possible deviations from the validation plan specified here, and for approval of the final tests.

In order to demonstrate the functionalities of the prototypes on pan-European and Russian systems, most test cases will consist in either:

- an EHV model of UCTE and TEIAS systems (around 10000 buses), possibly enriched with step-up transformers (around 13000 buses) and load transformers and HV capacitors (around 17000 buses)
- a model of IPS-UPS system
- an interconnection of the 2 previous systems, either through AC or DC links

Some more specific test cases will be used for state estimation in order to include a test case with PMU data (REE system), a test case for two-step state estimation with real measurements (merging of REE,REN and RTE snapshots), and a test case for substation level state estimation. Scenarios will basically consist in measurements sets. The focus will be on assessing the quality of estimates and their robustness to bad data.

Steady state optimization scenarios will basically consist in defining an objective function together with limits, control variables , and a list of contingencies for SCOPF. Test procedures will focus on assessing the results quality and robustness when using the innovative functionalities: contingency filtering, network compression, and discrete variable optimization.

Dynamic simulation scenarios will consist in sequence of events. For the full accuracy prototype, we will focus on checking the system trajectory, including complex equipments models, against the exact mathematical solution, while comparing performances with V4.4 of Eurostag. For the DSA prototype, the focus will be on performance, while ensuring a satisfying fidelity within normal operational limits and a conservative behaviour outside. For DTA, the focus will be on checking fidelity and the correct handling of detailed topology models and automata, while ensuring real-time execution.

Table of content

1 Scope of the document

This document describes the administrative procedures, the test cases specifications, and the test procedures needed to develop customized test cases in T6.3 and to run the test of the PEGASE software in T6.4.

The report is written in accordance with the information provided in the document "*Specification and Architecture the identification of the needs"* delivered by WP1 which defines the requirements for PEGASE software packages, which is the main input for this task.

The main objective of this document is to define a validation methodology for all functions developed in the project: State Estimation, Steady State Optimization, Full Time Scale Simulation, Quasi Steady State Simulation, DTS and MMI. This specification will be an input to the test scenario building task.

2 Organization of the verification and validation process

The testing to be done in WP 6 concerns only validation activities. That means that the tests to be done as part of T6.4 will check the different software packages against the requirements stated in WP1, but the verification activities defined in 2.1 remains a responsibility of each WP teams.

2.1 Verification activities

The verification activities for the WP2, WP3 and WP4 will be carried out by the WP teams. Each WP team will be responsible for conducting the activities and tasks for verifying the software. The verification activity includes:

- Verification of software. The software architectural design and the software detailed design shall be verified considering:
  - Internal an external consistencies
  - The design is correct with respect to requirements and interfaces
  - The design implements proper sequence of events, inputs, outputs interfaces, logic flow, error definition and recovery.
  - The chosen design can be derived from the requirements
- Verification of the code
  - The code is traceable to design and requirements, stable, correct.
  - Some alpha tests are carried out to verify that the code implements proper event sequence, consistent interfaces, correct data and control flow, completeness.
  - Feasibility of software testing
- Verification of software documentation. The documentation shall be verified considering the criteria listed below:
  - The documentation is adequate, complete and consistent

It shall be responsibility of the WP leaders to organize the verification activities and produce the reports of the verification activities to the validation Task 6.4 leader.

2.2 Validation activities

The validation process shall be done by the validation team for each of the software developed in WP2, WP3 and WP4. This team carrying out the validation process will consist of the following persons:

- The Task leader of the T6.3 here on the Testing Manager
- A member of the T6.4 team for each WP.
- The leader and key persons of the WP which software is being testing
- A member of the STC of the project
- Technical staff needed to run the test.

The systems shall be evaluated to ensure they are consistent with the baseline requirements from WP1 (included in Annex 1). For each software component to be validated, a set of test cases and test procedures shall be carried out. The validation activity shall take into consideration:

- Testing with stress, boundary and singular inputs
- Testing the software for its ability to isolate and minimize the effect of errors
- Testing that the software product performs successfully in a representative operation environment.

2.3 Validation plan structure

2.3.1 Definitions

A **test case** is a network description, in its structure and state, enriched with the necessary data and converted in the appropriate format, so that it is ready-to-run for a given prototype. Thus test cases are only the base **data** that will be used to build test scenarios. They differ from what is generally called a “test case” in software engineering.

A **test scenario** is a complete set of data and sequence of actions given as an input to the tested prototype, in order to exhibit one or several of its functionalities. They will be based on the test cases customized in Task6.3. They can consist for example in: a particular set of measurements or erroneous data for state estimation, a sequence of events for dynamic simulation, the definition of an optimization problem for state estimation…

A **test procedure** is a sequence of actions designed either to validate some prototype feature against the requirements, or to assess some characteristic of the prototype (performance, robustness, …). They will obviously be based on the execution of test scenarios previously designed. They need to clearly define:

- Test purpose: can be either validating a predefined requirement, or assessing some characteristic of the prototype (limits, performances, robustness…)
- Step-by-step specification of the method used to test the intended feature or characteristic, including: scenario specification, generation and presentation of the outputs…
- How to assess the quality of the generated outputs

While **main tests** aim at evaluating the ability of the prototypes to meet the requirements on the main features, **robustness tests** are intended to assess their limits, in terms of functionalities, quality or performance.

2.3.2 Validation plan structure

For all prototypes to be tested, the validation plan will have to provide the following elements:

- Traceability table relating requirements to prototype features or components
- Specifications for test cases
- Specifications for main test procedures
- Specifications for robustness test procedures

Depending on cases, quality assessment methodologies may be specified for all outputs of a prototype, or for individual test procedures.

3 Administrative Procedures

3.1 Anomaly reporting and resolution

Everything that does not conform to the requirements will be considered as anomaly. For each anomaly detected during the validation process, the Testing Manager shall report the problem to the corresponding WP leader and, depending on the severity of the anomaly, to the STB. Annex 2 includes an anomaly reporting template.

It is responsibility of the WP leader to enforce resolution of the anomaly and communicate to the Testing Manager the resolution of the anomaly. The WP leader should document the resolution of the anomaly and inform the Testing Manager of the resolution timing. The anomaly resolution report should include the following information:

- Reference to the anomaly identification report
- The name of the person(s) being responsible for the solution
- The description of the action to solve the anomaly
- The list of all elements affected by the changes(s)
- A description of the procedures in order to validate the change(s).

3.2 Task Iteration Policy

If during the validation process problems are encountered, concerning the correct working or non-conforming to the requirements, it has to be decided whether the validation task needs to be performed again because an input has changed, or not,. Guidelines for the following cases are:

- The testing team was unable to complete their task because of errors in the software or the output of the software does not comply with the requirements. In this case the Testing Manager should report the problem according to the anomaly and resolution process. Once the problem is solved the testing should be repeated. If necessary, the WP6 leader should schedule extra man-hours.
- A test procedure was not satisfied during the validation tasks. Depending on the severity of that test, the Testing Manager shall decide if the entire testing or only a part of the testing needs to be redone, or even that no action is taken. The decision shall be approved by WP6 leader.

3.3 Deviation Policy

This section describes the procedures for deviating from the validation plan and defines the authorisation required for approval of deviations, which have to be approved by the Coordinator.

The procedures described in this document should be followed. However if in the Testing Manager opinion, this endangers the completion of the project, then the Testing Manager should request to the WP6 leader the deviation of the procedures. The request for deviation should include:

- Requirement identification.
- Task identification.
- Deviation rationale.
- Impact on project objectives.

The WP6 leader shall submit the request to the Coordinator with a recommendation on the decision to be taken. It is the responsibility of the Coordinator to accept or reject the deviation request.

3.4 Software verification reporting

The results of the meetings, the testing activities will be reported and checked by the T6.4 leader, the WP6 leader, and the Coordinator, as a result of a test. It contains the reference test plan, the problems discovered during the testing and the solutions found, and the acceptance or disapproval of the software.

4 Validation Plan for State Estimation

The Validation Plan aims at proving the fulfilment of the requirements defined by WP1 for the applications developed within WP2, WP3, and WP4. As a general premise for the validation plan herein enclosed, the algorithms developed should not be modified or tuned according to the test case required. If so the software verification reporting should mention it.

We distinguish main tests from robustness tests. Main tests evaluate the ability of the prototypes to meet the requirements on the main features. Robustness tests evaluate the limits of the prototypes, in terms of functionalities, quality or performance.

This section includes the requirements and modules that need to be validated in T6.3 and T6.4 for the State Estimation software developed in WP2.

4.1 State estimation traceability table

The traceability table traces each software requirement to a component. In the first column the software requirements for State Estimation are listed. In the second column the component that implement them are listed.

Requirement	Component
The number of nodes to be estimated will be at least 5.000.	Two step estimation algorithm
The state estimator should be two step estimation. First step at TSO level, second step to ETN level.	Two step estimation algorithm (geographical decomposition)
The system should be parallelizable.	The Two step estimation algorithm (geographical decomposition) is parallelizable by nature
TSO state estimator should remain independent. The coordination at ETN should be done on a central computer.	Two step estimation algorithm (geographical decomposition)
The state estimator should be improved with the use of Phasor Measurements Units (PMUs)	Phasor Data Concentrator Two step estimation algorithm
The state estimator should allow redundancy capacity and filtering capability.	Two step estimation algorithm(Decomposition Architecture)
The state estimator should be able to ensure consistency between topology and measurements.	Substation Local Estimator TSSE topological changes in an invariant bus-branch structure
The state estimator should be designed to provide a solution at substation level.	Substation Local Estimator
A topology processor at substation level should be developed.	Substation Local Estimator
The topology processor should be able to detect topology errors.	Substation Local Estimator

Requirement	Component
The state estimator should perform PMUs data validation.	TSO SE with PMU improvement algorithm
The state estimator should deal with voltage and current phasors.	TSO SE with PMU improvement algorithm

4.2 Test case specification for state estimation validation

The main objective of the State Estimation testing is to validate the two step algorithm for estimating the state of a large system operated by various TSO and to improve the state estimation process by including PMU data.

The TSSE with geographical decomposition will be tested on three test cases:

- an EHV model of UCTE+TEIAS system (voltages above 100kV only),
- an EHV network built from a snapshot of REE, REN and RTE.
- an EHV model of IPS-UPS system

The measurement for these test cases will be created generating pseudo-measurements from a load flow solution or from the state estimation solution.

To validate the improvement brought by PMUs to the conventional TSO SE algorithm, a REE 400/220 kV system will be used. This case will be based on real measurements from traditional RTUs and PMUs measurements synchronized with the RTU measurements.

Finally the substation level State Estimation will be tested on a detailed topology of a complex substation and a solution of the SE for that substation including status of circuit breakers, disconnectors and analogue measurements will be estimated.

4.2.1 Test case EHV UCTE+TEIAS

a) Test Case Identifier: **EHV UCTE+TEIAS**

This case will be used for testing the TSO decomposition scheme of the TSSE algorithm developed in WP2. The decomposition approach where independent state estimations are performed at TSO level and a second step wherecoordination of that state estimation is done.

b) Case Specification:

For testing the two step algorithm a large network of at least 10.000 thousand nodes corresponding to UCTE system plus Turkey system will be used. The network will include at least network voltages of 110 kV and higher.

The network will be a branch/bus model including at least the following elements:

- OLTC transformers
- Lines
- Buses
- Capacitor and Shunt Reactances
- Phase Shifter Transformers
- HVDC links
- Electrical borders and tie lines connecting the different TSOs clearly identified in the network model

- Nodes connected via coupling devices will be merged
- Only the main connected part will be considered

On this network a load flow will be calculated. This Load flow solution will be a reference case for the TSSE solution. From this load flow solution several sets of pseudo-measurement are generated that ensure complete observability of the system, with different measurement patterns. The pseudo-measurements sets (voltage and current magnitudes, angles, flows (P,Q) and tap and shifter positions) will be the input for the TSSE algorithm for any of the architecture to be tested.

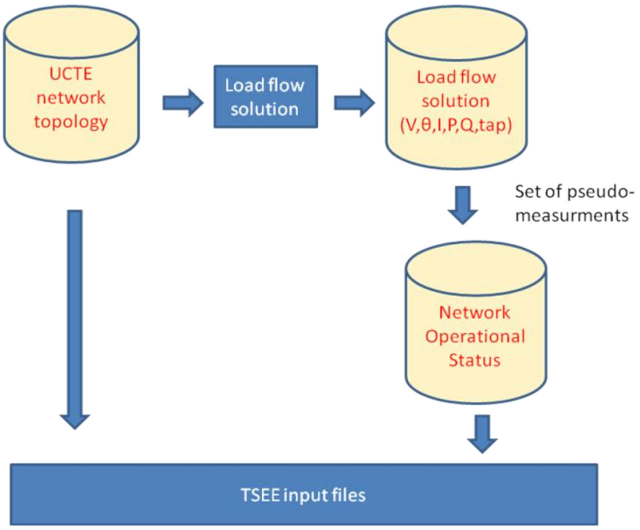


Figure 1. Large scale test case architecture

c) Output Specification:

- TSSE solution for the two architectures.
- Standard deviation of the measures and of the estimated values.
- Gain matrix at the end of the state estimation process.
- Comparative assessment of the solution versus a known system state (load flow solution/ result of a state estimation).
- Performance report including:
  - Iterations for a given accuracy
  - Cost of a single iteration
  - Number of bad data detected
  - CPU time execution for each part for a given accuracy.

4.2.2 Test case 3 TSOs

a) Test case identifier: 3TSO

The test case consists of a model formed by the union of the snapshots of the state estimators of three TSOs (REE, REN and RTE). It represents a real network.

b) Case specification:

This case is similar to the previous ones. The architecture of this case is exactly the same as the previous one with the only difference of having the real results of the state estimation as initial load flow solution.

c) Output specification:

- TSSE solution for the two architecture.
- Standard deviation of the measures and of the estimated values.
- Gain matrix at the end of the state estimation process.
- Comparative assessment of the solution versus a known system state (load flow solution/ result of a state estimation – case bis).
- Performance report including:
    - Iterations for a given accuracy
    - Cost of a single iteration
    - Number of bad data detected
    - CPU time execution for each part for a given accuracy

4.2.3 Test case. TSO SE PMUs improvement

a) Test case identifier: State Estimation SEPMUI

This case will be used for testing the improvement in the traditional SE algorithm by introducing the PMU technology. For this case real time conventional measurements and synchronized stream of PMU) will be used.

b) Case specification:

The network to be used for this case will be the REE network 400 and 220 kV network. For this network a snapshot of the REE SCADA measurements will be used. The network measurement will be mapped to a bus and Branch model including:

- OLTC transformer,
- Lines
- Buses
- Capacitor and Shunt Reactances
- The lower voltage levels not included in the case should be modelled as equivalent load transformers with the load connected to the secondary (low voltage level of the transformers)
- Electrical borders and tie lines connecting the different TSOs clearly identified in the network model

The SCADA snapshot will be linked with a stream of synchronized PMUs measurements (voltage and current phasors) stored at the PDC. At least 32 PMUs will be used in the case with a PMUs rate of sampling of 1 phasor/second.

As part of the system a set of weight for the PMU measurements will be provided. That set of weights will be fixed during the process of the State Estimation.

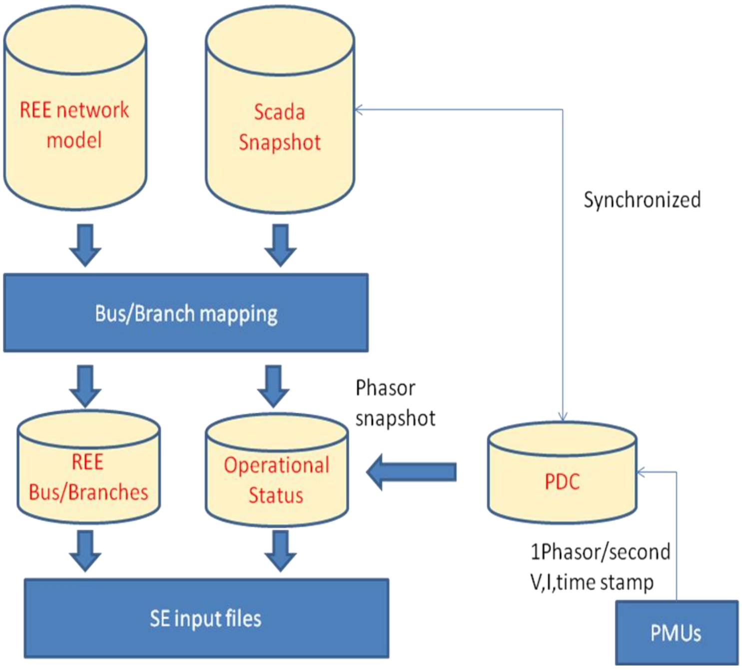


Figure 2. SE improvement using PMUs case architecture.

c) Output specifications:

- SE solution using PMUs measurement.
- SE solution without using PMU measurements.
- Statistical value of the χ^2 for the SE errors versus PMUs weights values.
- Performance report including:
    - Iterations for a given accuracy,
    - CPU time execution for a given accuracy

4.2.4 Test case Substation Local SE

a) Test case identifier: SLSE

This case will be used for testing the SE algorithm at the substation level and the topology processor developed in WP2.

b) Input specifications:

- A detailed substation topology will be used. The substation will have two voltages levels and both voltages level will be one and a half circuit breakers configuration. For that substation a connectivity solution from the SE will be used. That solution will include the status (open/closed) of circuit breakers and disconnectors and analogue pseudo-measurements.

From that SE solution a set of pseudo-measurements will be generated as the input for the substation level SE

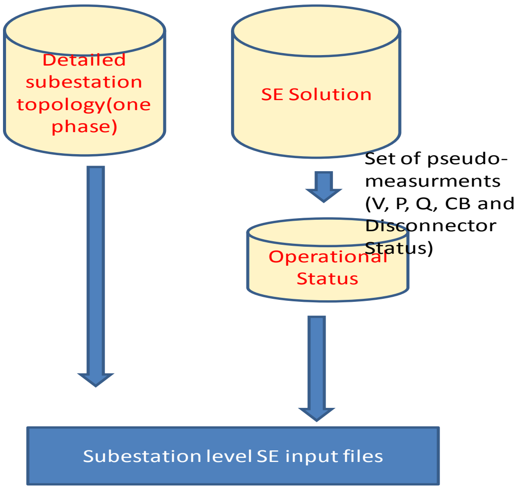


Figure 3. Substation level SE case architecture.

c) Output specifications:

- Substation State estimation solution including connectivity and status of the circuit breakers and disconnectors.
- Comparative assessment of the solution vs the SE solution.
- Outliers and topology errors report.

4.3 Test procedures for state estimation

This section defines the test to be carried out by T6.4 in order to test the different software modules produced by WP2, WP3, and WP4. Each test give a brief description of the purpose and include the case to be used from those defined in section 4.1.2 and the step to be followed by the testing team of T6.4

4.3.1 Test procedure 1. TSO decomposition Approach TSSE with noise and outliers

a) Test procedure identifier: TSSE TSO decomposition approach

b) Test purpose:

The purpose of the test is to demonstrate the feasibility of a hierarchical state estimator to solve a large transmission network as the European Transmission Network, assessing the accuracy and the performance of the state estimator algorithm. The test will follow a geographical system decomposition according the different TSOs networks. This test will consider that the network of each TSOs include the tie-lines connecting to other TSOs.

c) Test case:

Three test cases will be used in this procedure:

- EHV UCTE+TEIAS .
- 3 TSOs.

d) Procedure steps:

- The network defined in the case will be divided to create different TSO network subsystems, including the tie lines with neighbouring TSOs. Each TSO subsystem will have a different angle reference. In any case observability of each of the TSOs subsystems must be ensured during the decomposition process.

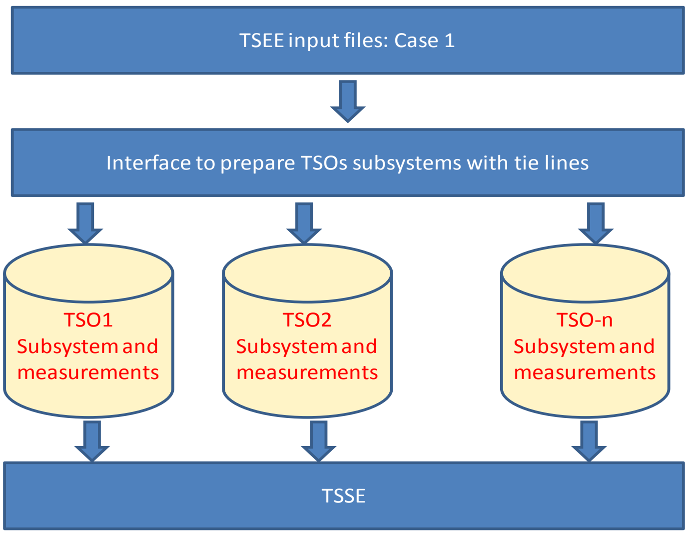


Figure 4. Interface to create the TSO network files for the Decomposition Scheme.

- For each TSO network the pseudo-measurement of the case will be associated to that network. For tie lines power flows pseudo-measurements and injection at boundary buses are included.

  In order to include the quality and noise of the measurement a white noise Gaussian noise will be added to each measurement to simulate quality of measurement. That noise will have different standard deviation for voltages and currents.

Outliers incorporated to the set of measurements for each TSO.

- A solution for the ETN network of the case will be found.
- The quality of the TSSE solution will be compared with the load flow solution (with the state estimation solution for 3TSO test case) of the case by comparing the state vector with the TSSE state vector (voltages and angles) for each TSO subsystem and for the TSO border nodes. Additionally the detected outliers by the TSSE should be the compared with the outliers introduced before running the SE and they must be the same.
- Several TSSE solutions will be performed with different measurement patterns.
- A test report will be prepared according to the annex 2.

4.3.2 Test procedure 3 Improvement of SE with PMUs

a) Test procedure identifier: PMUs Improvement

b) Test purpose:

The purpose of the test is to demonstrate the improvement in the traditional TSOs SE by incorporating the phasor measurements both voltages and currents.

c) Test Case: TSO SE PMUs improvement

d) Procedure steps:

- Run the traditional SE and get a SE solution for the case input. Evaluate for this solution the value of the objective function and the standard deviation associated to the state vector solution.
- Fix the weight of the PMUs measurements to be used for a new estimation using the PMUs data.
- Run the traditional SE incorporating the PMUs measurements and evaluate the value of the objective function and the standard deviation associated to the state vector solution.
- Compare the value of the objective function and standard deviation for the solution for both solutions. The improvement in the SE solution using PMUs should be reflected in a reduction in both variables in the case of the SE using PMUs.
- Repeat the SE process using PMUs by giving different weights to the PMU measurement. At least 10 different weights should be used. Compare solution for each weight with the base solution to analyze the impact of this variable on the quality of the SE solution.
- Prepare a test report according to the annex 2.

4.3.3 Test procedure 4 Local Substation SE

a) Test procedure identifier: Substation SE.

b) Test purpose:

The purpose of the test is to demonstrate the feasibility of the state estimation solution at substation level using measurements form PMUs and other IED used in a substation. This local SE will be able to estimate a state vector for the substation including switching devices status.

c) Test case: Substation Local SE

d) Procedure steps.

- From the input files of the test case generate a set of pseudo-measurements phasors.

- In order to represent the quality and noise of measurements a Gaussian noise will be added to each measurement. That noise will have different standard deviation for analogue pseudo-measurements and PMUs pseudo-measurements.
- Introduce topological errors by changing the status of the switching devices.
- Run the substation SE.
- The quality of the substation SE solution will be compared with SE solution of the case by comparing the state vector in both cases (voltages and angles and switching devices status). Additionally the detected topological errors by the Substation SE should be the compared with the topological errors introduced as part of the test. Both errors should be the same.

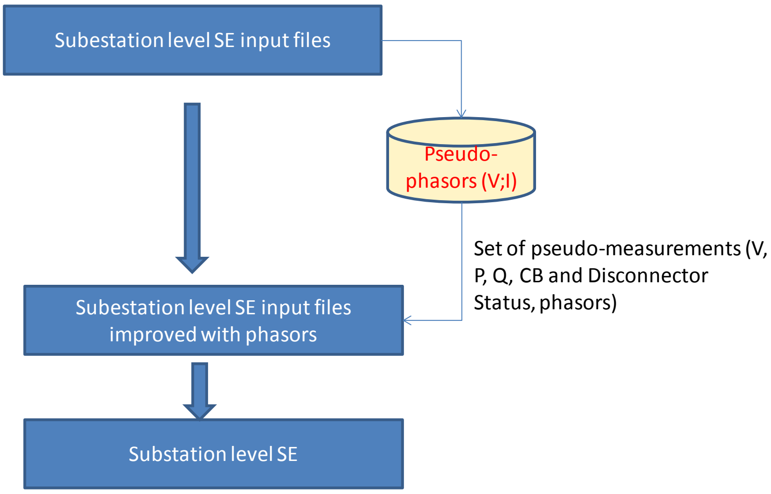


Figure 5. Test bed for Substation SE with PMUs.

4.4 Test procedures for robustness of state estimation

4.4.1 Test procedure 2. TSO decomposition Approach TSSE with noise and outliers and low measurement redundancy at TSOs borders

a) Test procedure identifier: **TSSE decomposition approach**

b) Test purpose:

The purpose of the test is to demonstrate the robustness of TSSE decomposition approach in case there is a low redundancy of the measurements in the borders between the different TSOs subsystems.

c) Test case:

Three test cases will be used in this procedure: the EHV UCTE+TEIAS**,** and the 3 TSOs.

d) Procedure steps:

- The network defined in the case will be divided to create different TSO network subsystem, including the tie lines with neighbouring TSOs. Each TSO subsystem will have a different angle reference. In this process the set of pseudo-measurements of the buses and tie-lines connecting each of the TSOs subsystems will have a low redundancy. By "low redundancy" we mean that a maximum of 1/3 of the pseudo-measurements will be available to the TSSE from the border nodes and tie-lines connecting the TSOs, but ensuring, in any case complete observability of each TSOs subsystem.

- For each TSO network the pseudo-measurement of the case will be associated to that network. For tie lines power flow pseudo-measurements and injection at boundary buses are included.

  In order to represent the quality and noise of the measurement a Gaussian noise will be added to each measurement. That noise will have different standard deviations for voltages and currents.

- Outliers incorporated to the set of measurements for each TSO and specifically on the TSO border nodes.
- A solution for the ETN network of the case will be found using the TSSE.
- The quality of the solution will be compared with the load flow solution of the case by comparing the load flow (the state estimation solution for the 3TSO test case) state vector with the TSSE state vector (voltages and angles) for each TSO subsystem. Additionally the detected outliers by the TSSE should be compared with the outliers introduced before running the SE and they must be the same.
- Several TSSE executions will be performed with different measurement patterns.

5 Validation plan for steady state optimization

The PEGASE research activities related to steady state optimization have as main objective to develop and validate new methodologies which will enable the optimization based calculations on the European Transmission Network (ETN) steady state model. One prototype will deal with a large number of continuous variables and another prototype developed will consider a more limited number of discrete variables only.

These new methodologies should be tested in the Tractebel Engineering prototype for which regards the continuous variables and on the RTE prototype for which regards the discrete variables.

The tests of the Tractebel Engineering prototype will be performed considering:

- a potentially huge amount of security contingencies representative of the targets of operating practices of all European TSOs in preventive and corrective mode.
- continuous variables.
- special devices like FACTS and HVDC .
- primary active and reactive control of the generating units.
- complex coordination control (automatic generation control).
- maximum computation time to be embedded in a global cycle of 15 min.

The tests of the RTE prototype will consider:

- a limited number of discrete variables : phase shifter taps or shunt capacitor switching or generating unit commitment (P=0 or $P \in [Pmin, Pmax]$).
- OPF only (no contingency)

5.1 Steady state optimization traceability table

The traceability table traces each software requirement to a component. In the first column the software requirements are listed. In the second column the components that implement them are listed.

Requirement	Component
R.3.1 should be able to handle around 10.000 nodes.	Tractebel SCOPF prototype and RTE MINLP prototype
R.3.2 Computation time for a given problem should not exceed 15 min.	
R.3.3 should be able to handle devices like FACTs, HVDC, AGC.	
R.3.4 should include a network compressor.	Tractebel SCOPF prototype
R.3.5 should be able to filter contingencies that are not binding for the problem.	
R.3.6 should be able to handle at least 10.000 contingencies	
R.3.7 should be able to handle automatic post-contingency regulations (frequency regulation, voltage regulation)	
R.3.8 should be able to handle simultaneously continuous and	RTE MINLP prototype

Requirement	Component
discrete variables	

5.2 Test case specification for steady state optimization

All test cases for steady state optimization should implement the following specifications:

- Nodes connected via coupling devices will be merged.
- Only the main connected part will be considered.
- Coherent limits on branches current
- Coherent limits on voltage operating ranges (by voltage levels – UCTE reference).
- Generation units limits for active generation: Pmin, Pmax, as well as frequency droop and cost.
- Generation units limits for reactive generation: Qmin, Qmax
- Transformer saturation coefficient.
- A load flow should be run and converge on the test case, within limits defined above. Its result should be stored in the test case, as a possible starting point for the prototypes.

5.2.1 Test case 1 EHV model of UCTE+TEIAS

a) Test case identifier: **EHVUCTETEIAS**

The test case is an EHV model of UCTE synchronously connected to Turkey including voltage levels above 100 kV (about 9.200 nodes).

b) Test case specification

In addition to the general specifications detailed above, this test case should implement the following specifications:

- EHV UCTE+TEIAS system.
- Generation units are aggregated at EHV level
- The system should contain special devices such as: phase shifter transformers, HVDC LCC and VSC links, FACTS (SVC and TCSC).

5.2.2 Test case 2 model of IPS/UPS

a) Test case identifier: **IPSUPS**

The test case is a model of IPS/UPS.

b) Test case specification

In addition to the general specifications detailed above, this test case should implement the following specification:

- EHV IPS/UPS system.
- Generation units are aggregated at EHV level

5.2.3 Test case 3 EHV model of UCTE+TEIAS with step-up transformers

a) Test case identifier: **EHVUCTETEIASWSUT**

The test case is an EHV model of UCTE synchronously connected to Turkey including voltage levels above 100 kV and step-up transformers (about 13.000 nodes).

b) Test case specifications

In addition to the general specifications detailed above, this test case should implement the following specification:

- Step-up transformers are described for all generation units.
- EHV UCTE+TEIAS system.
- The system should contain special devices such as: phase shifter transformers, HVDC LCC and VSC links, FACTS (SVC and TCSC).

5.3 Test procedures for steady state optimization

5.3.1 Test procedure 1 (TE prototype) : Network Compression

a) Test procedure identifier: NC

b) Test purpose:

Assess optimization results quality and computation performance when solving a SCOPF using compressed networks.

A SCOPF problem should be created including contingency constraints that affect the optimal result. An example is presented in the article “Large Scale Security Constrained Optimal Power Flow” [1]. The network compression should assess different compression ratios in order to evaluate thresholds leading to distort results.

c) Test case:

This test will be performed using the test case **EHVUCTETEIAS** specified in section 5.2.1 of this document.

d) Procedure steps:

Scenario 1: Quality assessment

- Set an optimization problem (objective function, voltage and power flow constraints).
- Choose a small number of contingencies that affect the optimal result (the optimal solution should be known in advance).
- First, solve the SCOPF problem with the full network (without compression).
- Define sets of thresholds to be used in the network compression procedure (sensitivity to voltage and power flow). Those thresholds impact directly the compression ratio.
- Second, solve the SCOPF problem with the compressed networks for the n sets of thresholds.
- Assessments for each set of thresholds:
    - Compare voltage and power flow profiles of the pre-contingency state for the active and inactive (reduced) region with respect to the full network optimization.
    - Compare the objective function result from the compressed SCOPF with the one from the full SCOPF.

- Simulate one incident in both optimal pre-contingency states. Compare nodal voltages and power flows issued from the full SCOPF and issued from the compressed SCOPF.
- Perform a security analysis on the pre-contingency state to make sure that security constraints are respected.

- For each optimization case the CPU time, the compression ratio of the compressed networks and the number of binding constraints should be noted.

Scenario 2: Number of contingency constraints assessment

- Set an optimization problem (objective function, voltage and power flow constraints).
- Create variants of the problem by adding 100, 1000, 5000 and 10000 contingencies. The selection of these contingencies is done using the contingency filtering method.
- Use the set of thresholds that presented the best ratio of results accuracy and network size, as well as a set of thresholds that gives a compression ratio slightly lower.
- Solve the SCOPF problem with those 2 sets of compression parameters for the n sets of contingency constraints.
- Assessments for each set of contingencies:
  - CPU times for each optimization.
  - Compare voltage and power flow profiles of the pre-contingency state for the active and inactive (reduced) regions between the 2 optimizations
  - Compare the objective function result between the 2 optimizations
  - Simulate one incident in both optimal pre-contingency states. Compare nodal voltages and power flows issued from the 2 SCOPFs.

5.3.2 Test procedure 2 (TE prototype) : Contingency filtering

a) Test procedure identifier: CF

b) Test purpose:

Assess optimization results quality and consistency, as well as computation performance when solving a SCOPF using contingency filtering techniques.

The objective function should progressively increase while adding contingencies, and modifications of the optimal state should be traceable to the binding contingencies resulting from the added contingencies.

c) Test case:

This test will be performed using the test case **EHVUCTETEIAS** specified in section 5.2.1 of this document.

d) Procedure steps:

- Set an optimization problem (objective function, voltage and power flow constraints).
- Create variants of the problem by adding 0, 100, 200…up to 1000 contingencies.
- Use the set of thresholds for Network Compression that presented the best ratio of results accuracy and network size.
- Solve the SCOPF problem with the compressed networks for the n sets of contingency constraints.
- Assessments:

- Objective function for each set of contingencies.
- CPU time for each set of contingencies.
- Number of contingencies selected as binding for each set of contingencies.
- Compare, for each set of contingencies, voltage and power flow profiles, as well as contingencies selected as binding, with the previous set of contingencies (slightly smaller).

5.3.3 Test procedure 3 (TE prototype): Power electronic devices: HVDC and FACTS

a) Test procedure identifier: PED

b) Test purpose:

Assess the correct functionality of power electronic devices such as HVDC and FACTS when embedded in large AC/DC network optimization problems.

c) Test case:

This test will be done using the test case **EHVUCTETEIAS** specified in section 5.2.1 of this document. On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

Scenario 1 - HVDC:

- Define an OPF problem where the actuation of HVDC has an impact on the optimum solution. Example: maximize the flow between two areas connected by an HVDC.
- The firing and extinction angles are to be used as control variables.
- Variants of the case shall be created by changing the limits of the firing and extinction angle in a way to restrain the optimal solution.
- Impose voltage and power flow constraints.
- The problem is solved for the different limits imposed to these control variables.
- At the end of each optimization it has to be observed if the HVDC presented the expected behaviour.
- For each optimization case the CPU time should be noted.

Scenario 2 - FACTS:

- Define an SCOPF problem where the actuation of one specific FACTS has an impact on the optimum solution.
- SVC example:
    - Set an objective function to maximize the individual range of the generators reactive power restrained by the loss of several generating units/several loads.
    - The SVCs should be the actuators to act in order to keep the desired reactive power alignment. Depending on the type of loss (generation/load) the SVCs will operate in injection or absorption.
    - Variants of the case shall be created by changing the limits of the SVCs control variables in a way to restrain the optimal solution.
    - Impose voltage and power flow constraints.
    - The problem is solved for the different limits imposed to these control variables.
    - At the end of each optimization it has to be observed if the specific FACTS showed the expected behaviour.

- For each optimization case the CPU time should be noted.

5.3.4 Test procedure 4 (TE prototype): Phase shifter transformers with variable impedance

a) Test procedure identifier: PST

b) Test purpose:

Assess the correct functionality of phase shifter transformers with variable impedance embedded in a large AC network.

c) Test case:

This test will be done using both the test cases 1 and 2 : **EHVUCTETEIAS** and **IPSUPS**. On these cases a load flow is run and the results must be in the input data.

d) Procedure steps:

- Define an OPF problem where the phase shifter transformer is needed to alleviate power flows crossing a specific area.
  - Example: minimization of power flows crossing Belgium acting only with the phase shifter transformers at the France/Belgium/Netherlands borders.
- Variants of the case shall be created by limiting the available tap positions of the PST.
- Impose voltage and power flow constraints.
- The problem is solved for the different limits imposed to these control variables.
- At the end of each optimization it has to be observed if the PST presented the expected behaviour (deviated flow with respect to available tap positions).
- For each optimization case the CPU time should be noted.

5.3.5 Test procedure 5 (TE prototype): Generator active limits (Pmin-Pmax)

a) Test procedure identifier: GAL

b) Test purpose:

Assess the optimizer capabilities to deal with generators maximum and minimum active power limits and frequency droop. In order to experience a significant frequency deviation it is proposed a limited number of generation units participate to the active power balance.

c) Test case:

This test will be done using the test case **EHVUCTETEIAS** defined in section 5.2.1 of this document. On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

Scenario 1: Pmax limit

- Define a SCOPF problem with the objective to increase the active power production of a group of generators. The remaining generators are declared as control variables.
- The problem is restrained by the loss of one or more generators declared in the objective function. The remaining generators will supply the required power according to their frequency droop.
- The problem should anticipate some generators to reach the maximum limit before meeting the required power. Consequently, the power that these generators could not supply will be again shared among the other generators.

Scenario 2: Pmin limit

- Define a SCOPF problem with the objective to maximize the active power flow crossing one region by increasing the generation in one side and decreasing the generation in the other side.
- The optimal solution should be met when the generators reach their minimum active power limit.

For both scenarios, it has to be observed if the generators indeed reach their minimum/maximum active power limit and if the active power lost due to the contingency is shared among the remaining generators according to their frequency droop.

5.3.6 Test procedure 6 (RTE prototype): Discrete variables I: phase shifter taps

a) Test procedure identifier: DVIPST

b) Test purpose:

Assess the algorithm capacity to find the same optimization solution having more or less discrete control variables available.

The aim is to compare the results of an OPF solved with n discrete control variables available to the results of an OPF solved with part of these discrete control variables fixed to the optimal solution. In both cases, the optimizer should find approximately the same solution. The discrete control variables will only be phase shifter taps as they would cause more significant impact on the optimal solution.

c) Test case:

This test will be done using the test case 1: **EHVUCTETEIAS**. On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

- Create an OPF problem involving the use of phase shifter taps only with the objective of minimizing the active generation from a given pattern.
- Impose voltage and power flow constraints as well as active and reactive power limits on generating units.
- The optimization problem should be solved several times considering:
  - all discrete control variables available .
  - part of the discrete control variables fixed to the optimal solution. In order to fix some control variables the result of the first optimization shall be used.
- Sensitivity analysis shall be performed by fixing different proportions of the total control variables (1/5, ¼, 1/3, etc…).
- The distance between the results of these optimizations and the first one represents an estimation of the MPEC NLP algorithm robustness.
- For each optimization case the CPU time and the number of fixed control variables should be noted.

5.3.7 Test procedure 7 (RTE prototype): Discrete variables II : reactive compensation banks

a) Test procedure identifier: DVIIRCB

b) Test purpose:

Assess the algorithm capacity to find the same optimization solution having more or less discrete control variables available.

The aim is to compare the results of a OPF solved with n discrete control variables available to the results of a OPF solved with part of these discrete control variables fixed to the optimal solution. In both cases, the optimizer should find approximately the same solution. The discrete control variables will only be reactive compensation banks.

c) Test case:

This test will be done using the test case 1: **EHVUCTETEIAS**. On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

- Create an OPF problem involving the use of reactive compensation banks only with the objective of minimizing the losses.
- Impose voltage and power flow constraints as well as active and reactive power limits on generating units.
- The optimization problem should be solved several times considering:
  - all discrete control variables available .
  - part of the discrete control variables fixed to the optimal solution. In order to fix some control variables the result of the first optimization shall be used.
- Sensitivity analysis shall be performed by fixing different proportions of the total control variables (1/5, ¼, 1/3, etc…).
- The distance between the results of these optimizations and the first one represents an estimation of the MPEC NLP algorithm robustness.
- For each optimization case the CPU time and the number of fixed control variables should be noted.

5.3.8 Test procedure 8 (RTE prototype): Discrete variables III : generation units start/stop

a) Test procedure identifier: DVIIIGUSS

b) Test purpose:

Check the ability of the RTE MPEC NLP prototype to deal with generation units start/stop (P=0 or P∈[Pmin, Pmax]), considering a limited number of units.

c) Test case:

This test will be done using the test case **EHVUCTETEIASWSUT** including step-up transformers (13000 nodes). On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

- Create an OPF problem involving the use of generating unit start/stop for a limited number of units only (no more than 100) with the objective of minimizing the distance to an initial active generation pattern.
- Impose voltage and power flow constraints as well as active and reactive power limits on generating units.
- The optimization problem should be solved several times considering:
  - all discrete control variables available .
  - part of the discrete control variables fixed to the optimal solution. In order to fix some control variables the result of the first optimization shall be used.
- Sensitivity analysis shall be performed by fixing different proportions of the total control variables (1/5, ¼, 1/3, etc…).

- The distance between the results of these optimizations and the first one represents an estimation of the MPEC NLP algorithm robustness.
- For each optimization case the CPU time and the number of fixed control variables should be noted.

5.4 Test procedures for steady state optimization Robustness

5.4.1 Robustness Test procedure 1 (TE prototype): Algorithm performance I

a) Test procedure identifier: AP I

b) Test purpose:

Assess the prototype algorithm performance (CPU time) with respect to the number of binding constraints in the OPF problem. In order to increase the number of binding constraints, the problem should be solved several times and at each time the constraints should have their limits narrowed. The optimal solution will change from case to case until it reaches unfeasibility.

The goal is to achieve a curve of CPU time versus number of constraints and verify if the algorithm presents a linear, quadratic or logarithmic behaviour. That test will be performed on TE prototype

c) Test case:

This test will be done using both test cases: **EHVUCTETEIAS** and **IPSUPS**.

d) Procedure steps:

- Create an OPF problem with a classical objective function - power flow maximization for example (maximize the active power flow crossing one region by increasing the generation in one side and increasing the load in the other side).
- Impose flow constraints on branches.
- Impose voltage constraints (UCTE or IPS/UPS standards).
- Progressively tighten voltage limits in a way to induce infeasibility, i.e. force the voltage at the generation side lower than at the load side.
- For each optimization case the CPU time and the number of active constraints should be noted.

The curve of CPU time vs number of binding constraints will give an estimation of the algorithm performance.

5.4.2 Robustness Test procedure 2 (TE prototype): Algorithm performance II

a) Test procedure identifier: AP II

b) Test purpose:

Assess the prototype algorithm performance (CPU time) with respect to the size of the system. The optimal solution will change from case to case until it reaches unfeasibility. The goal is to achieve a curve of CPU time versus number of nodes and verify if the algorithm presents a linear, quadratic or logarithmic behaviour.

c) Test case:

This test will be done using the three test cases.

d) Procedure steps:

- Create a SCOPF problem with a classical objective function - power flow maximization for example (maximize the active power flow crossing one region by increasing the generation in one side and increasing the load in the other side).
- Impose flow constraints on branches.
- Impose voltage constraints (UCTE or IPS/UPS standards).
- For each one of the three cases, define contingency sets of various sizes, and run the SCOPF.
- For each one of the three test cases, CPU time should be plotted against the number of binding contingencies. The three curves should then be compared.

5.4.3 Robustness Test procedure 3 (TE prototype): Algorithm robustness

a) Test procedure identifier: AR

b) Test purpose:

Assess the prototype algorithm capacity to find the same optimisation solution facing small perturbations of operating conditions. That test will be performed on TE prototype.

c) Test case:

This test will be done using both EHV UCTE+TEIAS and IPS/UPS test cases. On this case a load flow is run and the results must be in the input data.

d) Procedure steps:

- Set a SCOPF problem with a classical objective function (impose limits to voltage and power flow).
- Start from one point A and perform a first optimization to obtain point B.
- Than change the topology and perform a second optimization starting from point B to obtain point C.
- After that, change some constraints and perform a third optimization starting from point C to obtain point D.
- And finally, restore the initial topology and constraints and perform a last optimization starting from point D to obtain point E.

The distance between the results of this optimization (point E) and the first one (point A) represents an estimation of the algorithm robustness.

5.4.4 Robustness Test procedure 4 (TE prototype): Transformer saturation

a) Test procedure identifier: TS

b) Test purpose:

Assess the effect of transformer saturation in a SCOPF/OPF problem (Tractebel Engineering prototype). The optimization results of a SCOPF/OPF having transformers operating in the linear and in the non linear characteristic of the saturation curve are to be compared.

c) Test case:

This test will be done using the test case **EHVUCTETEIAS**. On these cases a load flow is run and the results must be in the input data.

d) Procedure steps:

- Define an OPF problem with the objective to increase the voltage profile in the system.

- Impose voltage and power flow constraints
- The OPF problem should be solved two times:
  - First time, with the network containing transformers with saturation characteristics. Be sure that the optimal solution found contains several transformers operating saturated.
  - Second time, with the network containing transformers without saturation characteristics.
- Both cases results are to be compared. Key indicators are: reactive power consumption (higher in case with saturation) and voltage profile (lower in the case with saturation).
- For each optimization case the CPU time should be noted.

5.4.5 Robustness test procedure 5 (RTE prototype): Discrete variables: generation unit start/stop

a) Test procedure identifier: DVRGUSS

b) Test purpose:

Check the ability of the RTE MPEC NLP prototype to deal with generating unit start/stop (P=0 or P∈[Pmin, Pmax]), considering a number of controllable units as large as possible.

c) Test case:

This test will be done using the test case **EHVUCTETEIASWSUT** including step-up transformers (13000 nodes).

d) Procedure steps:

- Create an OPF problem involving the use of generating unit start/stop only with the objective of minimizing the distance to an initial active generation pattern.
- Impose voltage and power flow constraints as well as active and reactive power limits on generating units.
- The problem will be solved with various sets of “controllable” units of increasing size.
- CPU time and the value of the objective function will be plotted against the number of controllable generation units. The objective function should be decreasing while increasing the number of units.

6 Validation plan for Full Dynamic Simulation

6.1 Full Dynamic Simulation traceability table

Requirement	Component
R.4.1 should be able to handle at least 10.000 nodes.	Full accuracy dynamic simulation prototype
R.4.2 should be able to handle VSC, HVDC and FACTS models.	
R.4.3 should be able to handle user defined models of protection systems.	
R.4.4 should be able to simulate the behaviour of complex generating unit controls	
R.4.5 should be able to integrate models describing fast numerical controls.	
R.4.6 should be able to handle 125.000 state variables. Detail: A simulation with fast dynamic transients during 20 s on a 125 000 variable system (at least 10 000 nodes) should run within 15 minutes (minimal requirement). Target : 5 minutes	
R.4.7 Integration algorithm should be A-stable.	

The mentioned timing constraints exclusively concern the simulation phase: Eurostag initialization and post-processing times will not be considered. Except otherwise specified, the CPU time mentioned hereafter are relative to a 20 seconds simulation.

6.2 Full accuracy dynamic simulation general methodology

6.2.1 Definition of fidelity and determination method

The fidelity of a simulation is defined with respect to the mathematical trajectory of the system. Fidelity as defined here then depends on the quality of the numerical process used to solve the system dynamic equations.

It is important to make clear that there is no reference here to the real system behaviour.

We will hereafter distinguish 2 notions related to simulation accuracy. The first is an input for the simulation, the second will be representative of the actual accuracy of the simulation:

- **Algorithm accuracy (input of the simulation):**

  In a variable step size with error control algorithm, the trajectory of the variables depends on the algorithm accuracy set by the user as a parameter of the numerical integration method. The more accuracy is required, the more CPU is required to solve the system but the closer the trajectory is to the reference.

- **Physical accuracy (output of the simulation):**

  Physical accuracy is defined as the maximum difference, for a given variable and over the simulation duration, between values of that variable in the reference trajectory on one side, and in the simulation output trajectory on the other side.

  We may for instance consider physical accuracy on one machine speed or voltage, critical clearing time etc…

Fidelity of the simulation will be assessed using the physical accuracy as numerical criterion. A method is proposed hereafter to find, by iteration, the minimum algorithm accuracy required to attain the required physical accuracy (10-4 pu on voltage or frequency for instance) :

- simulate the system with the highest algorithm accuracy (lowest error value)
- select the node at which the voltage has the greatest excursion (max-min) and the machine which speed has the greatest excursion during the simulation (without or before going to zero),
- simulate the system with the lowest algorithm accuracy (highest error value)
- compute for each time step (of stage 1) the voltage and speed difference at the selected components between the maximum accuracy trajectory and the current simulation
- if the maximum difference is greater than the required threshold for either the voltage or the speed, try a algorithm accuracy 10 times higher (10 times lower error value)
- When the voltage and speed difference are below the threshold, check if the simulation with algorithm accuracy 5 times lower (5 times higher error value) still keeps the differences below the threshold.

6.2.2 Criterion

For the physical accuracy requirements specified hereafter, typical scenarios should be simulated within 5 minutes whereas complex scenarios can take up to 15 min of CPU time:

- minimum algorithm accuracy ; 10-3
- maximum algorithm accuracy ; 10-8
- physical accuracy requirement on rotor speed : 10-4
- physical accuracy requirement on voltage : 10-4

The requirements on physical values are consistent with the accuracy expected from expert users of Eurostag software.

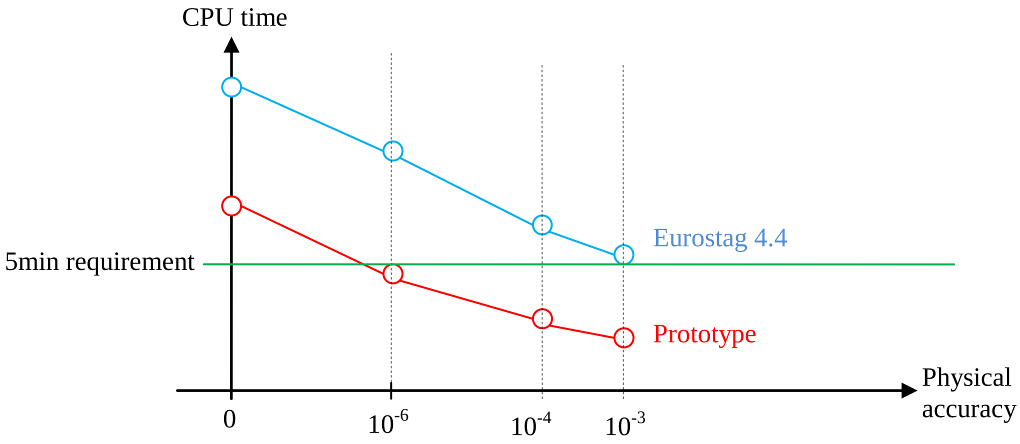
The CPU time requirement will be defined in each one of the test procedure specifications below.

6.2.3 Accuracy/Performance ratio evaluation

In addition to that criterion, and in order to emphasize the improvements of the algorithms, accuracy/performance ratios will be compared between simulations performed in the prototype and simulations performed in its commercial version : Eurostag 4.4.

In practice, **for all scenarios** a curve of the CPU time against physical accuracy should be plotted, for the prototype and Eurostag 4.4. That will be done using several values of algorithm accuracy as input for the simulation (that work can be done while determining the required algorithm accuracy from the method described above for instance).

The prototype curve should remain under Eurostag 4.4 curve:



Example of accuracy/performance assessment plot

6.3 Test case specification for Full accuracy Dynamic Simulation

6.3.1 Test case 1 Full UCTE+TEIAS model

a) Test case identifier: **UCTE+TEIAS**

b) Input specifications

- A bus and branch model of at least 10.000 nodes corresponding to different TSOs.
- The total size of the system is around 125.000 state variables, including automata.
- Generation controllers: simple avr, pss, governor controllers except for a small subset equipped with complex ones :
    - Rotor current limit management (long term dynamics). Timer interacting with the system through a commutation. Over-excitation, Under-excitation (with commutations).
    - Fast valving.
    - Coupled processes (ex. : fast valving+common water pipe with hammer effect, combined cycles) .
    - Other commutations (integral constraints).
- EHV Load model with 2 level transformers and motors at lowest voltage level and frequency dependency of the load.
- FACTS: SVC, TCSC, STATCOM.
- HVDC : HVDC links (LCC, VSC)
    - Several LCC and VSC links embedded in the European grid with several degrees of strength of the parallel AC system.
- Automata: on-load tap changers, phase shifter controller, tap changer blocking, under-voltage load shedding, under voltage generator tripping, automatic Switching of Capacitors/Self-Inductances, Secondary Voltage Control, AUFLS and SPSs.
- Wind farm models off shore and on-shore.
- Load model with voltage and frequency dependency, AGC, etc… should be modelled.

- Under Frequency Load Shedding (with Rate Of Change Of Frequency and frequency thresholds) relays.

6.3.2 Test case 2 IPS/UPS

a) Test case identifier: **IPSUPS**

b) Input specifications

The test case is a dynamic model of the IPS/UPS system

6.3.3 Test case 3 UCTE+TEIAS+IPS/UPS

a) Test case identifier: **UCTETEIASIPSUPS**

b) Input specifications

The test case is a dynamic model of the UCTE, TEIAS and IPS/UPS system all synchronously connected.

6.4 Test Procedures for Full accuracy Dynamic Simulation

6.4.1 Test procedure 1 Checking fidelity on machine stability and instability

a) Test procedure identifier: **CFMSI**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm.

c) Test case

d) This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

e) Procedure steps:

- Select a metallic 3-phase fault on the EHV grid close to generators.
- Search for the critical clearing time with a 10 ms accuracy with maximum algorithm accuracy.
- Simulate the marginally stable and unstable cases (unstable if one machine rotor angle goes beyond]-360°;360°[).
- Check the minimum CPU time is under 5 minutes in both cases. In case of unstability the simulation must be stopped before the usual 20 seconds to measure the CPU time.

6.4.2 Test procedure 2 Checking fidelity on voltage collapse

a) Test procedure identifier: **CFVC**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm on a voltage collapse scenario.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) 1.3 Procedure steps.

- Create a voltage collapse scenario activating at least 10 LTC and 2 generator field current limiter.
- generator field current limiter may be modified specifically for this scenario.

- Check the minimum CPU time is under 5 minutes.

6.4.3 Test procedure 3 Checking fidelity on generator set-point change

a) Test procedure identifier: **CFGSPC**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm on a generator set-point change.

c) This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS

d) Procedure steps.

- Select a generator in a low short-circuit power area.
- Modify the generator voltage set-point drastically (+10%) and observe the system during 10s after.
- Check the minimum CPU time is under 5 minutes.

6.4.4 Test procedure 4 Extreme case voltage collapse, system splitting and successful under-frequency load-shedding

a) Test procedure identifier: **ECVCUFLS**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm on a scenario with large voltage and frequency deviations.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Increase the load in a peninsular area compensated in active power from far away in the European system until provoking a slow voltage collapse.
- Key lines linking the peninsula to the rest of the continent are tripped, severing the system in two parts.
- The frequency then decreases in the peninsula triggering the under-frequency load shedding.
- The amount a load shed is enough and fast enough to stabilize the frequency in the peninsula.
- Check the minimum CPU time is under **15** minutes.

6.4.5 Test procedure 5 Extreme case thermal cascade leading to a system splitting, under-frequency load shedding and re-synchronization

a) Test procedure identifier: **TCLUFLS**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm on a scenario with large frequency deviations and re-synchronization.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Create a European wide East-West transit increase the load in the West and decreasing it in the East; modify some thermal limits so that this situation is acceptable in normal operation.
- Trip one or two major East-West lines, in order to trigger a thermal cascade splitting the whole European system in two big parts (key lines thermal limits may be modified to reach this goal).
- The frequency then decreases in the Western part triggering the under-frequency load shedding.
- The among a load shed is enough and fast enough to stabilize the frequency in the Western part.
- After stabilization, the two systems are re-synchronized successfully.
- Check the minimum CPU time is under **15** minutes.

6.4.6 Test procedure 6 Complex controls

a) Test procedure identifier: **CC**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with complex controls.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Tune the controls and create a scenario which activates commutations especially on long term dynamics, machine coupling (ex.: fast valving + common water pipe with hammer effect, combined cycles).
- If possible create global effects (out of step with cascading effect due to automata, voltage collapse induced by "rotor current limit management") activated with theses controls and not without.
- Check the minimum CPU time is under 5 minutes.

6.4.7 Test procedure 8 short circuit close to an embedded VSC HVDC

a) Test procedure identifier: **SCHVDC**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with a VSC HVDC embedded in the AC system.

c) Test case

This test will be done using the test case 1. UCTE+TEIAS .

d) Procedure steps.

- create a short circuit at one end that activates commutation inside HVDC control with influence the other side and has an influence on the AC network.
- Check the minimum CPU time is under **5** minutes.

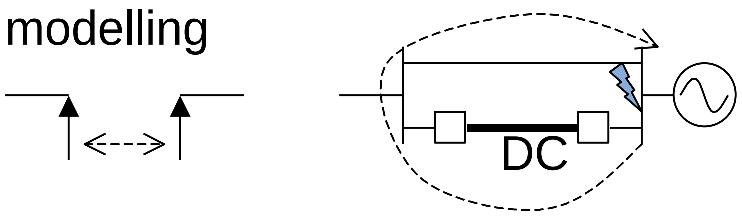


Figure 6. Short Circuit close to HVDC link model.

6.4.8 Test procedure 9 Changing the AC grid impedance in parallel with an embedded VSC HVDC

a) Test procedure identifier: **ACIHVDC**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with a VSC HVDC embedded in the AC system.

c) Test case

This test will be done using the test case 1. UCTE+TEIAS.

d) Procedure steps.

- HVDC link in parallel with a AC line; short AC line opened at the beginning and closed during the simulation.
- Check the minimum CPU time is under **5** minutes.

6.4.9 Test procedure 10 Centralized control by macroblocks

a) Test procedure identifier: **CCMB**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with a centralized control.

c) Test case

This test will be done using the test case 1. UCTE+TEIAS.

d) Procedure steps.

- Model a Secondary Voltage Control with macroblocks all over the system.
- Create a scenario activating a large number of centralized controls.
- Check the minimum CPU time is under **5** minutes.

6.4.10 Test procedure 11 Inter Area Oscillation

a) Test procedure identifier: **IAO**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with a badly damped inter-area oscillation.

c) Test case.

This test will be done using the test case 1. UCTE+TEIAS.

d) Procedure steps.

- Create a European wide poorly damped inter-area oscillation at frequency under 0.3 Hz: large power exchanges; topology with some disconnected tie-lines; tripping of a large generation.

- The PSS settings or the load model may be modified in order to lower the system damping.
- Check the minimum CPU time is under **5** minutes.

6.5 Robustness Test Procedures for Full accuracy Dynamic Simulation

6.5.1 Robustness Test procedure 1: Frequency dependency of impedances

a) Test procedure identifier: **FDI**

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with the dependency of impedances to the frequency.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) 5.3 Procedure steps.

- Activate the option in Eurostag and provoke a large frequency deviation.
- Analyse the difference of the results between prototype and Eurostag simulation.
- Check the minimum CPU, to be compared with a 5 minutes goal.

6.5.2 Robustness Test procedure 2: Transformers saturation

a) Test procedure identifier: TFS

b) Test purpose

The purpose of the test is to demonstrate the fidelity of the full accuracy algorithm with transformers saturation.

This test will be done using the test case 1 and 2.

c) Procedure steps.

- Add transformers saturation, and analyse their influence on results of a over-voltage scenario.
- Check the minimum CPU, to be compared with a 5 minutes goal.

6.5.3 Robustness Test procedure 3: Extra large system

a) Test procedure identifier: XLS

b) Test purpose

The purpose of the test is to demonstrate the feasibility of a dynamic simulation on an extra large system.

c) Test case

This test will be done using the test case 3. UCTE+TEIAS+IPS/UPS.

d) Procedure steps

- Select a metallic 3-phase fault on the EHV grid close to generators.
- Search for the critical clearing time with a 10 ms accuracy with maximum algorithm accuracy.
- Simulate the marginally stable and unstable cases (unstable if one machine rotor angle goes beyond]-360°;360°[).
- Check the minimum CPU time in both cases, to be compared with a 5 minutes goal.

7 Validation plan for simplified dynamic simulation for Security Analysis (DSA)

7.1 Objectives of the simplified simulation validation

The test should demonstrate that:

- the simulation method ensures a minimal fidelity within normal limits (voltages within 0.8-1.2pu, frequency above load shedding threshold and below over-speed machine tripping).
- there is no numerical problems on badly damped fast dynamics.
- the simulation method ensures a conservative assessment of unacceptable scenarios (instability or out of operational range):
  - Unacceptable scenarios must be simulated as unacceptable.
  - Some acceptable cases nearly unacceptable could be simulated as unacceptable.
- the CPU performances are according to the requirements.
- the method allows user modelling for simplified simulation for DSA (R.4.16).
- when possible, available information is provided to help understand the cause of the divergence (bad data, location,…).
- post-disturbance events are simulated.

All tests defined for validation of the simplified dynamic simulation (DSA) have to be performed also with the full dynamic simulation, for comparison purposes.

7.2 Simplified Dynamic Simulation traceability table

Requirement	Component
R.4.1 should be able to handle at least 10.000 nodes.	Simplified simulation prototype for DSA
R.4.2 should be able to handle VSC, HVDC and FACTS models.	
R.4.3 should be able to handle user defined models of protection systems.	
R.4.5 should be able to simulate the behaviour of complex generating unit controls	
R.4.6 should be able to integrate models describing fast numerical controls.	
R.4.9 should be able to be used as a computation engine to perform Dynamic Security Assessment with at least 65.000 state variables. In details : 2000 contingencies to simulate within 5 minutes. Assuming 100 CPU are available, each simplified simulation should run within 15s. The simulation is stopped when physical values go beyond normal values or when steady state is reached or when 1000s is reached.	

7.3 Test case specification for Simplified Dynamic Simulation

The cases to be used are the same than for Full dynamic simulation. See section 6.3

7.4 Test Procedures for simplified dynamic simulation for Security Analysis (DSA

7.4.1 Test procedure 1 N-1 400 kV line

a) Test procedure identifier: **N-1 400 kV line**

b) Test purpose

Check fidelity of simplified dynamic simulation and CPU time.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS

d) Procedure steps.

- Trip a 400 kV line generating significant voltage deviations over a large zone
- Compare to full accuracy simulation results.
- Check CPU time

7.4.2 Test procedure 1 N-2 400 kV line

a) Test procedure identifier: N-2 400 kV line (double circuit line)

b) Test purpose

Check fidelity of simplified dynamic simulation and CPU time.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Trip 2 lines on the same tower generating significant voltage deviations
- Compare to full accuracy simulation.
- Check CPU time

7.4.3 Test procedure 2: N-1 line under 400 kV with severe local impact on voltages

a) Test procedure identifier: N-1 line under 400 kV with severe local impact on voltages

b) Test purpose

Check fidelity of simplified dynamic simulation and CPU time.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Trip a line below 400 kV generating significant voltage deviations over a large zone
- Compare to full accuracy simulation results.
- Check CPU time.

7.4.4 Test procedure 3: N-1 generating unit

a) Test procedure identifier: **N-1 generating unit**

b) Test purpose

Check fidelity of simplified dynamic simulation and CPU time.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Trip a large generating unit generating significant voltage deviations over a large zone.
- Compare to full accuracy simulation results..
- Check CPU time.

7.4.5 Test procedure 4 Voltage collapse

a) Test procedure identifier: **VC**

b) Test purpose:

Test the following components, automata and controls: tap changer blocking, under-voltage load shedding, under voltage generator tripping automata, automatic Switching of Capacitors/Self-Inductances, Secondary Voltage Control.

c) Test case

This test will be done using the test case 1 and 2. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

Same as full accuracy simulation test procedure 2

Same as full accuracy simulation test procedure 5 after the system splitting.

7.5 Robustness test procedures for simplified dynamic simulation for Security Analysis (DSA)

7.5.1 Test procedure 7 Low frequency oscillations

a) Test procedure identifier: **LFO**

b) Test purpose

The purpose of the test is to clarify the limits of the simplification.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Create a European wide poorly damped inter-area oscillation at frequency under 0.3 Hz;
- Check how the algorithm reacts compared to the full accuracy simulation

7.5.2 Test procedure 8 Frequency-voltage coupling

a) Test procedure identifier: **FVC**

b) Test purpose

The purpose of the test is to check the ability of the method to cope with frequency and voltage coupling.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

- Use some 4 loop AVR and PSS that couple voltage and frequency.

- Use a load model with voltage and frequency dependency.
- Create a scenario with a frequency slowly drifting and observe the voltage response due to the frequency derivative (df/dt).

7.5.3 Test procedure 10 Transformer Saturation

a) Test procedure identifier: **TS**

b) Test purpose

The purpose of the test is to check the ability of the method to cope with transformer saturation that creates a voltage/frequency coupling.

c) Test case

This test will be done using the test case 1 and 2, section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

Same as full accuracy simulation procedure 13

7.5.4 Test procedure 5 Cascading thermal protections

a) Test procedure identifier: **CIOC**

b) Test purpose

Test the following components, automata and controls: phase shifters, SPS.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

Same as full accuracy simulation test procedure 5 until the system splits.

7.5.5 Test procedure 6 Frequency deviation

a) Test procedure identifier: **FD**

b) Test purpose

Test the following components, automata and controls: SPS.

c) Test case

This test will be done using the test case 1 and 2 section 6.3. UCTE+TEIAS and IPSUPS.

d) Procedure steps.

8 Dispatching Training Simulator (DTS) validation methodology

The test procedure for the DTS validation is focused on the validation of the ability of the DTS simulator to model and simulate the pan European Electrical network with performance that appears to the operators as real time and with acceptable fidelity for the purpose of the reproduction of the real phenomena. The effort is focused on the algorithms, as consequence the performances of the visualization application (SCADA) must not be benchmarked as these are outside the scope of PEGASE.

The validation procedure for DTS will assess the following requirements:

- Adequacy to all types of training purposes

The DTS simulator engine should be able to reproduce all the scenarios useful for the training of operators.

This requirement will be verified by simulating typical scenarios that are used in training dispatchers.

- Verify the fidelity

The DTS simulator engine has to be able to correctly represent the physical phenomena with an acceptable level of accuracy. In any case the model must be able to reproduce the behavior of the Electrical Power Network and must not miss any bifurcation in the behavior of the system.

This requirement will be verified by comparing the full scale simulation with the DTS simulation following the validation methodology specified for simplified simulation (DSA).

- Check the performances

The DTS simulator engine must be able to perform the needed calculations in a time that approaches the real time. This means that the users do not have to have the impression that the simulation is slow compared to the wall clock time.

This requirement will be verified calculating the average execution time on a moving time window of 20 s, the maximum value of the average execution time should not exceed 20% of the simulation time.

- Identify the remaining gap with respect to full modeling

If the DTS uses a simplified model, the validation should check that the simplifications introduced by the model do not endanger the reproduction of the physical phenomena that must be reproduced.

8.1 Dispatching Training Simulator traceability table

Requirement	Component
R.4.1 should be able to handle at least 10.000 nodes.	Simplified simulation prototype for DSA
R.4.2 should be able to handle VSC, HVDC and FACTS models.	
R.4.3 should be able to handle user defined models of protection systems.	
R.4.5 should be able to simulate the behaviour of complex generating unit controls	
R.4.6 should be able to integrate models describing fast numerical controls.	
R.4.7 should be able to handle 125.000 state variables.	
R.4.8 integration algorithm should be A-stable.	
R.4.10 For DTS prototype, the delay with respect to effective real time should be seamless.	
R.4.11 DTS prototype should be able to handle detailed topology substation and protection.	

8.2 Test case specification for Dispatching Training Simulator

8.2.1 Test case 1 Full UCTE_Enriched model

a) Test Case Identifier: **UCTE_Enriched**

b) Input Specifications

- A bus and branch model of at least 10.000 nodes corresponding to different TSOs..
- Generation controllers: simple avr, pss, governor controllers except for a small subset equipped with complex ones :
    - Rotor current limit management (long term dynamics). Timer interacting with the system through a commutation. Over-excitation, Under-excitation (with commutations!)
    - Fast valving
    - Coupled processes (ex. : fast valving+common water pipe with hammer effect, combined cycles)
    - Other commutations (integral constraints) ?
- EHV Load model with 2 level transformers and motors at lowest voltage level and frequency dependency of the load.
- Automata: on-load tap changers, under voltage generator tripping, over voltage generator trip, under speed generator trip, over speed generator trip.
- Load model with voltage and frequency dependency.
- Under Frequency Load Shedding (with Rate Of Change Of Frequency and frequency thresholds) relays.
- Detailed topology substation and protection systems enough to trigger, after an incident, a significant number of trips leading to the splitting of the system in different asynchronous islands.

c) Outputs Specifications

- System frequency in the different asynchronous areas.
- Enabling signals of the protections systems.
- Active and reactive powers on the devices that trips .
- Voltage amplitudes and angles on the devices that will be used to resynchronize the islands.
- Currents on the branches close to the resynchronization bus(es).
- Variables of the system that have to be displayed on the operator screens.
- CPU time (time needed to perform a complete simulation step) for each block of integration steps.

8.3 Test Procedures for Dispatching Training Simulator

8.3.1 Test case 1 Cascading leading to several islands and re-synchronization

a) Test Case Identifier: **IR1**

b) Test purpose

- Compare systematically with the full time domain simulation.
- Verify performances against real time.
- Check robustness :
  - Very bad conditions (voltage at 0.2pu).
  - Behaviour on wrong operator decision.

c) Test case

The case to be used is **UCTE_Enriched**.

d) Procedure steps

- Setup of the simulations and preparation of the one line diagram.
- The static and dynamic model for the DTS simulation and the respective one line diagrams must be prepared. The two lists of correct and incorrect actions to be performed on the system should be ready and their temporal sequence should be also defined.
- Perform the off line simulation using the full-scale dynamical model with maximum accuracy. The simulation should lead to a cascading tripping of relays splitting the system in at least two islands but without short circuit in the system.
- Perform the simulation and the actions using the DTS simulator. The same actions performed on the full scale simulation should be performed on the DTS trying to enable them in an instant as close as possible to the one used for the off-line full scale dynamical simulation.
- Post-process and analysis the data.
- Analysis of the quality of the results and of the performances of full dynamic simulation vs DTS simulation.

8.3.2 Test case 2 Black start to re-energization, to re-synchronization of a plant

a) Test Case Identifier: **IR2**

b) Procedure steps

- Setup of the simulations and preparation of the one line diagram of a black-out system.
- Prepare a list of actions to be performed that lead to the resynchronization of a power plant and create a system with two islands.
- Prepare a list of actions to be performed that lead to a failed resynchronization of the islands.
- Perform a simulation of the lists of actions that lead to successful synchronization and to the failed synchronization on the full scale dynamical model and on the DTS model.
- Post-process and analysis the data.
- Analysis of the quality of the results and of the performances of full dynamic simulation vs DTS simulation.

8.3.3 Test case 3 Short-circuit leading to voltage collapse potentially leading to black-out

a) Test case identifier: **SCLBCO**

b) Test purpose

Check the capability of the DST software to simulate a voltage collapse phenomenon leading to system blackout.

c) Test case

The case to be used is **UCTE_Enriched**

d) Procedure steps

- Setup of the simulations and preparation of the one line diagram. The initial state of the model should be tuned so that a short circuit leads to a voltage collapse with disconnection of the load.
- Include additional undervoltage relays for generation
- Prepare a list of actions that initiate a voltage collapse in the system.
- Prepare a list of actions that speed up the voltage collapse.
- Perform a simulation with the lists of actions on the full scale dynamical model and on the DTS model.
- Post-process and analysis the data.
- Analysis of the quality of the results and of the performances of full dynamic simulation vs DTS simulation.

8.3.4 Test case 4 Tripping of a line in a characteristic corridor, followed by load increase in the deficit area supplied through the corridor, leading to slow loading towards static stability limit, voltage decrease, slow oscillations, slow loss of synchronism.

a) Test Case Identifier: **CORRIDOR**

b) Test purpose

Check the capability of the DST software to simulate a loss of synchronism after an event

c) Test case

The case to be used is **UCTE_Enriched**

d) Procedure steps

- Setup of the simulations and preparation of the one line diagram. The initial state of the model should be tuned so that after the line tripping the specific corridor will be loaded with standard static stability reserve.
- Include additional undervoltage relays and overcurrent line/transformers relays for generation and loads.
- Prepare a list of actions that include generation increase to compensate load increase and leading to loss of stability in the corridor.
- Prepare a second list of actions including generation increase and/or load-shedding actions, leading to reducing power transfer in the corridor. Include also a list of actions to increase the stability limits of the corridor (shunt-compensation devices switching, topology changes, etc...).
- Perform a simulation with the lists of actions on the full scale dynamical model and on the DTS model.
- Post-process and analysis the data. Display indicators obtained by processing PMU measurements (i.e. voltage angle variation) to indicate the corridor/area under difficulty; and Loop parallel execution and result display of a QSS (if available), indicating static stability reserves.
- Analysis of the quality of the results and of the performances of full dynamic simulation vs DTS simulation.

ANNEX 1

The input to T6.2 testing methodology is the requirements identified in WP1 for the different applications to be developed in WP2, WP3 and WP4. The purpose of this annex is to clearly state those requirements, as understood from the WP1 document Identification of needs, for each prototype in order to develop the testing methodology that will be used to validate the fulfilment of those requirements.

Section 2 of this document covers the state estimation requirements, section 3 the steady state optimization requirements, and section 4 time domain simulation requirements.

State Estimation requirements

R2.1 The number of nodes to be estimated will be at least 5.000.

R.2.2 The state estimator should be two step estimation. First step at TSO level, second step at ETN level.

R.2.3 The system should be parallelizable.

R.2.4 TSO state estimator should remain independent. The coordination at ETN should be done on a central computer.

R.2.5 The state estimator should be improved with the use of Phasor Measurements Units (PMUs)

R.2.6 The state estimator should allow redundancy capacity and filtering capability.

R.2.7 The state estimator should allow tuning of the PMUs data weights for optimal on-line performance in one partner TSO system.

R.2.8 A Phasor Data Concentrator should be developed which should be able to handle at least 100 PMUs,

R.2.9 The state estimator should be able to ensure consistency between topology and measurements.

R.2.10 The state estimator should be designed to provide a solution at substation level.

R.2.11 A 3-phase model should be used with all circuit breakers represented for the estimation at substation level.

R.2.12 A topology processor at substation level should be developed.

R.2.13 The topology processor should be able to detect topology errors.

R.2.14 The state estimator at substation level should be able to send the measurements on states and statistical confidence.

R.2.15 The state estimator algorithm should be developed using the trust region approach.

R.2.16 The state estimator should perform PMUs data validation.

R.2.17 The state estimator should deal with voltage and current phasors.

R.2.18 The state estimator should check compliance with communications system by a real time algorithm.

R.2.19 The state estimator should assess delays on data flow.

Steady State Optimization requirements

R.3.1 Prototypes should be able to handle around 10.000 nodes.

R.3.2 Computation time for a given problem should not exceed 15 min.

R.3.3 The system should be able to handle devices like FACTs, HVDC, AGC.

R.3.4The system should include a network compressor.

R.3.5 The system should be able to filter contingencies that are not binding for the problem.

R.3.6 The system should be able to handle at least 10.000 contingencies

R.3.7 The system should be able to handle automatic post-contingency regulations (frequency regulation, voltage regulation)

R.3.8 The system should be able to handle simultaneously continuous and discrete variables.

Time Domain Dynamic Simulation requirements

R.4.1 Prototypes should be able to handle at least 10.000 nodes.

R.4.2 Prototypes should be able to handle VSC, HVDC and FACTS models.

R.4.3 Prototypes should be able to handle user defined models of protection systems.

R.4.4 Prototypes should be able to simulate the behaviour of complex generating unit controls

R.4.5 Prototypes should be able to integrate models describing fast numerical controls.

R.4.6 Full accuracy and DTS prototypes should be able to handle 125.000 state variables.

R.4.7 Full accuracy and DTS prototypes should include an integration algorithm A-stable.

R.4.8 The simplified simulation prototype should be able to be used as a computation engine to perform Dynamic Security Assessment with at least 65.000 state variables.

R.4.9 For DTS prototype, the delay with respect to effective real time should be seamless.

R.4.10 DTS prototype should be able to handle detailed topology substation and protection.

ANNEX 2

The records below provide the basic contents of the associated anomaly reports. The report puts the informacion in these records into context and provides the rationale for the decisions taken. Each anomaly identified is uniquely denoted by a reference number. This reference number follows the convention of the WP-integer number (eg. WP2-1 means anomaly 1 from WP2 (State Estimator)).

A.1 Anomaly identification record

A.1.1 Anomaly title

Textual sum-up of the anomaly as aide memoir

A1.2 Anomaly reporters

Names of the persons involved with the identification

A1.3 Date of the anomaly identification

Calendar date

A1.4 WP Application

WP where the anomaly has been detected

A.1.6 Anomaly Description

Free-text indication of the reason for assuming the presence of an anomaly

A.1.7 Priority

Preliminary assessment of the severity of the anomaly

A.2 Resolution action record

A.2.1 Resolution personnel

Name of persons involved with anomaly resolution

A.2.2 Items Changed

Parts of the system subjected to changes

A2.3 Resolution description

Free-text explanation of the change carried out

A2.4 Validation personnel

Name of persons involved with validation

A2.5 Validation justification

Free-Text indication of the reasons why the change can be considered to have been validated.