

Politecnico di Torino  
Master Degree in Electrical Engineering



Master Thesis

Remote interconnection of real-time  
simulators: performance analysis and case  
studies

**Supervisor:**

*Prof. Ettore Bompard* .....

**Co-Supervisors:**

*Prof. Enrico Pons* .....

*Prof. Andrea Mazza* .....

**Candidate:**

*Stefano Frittoli*  
.....

A.Y. 2020/2021



# Acknowledgements

Ringrazio il mio relatore, il Prof. Ettore Bompard, per avermi dato la possibilità di lavorare a questa Tesi.

Ringrazio in particolare i miei due co-relatori, il Prof. Enrico Pons e il Dr. Ing. Andrea Mazza, per la fiducia datami coinvolgendomi nel loro lavoro di ricerca e per essere stati sin dall'inizio dei lavori di Tesi fonte di ottimi consigli.

Ringrazio i miei genitori, mamma Grazia e papà Giovanni, per avermi insegnato il senso dell'umiltà, l'etica del lavoro e per avermi permesso in questi anni di perseguire il meglio. Un privilegio, questo, che ancora oggi in molti non hanno.

Ringrazio la mia sorella Lorena, per avere dimostrato nel tempo di avere nei miei confronti la migliore qualità che una persona possa avere: la pazienza.

Ringrazio le mie nonne, Elvira e Maria, incessanti sostenitrici della mia causa, nei bei giorni come nei periodi più bui.

*Dio perdona. La rete elettrica no.*

- un Professore saggio -

*Se la conoscenza può creare problemi, non è attraverso l'ignoranza che  
possiamo risolverli.*

- Isaac Asimov -

# Abstract

Within the global challenge to increase the energy production from Renewable Energy Systems (RES), a key role is and will be played by those plants connected to the Distribution System. However, a higher and higher penetration of RES connected at the Low and Medium Voltage grid requires redesigning the procurement of the Ancillary Services (AS) needed by the Transmission System Operator (TSO).

In this Thesis a set of Realtime Remote co-simulation is executed in collaboration with the Politecnico di Bari, within the *Living Grid* research project. Firstly, the quality of the communication between the two research facilities is investigated. A series of Realtime co-simulations with Remote Power Hardware In the Loop (PHIL) is implemented with the following aims: verify the feasibility of a Low Voltage (LV) flexible resource dispatchment and verify the stability of the proposed Remote co-simulation.

Within the *Living Grid* project, a further study is carried out with the aim to estimate the contribution of the distributed sources connected at the LV grid. The parameterization is achieved with a straight-forward non-linear regression. This study is realized merely with the Realtime simulator based in Turin and with no PHIL.

A further set of Realtime simulations is realized again with the simulator located in the Turin facility. Specifically, the Medium Voltage (MV) fault analysis is carried out, comparing the simulations with the expected results from the theoretical analysis. Secondly, the feasibility of the voltage regulation of a MV feeder by using uniquely the On Load Tap Changer (OLTC) at the High Voltage - Medium Voltage (HV-MV) transformer is carried out.

# Contents

<b>1</b>	<b>Introduction</b>	<b>17</b>
1.1	Realtime simulation and Power Hardware In the Loop concept	17
1.2	Environmental issues and trends in renewable energy . . . . .	23
1.3	Microgrids: definitions and classifications . . . . .	27
1.4	TSO-DSO interaction . . . . .	29
<b>2</b>	<b>Remote co-simulations between Turin and Bari</b>	<b>35</b>
2.1	Analysis of the communication latency . . . . .	36
2.2	OLTC transformer and its application in the grid . . . . .	45
2.3	R-PHIL co-simulation with the OLTC . . . . .	49
2.4	Stability of the R-PHIL co-simulation . . . . .	56
<b>3</b>	<b>Decoupling of a larger network</b>	<b>64</b>
3.1	State-Space method and Nodal Analysis . . . . .	65
3.2	State-Space Nodal method . . . . .	68
3.3	Application of parallelization for a small distribution grid . . .	70
3.4	Comparison with the RTDS simulator of the steady state results	76
<b>4</b>	<b>Characterization of the LV distributed sources</b>	<b>77</b>
4.1	Theoretical basis of the non-linear regression . . . . .	78
4.2	Implementing the regression with different cases . . . . .	82
4.2.1	Case study A: three parameters estimation with a five- step trajectory and high accuracy set of measures . . .	82
4.2.2	Case study B: three parameters estimation with a five- step trajectory and a set of measures with low accuracy	86
4.2.3	Case Study C: three parameters estimation with a three- step trajectory and white noise in the measures . . . .	88
4.2.4	Case study D: two parameters estimation with a five- step trajectory and a set of measures with low accuracy	91
4.2.5	Case study E: two parameters estimation with a three- step trajectory and noise in the measures . . . . .	94

<b>5</b>	<b>Fault analysis on a Distribution Network</b>	<b>98</b>
5.1	Method of symmetrical components . . . . .	99
5.2	Three-phase fault . . . . .	102
5.3	Line to Line fault . . . . .	105
5.4	Line-to-earth faults . . . . .	108
5.5	Comparison with the RTDS simulator of the faults' results . .	119
<b>6</b>	<b>Voltage regulation with high PV penetration</b>	<b>121</b>
6.1	Model of the PV generator . . . . .	122
6.2	MV feeder and results . . . . .	129
<b>7</b>	<b>Conclusion</b>	<b>134</b>
	<b>Appendix A</b>	<b>135</b>
A.0.1	Analisy of the communication latency: settings and practical aspects . . . . .	135
A.0.2	R-PHIL RT co-simulation with one feeder and the OLTC: settings and practical aspects . . . . .	144
A.0.3	Stability of the R-PHIL co-simulation: settings and practical aspects . . . . .	156
	<b>Appendix B</b>	<b>160</b>
B.0.1	Implementation of the SSN method: settings . . . . .	160
	<b>Appendix C</b>	<b>164</b>
C.0.1	Characterization of the distrubuted sources: Simulink settings . . . . .	164
C.0.2	Matlab scripts . . . . .	168
	<b>Appendix D</b>	<b>172</b>
D.0.1	Faults implementation: required settings . . . . .	172
D.0.2	Faults implementation: Matlab code . . . . .	175
	<b>Appendix E</b>	<b>185</b>
E.0.1	Overall model: an overview . . . . .	185
E.0.2	PV system: Simulink model and settings . . . . .	187
E.0.3	Matlab script with the required parameters . . . . .	193
	<b>Bibliography</b>	<b>206</b>

# List of Tables

1.1	INDC scenario and <i>below 2 degrees</i> scenario. Source: [2]. . . .	25
1.2	Percentage of PV installation based on the capacity (Germany case). Source: [19] (© IRENA 2016). . . . .	26
2.1	Statistics on data packets exchange from PoliTo to PoliBa . . .	41
2.2	Statistics on latencies of the PoliTo-PoliBa data exchange . . .	43
2.3	Statistics on latencies of the PoliTo-PoliTo data exchange . . .	43
2.4	OLTC Transformer main characteristics. . . . .	53
3.1	Type of load for each distribution bus. . . . .	75
3.2	ZIP coefficient for each type of load. Source: [6]. . . . .	75
3.3	Comparison between RTDS <sup>©</sup> and OPAL-RT <sup>©</sup> simulators: current amplitude. Partial source: [1]. . . . .	76
4.1	Parameters of the LV load and distributed generation. . . . .	82
4.2	Initial values of the three parameters. . . . .	83
4.3	Set of measures used for the implementation of this case. . . .	84
4.4	Estimation of the three parameters with percentage errors . . .	84
4.5	Set of measures used for the implementation of this case . . .	86
4.6	Estimated values of the parameters, with percentage errors. . .	86
4.7	Set of measures used for the implementation of this case . . .	89
4.8	Estimated values of the parameters, with percentage errors . .	89
4.9	Initial values of the two parameters . . . . .	91
4.10	Estimated values of the parameters, with percentage errors . .	91
4.11	Estimated value of the parameters and percentage errors . . .	93
4.12	Estimated values of the parameters, with percentage errors . .	94
4.13	Estimated values of the two parameters . . . . .	96
6.1	Main parameter of the photovoltaic system. . . . .	124
6.2	Nominal power of each PV system and PV penetration in the feeder. . . . .	130
6.3	List of cases considered. . . . .	132

E.1 Parameters of the PI regulators. . . . . 192

# List of Figures

1.1	RT-LAB <sup>©</sup> interface. . . . .	17
1.2	Top-level Simulink model:an example. . . . .	18
1.3	Qualitative scheme of the Host-Target interconnection. . . . .	19
1.4	Top-level Simulink model:an example with parallel computation. . . . .	19
1.5	OpComm block. . . . .	20
1.6	OpAsyncRec block. . . . .	21
1.7	OpAsyncSend block. . . . .	21
1.8	OpIPSocketCtrl block. . . . .	22
1.9	Hardware In the Loop principle scheme. . . . .	22
1.10	Power Hardware In the Loop principle scheme. . . . .	23
1.11	Primary sources of GHG emissions. Source: [10]. . . . .	25
1.12	Percentage of PV installation based on the voltage level con- nection (Italy case). Source: [16]. . . . .	26
1.13	A classification of microgrids based on user, application and key elements. Source: [18] (© IRENA 2015). . . . .	27
1.14	A classification of microgrids based on size, capability and complexity. Source: [18] (© IRENA 2015). . . . .	28
1.15	Proposed process to avoid HV-MV transformer congestion. Source: [20]. . . . .	30
1.16	Proposed process to avoid TSO line congestion. Source: [20]. . . . .	30
1.17	Proposed process to support grid voltage. Source: [20]. . . . .	31
1.18	General representation of a market with TSO-DSO interac- tion. Source: [21]. . . . .	31
1.19	(a) Centralized AS market model - (b) Local AS market model - (c) Shared balancing responsibility model - (d) Common TSO-DSO AS market model - (e) Integrated Flexibility mar- ket model. Source: [21]. . . . .	34
2.1	Simulink model: overall vision . . . . .	37
2.2	Simulink model: Master block scheme . . . . .	38
2.3	Exchanged signals prior to the time synchronization. . . . .	40

2.4	Exchanged signals: qualitative representation. . . . .	40
2.5	Latencies on the PoliTo-PoliBa data exchange: (Top-left) 1 ms; (Top-right) 5 ms; (Bottom-left) 10 ms; (Bottom-right) 20 ms. . . . .	42
2.6	Latencies on the PoliTo-PoliTo data exchange: (Top-left) 1 ms; (Top-right) 5 ms; (Bottom-left) 10 ms; (Bottom-right) 20 ms. . . . .	42
2.7	OLTC's schematic representation. . . . .	45
2.8	(a) Equivalent circuit of a transmission system with OLTC transformer; (b) Equivalent circuit with the parameters transferred on the secondary side (receiver) [28]. . . . .	46
2.9	OLTC control scheme. . . . .	47
2.10	Motor drive unit schematic representation. Source: [14]. . . . .	48
2.11	Schematic representation of the simulated distribution feeder. . . . .	49
2.12	Realtime co-simulation scheme. . . . .	50
2.13	SM_Master Simulink model. . . . .	52
2.14	OLTC Simulink model. . . . .	53
2.15	Low Voltage load's voltage amplitude. [8] . . . . .	54
2.16	Low Voltage load's active and reactive power. [8] . . . . .	54
2.17	Active and Reactive power in the TSO-DSO connection node. [8] . . . . .	55
2.18	active power, reactive power and voltage (phase-ground rms) in Turin. Scale factor = 1500, added delay = $0 T_s$ . . . . .	57
2.19	active power, reactive power and voltage (phase-ground rms) in Turin. Scale factor = 1500, added delay = $42 T_s$ . . . . .	58
2.20	active power, reactive power and voltage (phase-ground rms) in Turin. First row: scale factor = 1400, added delay = $0 T_s$ . Second row: scale factor = 1400, added delay = $84 T_s$ . Third row: scale factor = 1400, added delay = $336 T_s$ . . . . .	59
2.21	active power, reactive power and voltage (phase-ground rms) in Turin: scale factor = 1400, added delay = $672 T_s$ . . . . .	60
2.22	active power, reactive power and voltage (phase-ground rms) in Turin. First row: scale factor = 100, added delay = $0 T_s$ . Second row: scale factor = 100, added delay = $84 T_s$ . Third row: scale factor = 100, added delay = $336 T_s$ . Forth row: scale factor = 100, added delay = $833 T_s$ . . . . .	61
2.23	active power, reactive power and voltage (phase-ground rms) in Turin: scale factor = 1600, added delay = $0 T_s$ . . . . .	62
3.1	Example of an electrical network represented by a graph. . . . .	66

3.2	Norton equivalent of a generic one-port electrical element. Source: [11]. . . . .	67
3.3	Qualitative representation of the analysed electrical grid. . . . .	71
3.4	Stubline block and its interface window. . . . .	72
3.5	Schematic representation of the reactive compensation of the Stubline block. . . . .	72
3.6	SSN block (left) and SSN Interface block (right). . . . .	73
3.7	Qualitative representation of the analysed electrical grid with the decoupling in subsystems and groups. . . . .	74
4.1	Schematic representation of the LV node with equivalent exponential- form load and PV generator. . . . .	78
4.2	Flow chart representation of the non-linear regression. . . . .	80
4.3	Active power measured below the secondary substation (up- per); line-to-line voltage measured below the secondary sub- station (center); tap changer position of the HV-MV trans- former (lower). . . . .	83
4.4	Estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with high accuracy measures of voltage and active power. . . . .	85
4.5	Percentage errors of the estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with high accuracy mea- sures of voltage and active power. . . . .	85
4.6	Estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with low accuracy measures of voltage and active power. . . . .	87
4.7	Percentage errors of the estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with high accuracy mea- sures of voltage and active power. . . . .	87
4.8	Active power measured below the secondary substation (up- per); line-to-line voltage measured below the secondary sub- station (center); tap changer position of the HV-MV trans- former (lower). . . . .	88
4.9	Estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with measures spoiled by noise. . . . .	90
4.10	Percentage errors of the estimation of the parameters $P_{nom}$ (upper), $k_p$ (center) and $P_{gen}$ (lower) with measures spoiled by noise. . . . .	90
4.11	Estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures and with $k_p = 1.5$ . . . . .	92

4.12	Percentage errors of the estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures and with $k_p = 1.5$ . . . . .	92
4.13	Estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures, with $k_p = 1.45$ (left) and with $k_p = 1.55$ (right). . . . .	93
4.14	Percentage errors of the estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures, with $k_p = 1.45$ (left) and with $k_p = 1.55$ (right). . . . .	94
4.15	Estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with measures spoiled by noise and with $k_p = 1.5$ . . . . .	95
4.16	Percentage errors of the estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with measures spoiled by noise and with $k_p = 1.5$ . . . . .	95
4.17	Estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures, with $k_p = 1.45$ (left) and with $k_p = 1.55$ (right). . . . .	96
4.18	Percentage errors of the estimation of the parameters $P_{nom}$ (upper) and $P_{gen}$ (lower) with badly accurate measures, with $k_p = 1.45$ (left) and with $k_p = 1.55$ (right). . . . .	97
5.1	(Left) Balanced three-phase voltage; (Right) rotation operators.	99
5.2	(Left) Positive sequence; (Center) Negative sequence; (Right) zero sequence. . . . .	100
5.3	General three-pole representation. . . . .	101
5.4	Generic representation of the three symmetrical sequences. . .	102
5.5	Three-phase fault representation. . . . .	102
5.6	Thevenin equivalent of the positive sequence. . . . .	103
5.7	Identification of the position of the applied fault. . . . .	103
5.8	(Left): pre-fault condition; (Right): fault condition - (Upper) phase to ground voltage at the beginning of Magra feeder; (Lower) line currents at the beginning of Magra feeder. . . . .	104
5.9	Phase-to-ground voltage and current at the beginning of Magra feeder with different values of $\bar{Z}_f$ : (Left) $\bar{Z}_f = 0.001\Omega$ ; (Center) $\bar{Z}_f = 0.5\Omega$ ; (Right) $\bar{Z}_f = 2.5\Omega$ . . . . .	104
5.10	Line-to-line fault representation. . . . .	105
5.11	Composition of the symmetrical sequences. . . . .	106
5.12	Composition of the sequences with the Thevenin equivalent. .	106
5.13	Line-to-line fault condition: (Upper) phase-to-ground voltage at the beginning of Magra feeder; (Lower) line current at the beginning of Magra feeder. . . . .	107

5.14	Line-to-earth fault representation. . . . .	108
5.15	(Left) Composition of the symmetrical sequences; (Right) Thevenin equivalent. . . . .	109
5.16	(Left) Three-phase voltage pre-fault; (Right) three-phase voltage after line-to-earth fault with $Z_f = 0$ . . . . .	110
5.17	Variation of the voltages as a function of the $Z_f$ value. . . . .	110
5.18	(Left) Compensated neutral; (Center) Approximated compensated neutral; (Right) Circuit model of $\bar{Z}_o$ . . . . .	111
5.19	Generic distribution network model. . . . .	111
5.20	(Left) Model of a healthy feeder; (Center) Model of a faulted feeder; (Right) Model of the transformer with the neutral. . . . .	112
5.21	Model including the healthy feeder, the faulted feeder and the transformer's models. . . . .	112
5.22	Phasor representation with the insulated neutral. Source: [9] . . . . .	113
5.23	(Upper) Model of the compensated neutral and the healthy feeder; (Lower-left) Phasor representation with $P_f < 0$ and $Q_f > 0$ ; (Lower-right) Phasor representation with $P_f < 0$ and $Q_f < 0$ . Source: [9] . . . . .	113
5.24	Generic phasor representation with different $\phi$ angles. Source: [9] . . . . .	114
5.25	(Top-Left) Phase voltage at the beginning of Magra feeder; (Bottom-Left) Current at the beginning of Magra feeder; (Right) zoomed view of current at the beginning of Magra feeder. . . . .	115
5.26	Phase voltage of phase $b$ and phase $c$ with a line-to-earth fault on phase $a$ . . . . .	116
5.27	Phasor representation of a line-to-earth fault with isolated neutral. . . . .	116
5.28	(Top-Left) Phase voltage at the beginning of Magra feeder; (Bottom-Left) Current at the beginning of Magra feeder; (Right) zoomed view of currents at the beginning of Magra feeder. . . . .	118
5.29	Phasor representation of a line-to-earth fault with compensated neutral. . . . .	118
5.30	Three-phase fault: phase to ground voltage at the beginning of Magra feeder (Upper); line currents at the beginning of Magra feeder (Lower). Source: [1]. . . . .	119
5.31	Line-to-line fault: phase to ground voltage at the beginning of Magra feeder (Upper); line currents at the beginning of Magra feeder (Lower). Source: [1]. . . . .	120
6.1	Circuit model of a solar cell. . . . .	122

6.2	An example of Voltage-Current characteristic (left) and Power-Current characteristic (right) with different irradiances. . . . .	123
6.3	Scheme of the photovoltaic system. . . . .	124
6.4	Flow chart of the Perturb & Observe algorithm. . . . .	125
6.5	Scheme of the proposed control. Source: [5]. . . . .	126
6.6	Representation of the $(abc)$ , $(\alpha\beta)$ and $(dq)$ frames. . . . .	127
6.7	Vectorial representation of the PCC voltage with the synchronous reference frame $(dq)$ . . . . .	128
6.8	Schematic representation of the CHIARI feeder. . . . .	129
6.9	Phase-to-ground voltage (pu) along the CHIARI feeder: 0 % DG penetration (upper left); 59 % DG penetration (upper right); 38 % DG penetration (center left); 30 % DG penetration (center right); 24 % DG penetration (lower left); 15 % DG penetration (lower right). . . . .	131
6.10	Phase-to-ground voltage (pu) along the CHIARI feeder: 59 % DG penetration with Case 1.1 (upper); 59 % DG penetration with Case 1.2 (center); 59 % DG penetration with Case 1.3 (lower). . . . .	133
A.1	SC_Console subsystem. . . . .	136
A.2	Portion of the SM_Master. . . . .	136
A.3	Pulse generator block's settings. . . . .	137
A.4	Delay block's settings. . . . .	138
A.5	Signal specification block's settings. . . . .	138
A.6	AsyncIP proposed variation. . . . .	139
A.7	Loading of required files from RT-LAB <sup>®</sup> . . . . .	139
A.8	SC_Console subsystem of the model. . . . .	144
A.9	Pulse generator block's settings. . . . .	145
A.10	Amplification of the active and reactive power of the microgrid. . . . .	146
A.11	Settings of the OLTC transformer. . . . .	146
A.12	Simulink model: HV network and OLTC transformer Simulink model. . . . .	147
A.13	Simulink model: MV network Simulink model. . . . .	148
A.14	Simulink model: LV network Simulink model. . . . .	149
A.15	<i>SC_Console</i> subsystem of the model. . . . .	156
A.16	<i>SM_Master</i> subsystem: focus on the reset of the feedback from Bari. . . . .	157
A.17	Delay block's settings. . . . .	158
A.18	<i>SM_Master</i> subsystem: focus on the delay block. . . . .	158
A.19	<i>SM_Master</i> subsystem: focus on load model and the $(P, Q)$ scale factor. . . . .	159

B.1	Artemis Guide block. . . . .	160
B.2	Artemis Guide setup: General. . . . .	160
B.3	Artemis Guide setup: SSN Options. . . . .	161
B.4	Artemis Guide setup: Advanced. . . . .	161
B.5	SSN Interface Block and its settings: an example. . . . .	162
B.6	OLTC Transformer model ( <i>Left</i> ); Model of the adjustable windings for a single phase ( <i>Right</i> ). . . . .	163
C.1	Upper level of the implemented model. . . . .	164
C.2	Content of the <i>SC_Console</i> block. . . . .	165
C.3	Focus on the <i>OLTC transformer</i> model and the required adjustments. . . . .	165
C.4	Overall control scheme of the OLTC transformer (upper) and OLTC scheme (lower). . . . .	166
C.5	Focus on the <i>SS_FIAT_GROSSO</i> block. . . . .	166
C.6	Focus on the LV grid model of the FIAT feeder: first two implementations of the regression. . . . .	167
C.7	Focus on the LV grid model of the FIAT feeder: last two implementations of the regression. . . . .	167
C.8	<i>Moving Average</i> , <i>White Noise</i> and <i>Pulse Generator</i> Simulink blocks. . . . .	168
D.1	Upper level of the Simulink model. . . . .	172
D.2	Content of the <i>Slave</i> block. . . . .	173
D.3	Three-phase fault block in the chosen location. . . . .	173
D.4	Three-phase fault block: parameters. . . . .	174
D.5	Secondary winding connection setup. . . . .	175
E.1	Upper level of the Simulink model. . . . .	185
E.2	Content of the <i>SC_Console</i> subsystem. . . . .	186
E.3	Content of the <i>SS_CHIERI</i> subsystem. . . . .	186
E.4	Subsystem block of the PV system. . . . .	187
E.5	Overall Simulink scheme of the PV system. . . . .	187
E.6	Simulink block of the PV Array. . . . .	188
E.7	Simulink block of the three-phase inverter. . . . .	188
E.8	Overall Simulink scheme for the control. . . . .	189
E.9	Simulink scheme for the measures of voltage and current and the conversion to the ( <i>dq</i> ) reference frames. . . . .	191
E.10	Simulink scheme of the voltage DC link closed loop. . . . .	191
E.11	Simulink scheme of the current closed loop. . . . .	192

# Chapter 1

## Introduction

### 1.1 Realtime simulation and Power Hardware In the Loop concept

In order to understand in a more accurate and realistic way a physical phenomenon the realtime simulation is carried out. Such analysis approach, which is quickly increasing thanks to more and more powerful microprocessor, sees a wide range of application in the scientific world. The founding principle is that the simulation process will take as much time as the considered physical phenomenon will last: 10 seconds of a waveform will be simulated by our computer in exactly 10 seconds. Whenever the simulation time does not respect such condition (overrun simulation), the realtime simulating process fails.

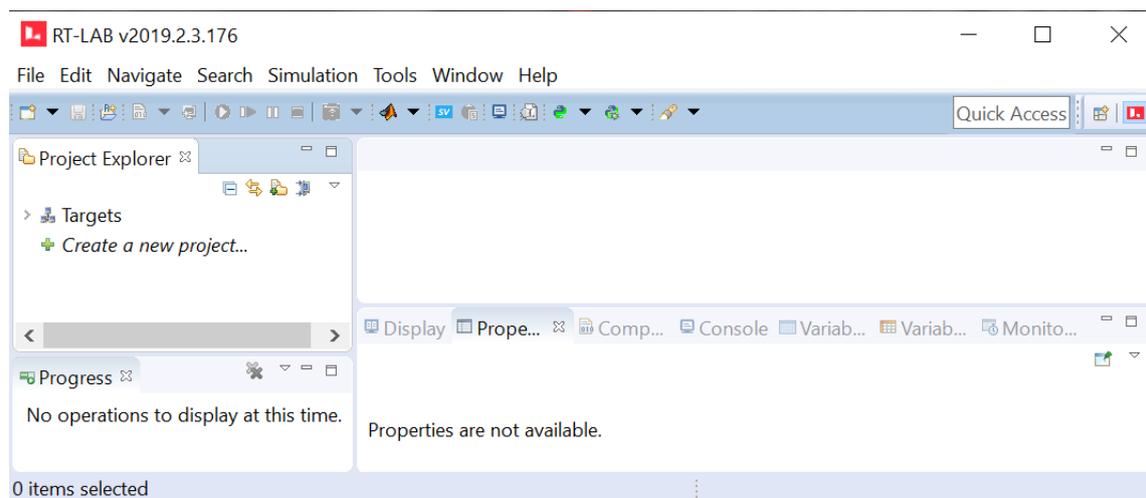


Figure 1.1: RT-LAB<sup>©</sup> interface.

For this thesis the OPAL-RT<sup>®</sup> hardware and software products have been used. In order to execute in realtime a Simulink model, the RT-LAB<sup>®</sup> software is executed. The main user interface is shown in Figure 1.1.

From such interface, a new model can be created. The model consists in a Simulink model that requires a certain number of blocks and component in order to run correctly.

Firtly, the top-level of the Simulink model requires only subsystem blocks with specific names and functions. An elementar case is depicted in Figure 1.2.

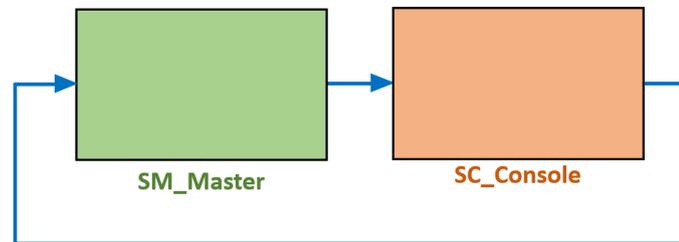


Figure 1.2: Top-level Simulink model:an example.

In the example illustrated above, the *SC\_Console* has the mere function of a Graphic User Interface (GUI) and is run by the *Host Computer*. The *SM\_Master*, instead, is in charge of the computational aspects. It will be executed by the realtime simulator, also called *Target Computer*. *Host* and *Target* computer will then be interfaced with a specific communication protocol (UDP/IP or TCP/IP). The Figure 1.3 summarizes the scheme here mentioned.

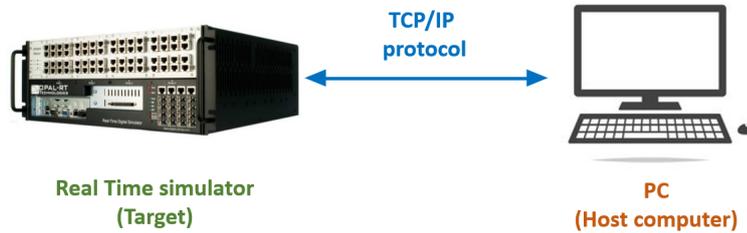


Figure 1.3: Qualitative scheme of the Host-Target interconnection.

Whenever a higher computational capacity is required, e.g. the case of a complex model to be simulated, the OPAL-RT<sup>®</sup> realtime simulator has the possibility to implement a parallel computation by using more CPU's cores of the simulator itself. An example of Simulink's top-level in which more CPU's cores are exploited is illustrated in Figure 1.3. The parallel computation requires to add a new subsystem block, which is the *SS\_Slave* block. In it, part of the computation is executed using a different CPU's core with respect to the one used for the *SM\_Master*. The amount of *SS\_Slave* block that can be implemented is as high as the number of available CPU's cores in the realtime simulator. A critical aspect concerning the naming choices of the subsystem blocks has to be brought to the attention of the reader: as noted from Figure 1.2 and Figure 1.4, the blocks are named in a specific way. The use of the prefixes *SM*, *SC* and *SS* is in fact mandatory: the software uses them to correctly distinguish the GUI application by the computational blocks and apply those at the correct CPU's cores.

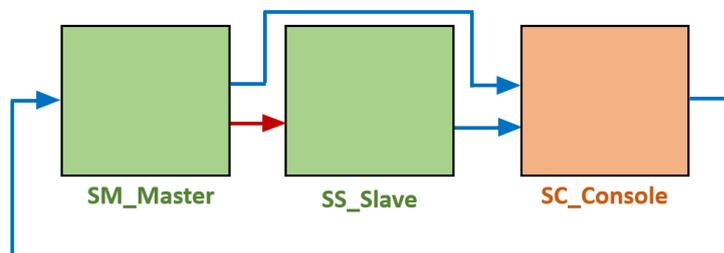


Figure 1.4: Top-level Simulink model: an example with parallel computation.

A further topic related to the implementation of the realtime simulation is the communication type between the different top-level subsystems. As mentioned in [25], the communication type are two: the *synchronous* and the *asynchronous* type. The former (red arrow in Figure 1.4) is used among CPU cores, it is a high-speed communication and thus able to support real time simulations. The latter (green arrow in Figure 1.4) is used between the *Host* and *Target* computer, it is a low-speed communication and thus only used for data logging.

As aforementioned, the top-level subsystems are implemented by different devices. In order to synchronize the signals in the simulation, within every subsystem the use of the *OpComm* block is mandatory. The block with its setup menu is shown in Figure 1.5. The *OpComm* has to be inserted after the subsystem creation and having correctly named.

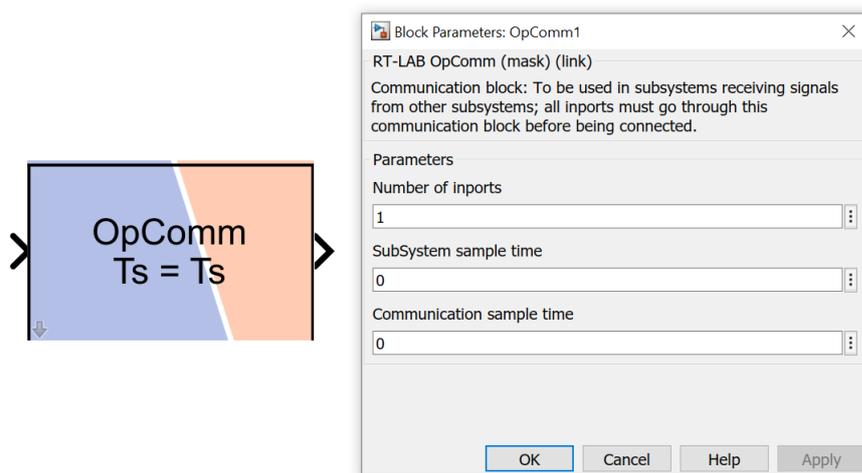


Figure 1.5: OpComm block.

From the setup menu shown in Figure 1.5 it can be noted that more inputs can be read by a single *OpComm* block.

To guarantee a correct implementation, a *OpComm* block cannot read both *synchronous* *asynchronous* signals. Whenever both type of signal enter in a subsystem, two *OpComm* will be required.

Because this thesis will also describe a *Remote Power Hardware In the Loop (R-PHIL)*, other critical blocks are here mentioned and illustrated. In order to receive data with a wireless connection from another realtime simulator, a model requires the use of the *OpAsyncRec* block, shown in Figure 1.6.

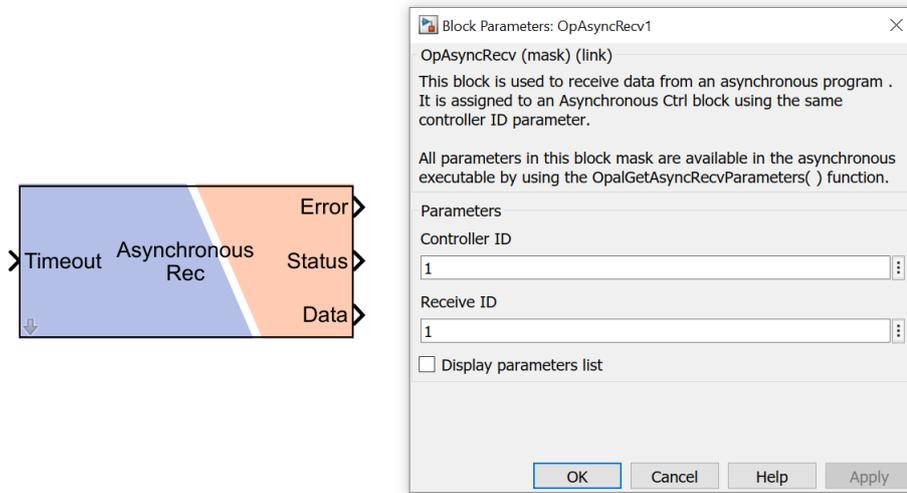


Figure 1.6: OpAsyncRecv block.

Similarly, in order to send data with a wireless connection to another real-time simulator, a model requires the use of the *OpAsyncSend* block, shown in Figure 1.7.

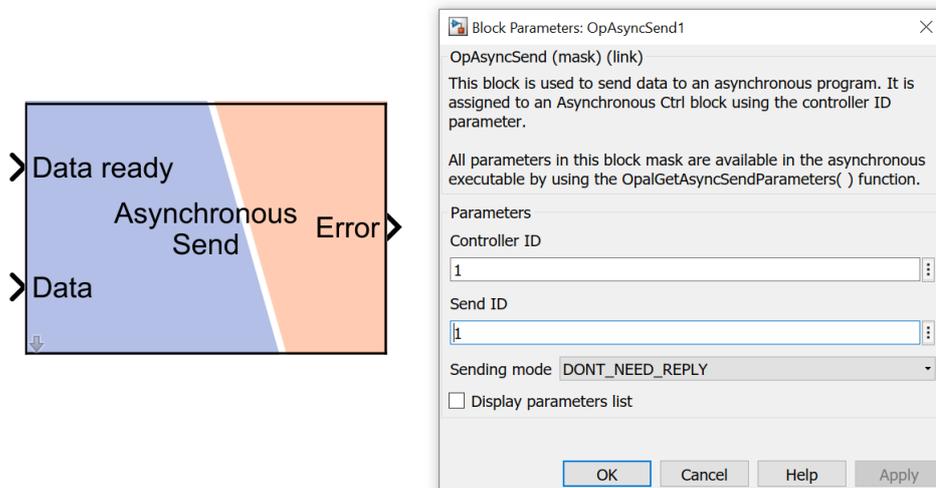


Figure 1.7: OpAsyncSend block.

Lastly, to setup a wireless communication between two real-time simulators the *OpIPSocketCtrl* block, shown in Figure 1.8 is required.

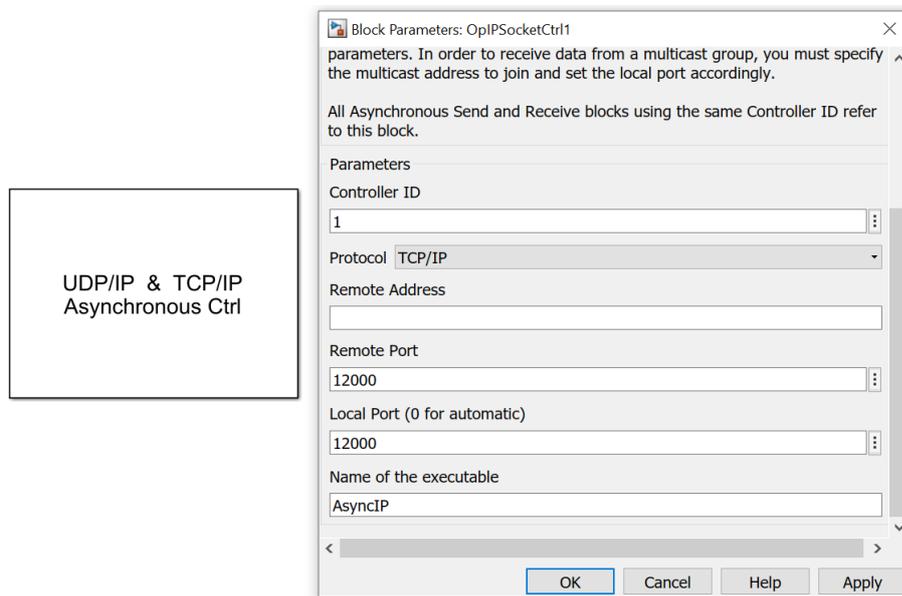


Figure 1.8: OpIPSocketCtrl block.

A last important distinction is related to the *Hardware In the Loop* and the *Power Hardware In the Loop* simulations, as done in [27]. The first one, whose working scheme is shown in Figure 1.9, included in the simulation loop hardware devices whose power required is low enough to make possible a direct connection with the realtime simulator. The Device Under Test may be a controller device, a circuit breaker, a sensor or an electronic component.



Figure 1.9: Hardware In the Loop principle scheme.

On the other hand, the *PHIL* simulation, whose working scheme is shown in Figure 1.10, provides a power hardware device to be included in the simu-

lation scheme. Here a power amplifier needs to be interposed between the realtime simulator and the power system.

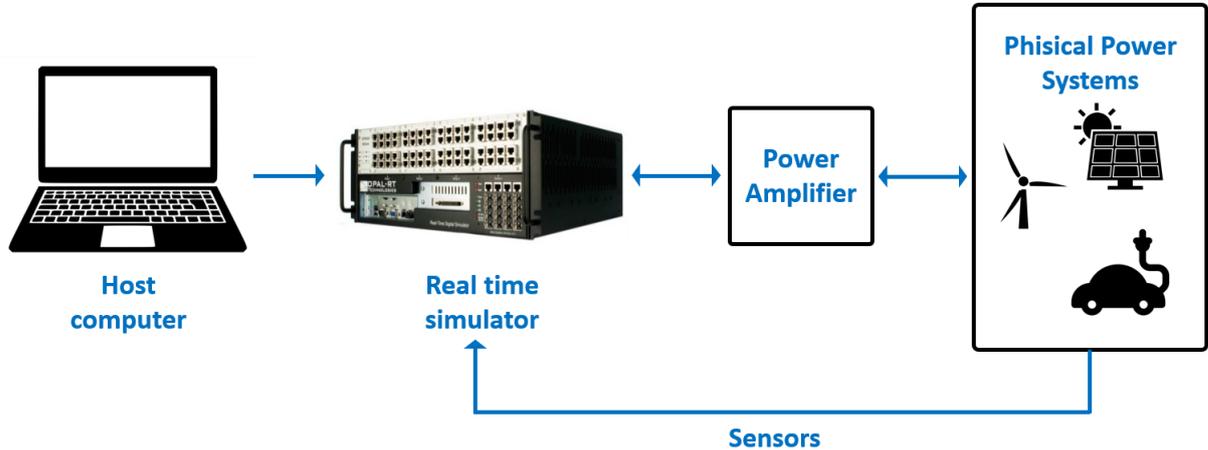


Figure 1.10: Power Hardware In the Loop principle scheme.

## 1.2 Environmental issues and trends in renewable energy

Since the creation of the *Club of Rome* in 1968, the international scientific community is aware, thanks to a continuous evaluation based on complex models of the biosphere, that the process of climate change is mostly due to the humans's activity. The Club of Rome's first contributions were pioneering at that time. The problem of a warmer planet was posed to the political class, without receiving a clear intention of response.

The activity of monitoring, modeling and reporting about the progression of climate change and its impact on the humankind has been carried out over the years by the *Intergovernmental Panel on Climate Change (IPCC)*, a United Nation working group composed by the best scientist and researcher of the field. The IPCC periodically publishes reports discussing the results of the monitoring activities, the forecasted amount of  $CO_2$  in the biosphere as a function of the policies implemented by the countries around the world. As the *IPCC Sixth Assessment Report* is expected to be completely published in 2022, a special report has been released in 2019 with the title *Global Warming of 1.5 °C* [24]. In [24] it is once again stated how the human activity is the main cause of *GreenHouse Gas (GHG)* emissions. It is also

asserted in [24] that the process of climate change has accelerated over the past 30 years, with a direct impact on:

- sea level rise;
- ocean acidification;
- polar and glacial ice loss;
- melting permafrost;

The main goal of [24] is to evaluate what might be the impact on the biosphere of an increase of  $1.5^{\circ}C$  of the average global temperature relative to the pre-industrial time (1850-1900). Considered in the past reports to be less harmful for humanity, the report [24] affirms that a  $1.5^{\circ}C$  increase would actually impacts negatively on humankind almost as much as the  $2^{\circ}C$  increase. According to [24], the  $1.5^{\circ}C$  threshold would be reached in 2040, while in the worst case scenario the same limit would be achieved prior to the 2030. The trends differ based on the implemented environmental and energy policies.

In order to keep the temperature increase below the aforementioned threshold, a reduction in  $CO_2$  emission is vital. According to [24], with the *higher overshoot* scenario, the emissions would be brought to net zero around the 2050. Other scenarios are also considered in [24] based on the implemented policies.

Given the target of reducing dramatically the GHG emissions, it is important to understand which are the main causes of such problem. The Figure 1.11, whose datas have been presented in [10], shows the primary sources of GHG emission worldwide. It is clear as the first and main contributor is the *Electricity & Heat* production sector with 31% of the contributions, followed by the *Transportation* sector with 15%.

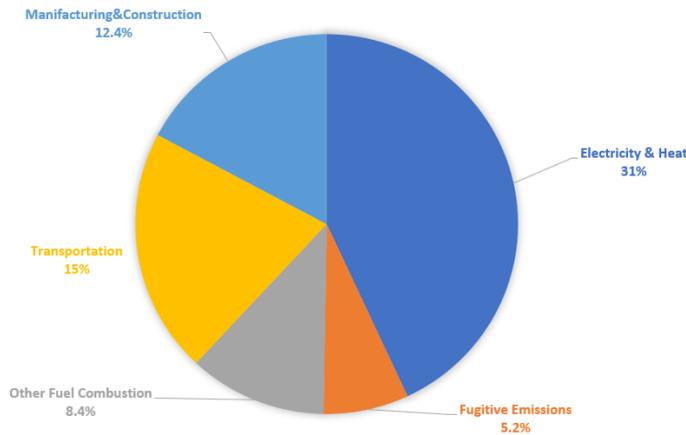


Figure 1.11: Primary sources of GHG emissions. Source: [10].

Being the main source of GHG emissions, the electricity production needs to be subjected to a dramatic change. The *Renewable Energy Sources (RES)* look as a solid candidate to play a key role in such process.

Long-term scenarios are mentioned in the *Renewable technologies in the EU electricity sector: trends and projections* report [2]. In [2] a list of scenarios related to the GHG emissions and the overall RES share are drawn up to 2050. Among those the INDC (Intended Nationally Determined Contributions) scenario and the *below 2 degree* scenario are here mentioned. The most significant targets are listed in Table 1.1 [2]. The INDC scenario implies all european countries to have a climate and energy policy, even those countries that previously did not have one. Although the results in the long-term are encouraging, they will not be enough to satisfy the ambitious climate targets set by the Paris Agreement and consequently by the European Commission. The *below 2 degree* scenario, however, seems promising in dramatically reducing the emissions, while increasing the contribution of the RES in the energy production.

Table 1.1: INDC scenario and *below 2 degrees* scenario. Source: [2].

	RES share (%)		GHG Index (%)	
	INDC scenario	<i>below 2 degrees scenario</i>	INDC scenario	<i>below 2 degrees scenario</i>
2030	27	-39	27	-44
2050	38	-50	55	-82

As abovementioned, a key aspect for a successful diffusion of the RES is the lowering in costs. The report [19] suggests that the solar PV technology has seen a sensible reduction in costs in the past decade, starting from 4 USD/W (utility scale) down to around 2 USD/W. A similar trend has been observed for PV systems on a residential level.

The faster-than-ever growth in RES installation will have a positive impact on reaching the *Sustainable Development Goals (SDG)* instituted by the United Nations. However, a more technical aspect needs to be highlighted, which is related to the impact of RES on the electric grid. More recent data show how the most of the RES newly installed systems has a small or very small capacity. This implies that the majority of the energy production from RES will be located in the Low Voltage (LV) and Medium Voltage (MV) networks. Table 1.2 summarizes what has just been described for the solar PV system installed in Germany.

Table 1.2: Percentage of PV installation based on the capacity (Germany case). Source: [19] (© IRENA 2016).

	<10 kWp	10-100 kWp	100-500 kWp	>500 kWp
%	13	44	14	29

The same trend can be found in the Italian electrical system, in which the 94.7% of the PV systems is connected whether to the LV or MV network [16]. The complete distinction is depicted in Figure 1.12.

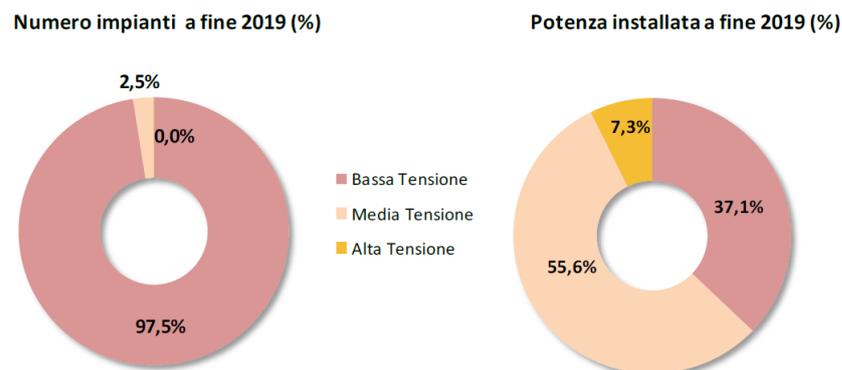


Figure 1.12: Percentage of PV installation based on the voltage level connection (Italy case). Source: [16].

This aspect imposes a drastic paradigm shift in how the overall electric grid

is operated. The *Load-following* paradigm, in which few production plants connected at the HV network are operated as a function of the predicted load, has to be abandoned in favour of a *Production-following* paradigm, where the electric system is operated based on the production profiles of the RES, mainly connected at the LV and the MV networks.

### 1.3 Microgrids: definitions and classifications

One of the major trends in the overall evolution of the electric grids is related to the *off-grid Renewable Energy System (off-grid RES)*, whose growth can be found both in the *developing* and in the *developed* countries. The advantages of the *off-grid RES* are many: smaller areas required compared to a utility-scale renewable project, lower losses related to the transmission of energy, and the possibility of integrating the renewable energy sources.

Among the *off-grid RES*, the *microgrids* play the major role, particularly in the *developed countries*. The application of the microgrids, whose growth is accelerating (it is estimated that in 2030 500 millions households will be connected to a microgrid, compared to 5 millions household in 2013 [18]), would be specifically useful in small islands and rural areas.

The concept of the microgrid does not actually represent all the possible cases. According to the size, the capacity and other specifics, we can define the *minigrid*, the *microgrid* and the *picogrid*. Figure 1.13 (© IRENA 2015) shows a first classification based on the users, the applications and the key elements [18].

	Grids		
	AC/DC		AC
<i>System</i>	Nano-grid Pico-grid	Micro-grid, Mini-grid	Full-grid
<i>Application</i>	Lighting, appliances, emergency power	all uses	all uses
<i>User</i>	Community; Commercial	Community; Commercial; Industry	
<i>Key component</i>	Generation + single-phase distribution	Generation + three-phase distribution + controller	Generation + three-phase distribution + transmission

Figure 1.13: A classification of microgrids based on user, application and key elements. Source: [18] (© IRENA 2015).

Furthermore, a classification based on the installed power, the capability and the complexity is proposed and highlighted in Figure 1.14 (© IRENA 2015).

	Size (kW)	Capability	Complexity
Stand-alone systems	0 - 0.1		
Pico-grid	0 - 1	<ul style="list-style-type: none"> <li>• Single controller</li> </ul>	
Nano-grid	0 - 5	<ul style="list-style-type: none"> <li>• Single voltage</li> <li>• Single price</li> <li>• Controllers negotiate with other across gateways to buy or sell power</li> </ul>	<ul style="list-style-type: none"> <li>• Both grid-tied and remote systems</li> <li>• Preference for DC systems</li> <li>• Typically serving single building or single load</li> <li>• Single administrator</li> </ul>
Micro-grid	5 - 100	<ul style="list-style-type: none"> <li>• Manage local energy supply and demand</li> <li>• Provide variety of voltages</li> <li>• Provide variety of quality and reliability options</li> <li>• Optimise multiple-output energy systems</li> </ul>	<ul style="list-style-type: none"> <li>• Incorporate generation</li> <li>• Varying pricing possible</li> </ul>
Mini-grid	0 - 100 000	<ul style="list-style-type: none"> <li>• Local generation satisfying local demand</li> <li>• Transmission limited to 11 kV</li> </ul>	<ul style="list-style-type: none"> <li>• Interconnected customers</li> </ul>

Figure 1.14: A classification of microgrids based on size, capability and complexity. Source: [18] (© IRENA 2015).

## 1.4 TSO-DSO interaction

As it has been mentioned in the previous paragraph a large number of solar and wind systems is connected at the Distribution Network (DN) at the MV and at the LV grid. Moreover, in the past few years and in the recent future the decommissioning of the thermal electric plants, with their synchronous generators, will be another key trend in the evolution of the electric grid. This second aspect implies a progressive reduction in the inertia of the grid and in the provision of Ancillary Services (AS), such as *frequency regulation*, *voltage regulation* and *congestion management*. The faster-than-ever growth in Distributed Generation (DG) systems and contemporary a reduction of the number of thermal electric plants requires a different approach for providing the AS.

A more innovative and more constructive interaction between the TSO and each DSO may result successful in achieving the abovementioned targets, as it is explained in the ISGAN's report "*TSO-DSO interaction: An Overview of current interaction between transmission and distribution system operators and an assessment of their cooperation in Smart Grids*". The report [20] analyzes a series of critical situation in which a TSO-DSO interaction may bring a positive impact and how TSO and DSO may intervene in any of these specific cases.

The first possible situation is the *Congestion of the TSO-DSO interface node*. Such node is represented by a HV-MV transformer (TRFO), usually provided with an OLTC system. It is worth noting that in case the TRFO is owned by the DSO the congestion problem can be managed by itself without the cooperation of the TSO. However, if such asset is owned by the TSO, a collaboration between the two operators will be required [20].

The proposed collaboration algorithm to safely manage this problem is shown in Figure 1.15. The TSO constantly monitors the HV-MV transformer load conditions and whenever a safety threshold is reached the DSO will be contacted. The DSO will evaluate, considering the DN constraints, the flexibility resource to be used to reduce the load requests. In the process of selecting the flexible resource to exploit, the DSO would implement the *Last-In, First-Out* principle: the last flexible resource to be connected at the DN will be the first one to be exploited in the congestion management [20].

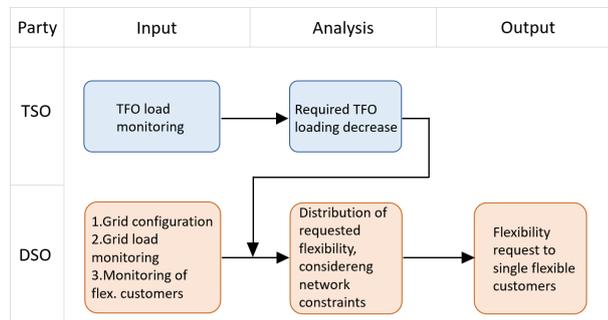


Figure 1.15: Proposed process to avoid HV-MV transformer congestion. Source: [20].

A second scenario proposed in [20] is the *Congestion of the transmission lines*. Such problem may be caused by more than one HV-MV transformer. The scheme proposed to face this problem is depicted in Figure 1.16. The process is an evolution of the one shown in Figure 1.15. In order to know the available flexibility, the TSO will require a real-time data exchange with the Transmission Network (TN) flexible resources and the ones connected at the DNs, which are aggregated at each TSO-DSO node. A sheer competition between flexible resources has to be guaranteed by a proper regulation. For such reason, a specific AS market has to be implemented in order to select the flexible resources merely based on their technical and economical competitiveness [20].

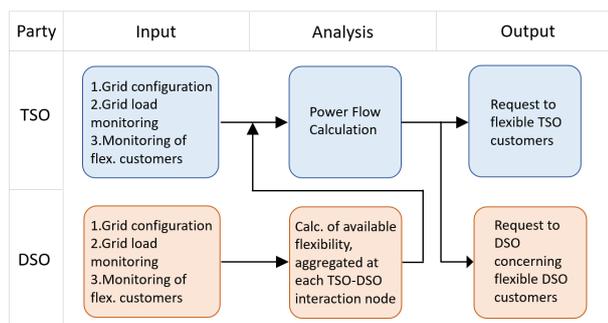


Figure 1.16: Proposed process to avoid TSO line congestion. Source: [20].

A third scenario debated in [20] is the *support between TSO and DSO for the voltage control*. In fact, the DN voltage can be managed by implementing the OLTC system, while the DSO can contribute to the TN voltage control by activating flexible resources of the DN, whether they are DG systems or capacitor banks.



Nowadays, in most of the cases the TSO discusses directly with the flexible resources connected at the distribution level, without involving the DSO. However, five possible market structures, which have a different level of feasibility, are presented in [21],[15]: the *Centralized AS Market model*, the *Local AS Market model*, the *Shared Balancing Responsibility Market model*, the *Common TSO-DSO market*, the *Integrated flexibility market*.

Those five schemes differ from each other based on which stakeholder plays a specific role. Those roles are defined as one or more actions that cannot be shared with other stakeholders [21].

The *Centralized AS Market model* sees the TSO as the Market Operator (MO), managing not only the TN's flexible resources, but also the DN's ones. In such structure, the TSO does not consider the DN's constraints, unless the DSO itself demands it. The DSO is required to send the requested data to the TSO, or otherwise guaranteeing the complete observability of the DN at the TSO. Finally, a pre-qualification system to verify that the DN's flexible resources do respect the DN's constraints may be included [21]. The schematic representation is depicted in Figure 1.19.a. This structure requires little communication between the DSOs and the TSO, with a consequent small amount of data exchange. This solution permits to limit the costs of market operation, while making the process to participate in the market simple. However, the DSO is not able to exploit the available flexibility and local constraints are not always respected. At the current state of the electricity market, the *Centralized AS Market model* would be the most compatible solution for the European context [21].

The *Local AS Market model* has the DSO managing the AS Market for the resources connected at the DN. The DSO will have priority in using the DN's flexible resources for local congestions. Whenever there are resources unused by the DSO, the DSO itself will aggregate them and after verifying the local constraints, transfer them to the TSO's AS Market. The activation of flexible resources for local congestions could cause local imbalances. For such reason the DSO notifies it in advance to the TSO, so that the TSO may be able to make corrections. The main disadvantage of this proposed scheme is the lack of economy of scale in case of an excessive number of small local markets. However, the small size may result in a higher capacity of satisfying the DSO's needs. This market system is represented in Figure 1.19.b [21].

The *Shared Balancing Responsibility Market model* slightly differs from the previous market system: here the TSO and DSO still operate two sepa-

rated AS markets. However, the exceeding DN's flexible resources cannot be transferred to the transmission's AS market. The TSO is not allowed to exploit the DN's flexible resources under all circumstances. The DSO has here the responsibility to manage non only the local congestions, but the local balancing as well. Like for the previous solution, this scheme may have a problem of liquidity due to AS market with an excessive small size. The market structure is shown in Figure 1.19.c [21].

The *Common TSO-DSO market* has a unique market run jointly by DSO and the TSO. Such market can acquire both transmission and distribution flexible resources. The flexibility is allocated to whom may have the higher need. This scheme could be run in two separated ways. In the first case, all the bids (transmission's and distribution's resources) are made together and cleared in a single market session. The clearing considers both the transmission and the distribution constraints. This approach may bring to a heavy computational burden. In the second approach there are decentralized markets. Those are run first by the DSOs with the goal to satisfy the DSOs' needs, while considering the local constraints. The outputs of the clearing of those markets are then sent to the second stage market run by the TSO. This proposed scheme permits to minimize the total system costs. The market structure is shown in Figure 1.19.d [21].

The *Integrated flexibility market* has a common market in which both regulated players (system operators) and non-regulated players (commercial parties) take part. To guarantee the respect of the competition within the market, an independent Market Operator will be required. The flexibility will be available for both the regulated and the non-regulated players. The market is cleared based on whom is willing to pay the most for the flexible resource. DSO and TSO are allowed to resell the unused flexibility previously bought in the market. The highest number of participants in the market increases the level of liquidity. The market structure is shown in Figure 1.19.e.

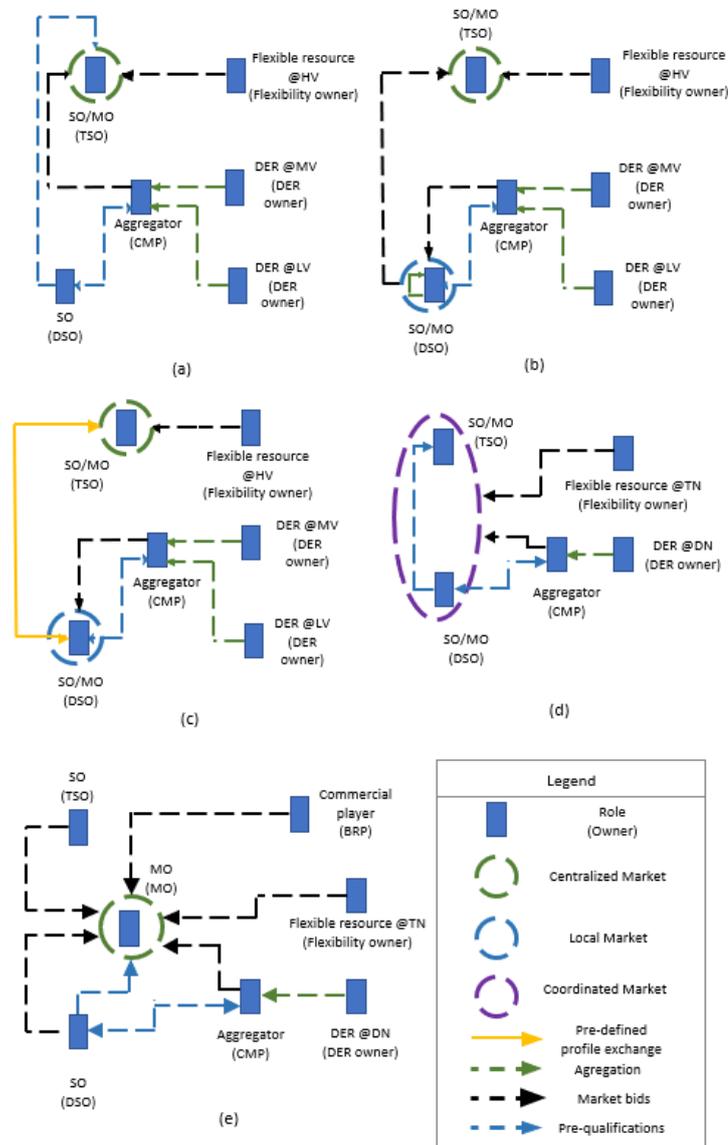


Figure 1.19: (a) Centralized AS market model - (b) Local AS market model - (c) Shared balancing responsibility model - (d) Common TSO-DSO AS market model - (e) Integrated Flexibility market model. Source: [21].

## Chapter 2

# Remote co-simulations between Turin and Bari

In this chapter a number of remote realtime co-simulations are discussed. Such simulations have been carried out within a collaboration between the Politecnico di Bari and the Politecnico di Torino. This kind of collaboration, and the sharing of both hardware and software resources, makes easier to implement a wide range of studies related to the electric grid, including a deeper analysis of the Transmission-Distribution coordination.

Specifically, the implementation of the R-PHIL simulation (Remote Power Hardware In the Loop) includes the PoliBa's LabZero facility and the PoliTo's G-RTSLAB facility. The facilities have already been exploited for the simulations described in [7]. The PoliBa's facility is composed by a OPAL-RT 5600<sup>©</sup> real time digital simulator, a 16 kVA 4-quadrants programmable power source (PMI15A30F60). Such controllable power output can be used to communicate either with a RLC local bank or to exchange active and reactive power with a microgrid, located in the same research facility. In the PoliTo's facility, which is equipped with a long list of device for realtime applications, only the digital realtime simulator OPAL-RT 5600<sup>©</sup> will be exploited.

The two laboratories are located at a distance of around 1000 km [7]. The considerable distance entails the presence of a latency in the remote communication between the two laboratories. The first test to be discussed in this chapter will be the assessment of such delay.

The two OPAL-RT<sup>©</sup> simulators exchange datas by means of a Virtual Private Network (VPN) tunnel, employing an IPsec encryption key for security reasons. The two simulators are set and programmed to exchange data using

asynchronous messaging with UDP protocol [7].

The second section of the chapter discusses a first experimental implementation of a R-PHIL realtime co-simulation. The model includes a distribution feeder with an OLTC HV-MV transformer and a RLC controllable load connected in the loop at the Bari's facility. The aim is to evaluate how the flexible resource connected at the LV grid could be dispatched using the tap changer of the HV-MV transformer when an external command is triggered.

The third section illustrates a R-PHIL realtime co-simulation of the same model described in the second section, but in which the transformer has not the tap changer. The aim is to assess the stability of the model and to find the conditions that bring the simulation results to diverge.

Some of the results listed in the following chapter have been later published in the paper [8].

## 2.1 Analysis of the communication latency

The first step has been the assessment of the latency of the remote communication between the simulators of the two laboratories. The latency is an aspect that can potentially disrupt a remote realtime co-simulation, specifically in high-dynamics transients. In order to determine the quality of the remote communication, a systematic study has been carried out, repeating the test at different hours of different days. The study includes two key parameters to characterize the latency: the sample time at which data are exchanged and the size of the exchanged data packets.

As already mentioned, the remote communication uses a UDP/IP protocol. Its main advantage is to always use the most updated data available. This aspect implies that when a data packet is missing or heavily delayed, the regular communication can be restored as soon as a new proper packet arrives [8].

The data exchanged consists in a stair case signal of integer numbers, which is created in the Turin Simulink model. The data is then sent to Bari and then sent back to the Turin simulator. This same test has been implemented a number of times with a variation in the sample time and in the data packet size. Particularly, those two parameters have been varied as follows:

- The size of the exchanged data packets has been varied from 2 up to 96 (2, 6, 12, 24, 48 and 96);

- The sample time of the communication has been varied from 1 ms up to 20 ms (1 ms, 5 ms, 10 ms and 20 ms);

The 96-sized data packet represents the maximum theoretical number of signals that can be transmitted from the realtime simulator at PoliBa onto the programmable power source [8].

This parameterization has enables us to identify possible patterns and evaluate which was the possible worst case scenario. Please note that the tests have been implemented at different moments without a predefined time schedule. This means that the results may be affected by how much the data network was congested in the moment in which a simulation test was being held. The obtained results are shown and discussed later in the chapter.

The Simulink model implemented for such test is here shown and described. In Figure 2.1 it is illustrated the top level of the model, merely composed by subsystem blocks. For this specific case, a SC\_Console block and only one SM\_Master block have been implemented. The former includes the scopes in which observing the signal and the commands to be used during the simulation, while the latter includes the scheme that generates the digital signal, the sending and the receiving operations.

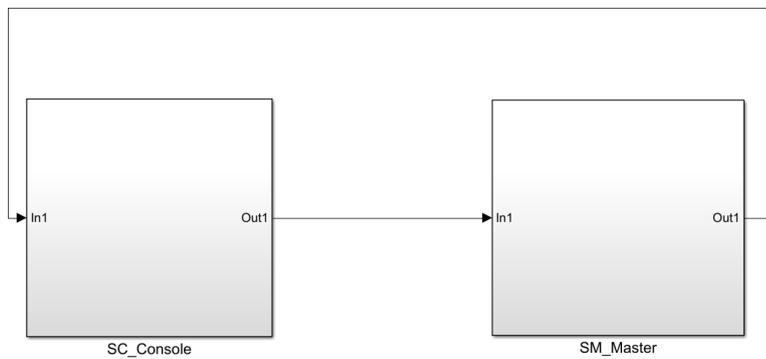


Figure 2.1: Simulink model: overall vision

The scheme included in the SM\_Master block is shown in Figure 2.2. The digital signal is generated with the components located in the right low part of the figure, while on the top there is the RT-LAB<sup>©</sup> block to save the data on file.

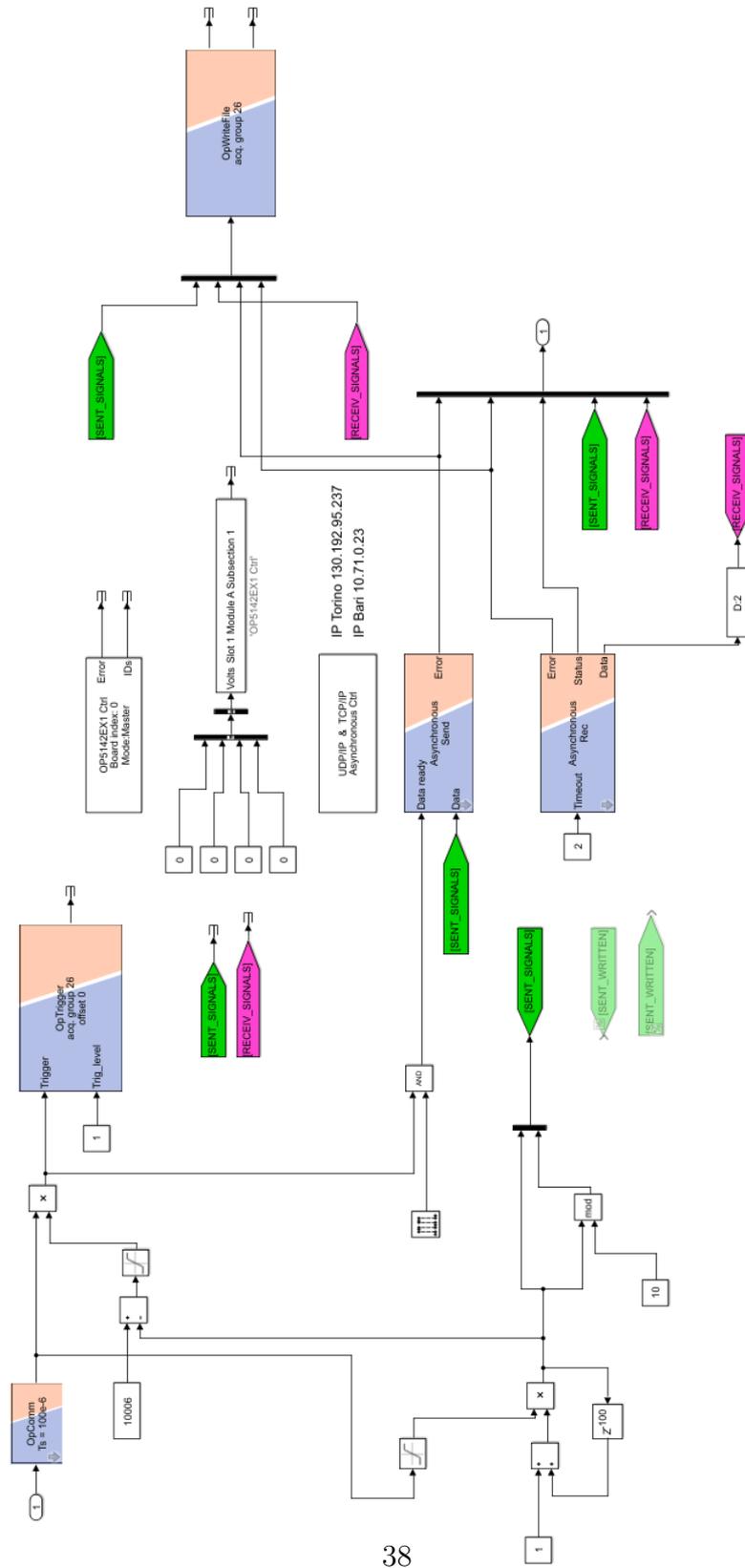


Figure 2.2: Simulink model: Master block scheme

On the left side of Figure 2.2 it is located a *OpTrigger* block. This is exploited during the simulation to interrupt the data saving-on-file process at the desired time. Figure 2.2 shows the simulation model for the case with 2 data exchanged and a time delay set at 10 ms.

As previously said, the test has been carried exchanging data packets of different size (2, 4, 6, 12, 24, 48 and 96 signals) and different values of sample time of the communication (1 ms, 5 ms, 10 ms and 20 ms) for an overall number of 24 simulations. The simulation has been executed with a realtime coordination between the PoliTo's simulator and the PoliBa's simulator, in order to correctly save the exchanged datas.

The data analysis of the aforementioned simulations have had the goal to evaluate the amount of time necessary to send the data from PoliTo to PoliBa and the amount of time to send the same data from PoliTo to PoliBa and back. Having the latency for each data packet sent/received, a series of statistically relevant information may be extracted (average value and standard deviation). Moreover, the number of packets correctly received from PoliBa is determined, and thus the number of the packets not received is easily deduced.

A critical aspect related to the UDP/IP communication protocol needs to be furtherly highlighted. To guarantee a high validation of the remote co-simulation, the lowest the number of lost packets, the better will be. However, in case a packet is lost, the two simulations (at PoliBa's facility and at PoliTo's facility) will use the last available packet. By doing so, the correctness of the overall simulation will not be excessively affected [7].

To correctly analyze the datas resulted by the simulations it's necessary to synchronize on the same timeline the PoliBa and the PoliTo datas. In fact, the realtime simulators in the two research facilities work on asynchronous timelines (see Figure 2.3). The algorithm, which is illustrated in the Appendix A, procedes by aligning the staircase exploiting the instants at which the two signals have the same amplitude. The procedure also takes care of the lost packets in order to avoid an uncorrect alignment.

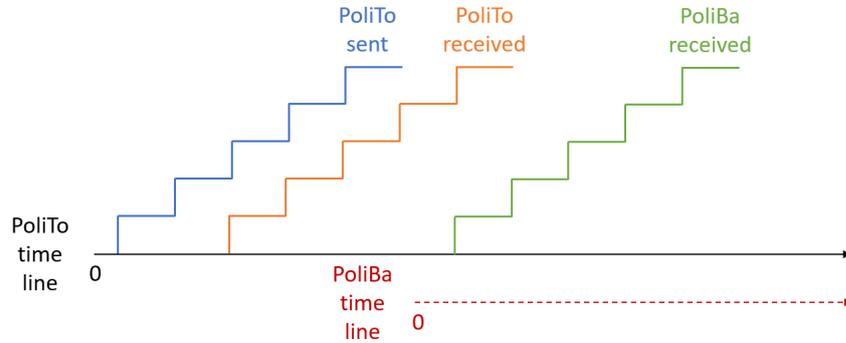


Figure 2.3: Exchanged signals prior to the time synchronization.

As previously mentioned, the not received packets from PoliBa are counted. The Matlab code implemented to do so is illustrated in the Appendix A. To correctly evaluate whether or not the packet has been lost it is essential to distinguish the cases in which such thing happens from the cases in which it does not. Figure 2.4 shows a case in which the sending of the packet has been fully executed (left picture) and a case in which the packet has not been received as expected (right picture).

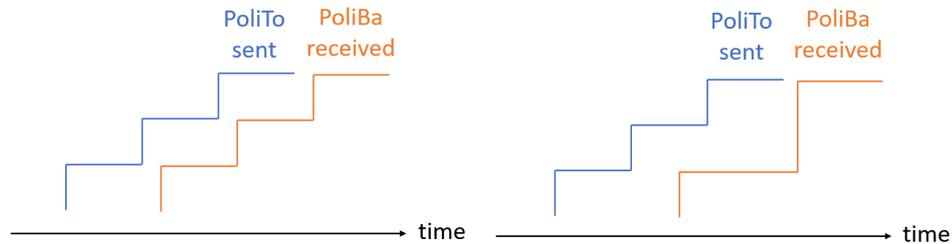


Figure 2.4: Exchanged signals: qualitative representation.

The above described algorithm has brought the following results. In Table 2.1 the statistical analysis regarding the sending of the packet from PoliTo to PoliBa is depicted, distinguishing the size of the exchanged data packets and the sample time for the communication. Secondly, the latencies generated in sending data from PoliTo to PoliBa are depicted in the boxplot of Figure 2.5, while the boxplot of Figure 2.6 shows the latencies of the signal sent by PoliTo and then received back by PoliTo (passing through PoliBa). Finally, Table 2.2 and Table 2.3 summarize the statistical specification of the PoliTo-PoliBa

latencies and the PoliTo-PoliTo latencies, respectively, distinguishing the size of the exchanged data packets and the sample time for the communication.

Table 2.1: Statistics on data packets exchange from PoliTo to PoliBa

sample time = 1 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
received packets	9907	9921	9899	9945	9796	9976
lost packets	93	79	101	55	204	24
lost packets (%)	0.93	0.79	1.01	0.55	2.04	0.24
sample time = 5 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
received packets	10000	9999	10000	9998	9999	10000
lost packets	0	1	0	2	1	0
lost packets (%)	0	0.01	0	0.02	0.01	0
sample time = 10 ms						
	2 data	6 data	12 data	24 data	48 data	96 data
received packages	9999	10000	9996	9998	10000	10000
lost packages	1	0	4	2	0	0
lost packages (%)	0.01	0	0.04	0.02	0	0
sample time = 20 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
received packets	10000	10000	9999	9999	10000	10000
lost packets	0	0	1	1	0	0
lost packets (%)	0	0	0.01	0.01	0	0

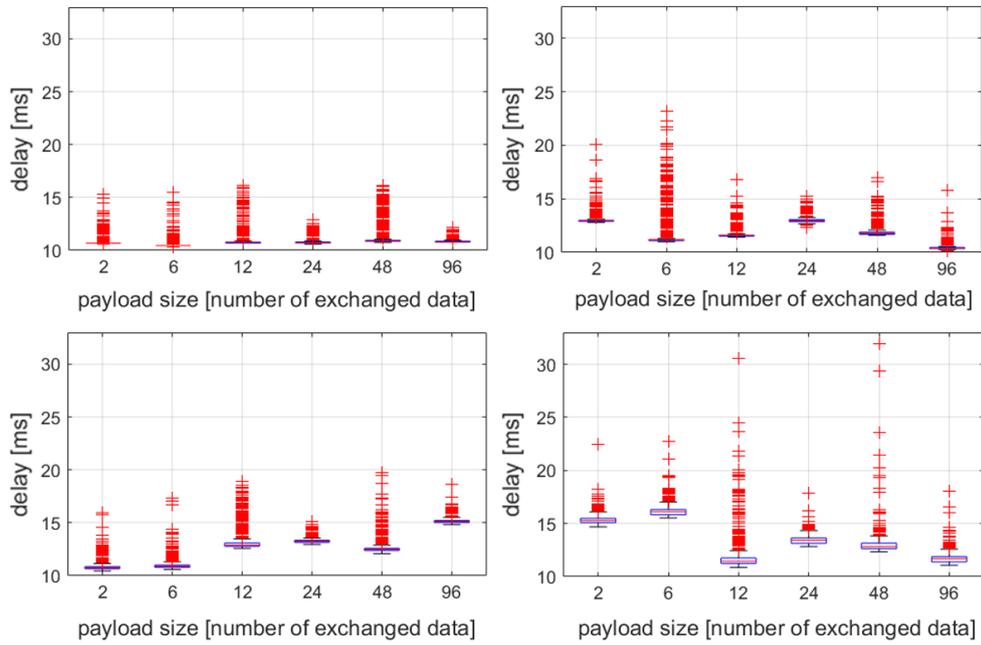


Figure 2.5: Latencies on the PoliTo-PoliBa data exchange: (Top-left) 1 ms; (Top-right) 5 ms; (Bottom-left) 10 ms; (Bottom-right) 20 ms.

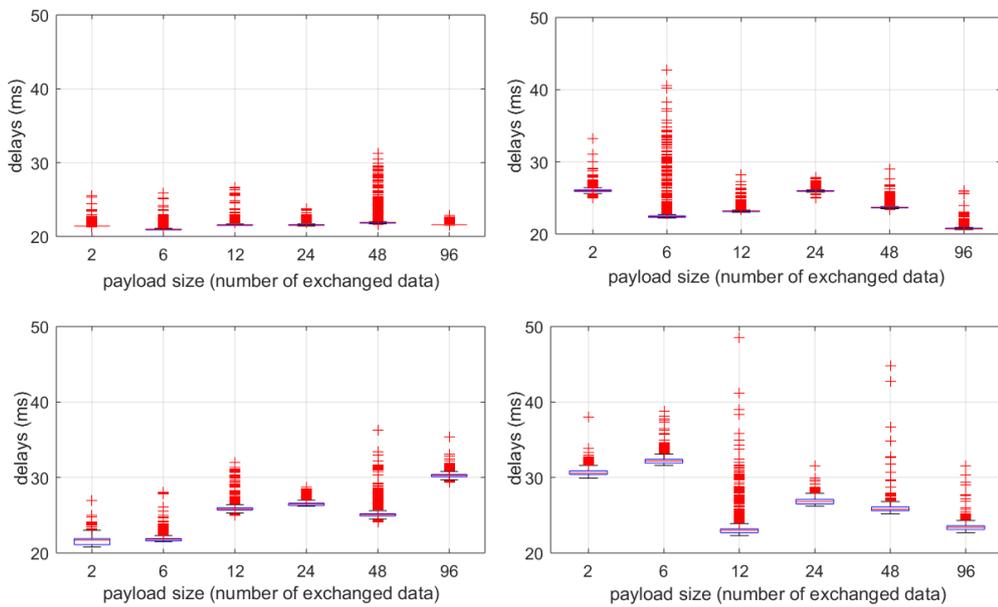


Figure 2.6: Latencies on the PoliTo-PoliTo data exchange: (Top-left) 1 ms; (Top-right) 5 ms; (Bottom-left) 10 ms; (Bottom-right) 20 ms.

Table 2.2: Statistics on latencies of the PoliTo-PoliBa data exchange

sample time = 1 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	10.72	10.51	10.8	10.77	10.97	10.81
std. deviation (ms)	0.19	0.19	0.25	0.17	0.42	0.11
sample time = 5 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	13	11.27	11.6	12.98	11.87	10.39
std. deviation (ms)	0.22	0.6	0.2	0.17	0.27	0.16
sample time = 10 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	10.79	10.9	12.94	13.25	12.53	15.12
std. deviation (ms)	0.24	0.24	0.39	0.19	0.34	0.2
sample time = 20 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	15.32	16.1	11.53	13.41	12.92	11.68
std. deviation (ms)	0.34	0.33	0.59	0.29	0.47	0.33

Table 2.3: Statistics on latencies of the PoliTo-PoliTo data exchange

sample time = 1 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	21.44	21.01	21.59	21.53	21.95	21.63
std. deviation (ms)	0.17	0.23	0.23	0.16	0.54	0.1
sample time = 5 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	26.02	22.54	23.2	25.97	23.73	20.78
std. deviation (ms)	0.27	1	0.22	0.19	0.31	0.23
sample time = 10 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	21.59	21.81	25.88	26.5	25.05	30.25
std. deviation (ms)	0.44	0.29	0.41	0.25	0.55	0.27
sample time = 20 ms						
payload size	2 data	6 data	12 data	24 data	48 data	96 data
average value (ms)	30.64	32.21	23.05	26.82	25.84	23.37
std. deviation (ms)	0.37	0.4	0.82	0.35	0.49	0.38

Firstly, it can be appreciated by Table 2.1 how the higher the time delay set for the data sending is, the lower number of unreceived packets there will be. In fact, there is a wider time range and thus it will be easier not to lose a data packet during the remote communication between the PoliBa's simulator and the PoliTo's one. Moreover, no pattern can be found between the number of data sent and the number of lost packets. Such result has a positive consequence on the future simulation: if a higher number of data is required to be exchanged there will not be a negative effect on the number of lost data packages.

Secondly, Figure 2.5 and Figure 2.6 do not show any relation between the number of data exchanged and the latencies that occurred between PoliTo and PoliBa [8]. Such result, like for the number of lost packets, highlights how in the analysed range the remote communication will not be sensitive to the amount of data exchanged by the two research facilities. The randomness in the correlation between the x-axis and the correspondent y-axis result in the four subplots can be found in the fact that those tests have been carried out at different times of the day. Specifically, it is the case of the simulations with:

- payload size of 6 and a time delay of 5 ms;
- payload size of 12 and a time delay of 20 ms;

For the two above listed tests the relevant standard deviation may be justified by the time of the day at which such tests have been executed. A specific event may have brought to an overloaded network, and thus to higher latencies [8].

Finally, Table 2.2 and Table 2.3 summarize the main information related to the two previous plots.

## 2.2 OLTC transformer and its application in the grid

An OLTC is a transformer whose winding ratio can be modified without interrupting the load current. Such adjustment can be also executed automatically, as it will be explained later in this paragraph. The OLTC transformer can be used in different conditions. However, the HV–MV application will be exclusively taken in consideration in this work.

The schematic representation of the OLTC transformer is shown in Figure 2.7, in which the regulator is already depicted.

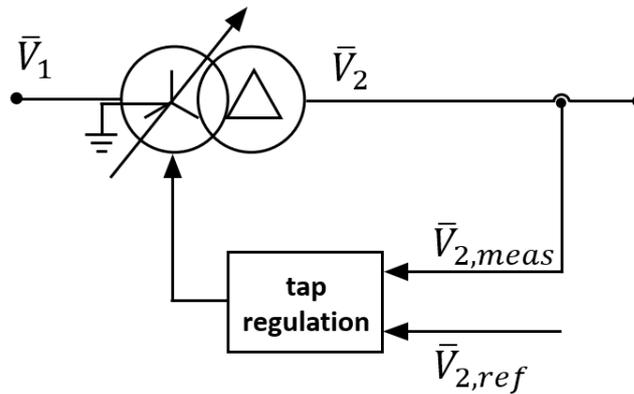


Figure 2.7: OLTC's schematic representation.

The OLTC transformer principle is based on the possibility to vary, in a discretized way, the number of winding turns (on the primary side or on the secondary side) and thus modifying accordingly the winding ratio. From a construction point of view the winding on which the regulation occurs will have a certain number of tap  $N_{tap}$ . The regulation of the taps can go from the position  $+N_{tap}$  to the position  $-N_{tap}$ , including position 0, on which the nominal winding ratio will be obtained.

The typology of OLTC transformer that will be analyzed more deeply enables the user to adjust the voltage magnitude, although in a limited way as the regulation range is around  $\pm 10\%$  of the nominal value.

The application of the OLTC plays a key role in the cooperation between the TSO and the DSO. In fact, the variation of the winding ratio with the con-

sequent voltage regulation enables to dispatch reactive power in the Transmission system. In order to understand analitically the phenomenon, let us consider the situation shown in Figure 2.8, in which two power systems are connected by a OLTC transformer [28]. In Figure 2.8.a  $V_s$  and  $V_r$  are the constant voltages of the two transmission systems. In order to obtain the analitical results more easily the circuit of Figure 2.8.b is deduced.

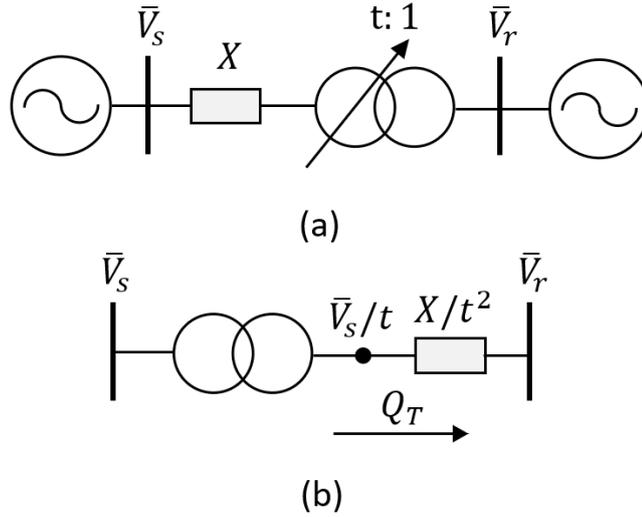


Figure 2.8: (a) Equivalent circuit of a transmission system with OLTC transformer; (b) Equivalent circuit with the parameters transferred on the secondary side (receiver) [28].

By the equivalent circuit of Figure 2.8.b the voltage drop between the two busbars can be obtained with the following procedure:

$$\Delta V = \left(\frac{V_s}{t}\right) - V_r = \frac{X}{t^2} \cdot \frac{Q_T}{V_r}$$

From which it can be deduced:

$$(V_s V_r t - V_r^2 t^2) \frac{1}{X} = Q_T$$

Finally, in case  $V_r = V_s = V$

$$t(1 - 1) \frac{V^2}{X} = Q_T \quad (2.1)$$

From equation 2.1 and considering the conventional direction of  $Q_T$  chosen in the circuit depicted in Figure 2.8.b, two following cases are possible:

- if  $t < 1$ ,  $Q_T$  will be positive, which means that the vars will flow from  $V_s$  into  $V_r$ ;
- if  $t > 1$ ,  $Q_T$  will be negative, which means that the vars will flow from  $V_r$  into  $V_s$ ;

The adjustment of the winding ratio can be obtained automatically, as already mentioned. To do so, a regulator will be required. The control block is highlighted in Figure 2.9.

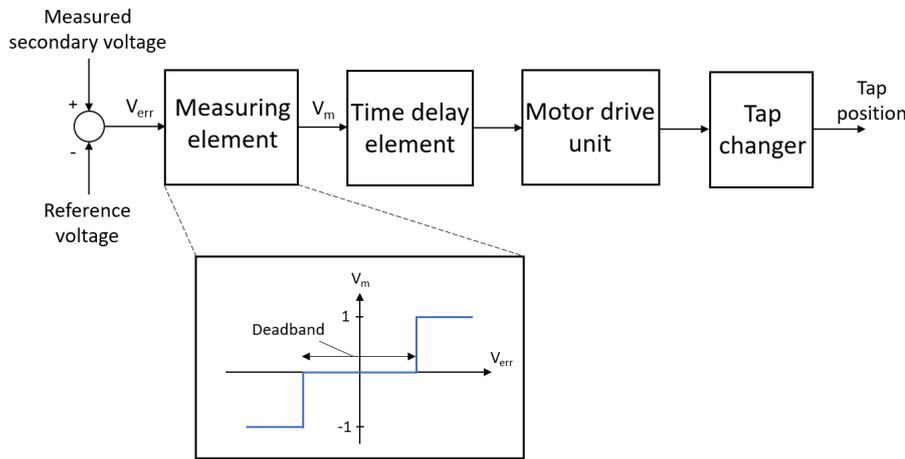


Figure 2.9: OLTC control scheme.

The control implementation is composed of a series of parts, as described with more details in [23], each one executing a specific task. The first step, after having compared the reference voltage  $V_{ref}$  and the measured secondary voltage  $V_{2,meas}$ , involves evaluating whether or not the voltage gap requires the winding tap adjustment. This operation implies confronting the  $V_{err}$  with a deadband: within the deadband the output command  $V_m$  will be 0, differently, the command  $V_m$  will be +1 or -1 depending if the winding ratio has to be increased or decreased [23].

The next step is related to determine the time delay to avoid unnecessary winding ratio adjustments as a response to transient phenomena. The output of such block,  $V_D$ , is normally equal to 0. However, if the accumulated time delay overcomes the preset  $T_D$ , the output  $V_D$  will be set to  $V_m$  [23].

The command  $V_D$  is sent to the motor drive unit block, in which a mechanical delay related to the tap adjustment is imposed. The motor drive unit is exploited to modify the position of a mechanical switch. By doing so, a specific tap is short circuited so to adjust the winding ratio accordingly. A schematic representation of this mechanism is depicted in Figure 2.10.

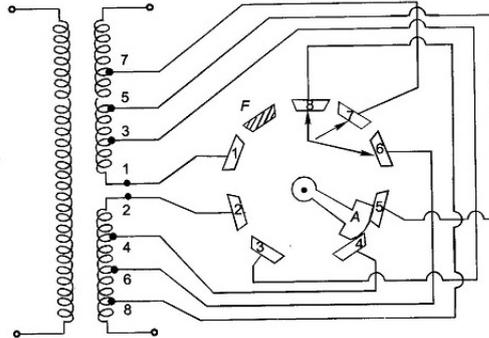


Figure 2.10: Motor drive unit schematic representation. Source: [14].

From Figure 2.10 it can be highlighted the rotating contact, which will move accordingly to the voltage control. The desired winding ratio is obtained bringing the rotating contact in the necessary position. The steady contact corresponding to the required winding ratio will be short circuited. By shifting from a steady contact to another, the rotating contact will temporarily short circuit two steady contacts.

Finally, the tap changer block imposes the new winding ratio as a function of the received command and of the previous winding ratio's value.

## 2.3 R-PHIL co-simulation with the OLTC

In this paragraph the first realtime co-simulation with PHIL is described. The aim of the co-simulation is to study how a flexible resource (a controllable RC load bank) can be dispatched when an external command is given. The increase in voltage of the flexible resource will trigger the tap changer of the HV-MV transformer, bringing the voltage at a new operating point. This study is implemented with the assumption to have a *Centralized AS market model*, in which the TSO plays the central role in dispatching both the transmission and the distribution flexible resources.

The implemented model simulates a MV feeder (22 kV) to which a microgrid and a flexible source are connected. The HV network (220 kV) is considered ideal and modeled with a Thevenin equivalent circuit. A schematic representation of the grid is shown in Figure 2.11.

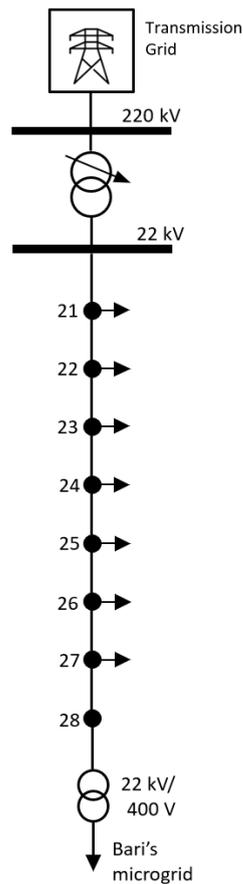


Figure 2.11: Schematic representation of the simulated distribution feeder.

The LV flexible source is an RC load bank located at the PoliBa's LabZERO facility that can be activated by a smart controller. The TSO can send a dispatch signal in order to activate an additional RC load of 64 var (capacitive) and 450 W. Such a small variation would be negligible for sustaining the transmission grid. For this reason and to better appreciate the voltage variation, the  $(P, Q)$  signals coming from the measures at Bari are amplified with a scale factor (x250) and then inserted in the simulated model of the distribution feeder (see Appendix A).

In the reality, the TSO may require an active and reactive contribution so to regulate the frequency and the voltage amplitude. The constant dismantling of thermal electric power plants needs to be counter balanced with new sources able to regulate the active and reactive power flows. As mentioned in the previous chapter, the presence of flexible sources connected to the LV grid may represent part of the solution to the previously mentioned problem.

The simulation has been fulfilled by modeling part of the electric network at the PoliTo's facility and the rest at the PoliBa's facility. Specifically, at PoliTo the HV level, the MV level, and part of the LV level have been simulated on software, while at PoliBa the rest of the LV level and the hardware in the loop have been implemented. The overall implementation scheme, as described in [7], is depicted in Figure 2.12.

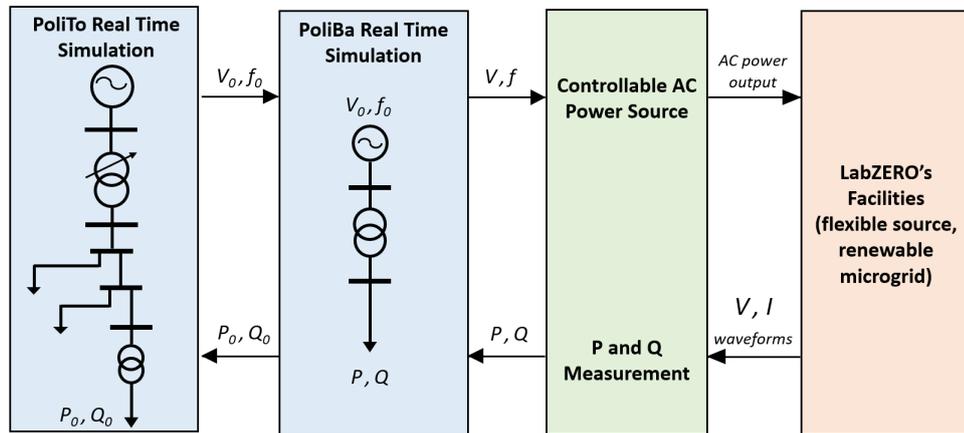


Figure 2.12: Realtime co-simulation scheme.

As it has been shown in the recent work in [7], it would not be feasible to exchange instantaneous waveforms between the two facilities. The sinusoidal signals would be heavily impacted by the data losses, obtaining an

unacceptable distortion of the signal itself with a THD excessively high. For such reason, PoliTo sends to PoliBa, besides the reference frequency  $f$ , the reference voltage amplitude  $V$ . Moreover, a loopback is implemented, with the active power  $P$  and the reactive power  $Q$  sent back from PoliBa to PoliTo.

In the PHIL implementation at PoliBa, the flexible source is controlled by a controllable AC power source, as already done in [7]. Also,  $P$  and  $Q$  of the flexible source are measured and inserted in the software model to be sent back to PoliTo.

The Simulink model's top level is merely composed by a SC\_Console and a SM\_Master, like the simulation concerning the analysis of the communication latency. Figure 2.13 shows the scheme inserted within the SM\_Master. The SM\_Master block contains the blocks necessary to fulfill the data exchange with PoliBa, the *OpTrigger* to correctly terminate the data saving at a desired instant and the electric grid simulated at PoliTo. Moreover, from the image it can be appreciated the loopback of  $P$  and  $Q$ , which are received from PoliBa and sent in the electric grid modeled at PoliTo. Specifically, the  $P$  and  $Q$  amplitudes are sent to the controllable load block simulating the LV flexible resource. The electric grid's scheme is illustrated in Appendix A.



highlighted later in the paragraph. The model will be improved for the simulations described in the following chapters.

The scheme of the implemented OLTC model is illustrated in Appendix A.

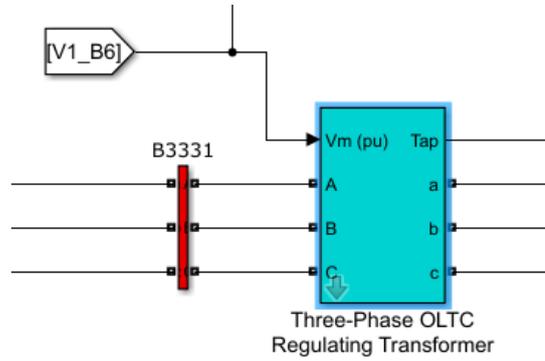


Figure 2.14: OLTC Simulink model.

The implemented OLTC transformer has a  $Yg-d$  connection and it is characterized by the parameters listed in Table 2.4.

Table 2.4: OLTC Transformer main characteristics.

<b>Transformer Parameters</b>	
Parameter	Value
Nominal power (MVA)	63
HV Nominal Voltage (kV)	220
MV Nominal Voltage (kV)	22
<b>OLTC Control Parameters</b>	
Initial Tap Position	-5
Voltage Deadband (pu)	0.025
Voltage Reference (pu)	0.978
Voltage Step per Tap (pu)	0.0125
Delay (s)	1

The results are highlighted in the following images. In Figure 2.15 the voltage amplitude of the LV flexible source is shown. The Figure 2.16 depicts the active power and the reactive power of the LV flexible source. Finally, Figure 2.17 illustrates the command trigger of the LV flexible source and the active and reactive power in the TSO-DSO connection node.

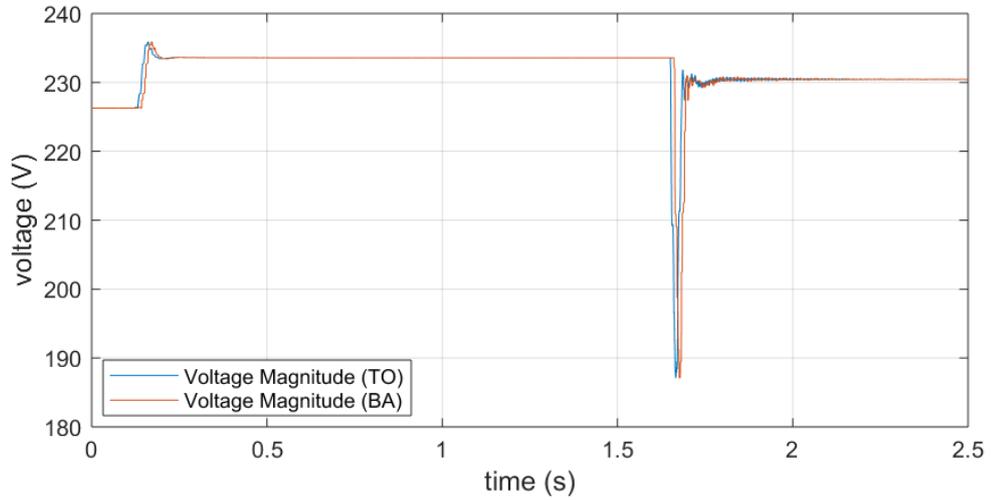


Figure 2.15: Low Voltage load's voltage amplitude. [8]

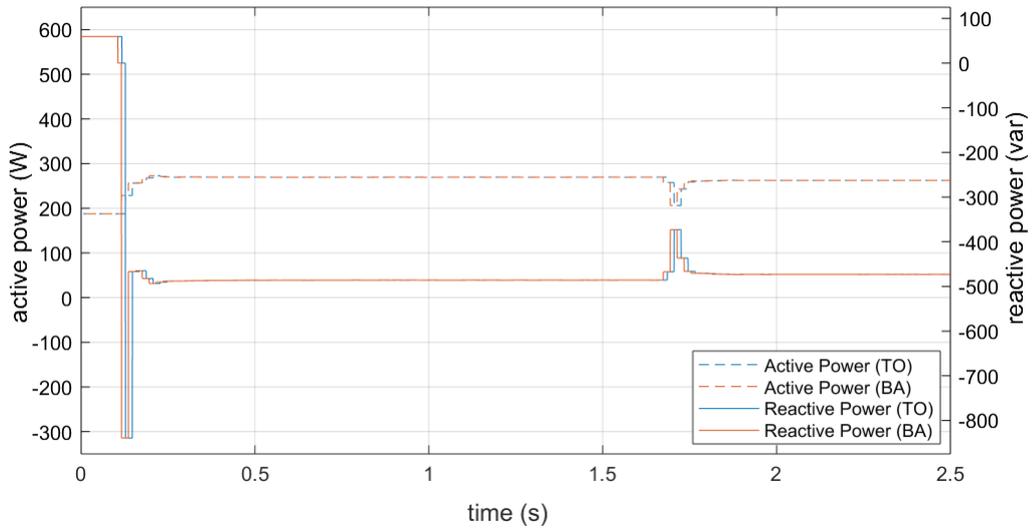


Figure 2.16: Low Voltage load's active and reactive power. [8]

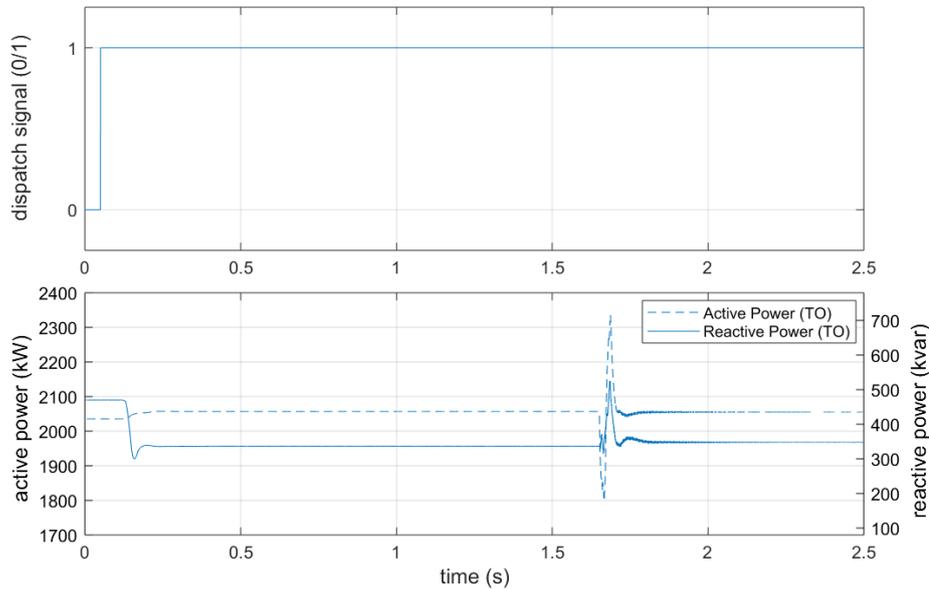


Figure 2.17: Active and Reactive power in the TSO-DSO connection node. [8]

From Figure 2.15 it can be appreciated the step variation in the instant in which the command trigger is given to regulate the flexible load. Such amplitude variation requires to be balanced: the tap of the OLTC transformer is adjusted and thus the voltage amplitude goes back to a value closer to 1 pu. However, the steady-state condition after the tap changing is preceded by a transient dynamics with a spike that does not describe the physical reality in a satisfying way. Such aspect will be reduced by improving the OLTC transformer model.

In Figure 2.16 it can be highlighted the first variation of  $P$  and  $Q$  in the instant in which the command trigger is given to the flexible source. Such variation presents a brief transient that does not impact in the same way of the voltage spike previously mentioned. A second noisy transient in active and reactive power occurs when the voltage spike happens, even though it's not disruptive.

Finally, in Figure 2.17 it has to be underlined how the voltage spike impacts heavily on the active and reactive power measured in the TSO-DSO connection. Furthermore, it has to be noted how the required variation of  $Q$  (around 150 kvar) is significantly higher than the  $P$  variation (around 22 kW). This is justified by the purpose of the regulation here described, which

is regulating the reactive power.

## 2.4 Stability of the R-PHIL co-simulation

In this section the same model of the previous co-simulation is implemented, although no OLTC is included in the HV/MV transformer. The aim of these tests is to evaluate the stability of the remote co-simulated model and to find which are the conditions to make the model diverging. Specifically, three aspects have been taken into account. Firstly, the addition of a *added delay* has been considered as a potential source of instability.

Secondly, the  $(P, Q)$  feedback from Bari is amplified with a certain scale factor and then inserted in the simulated distribution feeder. The variation of this *scale factor* has also been considered as a key aspect to bring the model at an instable condition.

In order to test the model and bring it to an instable condition an RL load bank with nominal load of 192.9 W and 64.3 var is used in Bari. The RL load is controlled to periodically activate and deactivate an additional load of 450 W and 128.6 var (inductive) as already done in [7]. However, differently from [7], the tests have been conducted varying the aforementioned *scale factor* and including an *added delay*. The settings for the co-simulation here discussed have been imposed with the awareness that a stable condition was achieved with the previous setting (scale factor of x250 and no added delay). For briefness' s sake only the results of Turin will be shown and discussed later in this section. Also, only the results linked to the most significant settings are later illustrated.

A further aspect has to be highlighted: increasing the scale factor implies simulating a LV load with a higher and higher power absorption. However, the limit in power imposed by the secondary substation needs to be remembered, as the nominal power is equal to 250 kVA. Considering that the RL load absorbs around 500 W, a scale factor larger than 500 would simulate an abnormal condition (eg. failure condition). Please note that simulating the whole model with a *scale factor* higher than 500 without obtaining any instability phenomenon means that the remote co-simulation can easily emulate an abnormal condition. A last technical aspect has so be brought to the reader's attention before descussing the results: the Simulink model of Turin has a different time step  $T_s$  of the Bari's model (150  $\mu$ s against 125  $\mu$ s). In order to add the same added value both in Turin and in Bari, it will be necessary to impose two different values of delay's lenght.

The first set of results includes a simulation in which the *scale factor* is equal to 1500, the *added delay* in Turin starts at  $0 T_s$  and it's increased to  $42 T_s$ . The results are shown in Figure 2.18 and Figure 2.19.

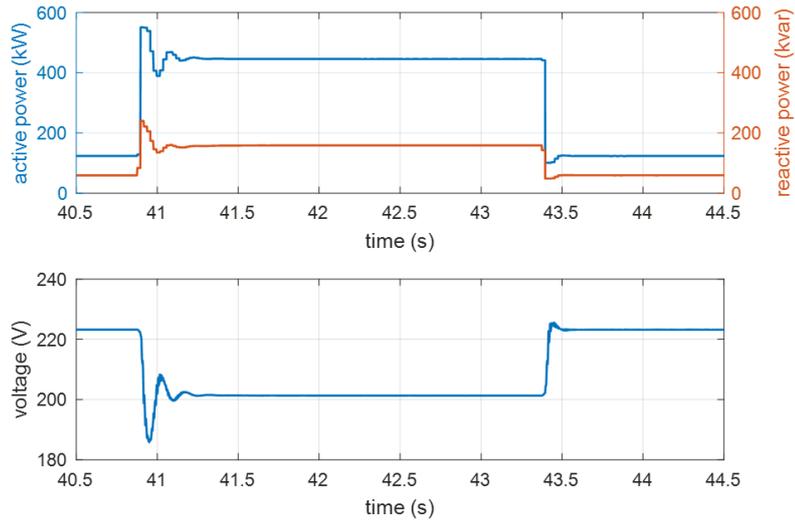


Figure 2.18: active power, reactive power and voltage (phase-ground rms) in Turin. Scale factor = 1500, added delay =  $0 T_s$ .

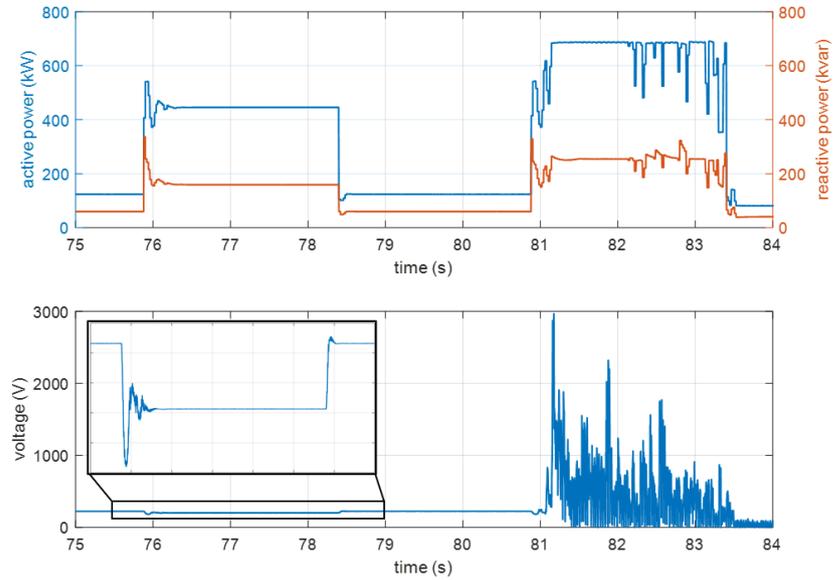


Figure 2.19: active power, reactive power and voltage (phase-ground rms) in Turin. Scale factor = 1500, added delay =  $42 T_s$ .

From the zoomed view of Figure 2.19 it can be seen that an instable condition was almost reached of the previous transient. If we compare such dynamic with the voltage transient in Figure 2.18, we see that the curve is less smooth. This first results are obtained with a quite high *scale factor*, meaning that the emulated load far exceeds the MV-LV transformer's thermal limits. Furthermore, reaching the instability with a low added delay may bring us at the conclusion that the most critical parameter may be the *scale factor*. Other simulations have been carried out to verify that this is true.

The second set of results includes a simulation in which the *scale factor* is equal to 1400 and the *added delay* in Turin varies from  $0 T_s$  up to  $672 T_s$ . The results are shown in Figure 2.20 and Figure 2.21.

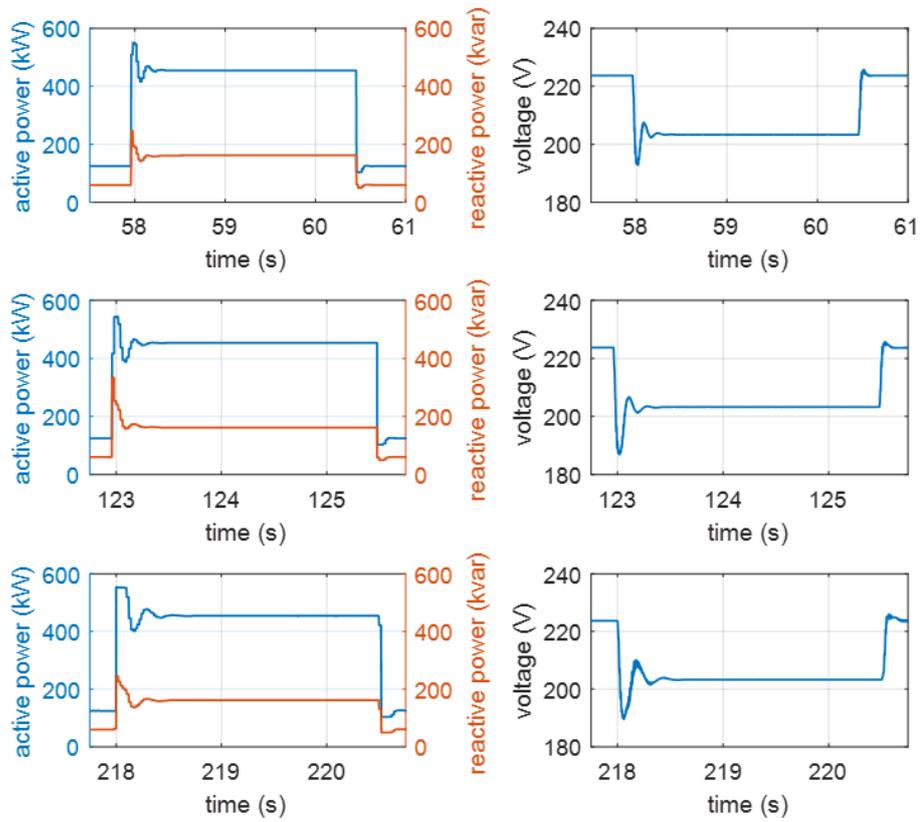


Figure 2.20: active power, reactive power and voltage (phase-ground rms) in Turin. First row: scale factor = 1400, added delay = 0  $T_s$ . Second row: scale factor = 1400, added delay = 84  $T_s$ . Third row: scale factor = 1400, added delay = 336  $T_s$ .

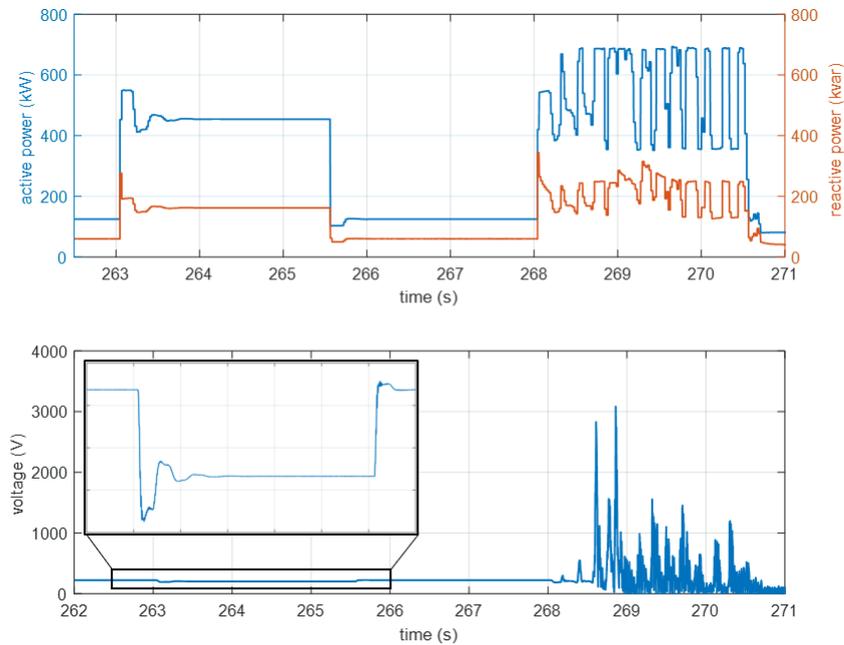


Figure 2.21: active power, reactive power and voltage (phase-ground rms) in Turin: scale factor = 1400, added delay =  $672 T_s$ .

A reduction of the *scale factor* compared with the previous set of results, guarantees a stable condition with higher added delays. Please note how the increase of the added delay leads to separate the transients of the closing/opening variations in the RL load. Although the instability is reached, it is not clear from this results whether the *added delay* or the *scale factor* plays the most critical role in obtaining the divergency. The next set of results may clear such doubt.

The third set of results includes a simulation in which the *scale factor* is equal to 100 and the *added delay* in Turin varies from  $0 T_s$  up to  $833 T_s$ . The results are shown in Figure 2.22. Please note that the initial condition of this simulation corresponds to the one already illustrated and debated in [7].

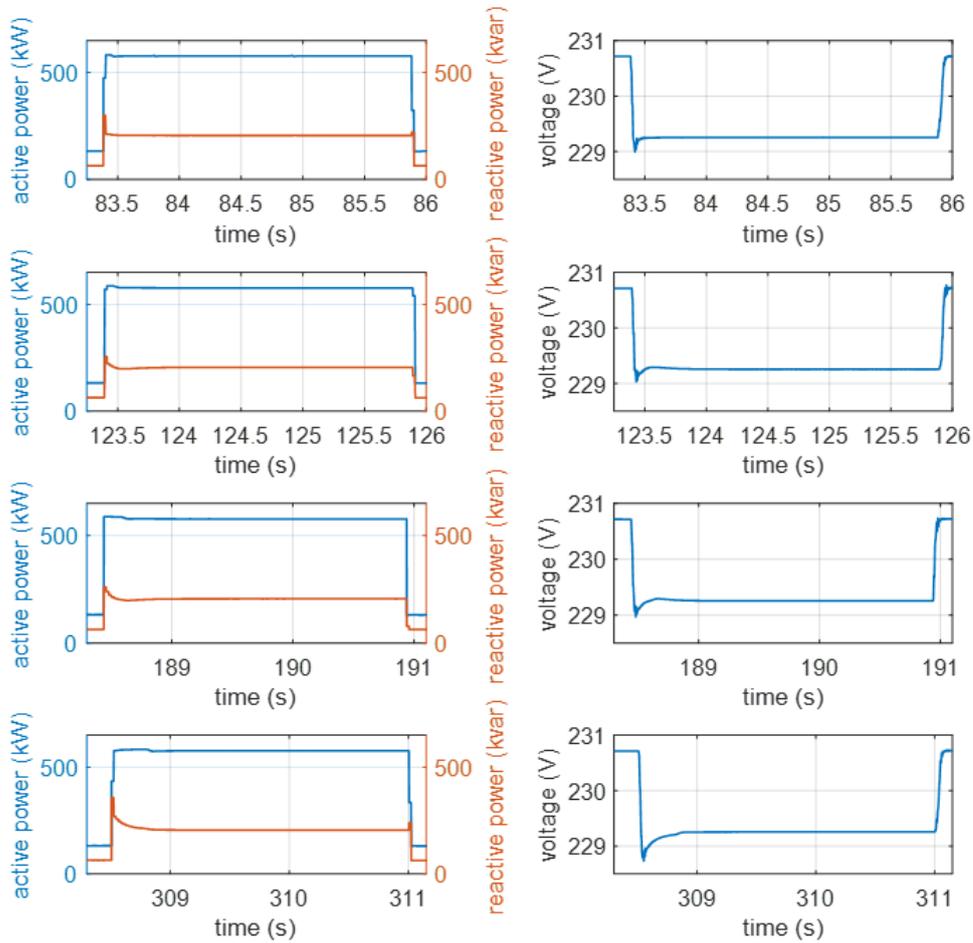


Figure 2.22: active power, reactive power and voltage (phase-ground rms) in Turin. First row: scale factor = 100, added delay =  $0 T_s$ . Second row: scale factor = 100, added delay =  $84 T_s$ . Third row: scale factor = 100, added delay =  $336 T_s$ . Forth row: scale factor = 100, added delay =  $833 T_s$ .

From these results we can immediately see that if the *scale factor* is small enough, the increase of the *added delay* will not bring the model to an unstable condition.

A fourth and last simulation is carried out. Now the simulation is set with a *scale factor* equal to 1600 and no added delay. The results are illustrated in Figure 2.23.

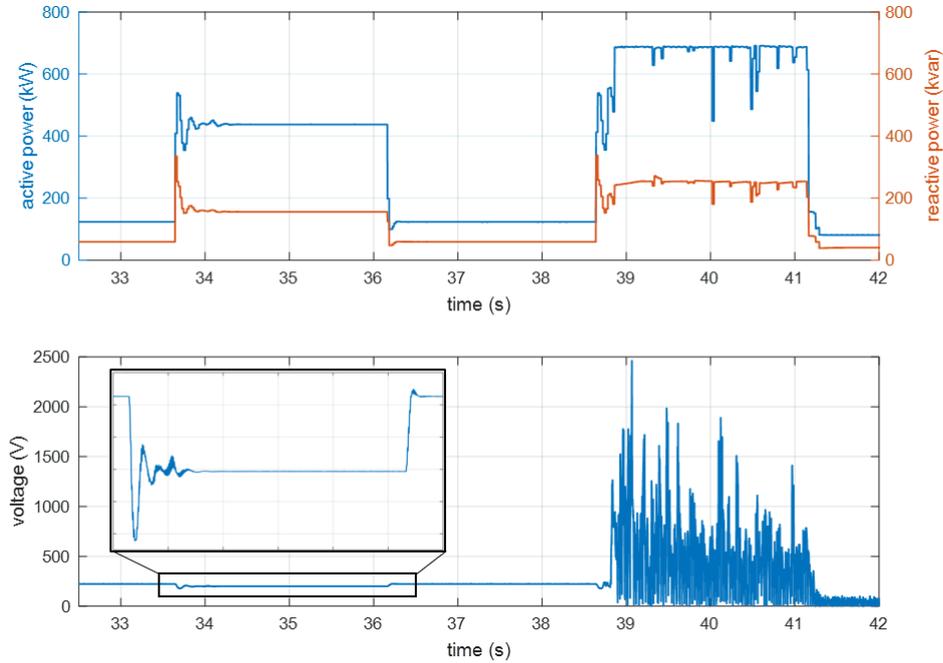


Figure 2.23: active power, reactive power and voltage (phase-ground rms) in Turin: scale factor = 1600, added delay =  $0 T_s$ .

As we have foresaw, an extremely high *scale factor* brings the model to instability, either with an added delay or without.

Having observed the different sets of simulations, we can draw the following conclusions:

- the *scale factor* results to be the most critical parameter in bringing the remote co-simulation to an instable condition. From a physical point of view, increasing this quantity means bringing a larger portion of grid out from the model simulated in Turin, thus increasing the ratio between the load's impedance and the main grid's impedance;
- the *added delay* does not have a heavy impact in the stability of the simulated model. Increasing its value means separating the transients of closing/opening process of the RL load. The most critical condition for the model can be found when the *scale factor* is high enough and the transients are closed to each other. However, increasing such delay may bring to the divergency only if the *scale factor* is high enough;
- the co-simulated model appears to be strongly stable, as we have to increase the aforementioned parameters enough to obtain the diver-

gency. This aspect can be justified by the physical quantities that are exchanged from Turin and Bari. As already mentioned, Turin receives from Bari the  $(P, Q)$  signals. They already operate as a filter, as they are computed as an average value of an integration. If we had exchanged the  $(V, I)$  signals we would not have obtained such a level of stability.

# Chapter 3

## Decoupling of a larger network

In the previous chapter the realtime analysis has been carried out for a single distribution network feeder. The computational burden was limited, and this allows to simulate the model on a single core of the OPAL-RT<sup>®</sup> simulator. The following part of the work aims to study a broader electrical network, whose technical specifications will be discussed later in this chapter, with a higher number of distribution feeders.

The increasing in the network's extension brings with itself an increase in the computational burden. A realtime simulation will then require the network to be divided in groups and decouple them in more than one core. A parallel computation will be executed.

In the following chapter the State-Space Nodal (SSN) method, which is implemented in RT-LAB<sup>®</sup> to execute a parallel computation, the mathematical theory on which the SSN method is based, and the methods from which the SSN method is deduced.

Moreover, the application of this concepts to our specific electric grid is discussed. The ZIP model used for the MV loads is presented, with the different coefficient for each type of MV load.

Finally the validation of the model is carried on by confronting some results with what is obtained with the RTDS<sup>®</sup> realtime simulator, using some of the results obtained by Dr. Andrea Aleo in [1]. Specifically, the steady state behavior is compared.

### 3.1 State-Space method and Nodal Analysis

From [3], given a physical model in a discrete time domain, the transfer function  $G(z)$  provides uniquely a correlation between the inputs  $u(k)$  and the outputs  $y(k)$ , where  $z$  is the independent variable in the complex domain while  $k$  is the discrete time independent variable. The function  $G(z)$  is defined as follows:

$$G(z) = \frac{y(k)}{x(k)} \quad (3.1)$$

However, to study what occurs within the physical model, which means understanding the states of the model, the *State-Space Model (SSM)* is applied. The *SSM* describes the physical model and its dynamics by a set of two equations [3]:

$$\begin{aligned} \mathbf{x}(\mathbf{k}+1) &= \mathbf{A}(\mathbf{k}) \cdot \mathbf{x}(\mathbf{k}) + \mathbf{B}(\mathbf{k}) \cdot \mathbf{u}(\mathbf{k}) \\ \mathbf{y}(\mathbf{k}) &= \mathbf{C}(\mathbf{k}) \cdot \mathbf{x}(\mathbf{k}) + \mathbf{D}(\mathbf{k}) \cdot \mathbf{u}(\mathbf{k}) \end{aligned} \quad (3.2)$$

where:

- $\mathbf{u}(\mathbf{k})$  is the vector of the inputs of the system;
- $\mathbf{x}(\mathbf{k})$  is the vector of the states of the system;
- $\mathbf{y}(\mathbf{k})$  is the vector of the outputs of the system;

The set of equations 3.2 is composed of matricial and vectorial elements. Specifically, the system has  $n$  states,  $l$  inputs and  $m$  outputs.

The first equation of 3.2 is called *States equation* which describes how the inputs impact on the states and the dynamics of the system. The second equation of 3.2 describes the correlation between the inputs/states and the outputs of the system.

Given the set of equations of 3.2, the evolution of the states can be expressed both when the system doesn't have inputs (*free evolution*) and when the system does have inputs (*forced evolution*) [3].

In case of a *free evolution* the states are described with the following equation:

$$\mathbf{x}(\mathbf{k}) = \Phi(\mathbf{k}, \mathbf{0}) \cdot \mathbf{x}(\mathbf{0}) \quad (3.3)$$

Instead, in case of a *forced evolution* the following equation is used to define the evolution of the states:

$$\mathbf{x}(\mathbf{k}) = \Phi(\mathbf{k}, \mathbf{0}) \cdot \mathbf{x}(\mathbf{0}) + \sum_{l=0}^{k-1} \Phi(\mathbf{k}, \mathbf{l} - \mathbf{1}) \cdot \mathbf{B}(\mathbf{l}) \cdot \mathbf{u}(\mathbf{l}) \quad (3.4)$$

where  $\Phi(\mathbf{k}, \mathbf{0})$  is the *Transition matrix* between the state  $k$  and the state  $0$ , while  $\Phi(\mathbf{k}, \mathbf{l} - \mathbf{1})$  is the *Transition matrix* between the state  $k$  and the state  $l-1$ .

Given an electrical network with  $n$  nodes and  $l$  branches, that can be represented by a graph as in Figure 3.1, we can define an *Incidence matrix*  $A$  to describe the beforementioned network, The matrix  $A$  will be of dimensions  $T \times M$ , where  $T = n-1$  and  $M = l-n+1$  [11].

Considering the generic row  $h$  and the generic column  $i$ , the element  $A(h, i)$  is defined as follows:

- $\mathbf{A}(\mathbf{h}, \mathbf{i}) = +1$ , if the *branch*  $i$  starts from *node*  $h$  and the conventional sign of the *branch*  $i$  is  $> 0$ ;
- $\mathbf{A}(\mathbf{h}, \mathbf{i}) = -1$ , if the *branch*  $i$  starts from *node*  $h$  and the conventional sign of the *branch*  $i$  is  $< 0$ ;
- $\mathbf{A}(\mathbf{h}, \mathbf{i}) = 0$ , if the *branch*  $i$  doesn't touch *node*  $h$ ;

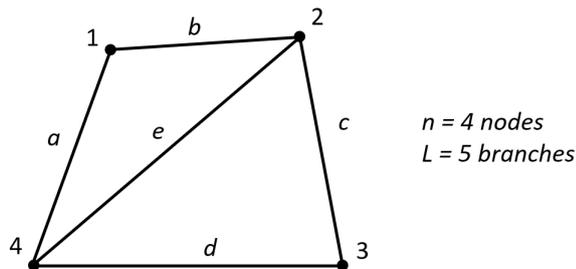


Figure 3.1: Example of an electrical network represented by a graph.

To study the electrical network in a sinusoidal steady-state, we define the following elements:

- $\mathbf{V}$  contains the phasors of the electrical potential difference between the extremes of each branch;
- $\mathbf{I}$  contains the phasors of the currents of each branch;

- $\mathbf{V}$  contains the phasors of the voltage of each node referred to the node of reference;

These elements are essential for the *Nodal Analysis*. If the considered network, merely composed by one-port electrical element, is in steady-state, the following equations are valid [11]:

$$\mathbf{A} \cdot \mathbf{I} = \mathbf{0} \quad (3.5)$$

$$\mathbf{A}^t \cdot \tilde{\mathbf{V}} = \mathbf{V} \quad (3.6)$$

Each one-port electrical element is studied considered with the Norton equivalent. For the  $k$ -th one-port element we define an admittance  $Y_k$ , a short-circuit current source  $I_{k0}$ , a voltage over the admittance  $V_k$  and the total current  $I_k$ . Figure 3.2 shows the Norton equivalent here described.

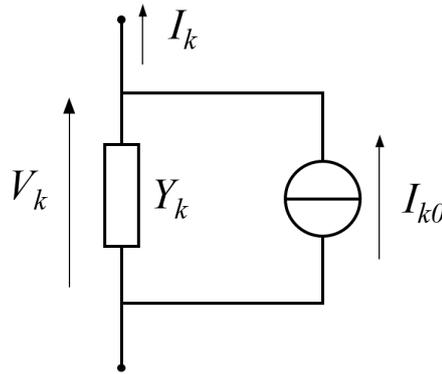


Figure 3.2: Norton equivalent of a generic one-port electrical element. Source: [11].

For each one-port electrical element the following equation can be written:

$$I_k = I_{k0} - Y_k \cdot V_k \quad (3.7)$$

where  $k = 1, 2, \dots, l$ .

The equation 3.7 can be written in a matricial form, thus describing the entire network rather than a single one-port element. The following equation is obtained:

$$\mathbf{I} = \mathbf{I}_0 - \mathbf{Y} \cdot \mathbf{V} \quad (3.8)$$

By inserting the equations 3.5 and 3.6 in the equation 3.8 the following is obtained the *Nodal Equation*:

$$\mathbf{A}\mathbf{Y}\mathbf{A}^t \cdot \tilde{\mathbf{V}} = \mathbf{A}\mathbf{I}_0 \quad (3.9)$$

where the unknown parameters to be determined are the electrical potential at each node of the network [11].

### 3.2 State-Space Nodal method

In order to solve large-sized electrical networks the *State-Space Method (SSM)* and the *Nodal Analysis (NA)* are combined so to create the *State-Space Nodal (SSN)* method [13].

The *SSN* is based on dividing the electrical network in groups. Each of these group will be described with the *SSM* and interfaced with the other groups by using the *NA*. The interface of the groups enable us to solve the groups independently from the others. Such advantage makes the resolution more efficient, specifically for networks that present switches.

This approach eases the resolution of networks with switches, the analysis of transients and of non-linear functions. Moreover, combining the two methods enables to reduce di disadvantages of both of them, exploiting the advantages.

The *SSN* divides the overall electrical network in a certain number of groups. For each of those groups a single *nodal admittance matrix* is defined. The equations within each groups are the *state-space equations* in the discrete domain. As already mentioned, the groups are interfaced with eachother. The interface is made by a interconnection node, whose electrical potential is unknown. This variable will be calculated by using the *NA*.

The set of equations for each group is the following. The bold characters describes a matrice or a vector [13].

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_k \mathbf{x} + \mathbf{B}_k \mathbf{u} \\ \mathbf{y} &= \mathbf{C}_k \mathbf{x} + \mathbf{D}_k \mathbf{u} \end{aligned} \quad (3.10)$$

By discretizing the 3.10 the following equation is obtained [13].

$$\mathbf{x}_{t+\Delta t} = \hat{\mathbf{A}}_k \mathbf{x}_t + \hat{\mathbf{B}}_k \mathbf{u}_t + \hat{\mathbf{B}}_k \mathbf{x}_{t+\Delta t} \quad (3.11)$$

where the  $\Delta t$  is the integration time step. The equations 3.10 and 3.11 are reworked, obtaining the following equations [13]:

$$\hat{\mathbf{x}}_{t+\Delta t} = \hat{\mathbf{A}}_k \mathbf{x}_t + \hat{\mathbf{B}}_k \mathbf{u}_t + [\hat{\mathbf{B}}_{k_i}, \dots, \hat{\mathbf{B}}_{k_n}] \cdot \begin{bmatrix} \mathbf{u}_{i_{t+\Delta t}} \\ \vdots \\ \mathbf{u}_{n_{t+\Delta t}} \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} \mathbf{y}_{i_{t+\Delta t}} \\ \vdots \\ \mathbf{y}_{n_{t+\Delta t}} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{k_i} \\ \vdots \\ \mathbf{C}_{k_n} \end{bmatrix} \mathbf{x}_{t+\Delta t} + \begin{bmatrix} \mathbf{D}_{k_{ii}} & \cdots & \mathbf{D}_{k_{in}} \\ & \vdots & \\ \mathbf{D}_{k_{ni}} & \cdots & \mathbf{D}_{k_{nn}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_{i_{t+\Delta t}} \\ \vdots \\ \mathbf{u}_{n_{t+\Delta t}} \end{bmatrix} \quad (3.13)$$

In the 3.12 and 3.13 the subscripts must be clearly distinguished: the subscript  $i$  defines source within a group, while the subscript  $n$  define a nodal injection in the group.

The equations 3.12 and 3.13 are put together to get the following equation [13]:

$$\mathbf{y}_{n_{t+\Delta t}} = \mathbf{C}_{k_n} (\hat{\mathbf{A}}_k \mathbf{x}_t + \hat{\mathbf{B}}_k \mathbf{u}_t + \hat{\mathbf{B}}_{k_i} \mathbf{u}_{i_{t+\Delta t}}) + \mathbf{D}_{k_{ni}} \cdot \mathbf{u}_{i_{t+\Delta t}} + (\mathbf{C}_{k_n} \hat{\mathbf{B}}_{k_n} + \mathbf{D}_{k_{nn}}) \mathbf{u}_{n_{t+\Delta t}} \quad (3.14)$$

The 3.14 is reworked so to have a lighter syntax [13]:

$$\mathbf{y}_{n_{t+\Delta t}} = \mathbf{y}_{k_{hist}} + \mathbf{W}_{k_n} \mathbf{u}_{n_{t+\Delta t}} \quad (3.15)$$

where the variable  $\mathbf{W}_{k_n}$  is defined as follows [13]:

$$\mathbf{W}_{k_n} = \mathbf{W}_{k_n} \hat{\mathbf{B}}_{k_n} + \mathbf{D}_{k_{nn}} \quad (3.16)$$

The 3.15 can be interpreted in two ways here listed [13]

- if  $\mathbf{y}_n$  contains the current injection in the group and  $\mathbf{u}_n$  contains the nodal electrical potentials, then  $\mathbf{y}_{k_{hist}}$  is the history of the current sources and  $\mathbf{W}_{k_n}$  is the admittance matrix. A group with those conditions is a *V-type SSN group* that will be described with a Norton equivalent;
- if  $\mathbf{y}_n$  contains the nodal voltage and  $\mathbf{u}_n$  contains the current injections, then  $\mathbf{y}_{k_{hist}}$  is the history of the voltage sources and  $\mathbf{W}_{k_n}$  is the impedance matrix. A group with those conditions is a *I-Type SSN group* that will be described with a Thevenin equivalent;

In the reality both types are presented in the network. The equation 3.15 is rewritten as follows [13]:

$$\begin{bmatrix} \mathbf{v}_{n_t+\Delta t}^I \\ \vdots \\ \mathbf{i}_{n_t+\Delta t}^V \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{k_{hist}} \\ \vdots \\ \mathbf{i}_{k_{hist}} \end{bmatrix} + \begin{bmatrix} \mathbf{W}_{II} \cdots \mathbf{W}_{IV} \\ \vdots \\ \mathbf{W}_{VI} \cdots \mathbf{W}_{VV} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{i}_{n_t+\Delta t}^I \\ \vdots \\ \mathbf{v}_{n_t+\Delta t}^V \end{bmatrix} \quad (3.17)$$

The 3.16 is reworked as follows [13]:

$$\begin{bmatrix} \mathbf{i}_{n_t+\Delta t}^I \\ \vdots \\ \mathbf{i}_{n_t+\Delta t}^V \end{bmatrix} = \mathbf{\Gamma}_{k_n} \begin{bmatrix} \mathbf{v}_{k_{hist}} \\ \vdots \\ \mathbf{i}_{k_{hist}} \end{bmatrix} + \mathbf{Y}_{k_n} \begin{bmatrix} \mathbf{v}_{n_t+\Delta t}^I \\ \vdots \\ \mathbf{v}_{n_t+\Delta t}^V \end{bmatrix} \quad (3.18)$$

In the equation 3.18 the variable  $\mathbf{Y}_{k_n}$  refers to a single group. In order to describe the overall network the following equation is introduced: [13]

$$\mathbf{i}_{N_t+\Delta t} = \mathbf{Y}_N \cdot \mathbf{v}_{N_t+\Delta t} \quad (3.19)$$

where  $\mathbf{i}_N$  contains the known nodal injections,  $\mathbf{v}_N$  contains the unknown nodal voltages and  $\mathbf{Y}_N$  is the global admittance matrix.

To obtain the unknown variables of the network, which are the nodal voltages, the 3.19 is solved. The resolution can be carried out using the LU factorization and the sparse matrix-based solver in case of large electrical systems [13].

### 3.3 Application of parallelization for a small distribution grid

The analysed electrical network, shown in Figure 3.3, is composed by an electrical primary substation with three HV-MV transformers. There are 5 distribution feeders departing from the three transformers. Specifically, the three transformers reduce the voltage from 220kV down to 22kV. Transformer 1 and Transformer 3 have a nominal power of  $S_N = 63$  MVA, while Transformer 2 presents a nominal power of  $S_N = 55$  MVA. From the first transformer the feeders MAGRA and BRENTA depart. From the second transformer the feeders FIAT and GROSSO depart. Finally, from the third transformer the feeder CHERI departs.

Because of the extension of the network taken into consideration, it becomes necessary to decouple the network in two or more subsystems in order to

implement correctly a realtime simulation. The defined subsystems will be solved in parallel on different cores of the OPAL-RT<sup>®</sup> simulator.

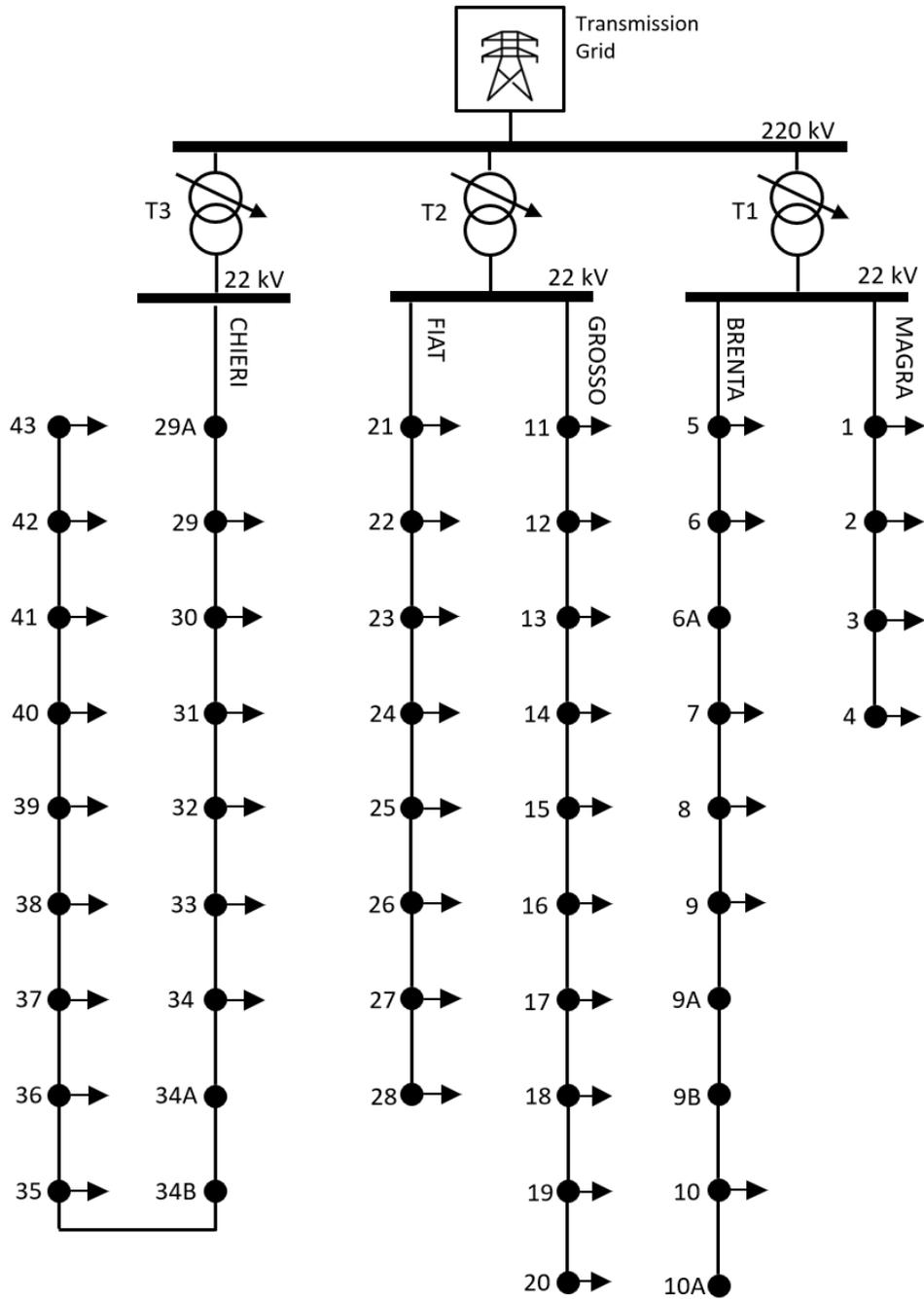


Figure 3.3: Qualitative representation of the analysed electrical grid.

Firstly, the electrical network is sectioned in four subsystems (one Master and three Slaves) by using the *Stubline* block of the Artemis library [17]. The block and its parameters are illustrated in Figure 3.4.

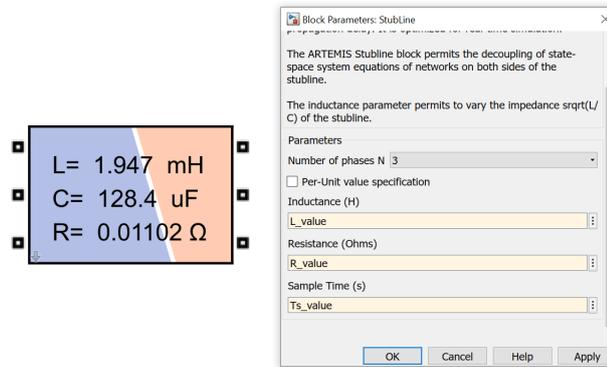


Figure 3.4: Stubline block and its interface window.

The Stubline block emulates the Bergeron line model with an adaptation so that the introduced propagation delay is equal to one time-step. The Stubline block is needed when the model does not include very long lines, which are typical of the transmission network [17].

In order to correctly use the block, the Stubline is applied to replace the secondary winding of a transformer: the resistance and the impedance of the secondary winding are set to zero, imposing those values in the Stubline block. A second possible application for the Stubline block is when we use such block to emulate very short lines.

Whether we are in the first or in the second case aforementioned, it will be required to compensate the capacitive contribution of the Stubline block. To solve this issue, a three-phase shunt inductance is placed in one of the two subsystems connected by the Stubline. A possible solution is shown in Figure 3.5.

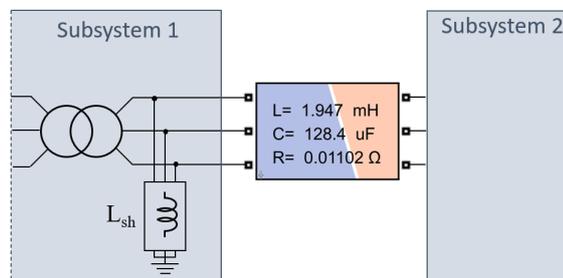


Figure 3.5: Schematic representation of the reactive compensation of the Stubline block.

The value of the shunt inductance is chosen applying the following equation:

$$Q_{L_{sh}} = 3 \cdot \frac{V^2}{X_{L_{sh}}} = Q_{C_{stub}} = 3 \cdot \frac{V^2}{X_{C_{stub}}} \quad (3.20)$$

From which we can easily deduce the following relation:

$$L_{sh} = \frac{1}{\omega^2 \cdot C_{stub}} \quad (3.21)$$

A further measure to improve the realtime simulation is the State-Space Nodal (SSN) method [17], already described in this chapter. To apply the SSN method on RT-LAB<sup>©</sup> the *SSN block* needs to be inserted. The block is shown in Figure 3.6(left). Moreover, we have to define the interconnection node between two groups. For this matter, the *SSN interface block*, from the Artemis library, is used. Such block is shown in Figure 3.6(right). Such block enables to choose the tipology of the group (V-type or I-type) that are interfaced.

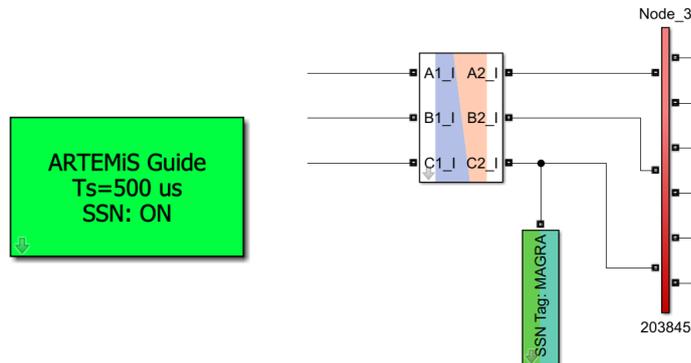


Figure 3.6: SSN block (left) and SSN Interface block (right).

The implementation of the *SSN method* in the RT-LAB<sup>©</sup> simulation environment does not add any delay.

The electrical grid is represented in Figure 3.7. The scheme highlights the division of the overall grid in subsystems (one Master and three Slaves) and in groups (green lines).

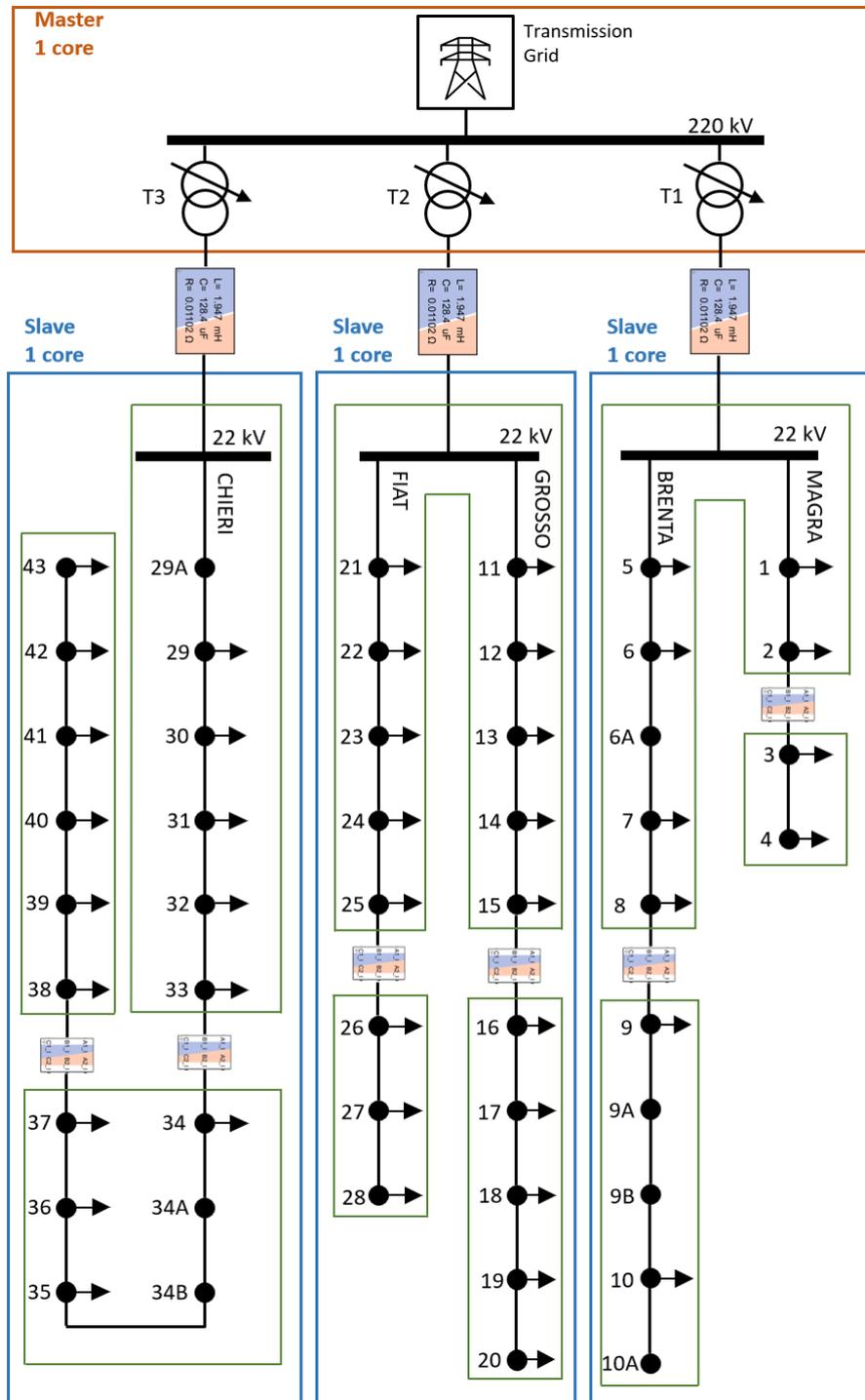


Figure 3.7: Qualitative representation of the analysed electrical grid with the decoupling in subsystems and groups.

In the analysis of the aforerepresented network the ZIP model has been used to implement the different types of load. The equations describing the ZIP load are the following:

$$P(V) = P_n \left[ P_p + I_p \left( \frac{V}{V_n} \right) + Z_p \left( \frac{V}{V_n} \right)^2 \right] \quad (3.22)$$

$$Q(V) = Q_n \left[ P_q + I_q \left( \frac{V}{V_n} \right) + Z_q \left( \frac{V}{V_n} \right)^2 \right] \quad (3.23)$$

in which  $P(V)$  and  $Q(V)$  are the value at a defined voltage  $V$ ,  $P_n$  and  $Q_n$  are the value at nominal voltage,  $P_p$  and  $P_q$  are the coefficients at constant power,  $I_p$  and  $I_q$  are the coefficients at constant current,  $Z_p$  and  $Z_q$  are the coefficients at constant impedance.

For each distribution bus the type of load has been chosen among the residential, commercial and industrial type. The assignation to each bus is listed in Table 3.1.

Table 3.1: Type of load for each distribution bus.

Type of load	Bus
Residential	5, 8, 10, 17, 19, 20, 26, 34, 35, 36, 37, 40 41, 42, 43
Commercial	4, 6, 7, 9, 13, 15, 16, 18, 23, 24, 27, 28, 31 32, 33, 38, 39,
Industrial	1, 2, 3, 11, 12, 14, 21, 22, 25, 29, 30

For the ZIP model, the coefficients for each type of load have been chosen according to the results of [6] for residential, commercial and industrial loads.

Table 3.2: ZIP coefficient for each type of load. Source: [6].

Type of load	$P_p$	$I_p$	$Z_p$	$P_q$	$I_q$	$Z_q$
Residential	1.21	-1.61	1.41	4.35	-7.08	3.72
Commercial	0.76	-0.52	0.76	6.92	-11.75	5.83
Industrial	1.5	-2.31	1.81	7.41	-11.97	5.55

The nominal active and reactive power ( $P_n$  and  $Q_n$ ) of each load are reported in the Appendix E, where the initialization Matlab code is reported.

### 3.4 Comparison with the RTDS simulator of the steady state results

Having set the model of the distribution network, including the MV loads, a comparison of the results at steady-state is done between the OPAL-RT<sup>©</sup> realtime simulator (used for the here discussed Thesis) and the RTDS<sup>©</sup> realtime simulator (used for [1]). The amplitude of the current is measured in a series of spots in the network. The comparison of the two simulators' results is reported in Table 3.3.

Table 3.3: Comparison between RTDS<sup>©</sup> and OPAL-RT<sup>©</sup> simulators: current amplitude. Partial source: [1].

Measure position	RTDS <sup>©</sup> current amplitude (A) [1]	OPAL-RT <sup>©</sup> current amplitude (A)	difference (%)
HV network	70	71	1.43
Transformer 1 (MV side)	154	158	2.53
Magra feeder	46	47	2.13
Brenta feeder	108	111	2.70
Transformer 2 (MV side)	368	369	0.27
Grosso feeder	198	198	0.00
Fiat feeder	172	172	0.00
Transformer 3 (MV side)	172	173	0.58

The negligible errors listed in Table 3.3 in the various measures show the validity of the proposed Simulink model.

# Chapter 4

## Characterization of the LV distributed sources

The chapter aims to describe a test implemented solely on the Turin-based OPAL-RT<sup>®</sup> realtime simulator. Firstly, the problem will be discussed from a theoretical viewpoint. The second part of the chapter will be dedicated to present the results of the simulations.

The five-feeder distribution network (see Chapter 3 for more details on the network) is simulated in realtime. However, the network presented in Chapter 3 is slightly modified in the FIAT feeder: a three-phase load is used to emulate a local photovoltaic generation, while a second three-phase load is included to emulate a load with the exponential mathematical form.

The aim of this test is to create a parameter characterization of the active power measured at the secondary substation. Specifically, having a load modeled with the exponential form, the power generated by the photovoltaic system, the rated power of the exponential-form load and the exponential coefficient  $k_p$  are estimated. In order to do so, the winding ratio of the HV-MV transformer 2 (T2) is manually changed via the tap changer, making the feeder voltage varying. On the LV side of the MV-LV transformer the line-to-line voltage and the active power are measured. This operation can be easily implemented for each secondary substation and thus giving an estimation for the overall distribution network at the HV-MV interface of the RES.

The advantages of this parameterization are quite evident, for the DSO as well as the TSO: for any instant in which this operation is carried out, the operator will know which is the contribution from the distributed energy

resource with a unique measure. In fact, below the secondary substation the power flows are not measured singularly for the aggregated load and for the local generation, but with a single measurement.

## 4.1 Theoretical basis of the non-linear regression

Let us consider the case represented in Figure 4.1, for each measure  $i = 1, \dots, M$  the measured active power can be analytically defined by the following equation:

$$P_i(v_i) = P_{nom} \cdot v_i^{k_p} - P_{gen} \quad (4.1)$$

where the variable  $v_i$ , measured in the  $i$ -th measure, is the ratio between the measured line-to-line rms voltage and the nominal voltage (400 V).

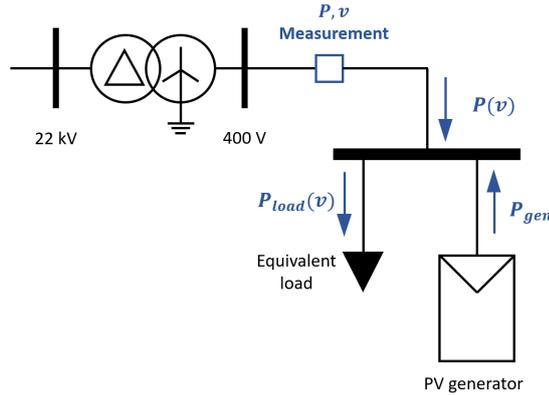


Figure 4.1: Schematic representation of the LV node with equivalent exponential-form load and PV generator.

The equation 4.1 can be generically defined as below:

$$y_i = f_i(v_i, \underline{x}) \quad (4.2)$$

where:

- $y_i$  and  $v_i$  represent the measured net active power and the measured voltage, respectively;
- $\underline{x} = [x_1, \dots, x_N]^T$  is the vector of the variables to be estimated (in our case we have  $N = 3$  and  $\underline{x} = [P_{nom} \ k_p \ P_{gen}]^T$ );

The element of  $\underline{x}$  cannot be obtained by a closed form, thus an iterative computation is required. The equation 4.2 is linearized around a certain point of its domain at the iteration  $k$ . Introducing the partial derivative concept we can obtain at the  $k$ -th iteration:

$$f_i(\underline{x}^{(k+1)}, v_i) \simeq f_i(\underline{x}^{(k)}, v_i) + \left. \frac{\partial f_i(\underline{x}, v_i)}{\partial \underline{x}} \right|_k \cdot \Delta \underline{x}^{(k)} \quad (4.3)$$

where  $f_i(\underline{x}^{(k+1)}, v_i)$  is the nodal active power computed by using the set of parameters  $\underline{x}^{(k+1)}$ .

In case of a multiple number of measures, we can generalize the mathematics using a vectorial equation. At the  $k$ -th iteration we get:

$$\underline{y}^{(k+1)} = \underline{y}^{(k)} + \mathbf{J}^{(k)} \cdot \Delta \underline{x}^{(k)}$$

where  $\underline{y}^{(k+1)}$  and  $\underline{y}^{(k)}$  are vectors with  $M$  elements (the number of measures), while  $\Delta \underline{x}^{(k)}$  is a vector with  $N$  elements. The matrix  $\mathbf{J}^{(k)}$ , noted as *Jacobian matrix* (with  $M \times N$  dimensions) has the following mathematical definition:

$$\mathbf{J}^{(k)} = \begin{bmatrix} \frac{\partial f_1(\underline{x}^{(k)}, v_1)}{x_1^{(k)}} & \dots & \frac{\partial f_1(\underline{x}^{(k)}, v_1)}{x_N^{(k)}} \\ \vdots & & \vdots \\ \frac{\partial f_M(\underline{x}^{(k)}, v_M)}{x_1^{(k)}} & \dots & \frac{\partial f_M(\underline{x}^{(k)}, v_M)}{x_N^{(k)}} \end{bmatrix} \quad (4.4)$$

At each iteration  $k$ -th of the non-linear regression we can define  $\Delta \underline{x}^{(k)}$  as below:

$$\Delta \underline{x}^{(k)} = (\mathbf{J}^T \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^T \cdot (\underline{y} - \underline{y}^{(k)}) \quad (4.5)$$

Thus, we can update the  $\underline{x}$  vector and procede with the iteration.

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \Delta \underline{x}^{(k)}$$

To more clearly understand the proposed algorithm, a flux diagram is illustrated in Figure 4.2. The proposed algorithm has two stop criterias: a first one is based on the maximum number of iteration  $iter_{max}$ , while the second is based on the tolerance  $toll$ , which is compared at each iteration with the  $\Delta \underline{x}^{(k)}$ . At the beginning of the algorithm the user is required to insert the initial condition of the parameters  $\underline{x}^{(0)}$ , beside the measures  $\underline{y} = [P_1 \dots P_M]^T$  and  $\underline{v} = [v_1 \dots v_M]^T$ .

Firstly, the tolerance  $toll$  and the maximum number of iterations  $iter_{max}$  are inserted, as well as the exponential load parameters and the vectors  $\underline{x}$ ,  $\underline{v}$  and  $\underline{y}$ . The iterative computation is abruptly stopped whether the iteration  $k$  reaches the  $iter_{max}$  value or the discrepancy between  $\underline{x}^{(k+1)}$  and  $\underline{x}^{(k)}$  is lower than the tolerance.

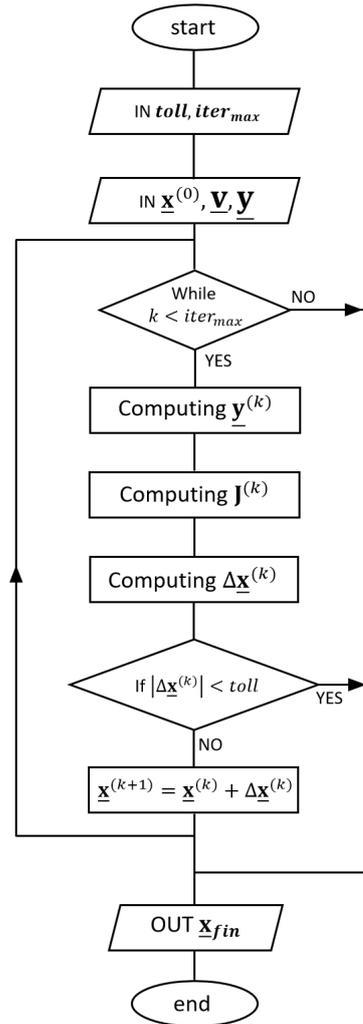


Figure 4.2: Flow chart representation of the non-linear regression.

In order to characterize the measured active power, a first approach included the estimation of all the three parameter ( $P_{nom}$ ,  $k_p$ ,  $P_{gen}$ ). To use such approach at least three measures are necessary.

The  $\underline{x}$  vector will have four elements, as indicated in equation 4.6.

$$\underline{x} = [x_1 \ x_2 \ x_3]^T = [P_{nom} \ k_p \ P_{gen}]^T \quad (4.6)$$

The non-linear function to be studied is:

$$f_i(\underline{x}, z) = x_1 \cdot v_i^{x_2} - x_3 \quad (4.7)$$

To define the matrix  $\mathbf{J}$ , the partial derivatives are computed:

$$\begin{cases} \frac{\partial f_i(\underline{x}, v_i)}{\partial x_1} = v_i^{x_2} \\ \frac{\partial f_i(\underline{x}, v_i)}{\partial x_2} = x_1 \ln(v_i) v_i^{x_2} \\ \frac{\partial f_i(\underline{x}, v_i)}{\partial x_3} = -1 \end{cases}$$

At the  $k$ -th iteration, the following is obtained:

$$\mathbf{J}^{(k)} = \begin{bmatrix} v_1^{x_2^{(k)}} & x_1^{(k)} \ln(v_1) v_1^{x_2^{(k)}} & -1 \\ \vdots & \vdots & \vdots \\ v_m^{x_2^{(k)}} & x_1^{(k)} \ln(v_m) v_m^{x_2^{(k)}} & -1 \\ \vdots & \vdots & \vdots \\ v_M^{x_2^{(k)}} & x_1^{(k)} \ln(v_M) v_M^{x_2^{(k)}} & -1 \end{bmatrix} \quad (4.8)$$

A second approach was then chosen, in which the  $k_p$  parameter is supposed to be known by the network operator.

The vector  $\underline{x}$  will then have merely two elements, as depicted below.

$$\underline{x} = [x_1 \ x_2]^T = [P_{nom} \ P_{gen}]^T$$

The non-linear function to be studied will be the following:

$$f_i(\underline{x}, v_i) = x_1 \cdot v_i^{k_p} - x_2 \quad (4.9)$$

The partial derivatives are then computed, from which the matrix  $\mathbf{J}$  can be easily defined.

$$\begin{cases} \frac{\partial f_i(\underline{x}, v_i)}{\partial x_1} = v_i^{k_p} \\ \frac{\partial f_i(\underline{x}, v_i)}{\partial x_2} = -1 \end{cases}$$

$$\mathbf{J}^{(k)} = \begin{bmatrix} v_1^{k_p} & -1 \\ \vdots & \\ v_m^{k_p} & -1 \\ \vdots & \\ v_M^{k_p} & -1 \end{bmatrix} \quad (4.10)$$

## 4.2 Implementing the regression with different cases

In this section the abovedescribed non-linear regression is implemented for a set of case studies. For each of these case studies the parameters are estimated, whether all three of them or only two out of three (parameter  $k_p$  is supposed to be known). The five case studies can be distinguished in two main groups: in three of the five cases a noise-free set of measures, with different accuracies, is analysed. On the other hand, the remaining two cases see a set of measures affected by noise.

For all of the following case study the LV load and the distributed resources are implemented with the parameters listed in the Table 4.1.

Table 4.1: Parameters of the LV load and distributed generation.

Parameter	Value
$P_{nom}$ (kW)	112.5
$k_p$ (-)	1.5
$P_{gen}$ (kW)	50

### 4.2.1 Case study A: three parameters estimation with a five-step trajectory and high accuracy set of measures

This first implementation sees the parameters initially set at the values listed in Table 4.2.

Table 4.2: Initial values of the three parameters.

Parameter	Value
$P_{nom}^{(0)}$ (kW)	90
$k_p^{(0)}$ (-)	1.3
$P_{gen}^{(0)}$ (kW)	35

As already mentioned, the test involves varying the tap changer of the HV-MV transformer above the GROSSO and FIAT feeders (Transformer T2). However, unlike the test carried out in Chapter 2, in which the tap changer variation is an output of the Automatic Voltage Regulator, the tap changer will be now changed manually.

For the first application, the tap changer is varied five times, obtaining below the secondary substation the profiles illustrated in Figure 4.3.

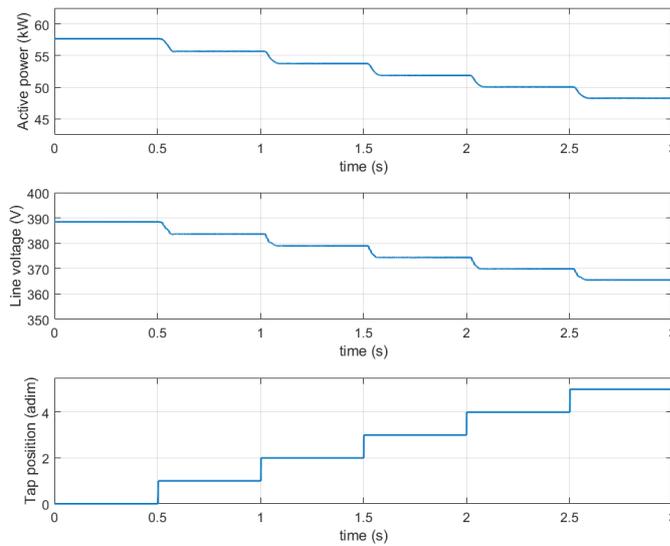


Figure 4.3: Active power measured below the secondary substation (upper); line-to-line voltage measured below the secondary substation (center); tap changer position of the HV-MV transformer (lower).

For this case study the set of measures listed in Table 4.3 is used. It is worth noting the decimal digits considered for the case study: three decimal digits for the active power and two decimal digits for the voltage.

Table 4.3: Set of measures used for the implementation of this case.

Measure	Active power (kW)	Voltage (V)
1	57.663	388.45
2	55.675	383.65
3	53.743	378.96
4	51.867	374.38
5	50.044	369.90
6	48.271	365.52

The final values of the three parameters, with the consequent percentage errors, are shown in Table 4.4. The profiles of both the parameters' values and the estimation errors are illustrated in Figure 4.4 and Figure 4.5, respectively. Although the initial values of the parameters are far from the real values, it can be observed a quick convergence to a set of values sufficiently close to the expected ones.

Table 4.4: Estimation of the three parameters with percentage errors

Parameter	Value	Error (%)
$P_{nom}$	113.95 kW	1.29
$k_p$	1.48	1.37
$P_{gen}$	51.45 kW	2.90

From the profiles shown in Figure 4.4, it can be appreciated the quick convergence of the method, obtaining an acceptable percentage error. The results are particularly important if we consider the simplicity of the non-linear regression.

A further aspect to underline is the number of iterations necessary to converge, making the proposed method not time-expensive.

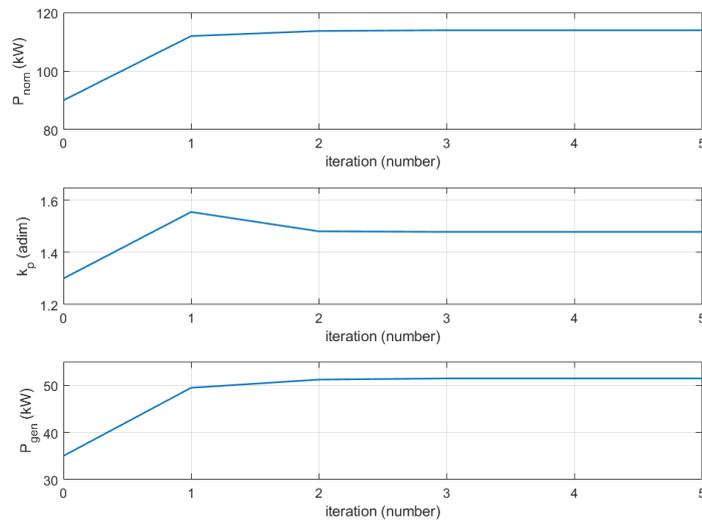


Figure 4.4: Estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with high accuracy measures of voltage and active power.

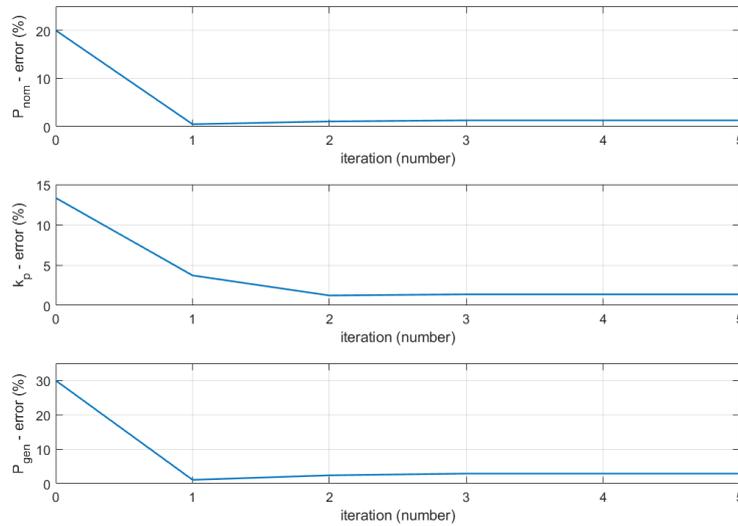


Figure 4.5: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with high accuracy measures of voltage and active power.

### 4.2.2 Case study B: three parameters estimation with a five-step trajectory and a set of measures with low accuracy

The abovedescribed estimation is now repeted considering measures with less accuracy (only one decimal digit for both voltage and active power measures). For a more proper comparison between the *Case Study A* and this case, the same initial values of the three parameters are used (see Table 4.2). The set of measures now used is listed in Table 4.5 below.

Table 4.5: Set of measures used for the implementation of this case

Measure	Active power (kW)	Voltage (V)
1	57.7	388.5
2	55.7	383.7
3	53.7	379.0
4	51.9	374.4
5	50.0	369.9
6	48.3	365.5

The results for the parameters with the percentage errors are listed in Table 4.6, while the profiles are shown in Figure 4.6 and Figure 4.7.

Table 4.6: Estimated values of the parameters, with percentage errors.

Parameter	Value	Error (%)
$P_{nom}$	81.0 kW	28.0
$k_p$	2.1	44.5
$P_{gen}$	18.3 kW	63.4

From the two figures below it is clear how a less accurate set of measures would lead to misleading results, whatever are the initial conditions. However, it is not far from the truth saying that the resolution of the measures required for the convergence is not unrealistic.

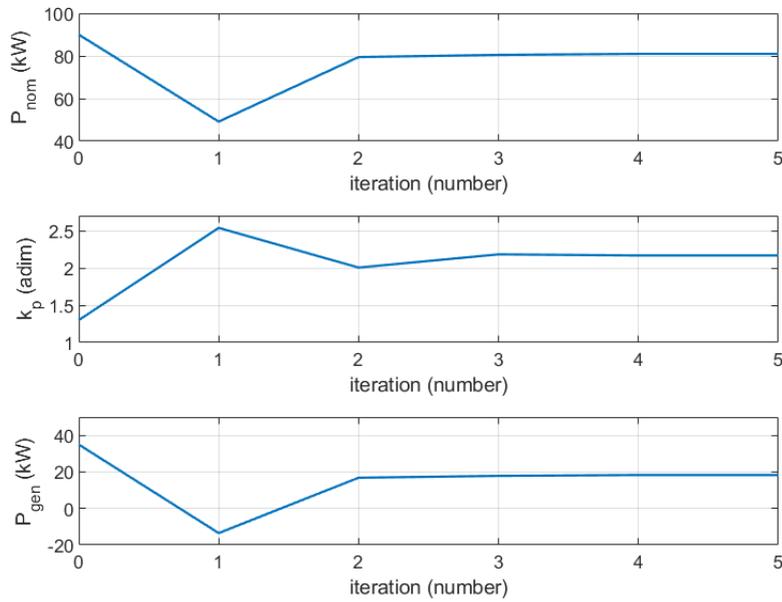


Figure 4.6: Estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with low accuracy measures of voltage and active power.

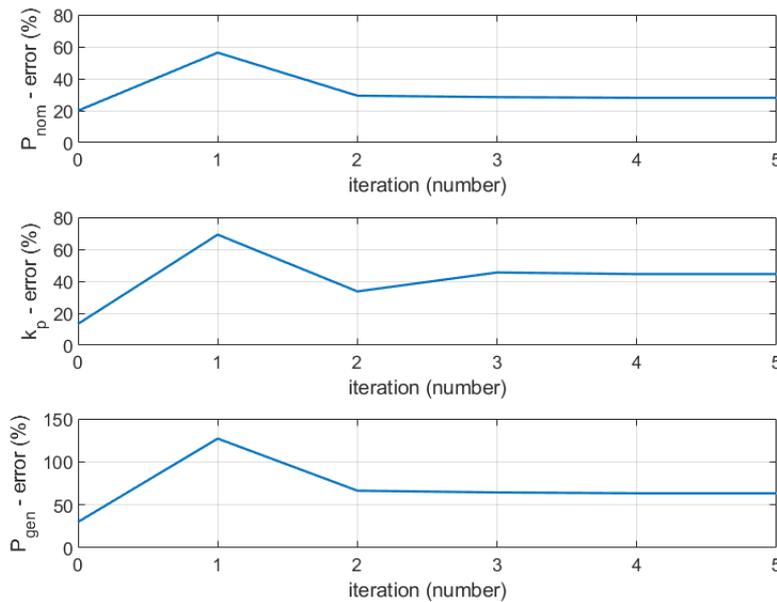


Figure 4.7: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with high accuracy measures of voltage and active power.

### 4.2.3 Case Study C: three parameters estimation with a three-step trajectory and white noise in the measures

The third application of the method involves varying the tap changer of the HV-MV transformer three times. To emulate the unavoidable noise in the measuring process, a white noise is arbitrary added at the signals of active power and line-to-line voltage. The obtained profiles are illustrated in Figure 4.8. To facilitate the measuring process of both the voltage and the active power, a sampling of value is implemented. In the abovementioned figure, beside the sampled measures (red squares) it is reported for both the active power and the voltage the expected profile without the noise.

The idea is to exploit the noise so to have more values to be inserted in the regression. Because a high number of measures is achieved thanks to the added noise, the tap changer is varied only three times.

The implementation on Simulink of the addition of noise and the data-sampling is properly described in Appendix C.

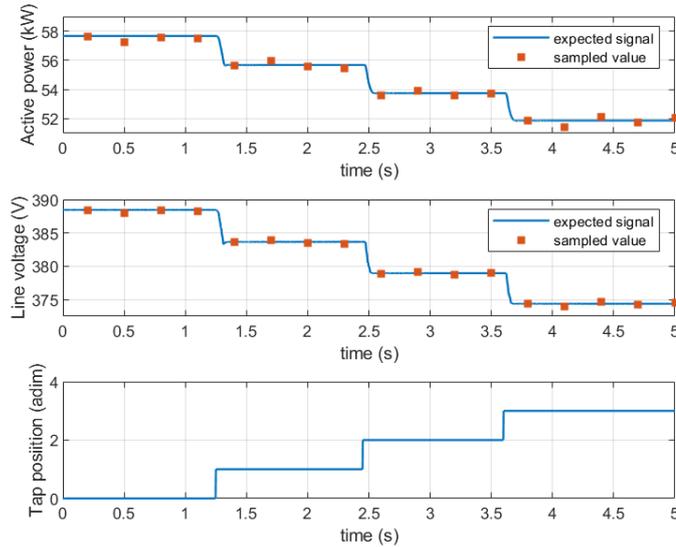


Figure 4.8: Active power measured below the secondary substation (upper); line-to-line voltage measured below the secondary substation (center); tap changer position of the HV-MV transformer (lower).

For this implementation the used set of measures is shown below in Table 4.7.

Table 4.7: Set of measures used for the implementation of this case

Measure	Active power (kW)	Voltage (V)
1	57.635	388.42
2	57.239	383.03
3	57.577	388.37
4	57.514	388.30
5	55.629	383.91
6	55.930	383.91
7	55.557	383.54
8	55.422	383.40
9	53.608	378.83
10	53.894	379.12
11	53.561	378.78
12	53.733	378.95
13	51.862	374.37
14	51.413	373.93
15	52.100	374.61
16	51.760	374.27
17	52.039	374.55

The initial values of the parameters are set as in the previous test (see Table 4.2). The results are listed in Table 4.8. The profiles are illustrated in Figure 4.9 and Figure 4.10.

Moreover, a higher number of iterations is required to obtain the convergence (ten iterations rather than five).

It is clear how the method's ineffectiveness in case of a set of measures spoiled by noise. In a following case study the same set of measures will be re-used in order to estimate only  $P_{nom}$  and  $P_{gen}$ .

Table 4.8: Estimated values of the parameters, with percentage errors

Parameter	Value	Error (%)
$P_{nom}$	-466.250 kW	514.450
$k_p$	-0.33	121.86
$P_{gen}$	-528.320 kW	1156.640

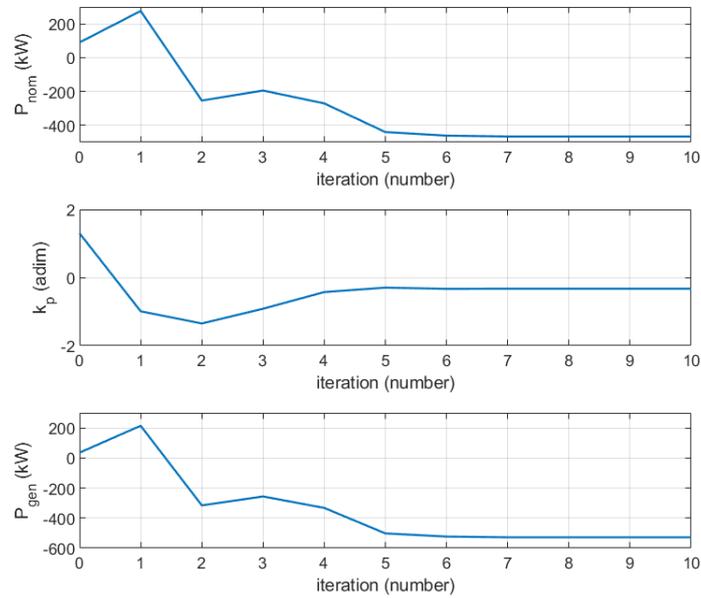


Figure 4.9: Estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with measures spoiled by noise.

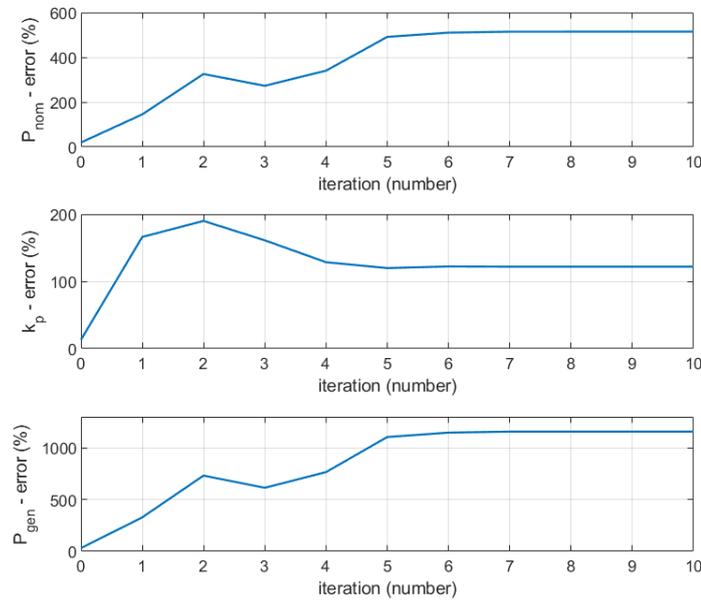


Figure 4.10: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper),  $k_p$  (center) and  $P_{gen}$  (lower) with measures spoiled by noise.

#### 4.2.4 Case study D: two parameters estimation with a five-step trajectory and a set of measures with low accuracy

The non-linear regression method is now implemented with the aim of estimate only two of the three parameters. The exponential coefficient  $k_p$  is supposed to be known and it is set to 1.5. The set of measures used in this case study is the same reported in Table 4.5.

The initial condition of the two parameters are listed in Table 4.9.

Table 4.9: Initial values of the two parameters

Parameter	Value
$P_{nom}^{(0)}$	90 kW
$P_{gen}^{(0)}$	35 kW

The estimated values at convergence are listed in the below Table 4.10, while the profiles are illustrated in Figure 4.11 and Figure 4.12.

The parameters converge rapidly to a set of values with an extremely low percentage error. Unlike a three-parameters estimation, the two-parameters analysis enables us to achieve a highly precise set of results.

Table 4.10: Estimated values of the parameters, with percentage errors

Parameter	Value	Error (%)
$P_{nom}$	112.5 kW	0.04
$P_{gen}$	49.9 kW	0.07

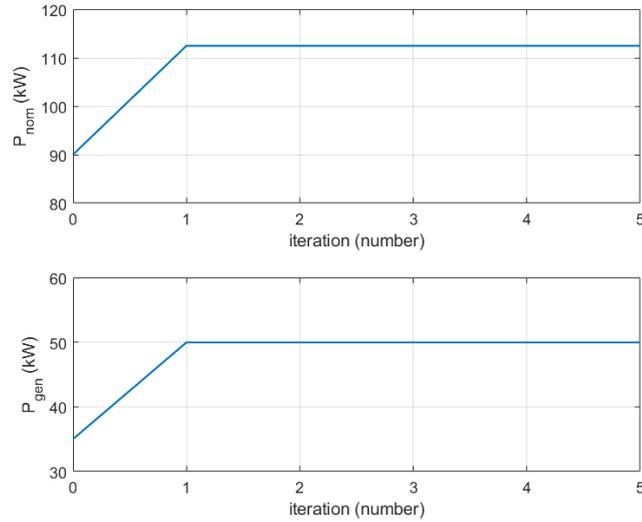


Figure 4.11: Estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures and with  $k_p = 1.5$ .

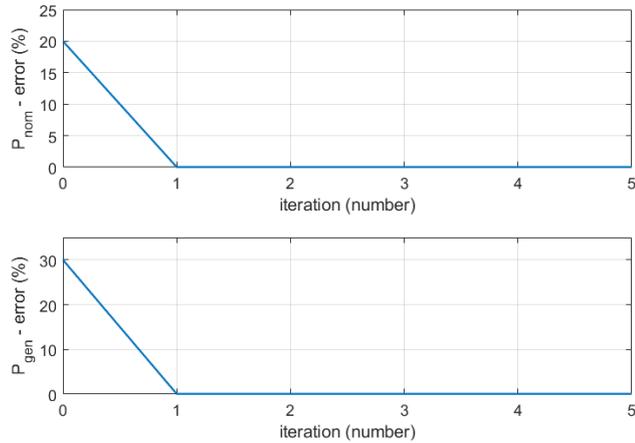


Figure 4.12: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures and with  $k_p = 1.5$

The previous analysis has been carried out supposing to know the parameter  $k_p$ . However, in a real case its knowledge may be not guaranteed. To evaluate the regression method with an uncorrect  $k_p$  value, two more implementation with the same set of measures are here described.

The estimation is executed with  $k_p$  set to 1.45 and 1.55 respectively. The results of the convergence are listed in Table 4.11, while the profiles of the two variables and the errors are illustrated in Figure 4.13 and in Figure 4.14.

A slightly wrong knowledge of  $k_p$  brings to a less accurate estimation of the two parameters, although the results can be still considered acceptable.

Table 4.11: Estimated value of the parameters and percentage errors

Parameter	$k_p = 1.45$		$k_p = 1.55$	
	Value	Error (%)	Value	Error (%)
$P_{nom}$	115.99	3.1	109.15	3.0
$P_{gen}$	53.51	7.0	46.65	6.7

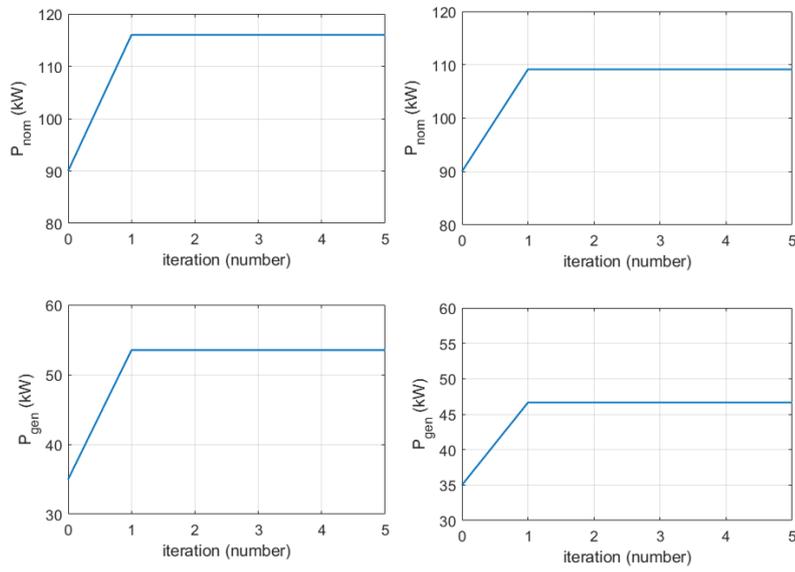


Figure 4.13: Estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures, with  $k_p = 1.45$  (left) and with  $k_p = 1.55$  (right).

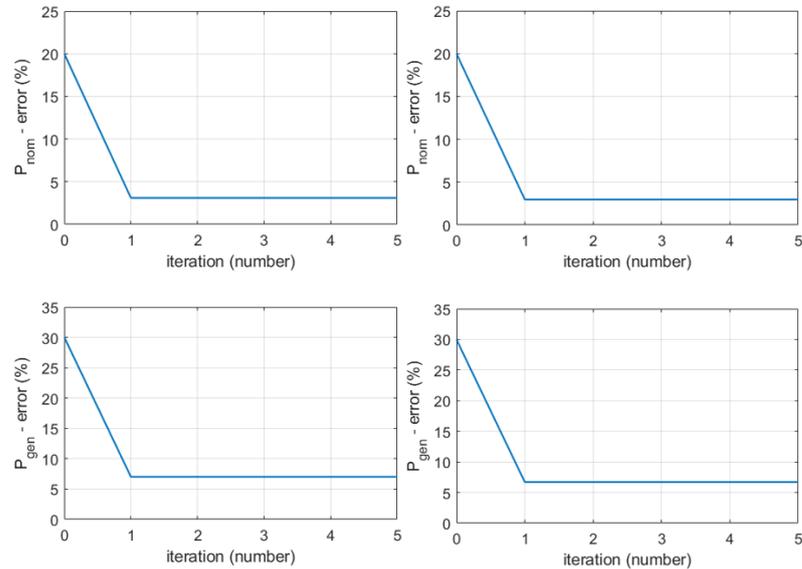


Figure 4.14: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures, with  $k_p = 1.45$  (left) and with  $k_p = 1.55$  (right).

#### 4.2.5 Case study E: two parameters estimation with a three-step trajectory and noise in the measures

The non-linear regression method is now implemented with the aim of estimate only two of the three parameters. The exponential coefficient  $k_p$  is a once again set to 1.5. The set of measures used in this case study is the same reported in Table 4.7.

The initial condition of the two parameters are the same reported in Table 4.9.

The estimated values at convergence are listed in the below Table 4.12, while the profiles are illustrated in Figure 4.13 and Figure 4.14.

Table 4.12: Estimated values of the parameters, with percentage errors

Parameter	Value	Error (%)
$P_{nom}$	111.33 kW	1.04
$P_{gen}$	48.95 kW	2.10

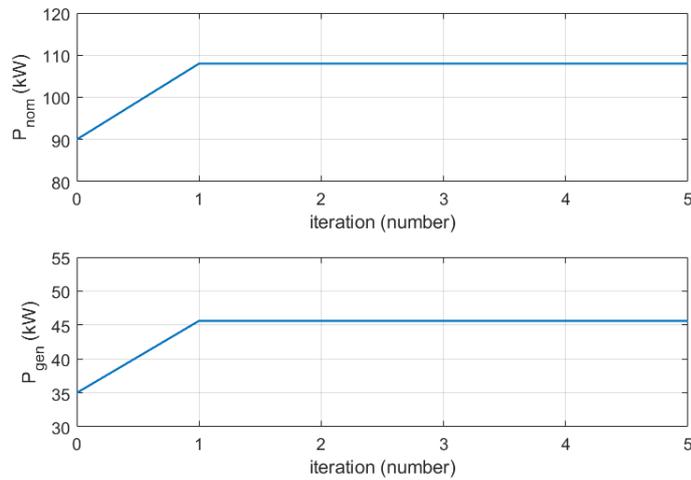


Figure 4.15: Estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with measures spoiled by noise and with  $k_p = 1.5$ .

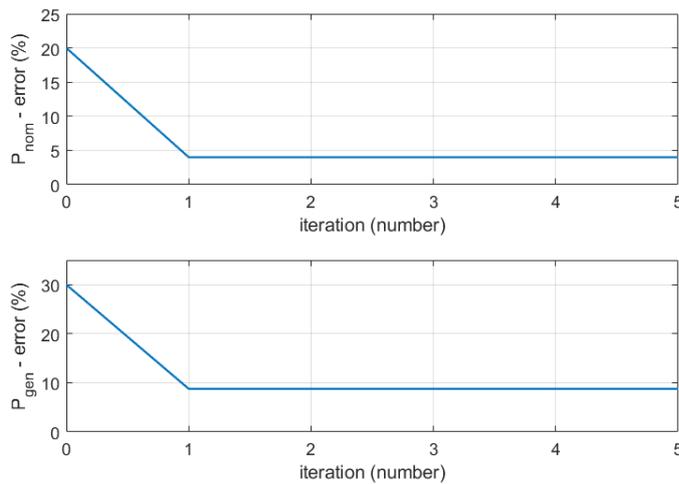


Figure 4.16: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with measures spoiled by noise and with  $k_p = 1.5$

The same estimation is now repeated with a wrong estimate of the parameter  $k_p$ . Similarly to what has been done in *Case Study D*, the  $k_p$  value is set to 1.45 and 1.55 respectively. As debated for the previous case study, a wrong  $k_p$  brings to encouraging results, even though the relative error rises if compared with what obtained with  $k_p = 1.5$ .

Table 4.13: Estimated values of the two parameters

Parameter	$k_p = 1.45$		$k_p = 1.55$	
	Value	Error (%)	Value	Error (%)
$P_{nom}$	107.99	4.0	114.89	2.1
$P_{gen}$	45.61	8.8	52.52	5.0

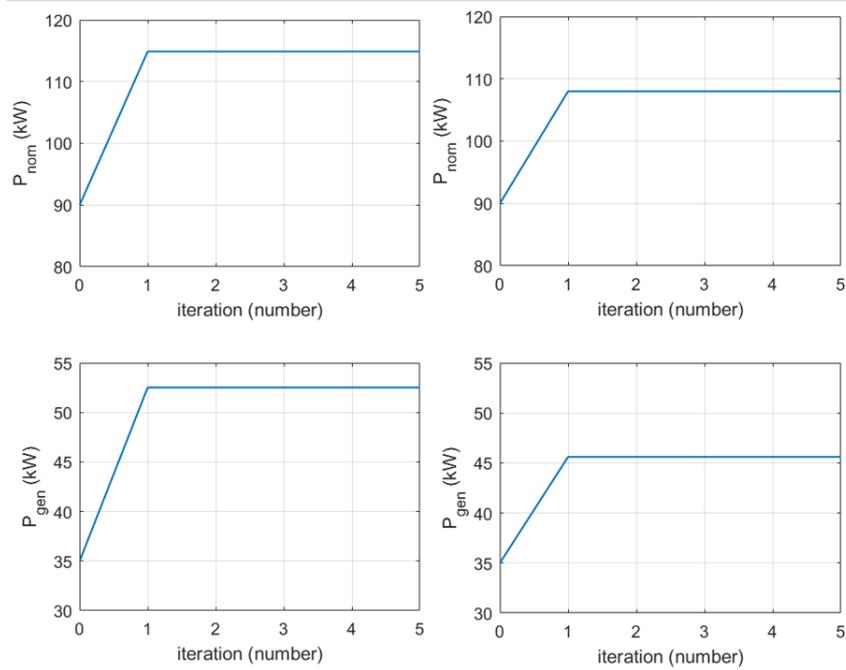


Figure 4.17: Estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures, with  $k_p = 1.45$  (left) and with  $k_p = 1.55$  (right).

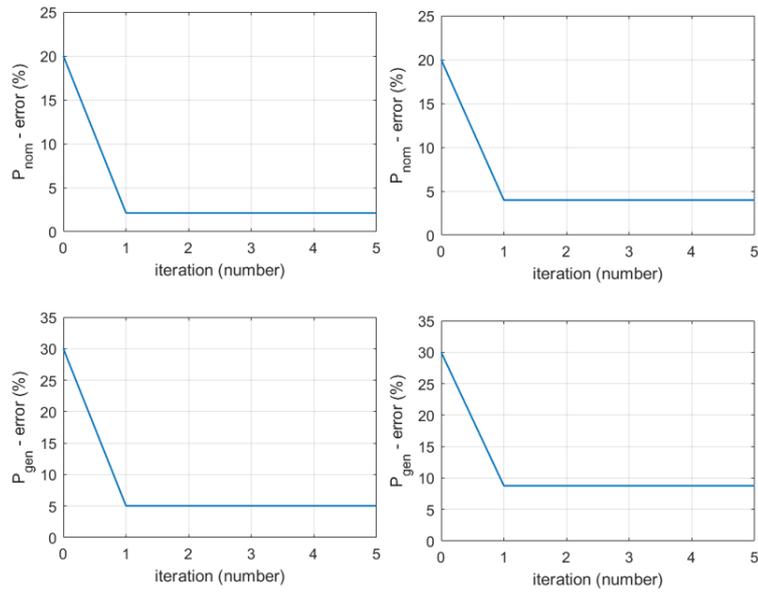


Figure 4.18: Percentage errors of the estimation of the parameters  $P_{nom}$  (upper) and  $P_{gen}$  (lower) with badly accurate measures, with  $k_p = 1.45$  (left) and with  $k_p = 1.55$  (right).

# Chapter 5

## Fault analysis on a Distribution Network

In this Chapter the electric grid already discussed in Chapter 3 and Chapter 4 will be further analysed. The grid will be now simulated exclusively on the Turin-based simulator, thus without a remote connection. Moreover, no PHIL will be included.

In this Chapter the main topic for which the grid is implemented will be the different types of electric faults occurring on the MV network. For each type of fault the results of the simulation will be evaluated with the aim to have a confirmation from the theoretical aspects of the fault analysis. The grid analysis is anticipated by a theoretical discussion of the method of the symmetrical components.

To appreciate some of the aspects of the fault analysis it is necessary to have more feeders powered by the same HV-MV transformer. For such reason, the fault analysis will be concentrated on the MAGRA and BRENTA feeders, which are powered by the Transformer  $T1$ . Please note that for the analysis' sake the transformer  $T1$  will not have the tap changer installed.

To better understand the properties of each fault type, the neutral of the secondary winding in the HV-MV transformer will be studied considering a certain variety of connections (i.e.: isolated neutral, compensated neutral).

Specifically, the following types of fault will be imposed and analysed:

- Three-phase fault with insulated neutral in the transformer;
- *Line to Line* fault with insulated neutral in the transformer;

- *Line to Earth* fault with insulated neutral in the transformer;
- *Line to Earth* fault with compensated neutral in the transformer;

Finally the validation of the model is carried out by confronting some results with what is obtained with the RTDS<sup>©</sup> realtime simulator, using some of the results obtained by Andrea Aleo in [1]. Specifically, some of the results concerning the faults are compared.

## 5.1 Method of symmetrical components

As it is shown in the list above, the faults occurring in an electric power system may be related just to one phase or two phases, imposing an unbalanced condition in the network. Because of this intrinsic feature of faults, a traditional circuit analysis would not be appropriate. Thus, the method of symmetrical components is introduced.

First, let us consider a balanced three-phase voltage, like the one shown in Figure 5.1 (left). The phasors are defined with the parameter  $\alpha$ , whose value is set as a definition [9]:

$$\alpha = e^{j\frac{2\pi}{3}} \quad (5.1)$$

Because of how it is defined, the  $\alpha$  parameter follows this mathematical relation:

$$1 + \alpha + \alpha^2 = 0 \quad (5.2)$$

The Figure 5.1 (right) depicts clearly the idea of 5.2.

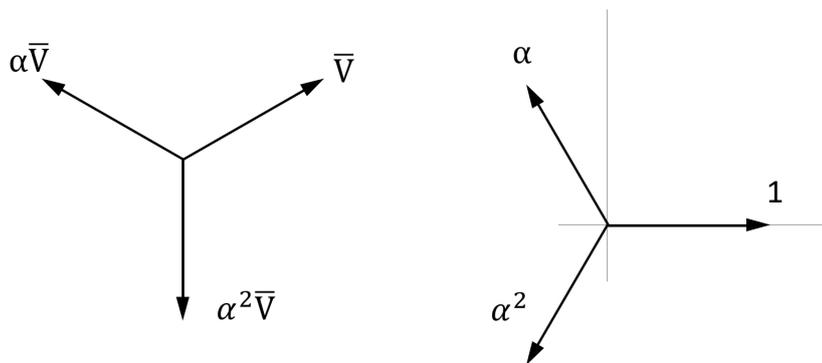


Figure 5.1: (Left) Balanced three-phase voltage; (Right) rotation operators.

Let us now define the three possible sequences: the *positive sequence*, the *negative sequence* and the *zero sequence*. The three sequences are shown in Figure 5.2 [9].

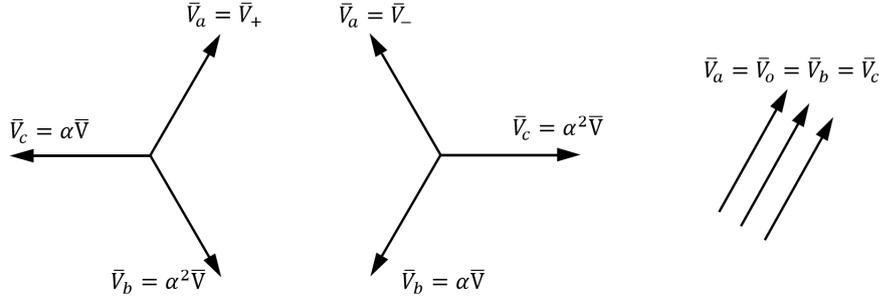


Figure 5.2: (Left) Positive sequence; (Center) Negative sequence; (Right) zero sequence.

It is now required to introduce the *Fortesque Theorem*, which said that *each three phasors  $\bar{V}_a, \bar{V}_b$  and  $\bar{V}_c$  of a three-phase system can be represented as a combination of a positive sequence-three phasors, a negative sequence-three phasors and a zero sequence-three phasors*. This combination can be expressed with a matrix notation [9]:

$$\begin{bmatrix} \bar{V}_a \\ \bar{V}_b \\ \bar{V}_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ \alpha^2 & \alpha & 1 \\ \alpha & \alpha^2 & 1 \end{bmatrix} \cdot \begin{bmatrix} \bar{V}_+ \\ \bar{V}_- \\ \bar{V}_o \end{bmatrix} \quad (5.3)$$

For briefness' sake, the 5.3 can ben written in the following format:

$$\bar{\mathbf{v}}_{abc} = \mathbf{T} \bar{\mathbf{v}}_{+-o} \quad (5.4)$$

where  $\mathbf{T}$  is the *Transformation Matrix*. The sequences can be easily determined from the *abc* components by working on the 5.4. We obtain the following:

$$\bar{\mathbf{v}}_{+-o} = \mathbf{T}^{-1} \bar{\mathbf{v}}_{abc} \quad (5.5)$$

where  $\mathbf{T}^{-1}$  will be:

$$\mathbf{T}^{-1} = \frac{1}{3} \begin{bmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \\ 1 & 1 & 1 \end{bmatrix} \quad (5.6)$$

As we obtain the 5.4, we can write the following for the currents of a three-phase system:

$$\bar{\mathbf{i}}_{abc} = \mathbf{T}\bar{\mathbf{i}}_{+ - o} \quad (5.7)$$

It is now necessary to introduce a representation of the grid that will simplify the study of the unbalance faults. Let us define a generic active three-poles element [9], shown in Figure 5.3 (left). The element has the *phase a*, the *phase b* and the *phase c* poles, beside the reference pole, whose function may be the ground or the neutral of the system. This representation can only be achieved with the hypothesis of physical symmetry of the system.

If we consider the generic element in a no-load condition using the Thevenin equivalent, we will get the *electromotive force (emf)*. Specifically, we obtain the voltages shown in Figure 5.3 (right), in which  $\bar{V}_{a0} = \bar{E}_+$ ,  $\bar{V}_{b0} = \alpha^2\bar{V}_{a0}$  and  $\bar{V}_{c0} = \alpha\bar{V}_{a0}$ . It is important to note that once we pass onto the sequences, the negative and the zero sequences will have  $\bar{E}_- = 0$  and  $\bar{E}_o = 0$ , while the positive sequence will be the only one with an emf different than zero ( $\bar{E}_+ \neq 0$ ).

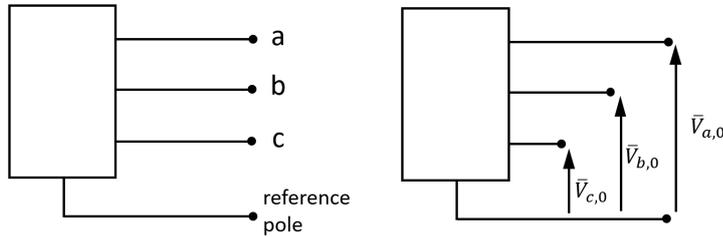


Figure 5.3: General three-pole representation.

The last structures to be introduced are the sequences' networks (Figure 5.4), which are defined with the Thevenin equivalent [9]. The sequence's networks are used in the faults analysis once the fault has been defined: according to the voltages and currents relationships, the sequence's networks are connected in series or in parallel in order to extract the fault current. Moreover, this structures are studied separately to define the voltages after the fault.

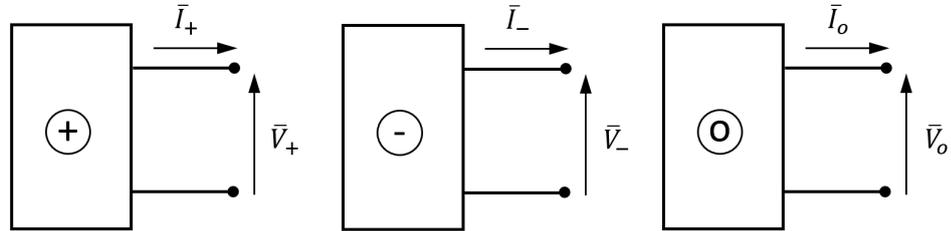


Figure 5.4: Generic representation of the three symmetrical sequences.

## 5.2 Three-phase fault

The three-phase fault, illustrated in Figure 5.5 is the easiest typology to analyze from a theoretical point of view, as the balanced condition does not force to use the symmetrical components. It is not the most frequent type of fault in the real electrical network. However, it is still largely studied for its practicality and to be used as a benchmark for the other case-scenarios of fault.

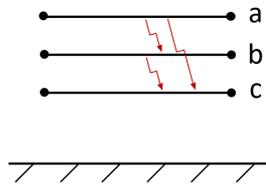


Figure 5.5: Three-phase fault representation.

Because of the symmetry of the network with such fault, the fault impedances of the three phases will be the same and equal to  $\bar{Z}_f$ . The fault currents are usually of the positive sequence. The negative and zero sequence do not give any contribution. Therefore, the current  $\bar{I}_a$  can be written as follows:

$$\bar{I}_+ = \frac{\bar{V}_+}{\bar{Z}_f} \quad (5.8)$$

By studying separately the positive sequence with its Thevenin equivalent (shown in Figure 5.6), we obtain:

$$\bar{I}_+ = \frac{\bar{E}_+}{\bar{Z}_f + \bar{Z}} \quad (5.9)$$

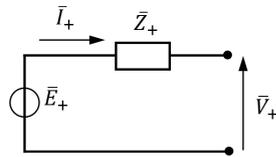


Figure 5.6: Thevenin equivalent of the positive sequence.

where  $\bar{E}_+$  is the prior-to-the-fault voltage. In the case on the network under study, the three-phase fault has been applied in the bus number 4 of the feeder Magra, as illustrated in Figure 5.7.

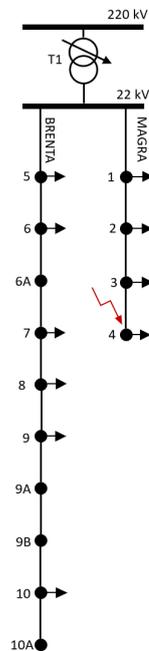


Figure 5.7: Identification of the position of the applied fault.

The pre-fault condition in the Magra feeder are shown in Figure 5.8 (left). The fault condition is instead illustrated in Figure 5.8 (right). Some aspects are worth being discussed. Firstly, the symmetrical fault creates a symmetrical system after the fault, as expected from theory. Secondly, as the currents rise abruptly, the voltage amplitude drops.

Please note that the values reported in Figure 5.8 and 5.9 are measured at the beginning of the MAGRA feeder.

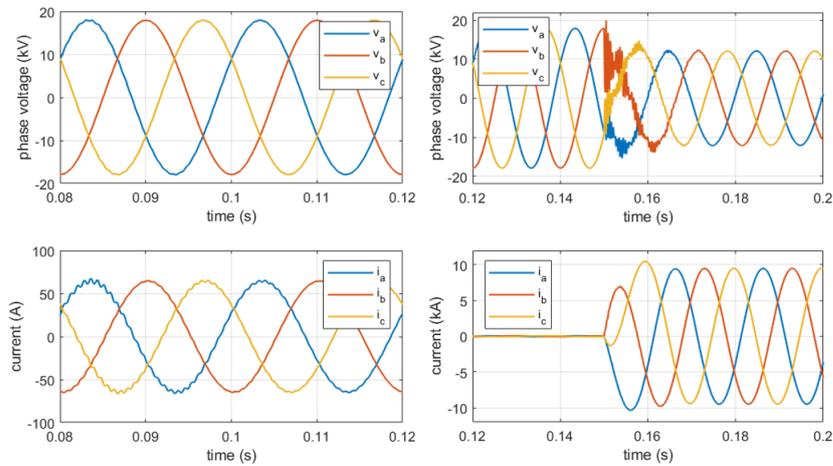


Figure 5.8: (Left): pre-fault condition; (Right): fault condition - (Upper) phase to ground voltage at the beginning of Magra feeder; (Lower) line currents at the beginning of Magra feeder.

A final aspect is the influence of the  $\bar{Z}_f$  on the current amplitude. The fault has been simulated with different values of fault impedance. The results confirm the expectation from 5.9: the higher the  $\bar{Z}_f$ , the lower will be the current. The results are illustrated in Figure 5.9.

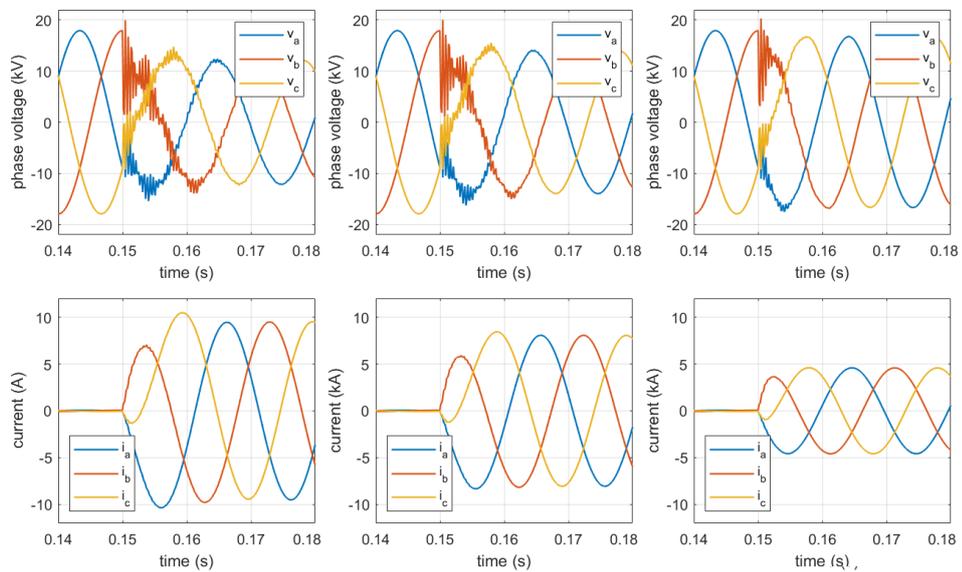


Figure 5.9: Phase-to-ground voltage and current at the beginning of Magra feeder with different values of  $\bar{Z}_f$ : (Left)  $\bar{Z}_f = 0.001\Omega$ ; (Center)  $\bar{Z}_f = 0.5\Omega$ ; (Right)  $\bar{Z}_f = 2.5\Omega$ .

### 5.3 Line to Line fault

The line-to-line fault is now considered. The fault is supposed to happen between phase  $b$  and phase  $c$ , as shown in Figure 5.10.

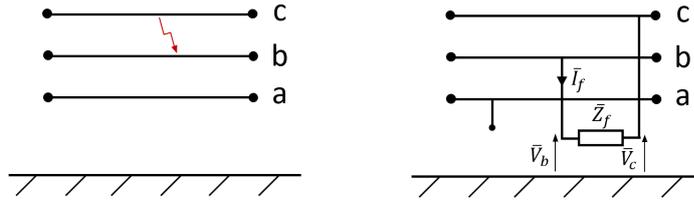


Figure 5.10: Line-to-line fault representation.

The fault is defined by the following equations [9]:

$$\left\{ \begin{array}{l} \bar{I}_a = 0 \\ \bar{I}_b = -\bar{I}_c = \bar{I}_f \\ \bar{V}_b - \bar{V}_c = \bar{Z}_f \bar{I}_f \end{array} \right.$$

We now determine the symmetrical sequences by applying the aforementioned transformations:

$$\left\{ \begin{array}{l} \bar{I}_o = 0 \\ \bar{I}_+ = -\bar{I}_- \\ \bar{V}_o = 0 \\ \bar{V}_+ - \bar{V}_- = \bar{Z}_f \bar{I}_f \end{array} \right.$$

Moreover, we can define a relation between  $\bar{I}_f$  and  $\bar{I}_+$ :

$$\bar{I}_f = (\alpha^2 - \alpha) \bar{I}_+ \tag{5.10}$$

Given the previous formulas, we can create the connexion among the network sequences [9] (Figure 5.11).

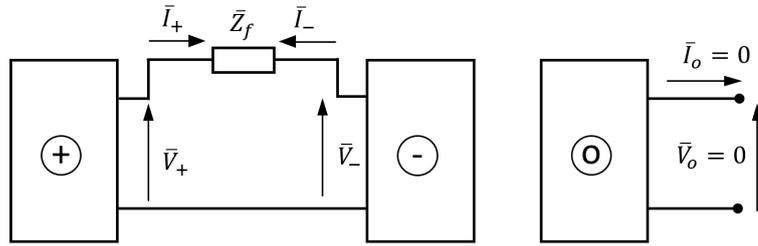


Figure 5.11: Composition of the symmetrical sequences.

We now use the Thevenin equivalent, obtaining the circuit of Figure 5.12 [9].

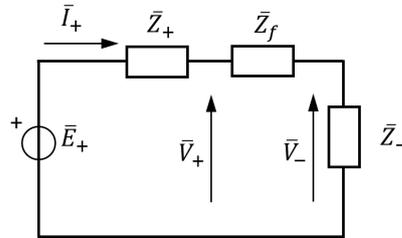


Figure 5.12: Composition of the sequences with the Thevenin equivalent.

From the above circuit we can extract the following results:

$$\bar{I}_+ = -\bar{I}_- = \frac{\bar{E}_+}{\bar{Z}_+ + \bar{Z}_- + \bar{Z}_f} \quad (5.11)$$

$$\bar{I}_f = (\alpha^2 - \alpha) \bar{I}_+ = \frac{(\alpha^2 - \alpha) \bar{E}_+}{\bar{Z}_+ + \bar{Z}_- + \bar{Z}_f} \quad (5.12)$$

If we consider  $\bar{Z}_f = 0$  and  $\bar{Z}_+ = \bar{Z}_+$ , we obtain:

$$\bar{I}_f = \frac{-\sqrt{3}j\bar{E}_+}{2\bar{Z}_+}$$

Considering that for a three-phase fault with  $\bar{Z}_f = 0$  we have the following

$$\bar{I}_{f3ph} = \frac{\bar{E}_+}{\bar{Z}_+}$$

We can extract the following relation [9]:

$$\left| \frac{\bar{I}_f}{\bar{I}_{f,3ph}} \right| = \frac{\sqrt{3}}{2} \simeq 0.866 \quad (5.13)$$

The line-to-line fault is applied at the same bus of the abovementioned three-phase fault. The results are illustrated in Figure 5.13. Firstly, the plots show how the currents follow the theoretical analysis, as the current of phase  $a$  is negligible and the other two phases have current equal in amplitude but with opposite sign. Secondly, if we compare the current amplitude with the three-phase fault we obtain the following ratio:

$$\left| \frac{\bar{I}_f}{\bar{I}_{f,3ph}} \right| \simeq \frac{8268.4}{9497.4} \simeq 0.870 \quad (5.14)$$

which is close to the ratio determined in 5.13, confirming the validity of the simulation results.

Please note that the values shown in Figure 5.13 are measured at the beginning of the MAGRA feeder.

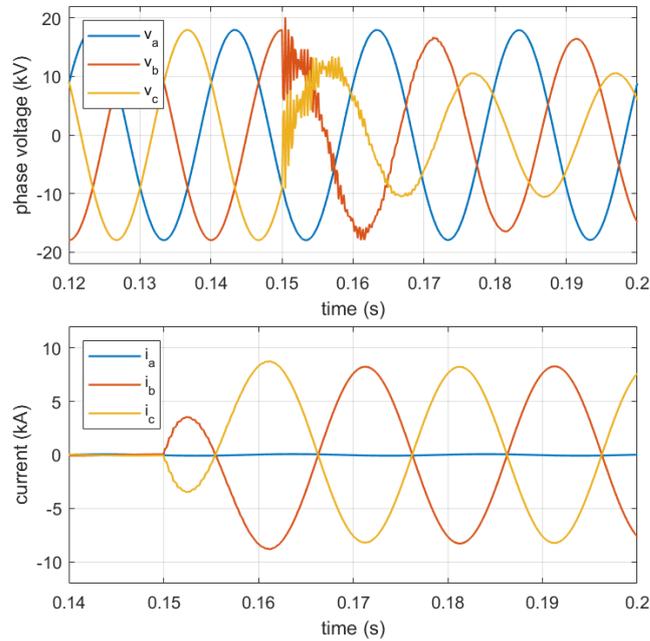


Figure 5.13: Line-to-line fault condition: (Upper) phase-to-ground voltage at the beginning of Magra feeder; (Lower) line current at the beginning of Magra feeder.

## 5.4 Line-to-earth faults

This type of fault is statistically the most frequent. In this section it will be studied both with a insulated neutral and with a compensated neutral.

Suppose a line-to-earth happens with phase  $a$ , as shown in Figure 5.14.

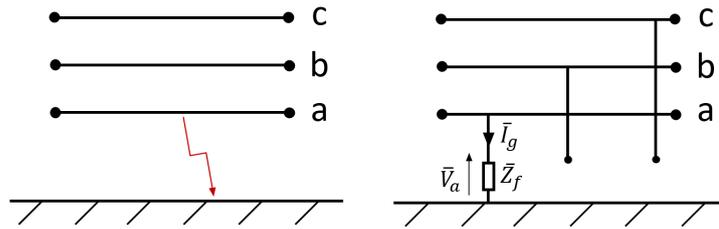


Figure 5.14: Line-to-earth fault representation.

The fault can be defined by the following set of equations [9]:

$$\begin{cases} \bar{I}_a = \bar{I}_f \\ \bar{I}_b = -\bar{I}_c = 0 \\ \bar{V}_a = \bar{Z}_f \bar{I}_f \end{cases}$$

The successive step is to determine the symmetrical sequences:

$$\begin{cases} \bar{I}_+ = \bar{I}_- = \bar{I}_o = \bar{I}_f \\ \bar{V}_a = \bar{V}_+ + \bar{V}_- + \bar{V}_o = 3\bar{Z}_f \bar{I}_+ \end{cases}$$

Using the last set of equations we can easily combine the three sequences' networks (Figure 5.15 (left)) and use the Thevenin equivalent to obtain the circuit depicted in Figure 5.15 (right) [9].

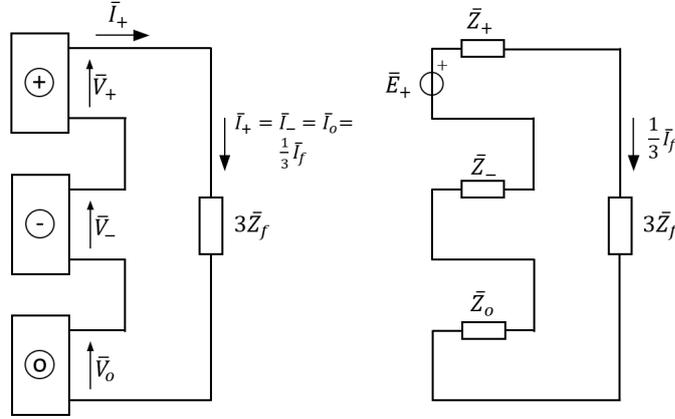


Figure 5.15: (Left) Composition of the symmetrical sequences; (Right) Thevenin equivalent.

From the abovesshown circuit we determine the current  $\bar{I}_+$ :

$$\bar{I}_+ = \bar{I}_- = \bar{I}_o = \frac{\bar{E}_+}{\bar{Z}_+ + \bar{Z}_- + \bar{Z}_o + 3\bar{Z}_f}$$

And thus the current  $\bar{I}_f$ :

$$\bar{I}_f = \frac{3\bar{E}_+}{\bar{Z}_+ + \bar{Z}_- + \bar{Z}_o + 3\bar{Z}_f} \quad (5.15)$$

If the impedance  $\bar{Z}_f$  is negligible, we obtain:

$$\bar{I}_f = \frac{\bar{E}_+}{\frac{\bar{Z}_+ + \bar{Z}_- + \bar{Z}_o}{3}} \quad (5.16)$$

Finally, if the  $\bar{Z}_+$  and  $\bar{Z}_-$  are negligible compared to  $\bar{Z}_o$ , we can write the following:

$$\bar{I}_f \simeq \frac{3\bar{E}_+}{\bar{Z}_o} = 3j\omega C_{o,tot}\bar{E}_+ \quad (5.17)$$

When studying the line-to-earth fault, a focus on the voltage of healthy phases is required. Let us suppose to have the fault on phase  $a$ , we obtain:

$$\begin{aligned} \bar{V}_b &= \alpha^2\bar{V}_+ + \alpha\bar{V}_- + \bar{V}_o = \alpha^2\bar{E}_+ - \bar{E}_+ = (\alpha^2 - 1)\bar{E}_+ \\ |\bar{V}_b| &= |\alpha^2 - 1|\bar{E}_+ = \sqrt{3}\bar{E}_+ \end{aligned} \quad (5.18)$$

Similarly for phase  $c$ :

$$\bar{V}_c = \alpha \bar{E}_+ - \bar{E}_+ = (\alpha - 1) \bar{E}_+$$

$$|\bar{V}_c| = \sqrt{3} \bar{E}_+ \tag{5.19}$$

The variation, also shown in Figure 5.16, is an extreme case, as we have considered to have  $\bar{Z}_f = 0$ , and thus having the ground potential coinciding with the potential of phase  $a$ .

However, varying the fault impedance will impact on the ground potential and thus on the voltage on the healthy phases, as depicted in Figure 5.17. The ground potential a semicircle, creating different scenarios for the voltage of healthy phases.

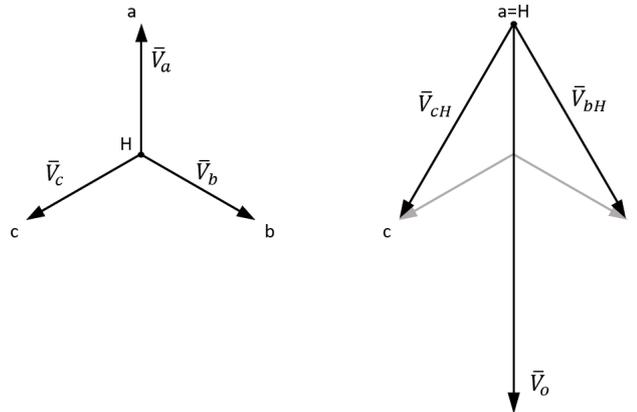


Figure 5.16: (Left) Three-phase voltage pre-fault; (Right) three-phase voltage after line-to-earth fault with  $Z_f = 0$ .

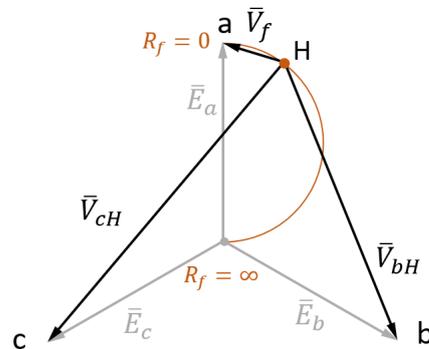


Figure 5.17: Variation of the voltages as a function of the  $Z_f$  value.

Please note that the concept previously exposed in this section can be considered valid whether the neutral is insulated or compensated.

It is introduced now the compensated neutral, starting from the components shown in Figure 5.18 (left).

Firstly, there is the impedance  $X_b$ , that is adapted to the  $C_o$  of the network. The  $R_s$  helps reducing the transient components. Finally, the  $R_p$  introduces a resistive component in  $\bar{I}_f$ .

However, if  $R_p \gg X_b$  and  $R_s \ll X_b$ , we can approximate the circuit to the configuration shown in Figure 5.18 (center). Finally, the overall zero impedance  $\bar{Z}_o$  can be represented with the circuit shown in Figure 5.18 (right) [9].

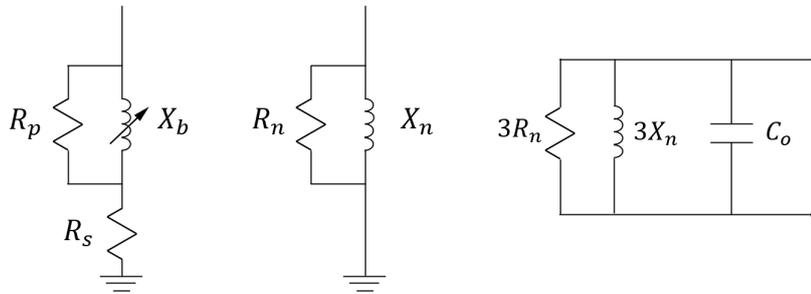


Figure 5.18: (Left) Compensated neutral; (Center) Approximated compensated neutral; (Right) Circuit model of  $\bar{Z}_o$ .

Let us now consider the elementar grid illustrated in Figure 5.19. Of the two distribution feeders only one is impacted by a line-to-earth fault [9].

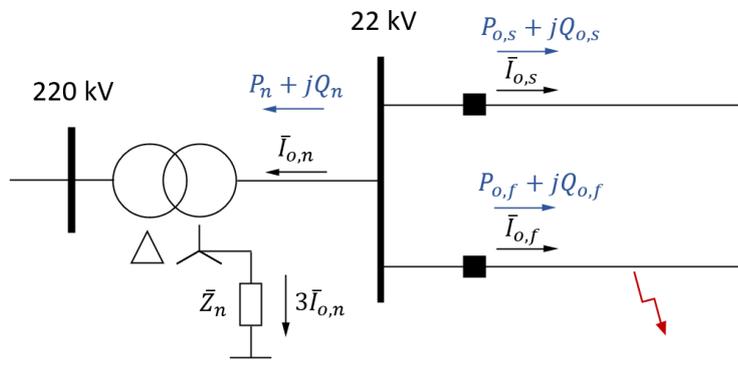


Figure 5.19: Generic distribution network model.

We now represent a healthy feeder, a faulted feeder and the transformer with an equivalent circuit. The three component's equivalent circuit are depicted in Figure 5.20 [9].

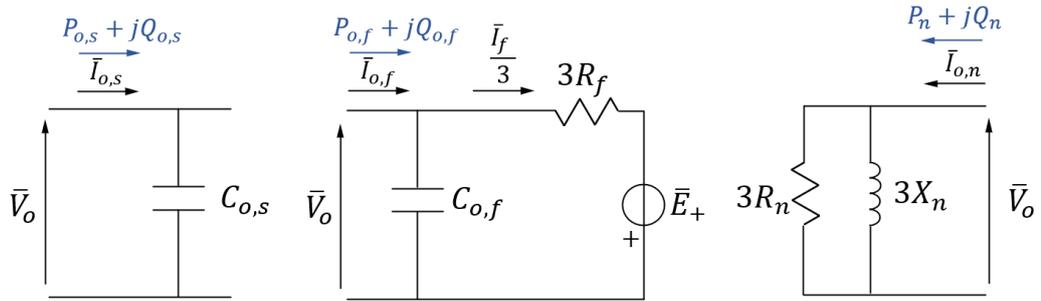


Figure 5.20: (Left) Model of a healthy feeder; (Center) Model of a faulted feeder; (Right) Model of the transformer with the neutral.

We can now unify the three models in a single one, representing the overall grid [9].

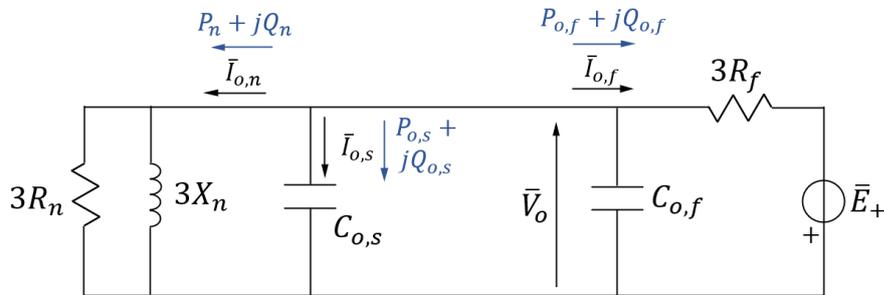


Figure 5.21: Model including the healthy feeder, the faulted feeder and the transformer's models.

Having introduced the model of the grid, it can be now discussed what happens in a line-to-earth fault with an insulated neutral.

In this condition  $\bar{I}_{o,N} = 0$ ,  $P_N = 0$  and  $Q_N = 0$ , while  $R_N$  and  $X_N$  tend to infinite.

Moreover, we can write  $Q_s = -Q_f$ , implying that  $\bar{I}_{o,f} = -\bar{I}_{o,s}$ . The criterium to recognize the faulted feeder will be then determining whether the current

is capacitive (*healthy feeder*) or inductive (*faulted feeder*), as illustrated in Figure 5.22.

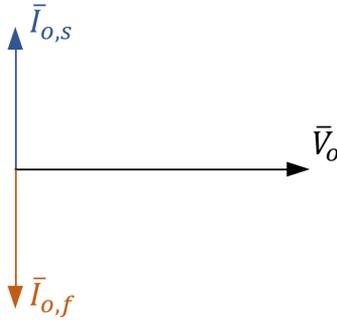


Figure 5.22: Phasor representation with the insulated neutral. Source: [9]

We consider now the case of a compensated neutral, in which the equivalent circuit is the one illustrated in Figure 5.23 (upper center). The possible scenarios are possible.

Firstly, if  $P_f < 0$  and  $Q_f > 0$  we will have the case shown in Figure 5.23 (lower left). Secondly, if  $P_f < 0$  and  $Q_f < 0$  we will have the case shown in Figure 5.23 (lower right).

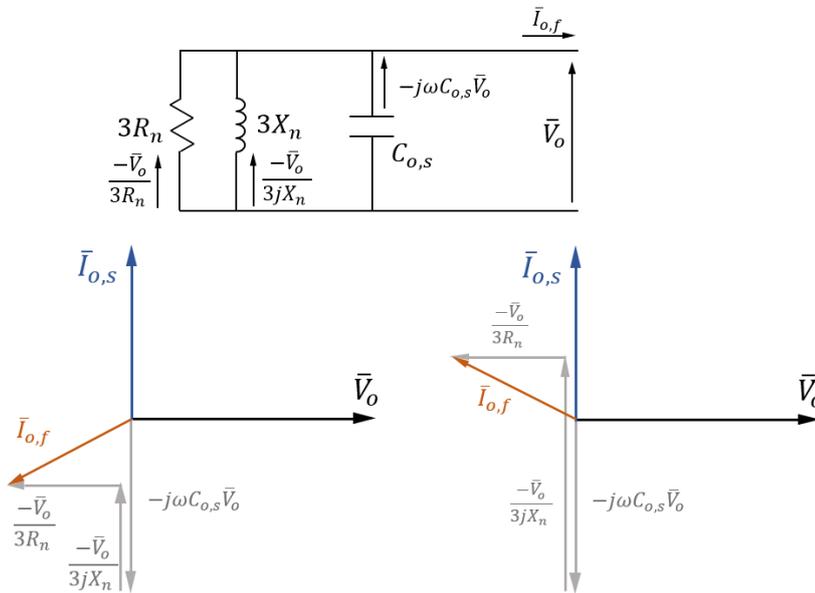


Figure 5.23: (Upper) Model of the compensated neutral and the healthy feeder; (Lower-left) Phasor representation with  $P_f < 0$  and  $Q_f > 0$ ; (Lower-right) Phasor representation with  $P_f < 0$  and  $Q_f < 0$ . Source: [9]

In conclusion, to recognize the faulted feeder it is required a range with a maximum angle  $\phi$  and a minimum angle  $\phi$ .

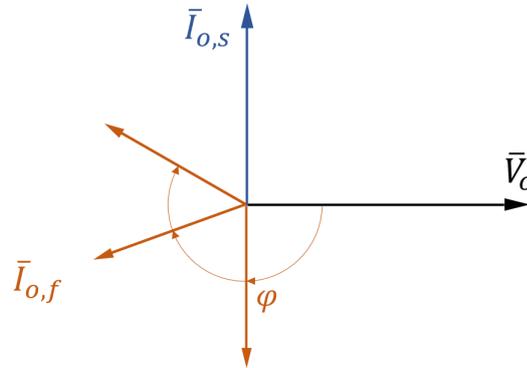


Figure 5.24: Generic phasor representation with different  $\phi$  angles. Source: [9]

The line-to-earth fault is now implemented in the evaluated network. The fault is imposed at the same bus of the previous simulations in the MAGRA feeder. Firstly, the neutral is isolated. The fault is imposed with a fault impedance  $Z_f = 0.001\Omega$ . The phase voltages and the line currents of Magra feeder are illustrated in Figure 5.25.

The values of Figure 5.25 are measured at the beginning of the MAGRA feeder.

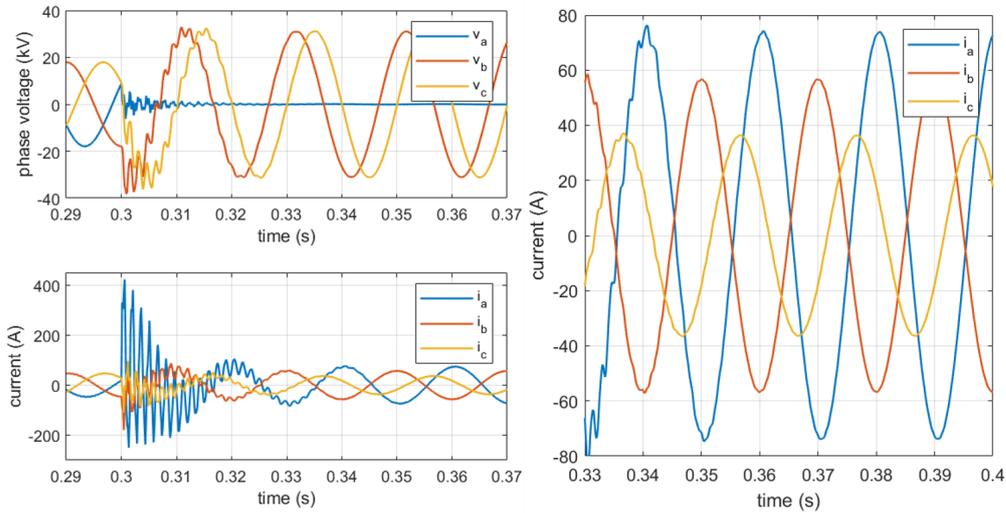


Figure 5.25: (Top-Left) Phase voltage at the beginning of Magra feeder; (Bottom-Left) Current at the beginning of Magra feeder; (Right) zoomed view of current at the beginning of Magra feeder.

As the imposed fault impedance is negligible, the voltage  $V_a$  drops almost to zero, while  $V_b$  and  $V_c$  rise to the line-to-line voltage value, confirming what has been explicated in Figure 5.16. The current of the faulted phase rises at almost 80 A, as depicted in 5.25 (right).

It is now simulated the same fault varying the value of the fault impedance  $Z_f$ . The result is illustrated in Figure 5.26, where the phase voltages of phase  $b$  and phase  $c$  are shown as a function of the inverse of  $Z_f$ .

The plot confirms what has been graphically explained with Figure 5.17. At the extreme right side of the graph we impose the fault with a value of  $Z_f$  close to zero, thus having the phase voltages converging to the phase-to-phase value. On the left side we have instead high values of  $Z_f$  (the extreme case is having no fault) and thus the two values converge to  $1/\sqrt{3}$ , which is the nominal value of the phase voltage.

There is a value of  $Z_f$ , however, by which we obtain a more critical condition for phase  $c$ . We obtain indeed a value higher than 1, thus higher than the phase-to-phase voltage value. From the graph such value is situated at around 1.05 pu.

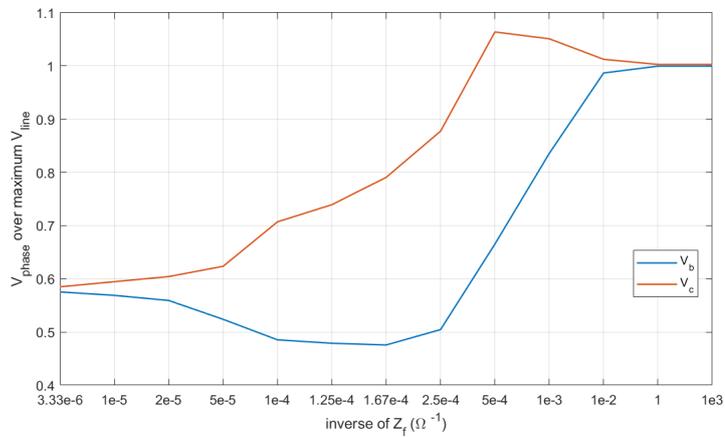


Figure 5.26: Phase voltage of phase  $b$  and phase  $c$  with a line-to-earth fault on phase  $a$ .

As mentioned in the theoretical analysis, the criterium to recognize the faulted feeder is by looking at the phases of the zero voltage and the zero current of both the faulted and the healthy feeder. The three variables here mentioned are shown in a phasorial form in Figure 5.27. The Simulink structure to define such values is shown in the Appendix D.

The results confirm that while the healthy feeder will have a capacitive zero current, the faulted feeder will present a inductive zero current.

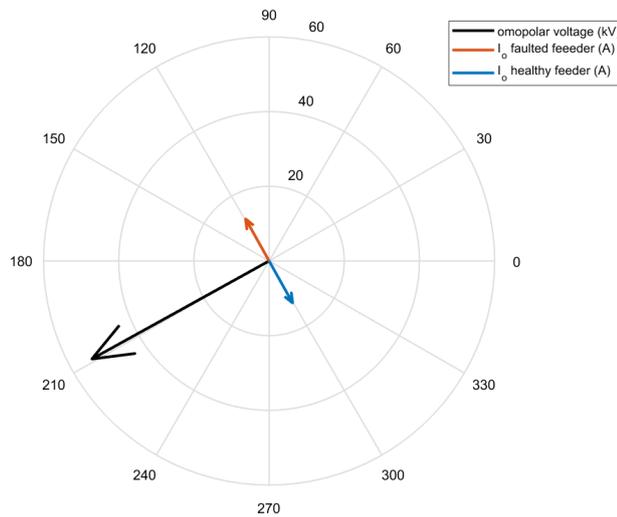


Figure 5.27: Phasor representation of a line-to-earth fault with isolated neutral.

The line-to-earth is now implemented, at the same bus, having a compensated

neutral grounding. Its elements are shown and described in Figure 5.18 (left). The values of  $R_p$  and  $R_s$  are chosen so to respect the condition to simplify the model. On the other hand, the value of  $X_b$  has been chosen in consideration of the Magra and Brenta's parameter.

In order to reduce the overcurrent caused by the fault,  $X_b$  is sized considering the *zero capacitances* of the two feeders abovementioned. By neglecting the longitudinal parameters, the capacitance can be considered in parallel. The inductance of the neutral grounding is then defined by the following:

$$X_b = \frac{X_c}{3} = \frac{1}{3\omega C_{o,Magra,Brenta}} \quad (5.20)$$

where  $C_{o,Magra,Brenta}$  is computed by summing the zero capacitances of the two feeders.

In conclusion, the values are set to:

- $R_p = 5 \cdot 10^4 \Omega$ ;
- $R_s = 50 \Omega$ ;
- $L_b = 0.771 \text{ H}$ ;

The fault is implemented with a fault impedance  $Z_f = 0.001\Omega$ . The first results are shown in Figure 5.28. While the voltage waveform has not changed with respect to the isolated neutral, the currents see a change due to the imposition of a compensated grounding neutral. Specifically, thanks to the inductance  $L_b$  we see a lower overcurrent in phase  $a$ , as we can see in Figure 5.28 (right). While with a isolated neutral the phase  $a$  reaches 80 A, the current now rises to around 60 A.

The values of Figure 5.28 are measured at the beginning of the MAGRA feeder.

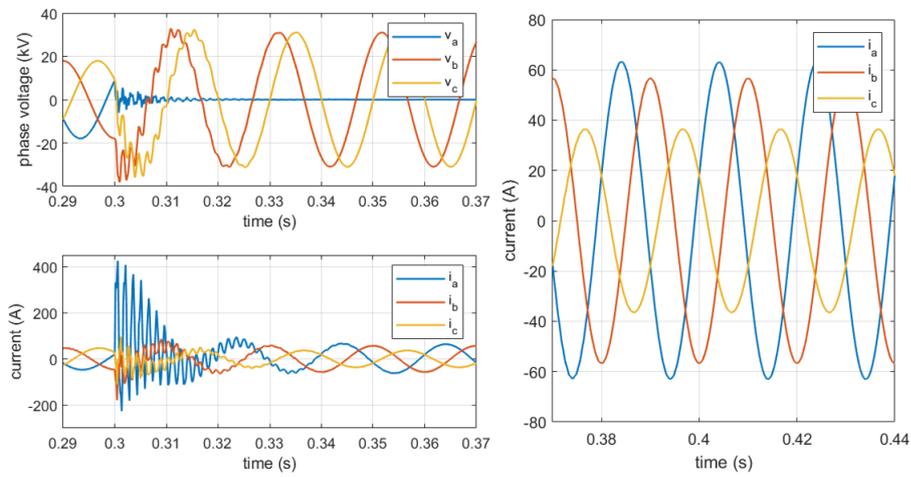


Figure 5.28: (Top-Left) Phase voltage at the beginning of Magra feeder; (Bottom-Left) Current at the beginning of Magra feeder; (Right) zoomed view of currents at the beginning of Magra feeder.

The zero components, and their phase displacement, are now investigated. The results, shown in Figure 5.29, confirm the idea expressed theoretically in Figure 5.23 and Figure 5.24.

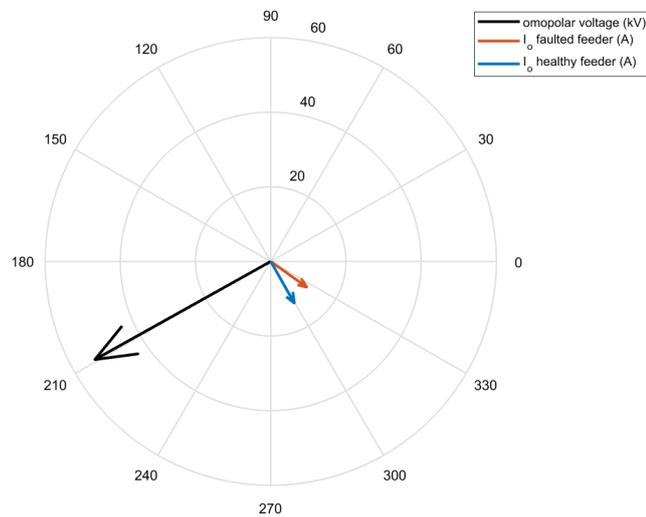


Figure 5.29: Phasor representation of a line-to-healthy fault with compensated neutral.

## 5.5 Comparison with the RTDS simulator of the faults' results

Having set the model of the distribution network, including the MV loads, a comparison of the results at steady-state is done between the OPAL-RT<sup>®</sup> realtime simulator (used for the here discussed Thesis) and the RTDS<sup>®</sup> realtime simulator (used for [1]).

A series of results of the simulations executed with the RTDS<sup>®</sup> simulator are presented. Specifically, the simulation of the faults already discussed in this chapter have been implemented so to compare with the previous graphs.

Firstly, the three-phase fault results are illustrated in Figure 5.30. The results can be compared with what illustrated in Figure 5.8. It can be appreciated that the transient seems to be similar with the two simulators' results. However, the steady-state amplitude of both current and voltage results to be slightly different.

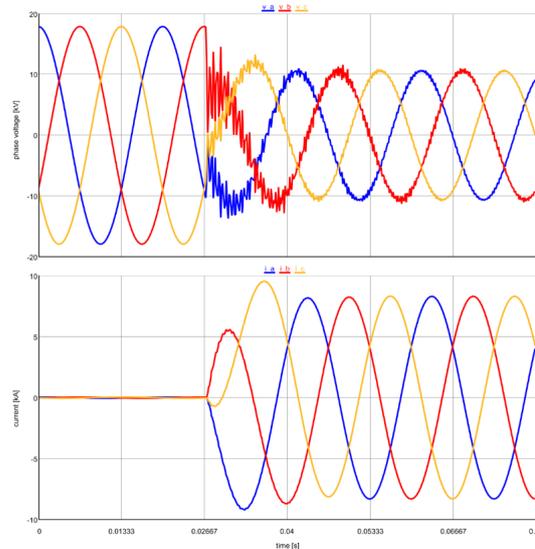


Figure 5.30: Three-phase fault: phase to ground voltage at the beginning of Magra feeder (Upper); line currents at the beginning of Magra feeder (Lower). Source: [1].

Secondly, the phase-to-phase fault results are illustrated in Figure 5.31. The results can be compared with what illustrated in Figure 5.13. As in the previous comparison, the transient results to be the same, although the steady-state values are slightly different in the two cases.

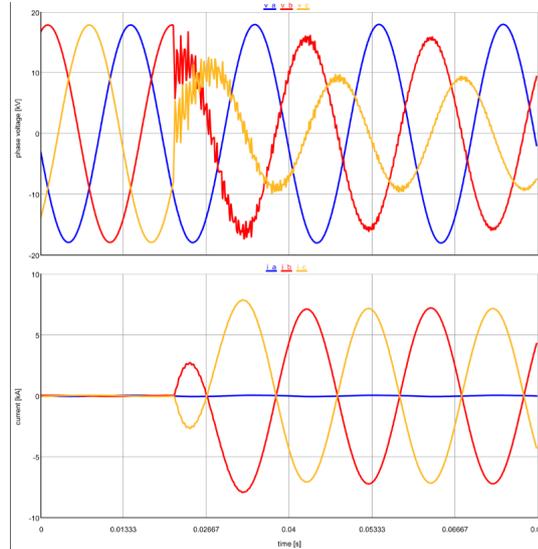


Figure 5.31: Line-to-line fault: phase to ground voltage at the beginning of Magra feeder (Upper); line currents at the beginning of Magra feeder (Lower). Source: [1].

# Chapter 6

## Voltage regulation with high PV penetration

In the previous chapters the critical role played by the OLTC transformer has been discussed related to two specific applications:

- Voltage regulation following the activation of a low voltage flexible resource;
- Voltage regulation in order to characterize the distributed resources placed in the low voltage grid.

The aim of this chapter is to evaluate the implementation of an OLTC transformer in the primary substation in order to regulate the voltage of a MV feeder which presents a high penetration of PV generation. The study is carried out with a similar approach of [22], although the OLTC transformer here implemented has a mechanical tap changer.

The analysis is conducted in the most critical condition, which can be found in the middle of a sunny day, with high production from the PV generators and a low power request from the loads. The studied feeder is the CHIARI feeder, which presents the highest rate of residential loads.

The surplus of PV generation causes a rise in the bus voltages, which need to be managed by using the tap changer.

The evaluation of this phenomenon is carried out with different levels of PV penetration. Also, the PV plant are distributed in different manners:

- in each bus of the MV feeder;
- at the beginning of the MV feeder;

- in the middle of the MV feeder;
- at the end of the MV feeder.

By doing so, the effectiveness of the centralized voltage regulation is tested for the different scenarios above listed.

The PV generators are simulated implementing the PV arrays, the three-phase VSC inverter (average model) with a trivial control consisting of two closed loops (DC-link voltage loop and a current loop), the MPPT algorithm and the step-up LV-MV transformer.

The analysed grid is the same already discussed in the previous chapters.

The details of how the simulation has been conducted are reported in Appendix E.

## 6.1 Model of the PV generator

The implemented model of the PV generator includes the array of PV panels, the three-phase VSC inverter (with the required control loops and the MPPT algorithm) and the step-up LV-MV transformer.

The model of the PV array is based on the single solar cell circuitual model [12], shown in Figure 6.1.

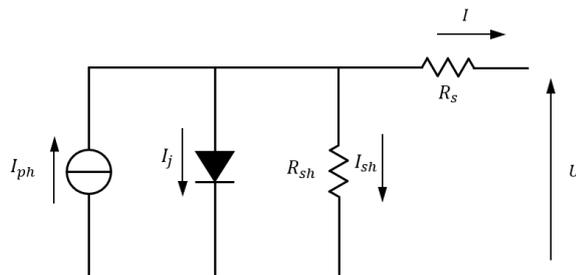


Figure 6.1: Circuitual model of a solar cell.

From the Figure 6.1, the parameter  $I_{ph}$  needs to be more deeply defined:

$$I_{ph} = kSG$$

The current generated by the solar cell is indeed a function of the *irradiance*  $G$  ( $W/m^2$ ). The implication of such correlation is later discussed. Other

important parameters are the *diode*  $D$ , which describes the recombination inside the solar cell, the *series resistance*  $R_s$ , of the two layers of silicon and the metal pins, and the *shunt resistance*  $R_{sh}$ , which emulates the leakage currents.

For the single solar cell the *V-I characteristic* is introduced. Along the *V-I* profile, the power as a function of the voltage is represented. The plot, shown in Figure 6.2, can be transposed for an entire PV array supposing the panels are exposed at the same environmental conditions (temperature and irradiance).

From the graph the correlation between the irradiance and the current is now clearly depicted in the plot: the higher the irradiance, the higher the generated current. Moreover, the point at which we obtain the maximum generated power is called *maximum power point (MPP)*. This is extremely important in order to exploit the solar panels in the best way. This topic will be discussed later in the control paragraph.

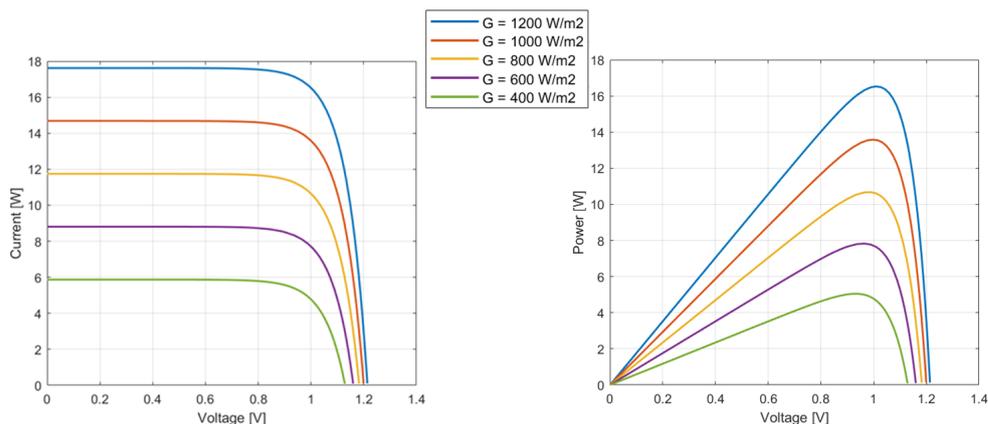


Figure 6.2: An example of Voltage-Current characteristic (left) and Power-Current characteristic (right) with different irradiances.

The solar array is connected to the utility grid via a three-phase VSC inverter, whose control is implemented through the *Maximum Power Point Tracker* (MPPT) algorithm, a DC-link closed loop and a current closed loop. Although the proposed control scheme for this application may result trivial, it is more than enough for the analysis here proposed. Specifically, an average model of a three-phase inverter is implemented. The reason is straight-forward: as we do not need to analyze high-frequency phenomenon, an average model is preferred as it requires a lower sample time, thus a lower risk of obtaining an overrun in a realtime simulation.

The scheme of the overall system, composed by the solar array, the inverter

and the step-up transformer is illustrated in Figure 6.3.

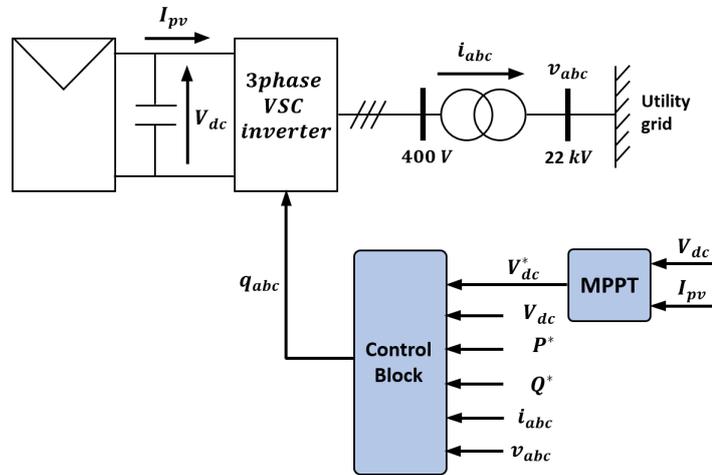


Figure 6.3: Scheme of the photovoltaic system.

The main characteristics of the system are listed in Table 6.1.

Table 6.1: Main parameter of the photovoltaic system.

Parameter	Value
Irradiance ( $W/m^2$ )	1000
Temperature ( $^{\circ}C$ )	30
$U_{MPP}(V)$	72.9
$I_{MPP}(A)$	5.69
$V_{dlink,nom}(V)$	480
$V_{ac,nom,rms}(V)$	400

The main components from the scheme of Figure 6.2 are now discussed more in detail.

Firstly, the MPPT algorithm is debated. In order to exploit the solar cells as much as possible, the MPPT is included in the control. Specifically, the Perturb & Observe (P&O) algorithm is applied. This algorithm is quite straightforward, which implies an easy application. However, its lack of robustness makes it not the mostly used algorithm.

The P&O process consists in varying the voltage of a solar array, computing the new value of power and comparing it with the previous value. Such variation is applied in order to reach a working condition as close as possible

to the peak value of power. The amplitude of voltage variation, imposed at each iteration of the algorithm, influences directly the accuracy of the final result. Specifically, there is a lack of accuracy in case of rapid variations of the environmental conditions and whenever the amplitude of the voltage variation results to be too large.

The proposed algorithm is illustrated with a flowchart in Figure 6.4. The Matlab code used to implement the P&O algorithm is reported in the Appendix E.

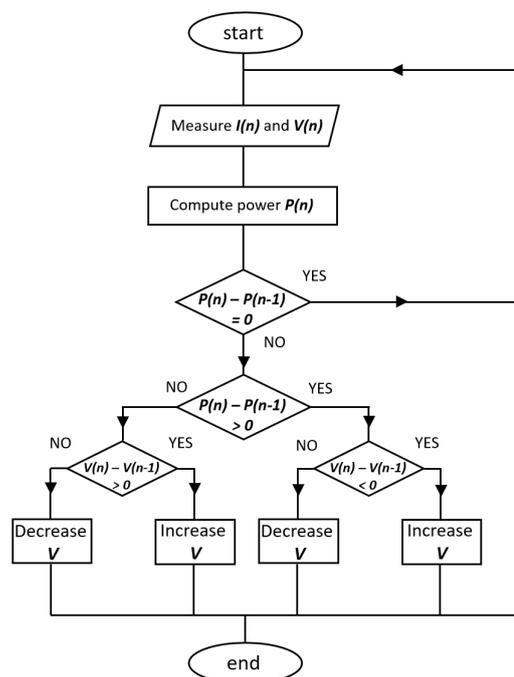


Figure 6.4: Flow chart of the Perturb & Observe algorithm.

The control of the three-phase inverter, with its two closed loops, is now discussed. Although the proposed control scheme is trivial and straightforward, it is more than enough for what is the aim of the study discussed in this chapter. The control scheme is illustrated in Figure 6.5.

The first component to be discussed is the *Phase Locked Loop (PLL)* block. Its function is to define the angle  $\theta$  from the voltage (in the  $\alpha\beta$  frames, obtained with an *abc-to- $\alpha\beta$*  transformation) of the utility grid. Such value of  $\theta$  will then be used to execute the *abc-to-dq* transformations and viceversa. Using the  $\theta$  angle from the utility grid enables the control to be synchronous with the grid itself, obtaining a so called *Vector Oriented Control (VOC)*. In

the Simulink environment, the *PLL* parameter block is used to obtain the  $\theta$  angle. A deeper description of this block is reported in Appendix E.

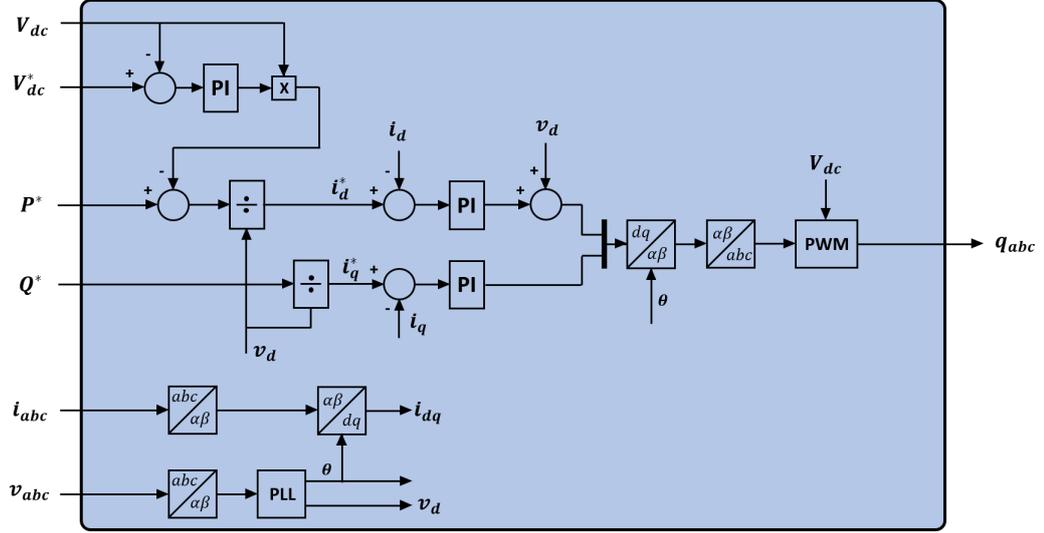


Figure 6.5: Scheme of the proposed control. Source: [5].

Please note that the schemes of Figure 6.3 and Figure 6.5 are represented as a real VSC three-phase inverter is included, thus having the *Pulse Width Modulation* (PWM) block as well. However, as explained in Appendix E this is not the case when the average model of the inverter is implemented.

The *abc-dq* transformation is now debated. The main goal of this transformation is to pass from a three-axis reference system to a two-perpendicular-axis reference system which are synchronous with the utility grid. To achieve that, two transformations are applied.

Firstly, the *abc- $\alpha\beta$*  is implemented. It does not require the knowledge of the  $\theta$  angle and it consists in defining a two-perpendicular-axis reference system,  $\alpha$  and  $\beta$  indeed, where the  $\alpha$  axis is aligned with the  $a$  axis. The mathematics behind it can be summarised in the following expression, where  $\bar{x}$  is a generic phasor:

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_o \end{bmatrix} = [T] \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (6.1)$$

A vectorial representation of this transformation is shown in Figure 6.6 (left).

The second step is the rotating transformation, which requires to know the  $\theta$  angle. The transformation is achieved with the following equation:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = [A(\theta)] \cdot \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (6.2)$$

A vectorial representation of the rotating transformation is shown in Figure 6.6 (right). In the Simulink environment, the  $abc$ - $dq$  transformation is obtained by using the *abc-to-dq0 Transformation* block. A deeper description of this block is reported in Appendix E.

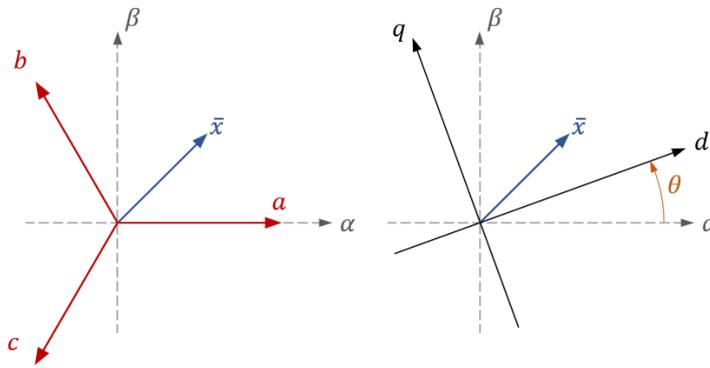


Figure 6.6: Representation of the  $(abc)$ ,  $(\alpha\beta)$  and  $(dq)$  frames.

The  $dq$  reference frame is applied in order to implement a VOC, in which the  $d$ -axis alignment is defined by the *Point of Common Coupling* (PCC) vector. Specifically, the aim of this control is to impose the  $q$ -axis component of the voltage equal to zero, as depicted in Figure 6.7.

$$\begin{cases} v_d = \bar{v}_{PCC,\alpha\beta} \\ v_q = 0 \end{cases}$$

By imposing  $v_q = 0$ , the active and the reactive power are computed as follows:

$$p = v_d \cdot i_d$$

$$q = v_d \cdot i_q$$

By these equations, the reference values of  $i_{dq}$  can be defined by the desired value of the active and reactive power.

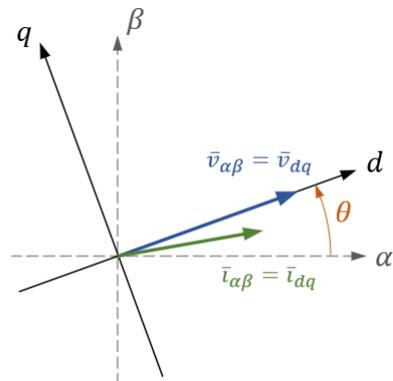


Figure 6.7: Vectorial representation of the PCC voltage with the synchronous reference frame ( $dq$ ).

## 6.2 MV feeder and results

The feeder to be used for the analysis is the CHIARI feeder, as only one line is powered by the HV-MV transformer. Unlike the previous studies, it is now considered for the CHIARI feeder a transformer provided with a OLTC system. The model is the same already used for the previous chapters, with a different rated power (63 MVA instead of 55 MVA).

Firstly, the scenario with the PV plants located at each bus of the feeder is considered. A schematic representation of the feeder, with the loads and the distributed generation (15 PV power systems), is reported in Figure 6.8.

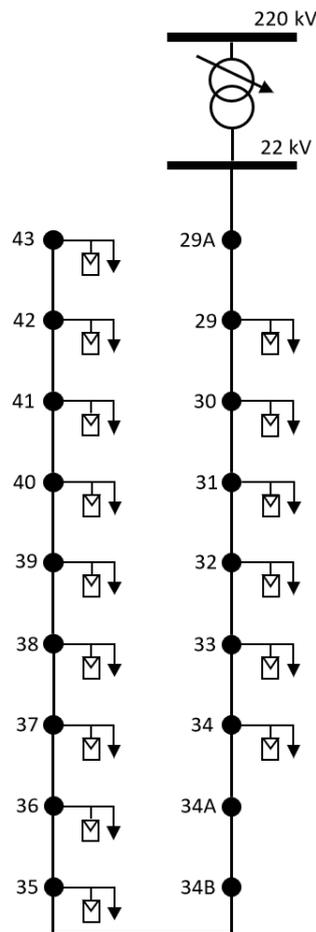


Figure 6.8: Schematic representation of the CHIARI feeder.

The set of MV loads, which are simulated with a ZIP model, have been set

with a rated power equal to half of the values used for the previous evaluations. By doing so, the desired condition (midday with sunny condition and low load) is emulated. Such condition brings to rising values of bus voltage along the feeder, with the possibility to obtain unacceptable overvoltages ( $> 1.1$  pu). To avoid this scenario, the voltage regulation is required.

Many viable options can be considered. They can be divided between local and centralized approaches. A first option would be injecting reactive power in the grid from the PV systems. However, in this study such approach is not used. Indeed, the desired injected reactive power is set to zero, using the control scheme shown in Figure 6.5. On the contrary, the reference active power is set to 1 pu. The approach to not inject reactive power has also been used in [22].

As considered in [22], the voltage regulation is carried out using the tap changer of the HV-MV transformer. Although the OLTC regulation parameters are the same already used in Chapter 2, the tap position is manually varied.

The regulation has been carried out in different scenarios with different PV penetration in the MV feeder. The five considered cases, with the relative PV penetrations, are listed in Table 6.2. The PV penetration is computed as the ratio between the overall installed PV power in the feeder and the rated power of the HV-MV transformer (63 MVA).

Table 6.2: Nominal power of each PV system and PV penetration in the feeder.

	Nominal power of each PV plant	PV penetration in the feeder
Case 1	2.50 MWp	59 %
Case 2	1.60 MWp	38 %
Case 3	1.25 MWp	30 %
Case 4	1.00 MWp	24 %
Case 5	0.63 MWp	15 %

Along the five cases above listed, a scenario with no PV installed is also shown to verify the voltage profile along the feeder. The overall set of results is illustrated in Figure 6.9.

Firstly, the profile in the no-PV scenario is quite expected, as a low load case implies a negligible voltage drop along the feeder.

A second observation is that only the Case 1 brings to an overvoltage situation ( $> 1.1$  pu), while the others do not.

The first scenario, which results to be the most critical, is the only one where four variations of the tap changer are required. By doing so, the voltage is reduced from 1.101 pu to 1.059 pu in the most critical bus.

The second and third scenarios required three variations of the tap position (from position 0 up to position 3). The regulation brings the higher value to 1.03 pu (Case 2) and 1.02 pu (Case 3), resulting far closer to the nominal condition. On the other hand, the lowest value obtained at the first bus results to be equal to 0.97 pu, thus not a critical condition.

Finally, the last two scenarios (Case 4 and Case 5) simply require two variations of the tap position, obtaining as a critical value to 1.019 pu (Case 4) and 0.999 pu (Case 5). On the other hand, the lowest value obtained at the first bus results to be equal to 0.984 pu (Case 4) and 0.98 pu (Case 5).

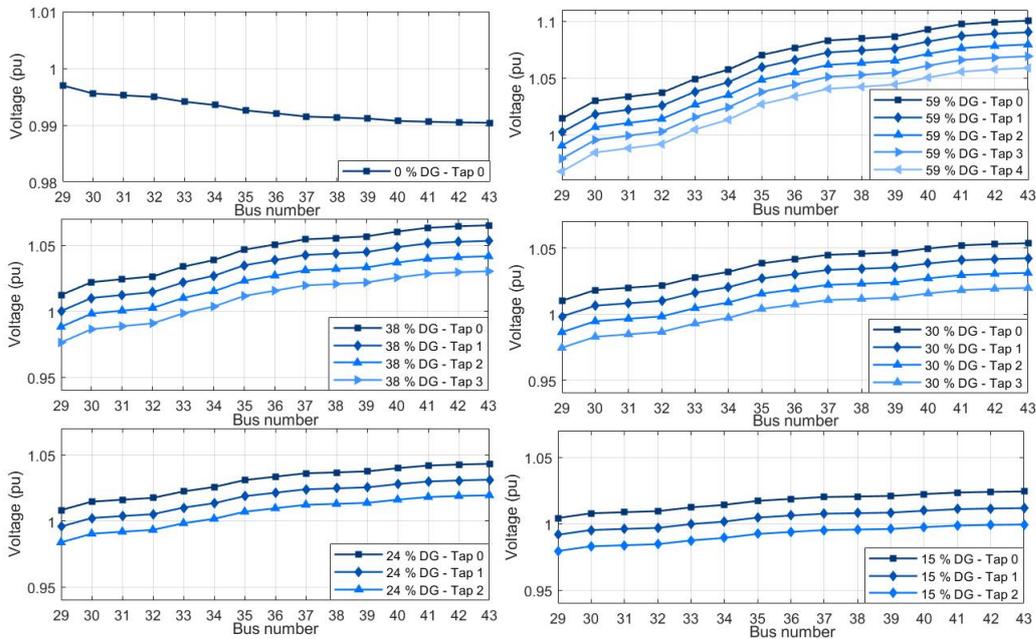


Figure 6.9: Phase-to-ground voltage (pu) along the CHERI feeder: 0 % DG penetration (upper left); 59 % DG penetration (upper right); 38 % DG penetration (center left); 30 % DG penetration (center right); 24 % DG penetration (lower left); 15 % DG penetration (lower right).

From the results shown in Figure 6.9, the centralized voltage regulation appears to be effective in preventing overvoltages along the feeder. However, the scenario with a homogeneous location along the feeder of the PV plants is not the only possible case.

The PV plants are now positioned only on specific buses, keeping the same PV penetration of Case 1, as shown in Table 6.2. Specifically, the buses considered are listed in Table 6.3.

Table 6.3: List of cases considered.

	MV bus with PV plant
Case 1.1	29, 30, 31
Case 1.2	35, 36, 37
Case 1.3	41, 42, 43

The results are reported in Figure 6.10.

It is clear from Case 1.3 (Figure 6.10 (lower)) how the only centralized voltage regulation will not be sufficient to prevent overvoltages along the feeder, as the last three MV buses still present an overvoltage after four variation by the HV-MV tap changer.

A local voltage regulation, consisting in injecting reactive power by the PV plants, would be required. Moreover, bringing every bus' voltage below the 1.1 pu may cause a voltage sag ( $< 0.9pu$ ) in the buses closer to the HV-MV transformer.

On the other hand, Case 1.1 (Figure 6.10 (upper)) and Case 1.2 (Figure 6.10 (center)) seem not to have any critical aspect.

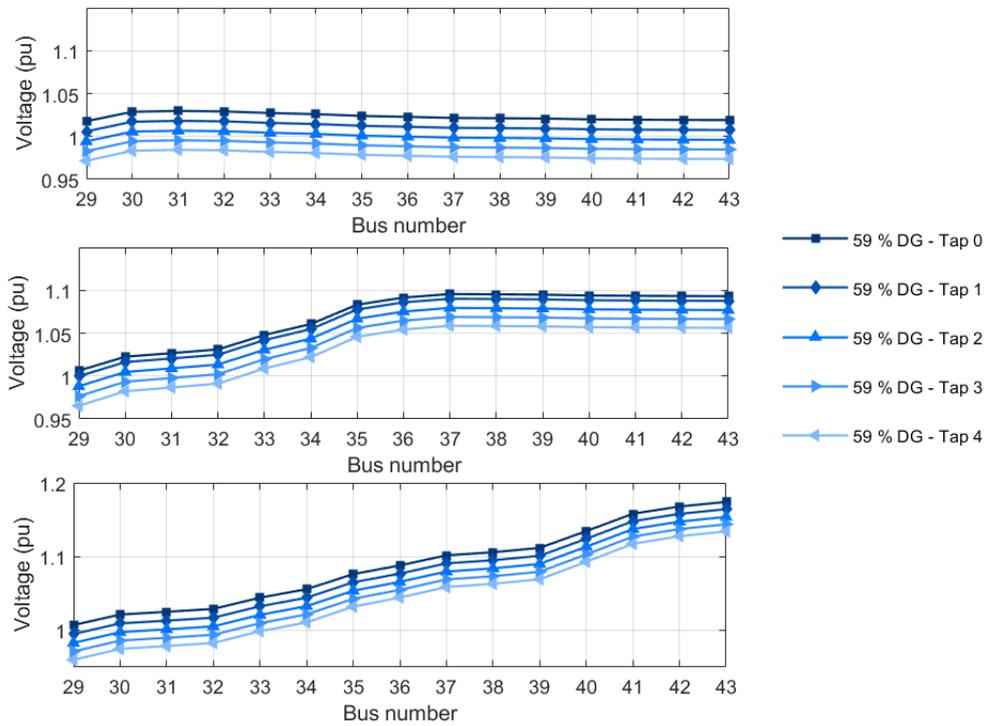


Figure 6.10: Phase-to-ground voltage (pu) along the CHIARI feeder: 59 % DG penetration with Case 1.1 (upper); 59 % DG penetration with Case 1.2 (center); 59 % DG penetration with Case 1.3 (lower).

# Chapter 7

## Conclusion

The Realtime Remote co-simulations realized within the *Living Grid* project show the validity of a geographically distributed research facility. Specifically, the systemic analysis of the latency between Turin and Bari indicates that the communication is not significantly impacted by the size of the exchanged data packet.

Secondly, the integration of a Power Hardware In the Loop shows the feasibility to investigate problems related to the distribution network, such as the dispatchment of low voltage flexible sources.

Finally, the robustness of the Remote Power Hardware In the Loop co-simulation is verified by looking at the conditions that bring the simulation to instability.

The proposed method to estimate distributed sources connected to the LV grid looks to be valid for the intended parameterization. The validity is important if the simplicity of the method is considered. Moreover, a good estimation is achieved still with a bad knowledge of the  $k_p$  parameter.

The Realtime simulation of the analysed distribution network has been then implemented to analyse the MV faults and verify their coherence with the theory.

Finally, the same model has been simulated to verify the feasibility of a voltage regulation, with a significant photovoltaic penetration in the MV feeder, by only using the OLTC situated in the HV-MV transformer. The regulation is effective even with no reactive injection by the PV plants, as long as the PV plants are evenly located on the MV buses. On the other hand, a concentration of PV plants in the terminal part of the feeder shows how the only centralized control is not enough to avoid overvoltages.

# Appendix A

The Appendix A describes in a deeper way the Remote Power Hardware In the Loop (R-PHIL) realtime simulations, which have been already discussed in Chapter 2. Specifically, the Appendix A illustrates the most critical settings for the models implemented in Turin. Moreover, the Matlab scripts implemented to whether analyze the datas or initialize the models are illustrated.

## **A.0.1 Analisis of the communication latency: settings and practical aspects**

This section describes the most critical aspects and settings of the model implemented in Turin to test the communication latency between Turin and Bari's laboratories.

Firstly, the *SC\_Console* subsystem is illustrated in Figure A.1. Among the variables to be monitored we have the status and the errors related to the communication between Turin and Bari, and the datas exchanged between the two laboratories. Furthermore, it can be seen a manual switch whose discrete possible values are 0 and 2. It will be used during the simulation to start sending datas from Turin and saving the desired variables at the same time.

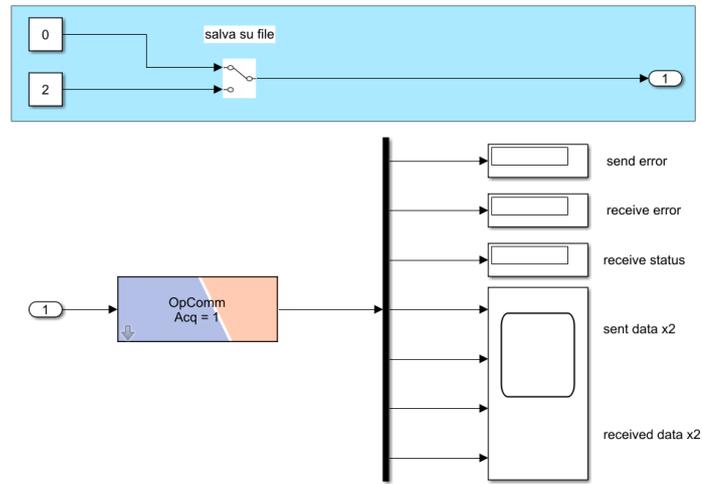


Figure A.1: SC\_Console subsystem.

The choice of the two values, 0 and 2, is related to the *OpTrigger* block and its settings. Such block, shown in Figure A.2, is located in the *SM\_Master* subsystem. Beside this block, which has been assigned to the *Acquisition Group 26*, there is a mechanism to saturate the signal to be sent at a value around 10000 (10006, to be precise). Also in Figure A.2 the *OpAsyncSend* block is shown, which has as input the signal to send to Bari. However, the signal is sent only when the state *Data ready* is equal to 1, which occurs when the manual switch is moved to 2 and the pulse generator gives a value equal to 1.

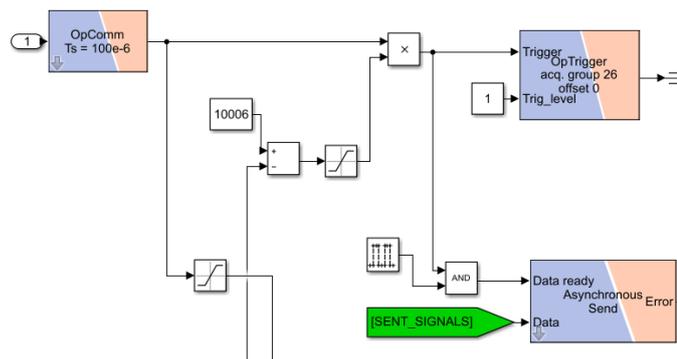


Figure A.2: Portion of the SM\_Master.

The aforementioned pulse generator is used to impose the *Sample Time* of the exchange of data. The *Sample Time* is, along with the *Data Packet size*, the

parameter with which the communication latency is evaluated. The sample time of the simulation is set by the *Period (number of samples)* parameter, as shown in Figure A.3.

Considering that the *Time step* of the model used in Turin is equal to 100 microseconds, we can obtain that:

- for a *sample time* = 1 ms, we set *Period* = 10 number of samples;
- for a *sample time* = 5 ms, we set *Period* = 50 number of samples;
- for a *sample time* = 10 ms, we set *Period* = 100 number of samples;
- for a *sample time* = 20 ms, we set *Period* = 200 number of samples;

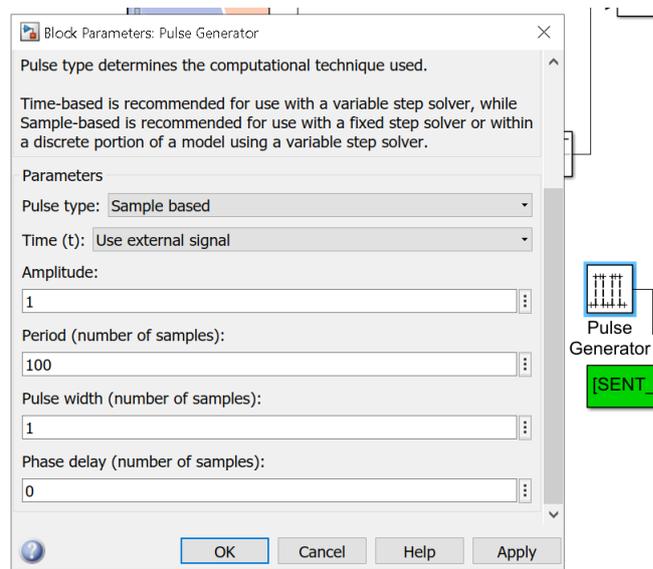


Figure A.3: Pulse generator block's settings.

Please note that the value set in the *Period* variable has to be inserted also in the *Delay* block, which is used to create the signal to send to Bari. The value has to be inserted in the parameter *Delay length*, as shown in Figure A.4

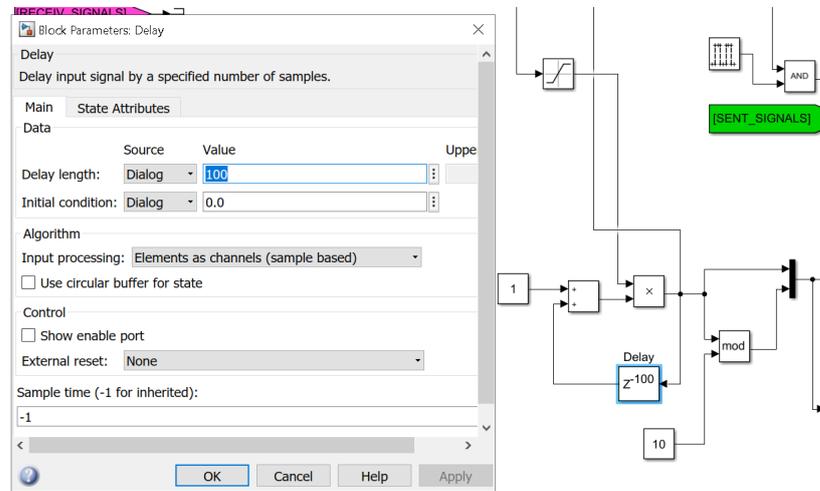


Figure A.4: Delay block's settings.

The other parameter used to characterize the latency is the *Data Packet size*, which has been set at different values during the various simulations (2, 6, 12, 24, 48 and 96). The value is inserted in the *Signal specification* block, which has as the input the signal received by Bari and passed through the *OpAsyncRec* block, as shown in Figure A.5.

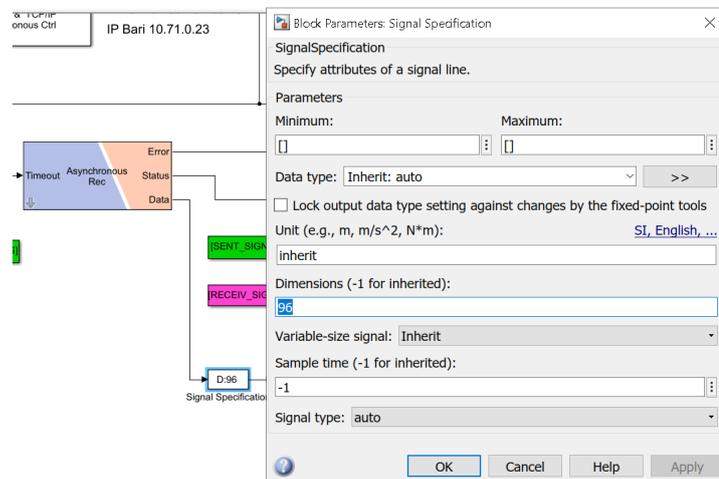


Figure A.5: Signal specification block's settings.

Because of the high number in exchanged data, specifically when setting the data packet size at 96, it is necessary to vary some parameters in the *AsyncIP.c* file. The variables *MAXSENDSIZE* and *MAXRECSIZE* are incremented from 64 to 128, in order to enable the 96-sized data packets to be

correctly exchanged between Turin and Bari. The abovedescribed variation is shown in Figure A.6

```
// This defines the maximum number of signals (doubles) that can be sent
// or received by any individual Send or Recv block in the model. This
// only applies to the "model <-> asynchronous process" communication.
#define MAXSENDSIZE 128
#define MAXRECVSIZE 128

// Set the stack size of each thread.
#define STACKSIZE 4096

// Use the smallest possible memory footprint to store the data_out
// and data_in structures.
#define...
```

Figure A.6: AsyncIP proposed variation.

In order to correctly execute such remote realtime co-simulation, a series of ASCII files are required. Among those there is the already mentioned *AsyncIP.c* file. Following the *Build* operation of the model, the required files can be added by the *Files* window of RT-LAB<sup>©</sup> as shown in Figure A.7.

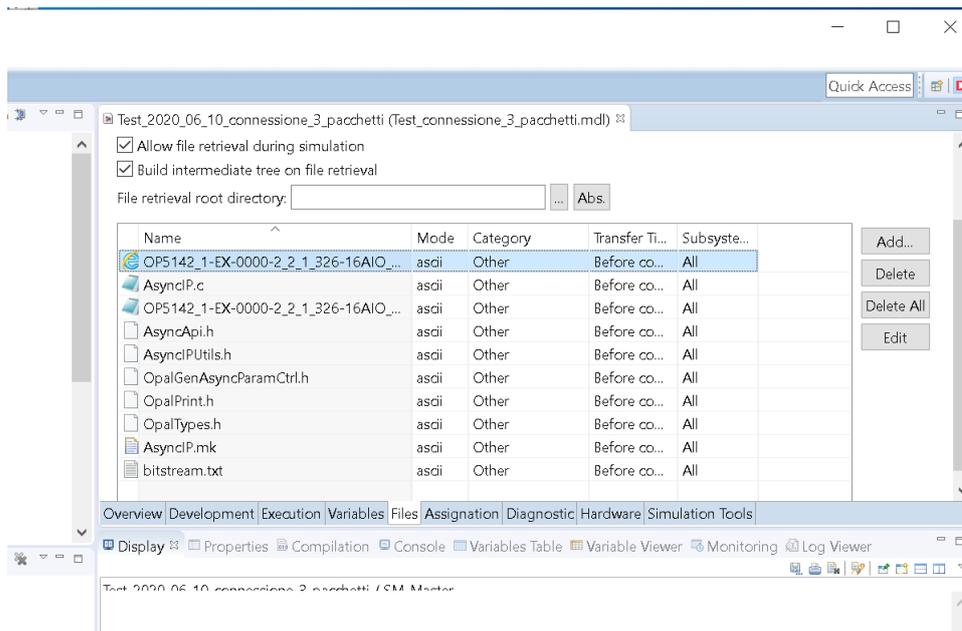


Figure A.7: Loading of required files from RT-LAB<sup>©</sup>.

The data obtained from the simulations have been analysed so to obtain the results discussed in Chapter 2.

The algorithm, which has been used for the dataset of each simulation, has been constructed to execute the following tasks:

- distinguishing datas lost in the communication from data simply affected by heavy latency;
- counting the lost data packets;
- computing the latency of each data packet and analyzing them so to obtain the mean value, the standard deviation and the boxplot;

The Matlab code is here shown.

```
function [twoWaysDelay, twoWaysDelayAVG, stdDelayToTo,
nOutlierToTo, oneWaysDelay, oneWaysDelayAVG, stdDelayToBa,
nOutlierToBa, packSentCnt, packNotSentCnt, packNotSentpercent] =
syncroToBaRitardi(Length, timeTo, stairToSent, stairToReceived,
timeBa, stairBaReceived)

% ----- %
% Calcolo dei ritardo Andata/Ritorno
% Ritardo calcolato con il primo valore di ogni pacchetto
% (lo "spigolo" del gradino)
posCorner = zeros(1,Length);
deltaTimeBaTo = zeros(1,Length);
packSentCnt = 0;
packNotSentCnt = 0;
% posCorner() = 1 -> è primo valore del "gradino"
% usato per calcolo del delay
% posCorner() = 0
% é valore nel "piano" del gradino
twoWaysDelay = zeros(1,Length); % vettore con il ritardo A/R per
% ogni pacchetto correttamente ricevuto
oneWaysDelay = zeros(1,Length); % vettore con il ritardo To-Ba per
% ogni pacchetto correttamente ricevuto

% ricerca "spigoli" nei dati PoliTo-sent

for k1=2:1:Length

    if (k1 == 2)
        posCorner(1) = 1;
    end

    if (k1 > 2)
```

```

        if (stairToSent(k1) == 1+stairToSent(k1-1))
        % è il primo valore del gradino
            posCorner(k1) = 1;
        else
            posCorner(k1) = 0;
        end
    end
end

% Ricerca dei valori corrispettivo nei dati PoliTo-received
for k1=1:1:Length
    if(posCorner(k1)) % il valore è uno "spigolo"
        valueToFind = stairToSent(k1);
        time1 = timeTo(k1); % istante del valore di PoliTo-sent
        posFound = find(stairToReceived(:)== valueToFind);
        if(isempty(posFound)) % pacchetto non ricevuto
            twoWaysDelay(k1) = 0; % ritardo non viene calcolato
        else % pacchetto ricevuto
            posFoundCorner = posFound(1);
            time2 = timeTo(posFoundCorner);
            twoWaysDelay(k1) = time2 - time1; % ritardo calcolato
        end
    end
end

twoWaysDelayNotZero=twoWaysDelay(twoWaysDelay~=0);
% tengo solo i valori !=0
lengthNotZeroToTo = length(twoWaysDelayNotZero);
% conto i valori !=0
twoWaysDelayAVG = sum(twoWaysDelayNotZero)/lengthNotZeroToTo;
% valore medio di ritardo A/R
oneWayDelayAVGnottrue = 0.5*twoWaysDelayAVG;
% valore medio di ritardo one way

stdDelayToTo = std(twoWaysDelayNotZero);
delayToTooutlier = isoutlier(twoWaysDelayNotZero,'quartiles');
delayToTooutlier = delayToTooutlier(delayToTooutlier~=0);
nOutlierToTo = length(delayToTooutlier);

for k1=1:1:Length
    if(posCorner(k1)) % il valore è uno "spigolo"

```

```

        valueToFind = stairToSent(k1);
        time1 = timeTo(k1);
        % istante del valore di PoliTo-sent
        posFoundBa = find(stairBaReceived(:) == valueToFind);
        if isempty(posFoundBa)
            % pacchetto non ricevuto
            deltaTimeBaTo(k1) = 0;
            % ritardo non viene calcolato
        else % pacchetto ricevuto
            posFoundBaCorner = posFoundBa(1);
            time2 = timeBa(posFoundBaCorner);
            deltaTimeBaTo(k1) = time2 - time1;
            % ritardo calcolato
        end
    end
end

notZero = deltaTimeBaTo(deltaTimeBaTo~=0);
% tengo solo i valori !=0
lengthNotZero = length(notZero);
% conto i valori !=0
deltaTimeBaToAVG = sum(deltaTimeBaTo)/lengthNotZero;
% valore medio di ritardo A/R

% Sincronizzazione
if(deltaTimeBaToAVG > 0)
    newPoliBaTime = timeBa - deltaTimeBaToAVG + oneWayDelayAVGnottrue;
elseif(deltaTimeBaToAVG < 0)
    newPoliBaTime = timeBa + deltaTimeBaToAVG + oneWayDelayAVGnottrue;
elseif(deltaTimeBaToAVG == 0)
    newPoliBaTime = timeBa + oneWayDelayAVGnottrue;
end

%   packNotReceived = 0;

% Calcolo dei ritardi To-Ba post sincronizzazione
for k1=1:1:Length
    if(posCorner(k1)) % il valore è uno "spigolo"
        valueToFind = stairToSent(k1);
        time1 = timeTo(k1); % istante del valore di PoliTo-sent
        posFound = find(stairBaReceived(:) == valueToFind);
    end
end

```

```

        if(isempty(posFound)) % pacchetto non ricevuto
            oneWaysDelay(k1) = 0; % ritardo non viene calcolato
        else % pacchetto ricevuto
            posFoundCorner = posFound(1);
            time2 = newPoliBaTime(posFoundCorner);
            oneWaysDelay(k1) = time2 - time1; % ritardo calcolato
        end
    end
end

oneWaysDelayNotZero=oneWaysDelay(oneWaysDelay~=0);
% tengo solo i valori !=0
lengthNotZeroToBa = length(oneWaysDelayNotZero);
% conto i valori !=0
oneWaysDelayAVG = sum(oneWaysDelayNotZero)/lengthNotZeroToBa;
% valore medio di ritardo A/R

stdDelayToBa = std(oneWaysDelayNotZero);
delayToBaoutlier = isoutlier(oneWaysDelayNotZero,'quartiles');
delayToBaoutlier = delayToBaoutlier(delayToBaoutlier~=0);
nOutlierToBa = length(delayToBaoutlier);

% Contare pacchetti spediti correttamente To-Ba
for k1=1:1:Length
    if(posCorner(k1))
        valueToFind2 = stairToSent(k1);
        posFound2 = find(stairBaReceived(:) == valueToFind2);
        if(isempty(posFound2) || length(posFound2)==1)
            packNotSentCnt = packNotSentCnt + 1;
        elseif(length(posFound2)>1)
            packSentCnt = packSentCnt + 1;
        end
    end
end
packNotSentpercent = 100*packNotSentCnt/(packSentCnt+packNotSentCnt);
end

```

### A.0.2 R-PHIL RT co-simulation with one feeder and the OLTC: settings and practical aspects

This section describes more accurately the model implemented in Turin and used for a first R-PHIL realtime co-simulation. The model includes a 22 kV distribution feeder, an OLTC transformer and the connexion in the loop with the R-L-C load bank located in Bari's facility.

Firstly, the *SC\_Console* is shown in Figure A.8. The figure highlights the variable to be monitored besides a manual switch used to save datas on file. The reasoning behind this switch is the same of what has been discussed in the previous section (i.e., in order to start saving a signal on file the switch is moved from 0 to 2. The save-on-file procedure is stopped once the switch is moved back on the 0 value).

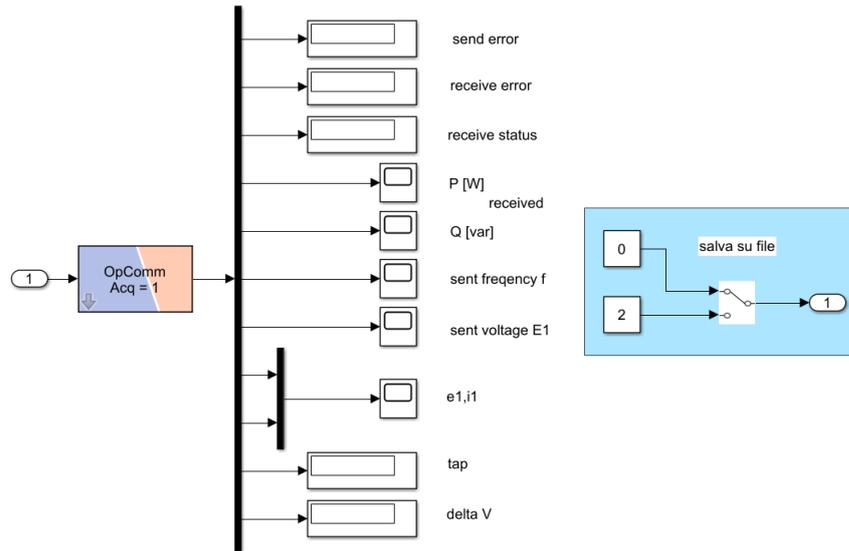


Figure A.8: *SC\_Console* subsystem of the model.

In the *SM\_Master* the blocks used to exchange datas with Bari are the same of the previous test. However, the pulse generator is set differently. The period is now set to 6 number of samples, as shown in Figure A.9. Considering that the time step of this model is equal to 500 microseconds, with such setting of the pulse generator a sample time of 3 ms is obtained.

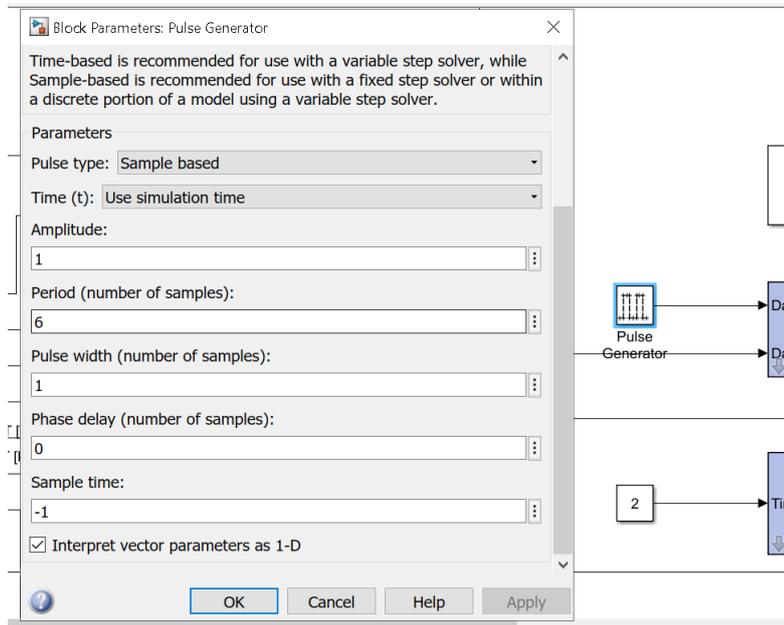


Figure A.9: Pulse generator block's settings.

Inside the distribution feeder subsystem there are a couple of aspect to highlight.

Firstly, the active and reactive power of the Bari's microgrid are amplified with a specific gain after receiving them from the loopback. By doing so we are simulating a higher contribution of the microgrid itself in the distribution feeder. For this specific application, a gain equal to 250 is imposed. Moreover, both active and reactive power pass through a low-pass filter to make the co-simulation more stable. Both the gains and the low-pass filters are depicted in Figure A.10.

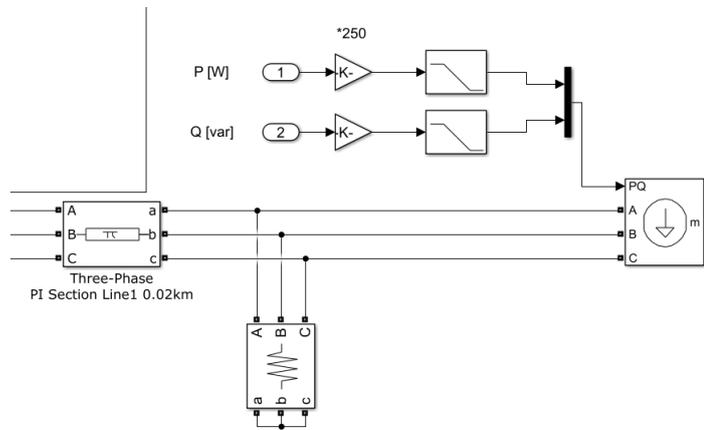


Figure A.10: Amplification of the active and reactive power of the microgrid.

Secondly, the OLTC parameter are set as shown in Figure A.11. The choice of these value is linked to the specific application and to the expected contribution of the microgrid in terms of active and reactive power. Moreover, the values are chosen to avoid the tap to change without a variation on the MV network.

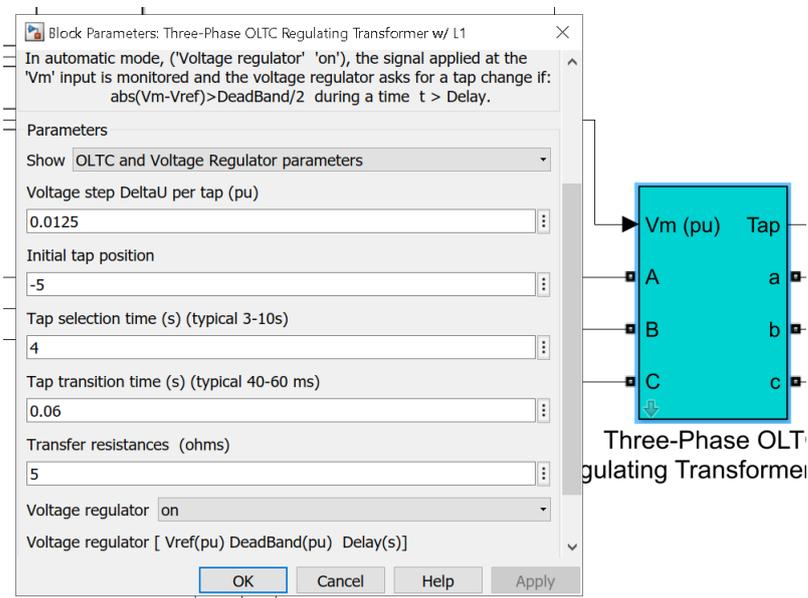


Figure A.11: Settings of the OLTC transformer.

The model of the distribution feeder is shown in Figure A.12, Figure A.13 and Figure A.14.

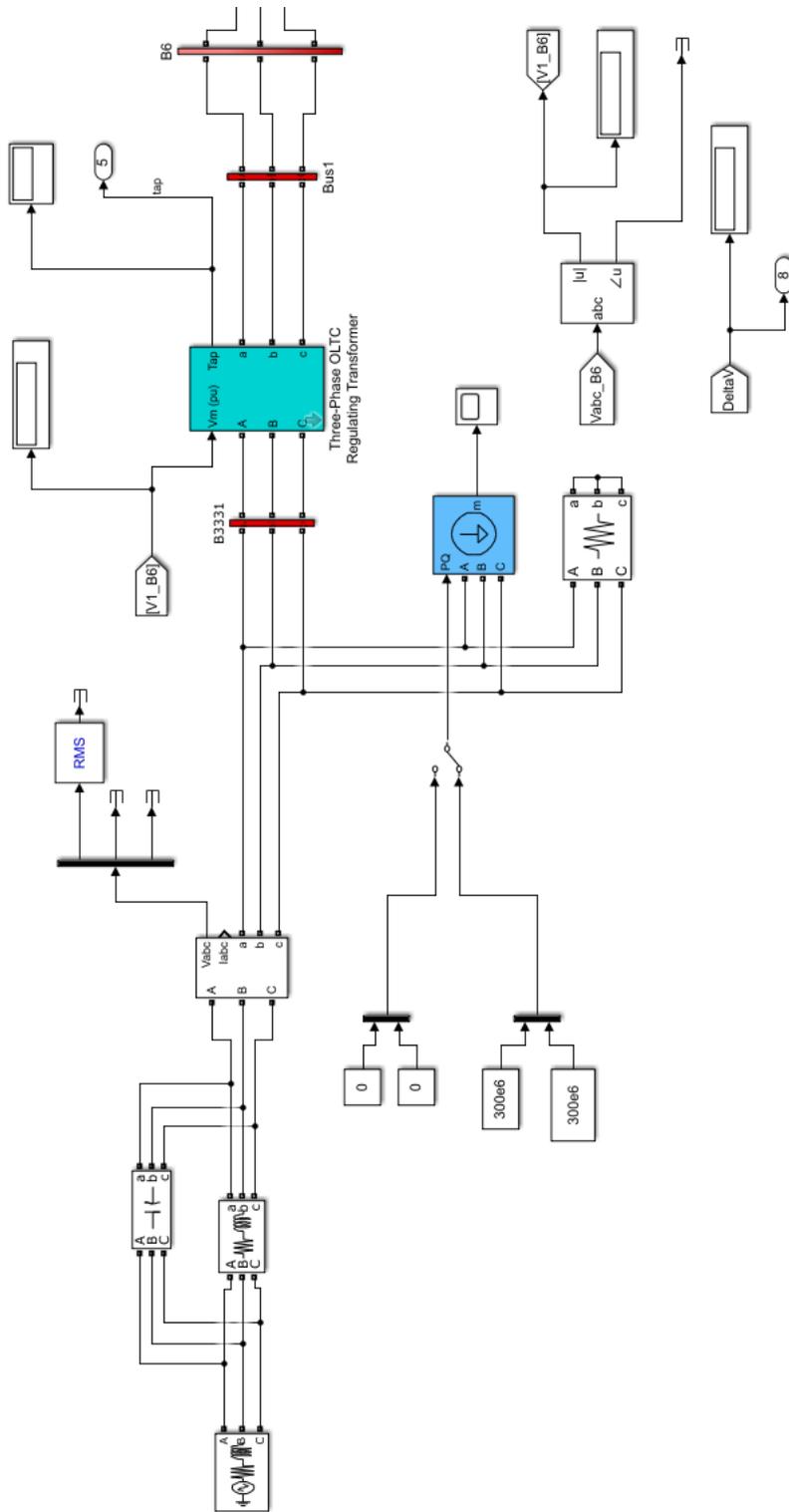


Figure A.12: Simulink model: HV network and OLTC transformer Simulink model.

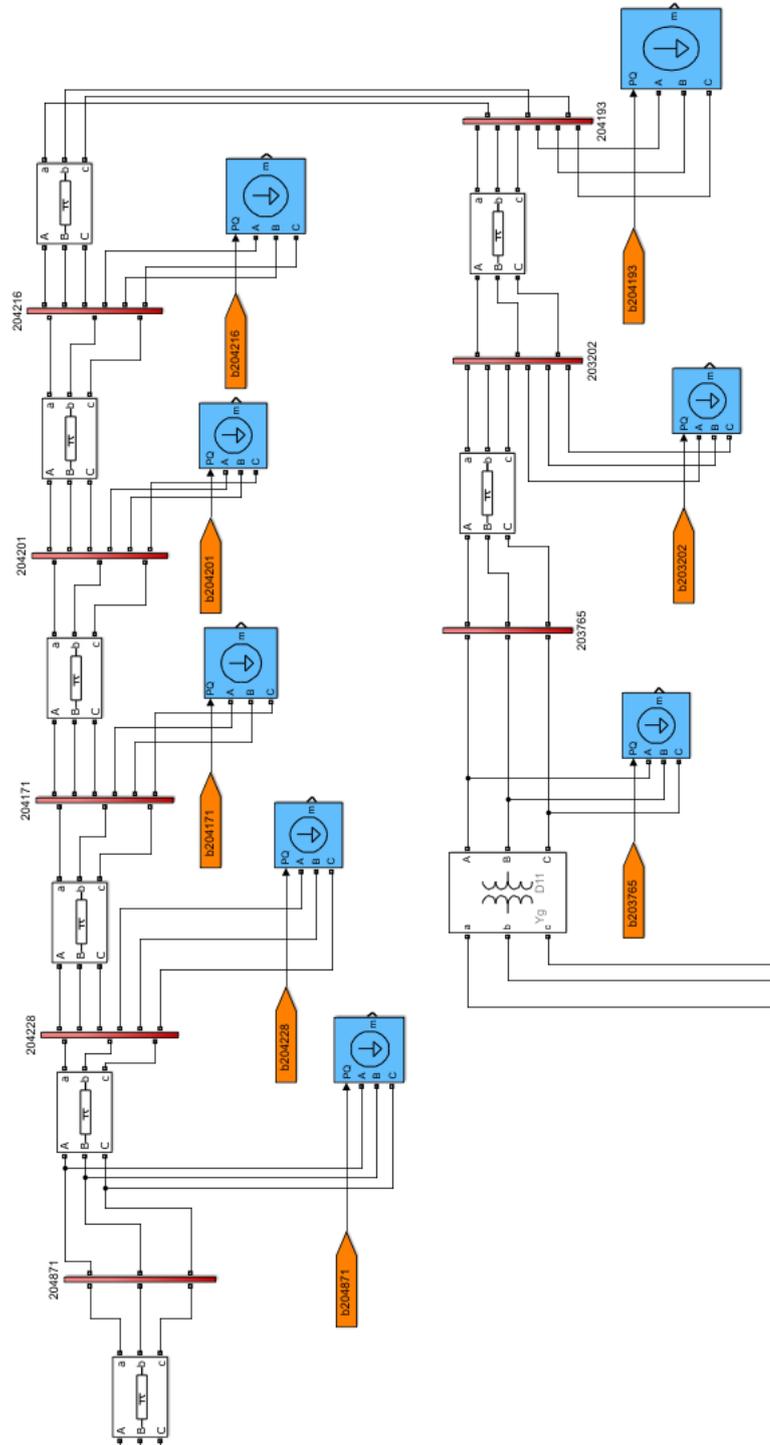


Figure A.13: Simulink model: MV network Simulink model.

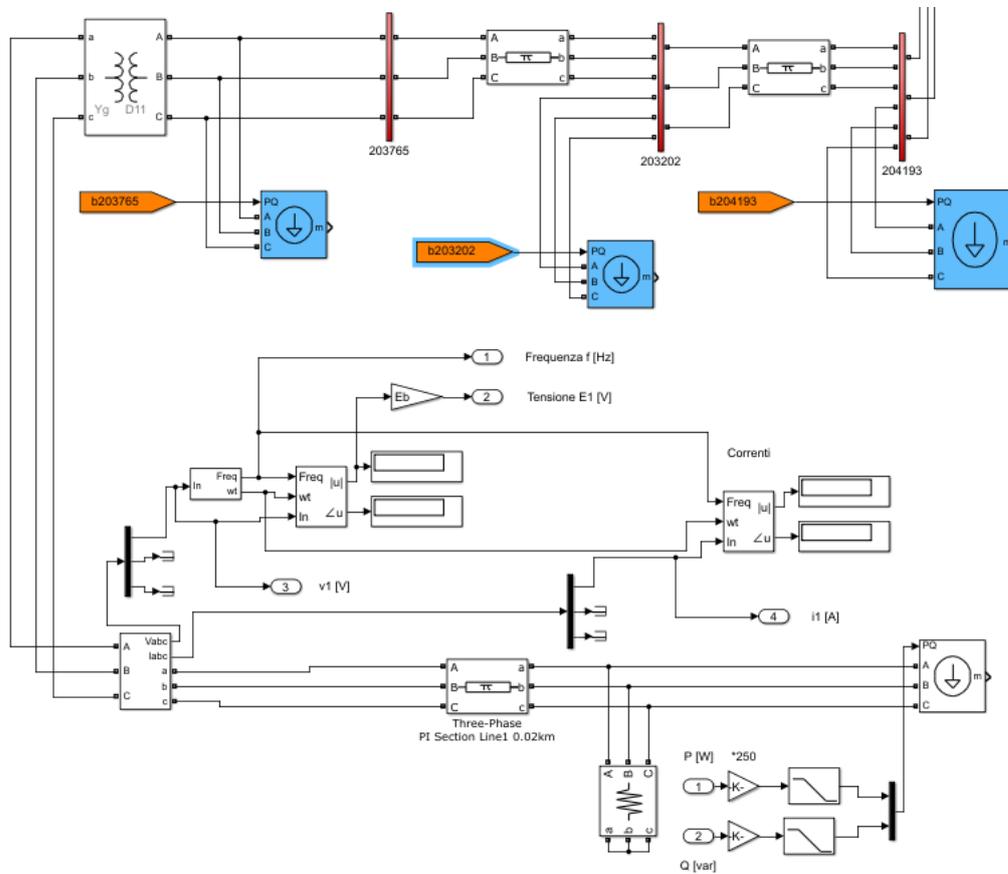


Figure A.14: Simulink model: LV network Simulink model.

Finally, the Matlab code used to initialize the Simulink model is attached.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f=50; %Hz, the nominal frequency of residential networks
w=2*pi*f;

% Zone 1: High-Voltage network
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V_hi = 220e3;
% VLL_rms for the High-Voltage zone
%(equivalent transmission grid)
SC_current = 30e3;
% Rated short-circuit current of the
% equivalent network impedance
X_hi = V_hi/(sqrt(3)*SC_current);

```

```

% Equivalent reactive impedance value
%according to the SC-current (base impedance)

X_over_R = 20; % X/R ratio
L_hi = X_hi/(2*pi*f); % Equivalent inductance
R_hi = X_hi / X_over_R; % Equivalent resistance value
C_hi1 = 1/(2*pi*f*X_hi*1e5);
% Small parallel capacitor of network
%impedance (High impedance at fundamental freq)
C_hi=[1,1,1;1,1,C_hi1;1,1,1];

%% HV/MV Trans Param
% 63MVA Trans:
MVA_1 = 63e+6;
Pcc_1 = 180.77e+3;
Vcc_1 = 15.93;%
Po_1 =41.7e+3;
Io_1 = 0.07;%

Zcc_1 = Vcc_1/100;
Pcc_1 = Pcc_1 / MVA_1; % per unit
R_1 = Pcc_1;
X_1 = sqrt(Zcc_1^2-R_1^2);
R1_1 = R_1/2;
R2_1 = R_1/2;
L1_1 = X_1/2;
L2_1 = L1_1;

Zm_1 = Io_1/100;
Po_1 = Po_1 / MVA_1;
Gm_1 = Po_1;
Ym_1 = Io_1/100;
Bm_1 = sqrt(Ym_1^2-Gm_1^2);
Rm_1 = Gm_1/Ym_1^2;
Lm_1 = Bm_1/Ym_1^2;

% 55MVA Trans:
MVA_2 = 55e+6;
Pcc_2 = 182.84e+3;
Vcc_2 = 11.46;%
Po_2 =74.93e+3;

```

```

Io_2 = 0.63;%

Zcc_2 = Vcc_2/100;
Pcc_2 = Pcc_2 / MVA_2; % per unit
R_2 = Pcc_2;
X_2 = sqrt(Zcc_2^2-R_2^2);
R1_2 = R_2/2;
R2_2 = R_2/2;
L1_2 = X_2/2;
L2_2= L1_2;

Zm_2 = Io_2/100;
Po_2 = Po_2 / MVA_2;
Gm_2 = Po_2;
Ym_2 = Io_2/100;
Bm_2 = sqrt(Ym_2^2-Gm_2^2);
Rm_2 = Gm_2/Ym_2^2;
Lm_2 = Bm_2/Ym_2^2;

%%
dLine_hi = 20; % (km) Transmission line distance

%% 3-phase Distribution Network
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S_Xfo_1 = 63e6; % Rated power of the step-down transformer (VA)
S_Xfo_2 = 55e6; % Rated power of the step-down transformer (VA)
V_dis=22000; %V,The nominal voltage of the distribution network

% Neutral Impedance calculation
Rneutral_pu = 0.9; % percentage of neutral impedance
Rneutral_SI = Rneutral_pu * V_dis^2/S_Xfo_1; % Neutral impedance value

R_dl=0.313; %ohms per kilometre, the resistance of 6.6kV distribution
%line according to Japanese standard
X_dl=0.377; %ohms per kilometre, the impedance of 6.6kV distribution
%line according to Japanese standard
L_dl=X_dl/w; %6.6kv line inductance
dLine_dis = 10; % (km) Distribution line distance

```

```
% For HIL Interface
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V_base=100; %Volts(V), the base rms value phase to ground voltage

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% LINE FIAT
% StartNode,EndNode,R1,R0,L1,L0,C1,C0,Length, Imax[A]
PIS_Param_Fiat = [
204871 204228 0.144000164 0.216000246 0.000327859
0.000491788 4.16999E-07 4.16999E-07 2.801 340;

204871 204228 0.144000164 0.216000246 0.000327859
0.000491788 4.16999E-07 4.16999E-07 0.233 340;

204228 204171 0.144000164 0.216000246 0.000327859
0.000491788 4.16999E-07 4.16999E-07 0.357 340;

204171 204201 0.201378848 0.302068272 0.000345027
0.000517541 3.67358E-07 3.67358E-07 0.409 265;

204201 204216 0.201499848 0.302249772 0.000345067
0.000517601 3.6728E-07 3.6728E-07 0.416 265;

204216 204193 0.18800012 0.28200018 0.00034059
0.000510885 3.75999E-07 3.75999E-07 0.287 315;

204193 203202 0.18800012 0.28200018 0.00034059
0.000510885 3.75999E-07 3.75999E-07 0.280 315;

203202 203765 0.144000164 0.216000246 0.000331933
0.0004979 3.72833E-07 3.72833E-07 0.414 340;
];

% Node,Active power, Inductive Reactive Power
Load_Fiat = [
204871 0 0
204228 1347.00 426.87
204171 1347.00 426.87
204201 255.73 47.93
204216 698.25 381.85
```

```
204193 698.25 381.85
203202 637.65 225.00
203765 77.73 -6.60
];
```

```
MV_Load_Fiat = [
204871 33 13.2
204228 0 0
204171 0 0
204201 0 0
204216 0 0
204193 0 0
203202 198 79.2
203765 0 0
];
```

```
load('GP_Sunny.mat');
```

```
GP1 = [];
for i=1:length(GP(1,:))
    GP1 = [GP1, GP(:,i), zeros(size(GP(:,1)))];
end
GP = GP1;
```

```
LP = GP;
load('PowerMatrix_P.mat');
load('PowerMatrix_Q.mat');
deltatime = 0;
LP(:,1) = PowerMatrix_P(16500+deltatime:16595+deltatime,36);
LP(:,2) = PowerMatrix_Q(16500+deltatime:16595+deltatime,36);

LP(:,21) = PowerMatrix_P(16500+deltatime:16595+deltatime,37);
LP(:,22) = PowerMatrix_Q(16500+deltatime:16595+deltatime,37);

LP(:,41) = PowerMatrix_P(16500+deltatime:16595+deltatime,32);
LP(:,42) = PowerMatrix_Q(16500+deltatime:16595+deltatime,32);

LP(:,43) = PowerMatrix_P(16500+deltatime:16595+deltatime,35);
LP(:,44) = PowerMatrix_Q(16500+deltatime:16595+deltatime,35);

LP(:,45) = PowerMatrix_P(16500+deltatime:16595+deltatime,33);
```

```

LP(:,46) = PowerMatrix_Q(16500+deltatime:16595+deltatime,33);

LP(:,47) = PowerMatrix_P(16500+deltatime:16595+deltatime,34);
LP(:,48) = PowerMatrix_Q(16500+deltatime:16595+deltatime,34);

LP(:,51) = PowerMatrix_P(16500+deltatime:16595+deltatime,31);
LP(:,52) = PowerMatrix_Q(16500+deltatime:16595+deltatime,31);

LP(:,69) = PowerMatrix_P(16500+deltatime:16595+deltatime,30);
LP(:,70) = PowerMatrix_Q(16500+deltatime:16595+deltatime,30);

LP = LP*1000; % To have Watt

% LP1 = [
% 763.53 240.07 574.20 249.00 34.17 5.69 476.33 322.00 40.95 50.72 ...
% 77.73 -6.60 217.80 123.47 189.67 47.93 517.05 193.50 147.88 80.85 ...
% 485.10 255.15 81.95 23.28 53.856 14.432 105.6 39.6 135.52 44 ...
% 19.8 6.6 135.168 30.624 0 0 1050.110 90.680]*1000;
%
% LP2 = [
% 1295.33 395.33 709.80 414.60 65.83 18.61 568.07 379.07 42.42
53.03 172.67 100.13 257.93 167.87 287.20 51.33 628.20 224.10
185.33 105.70 663.95 372.40 109.08 45.28 73.216 32.384 171.6
79.2 369.6 170.72 33 13.2 153.384 42.768 0 0 100.736 40.56
% ]*1000;
%
% load('LP.mat');
% load('GP_Sunny.mat');
%
% LP(:, [1:2:86])=LP(:, [1:2:86])-GP;
%
% % Para anadir solar.
% load('GP_Cloudy.mat');
% GP1 = [];
% for i=1:1:length(GP(1,:))
%     GP1 = [GP1, GP(:,i), zeros(size(GP(:,1)))];
% end
% GP = GP1;
% % Profile = [LP1;LP2];
% % TimePoints = [0 1];

```

```
% %%
% % Do not forget to check KW or W in the profiles; you may need *1000
% %% for SEB we comment loading Profile and put it equal to LP
% % load('Profile.mat');
Profile = LP;
% % Profile(:,[37,38]) = 10 * Profile(:,[37,38]) ;
TimePoints = [0:size(Profile,1)-1];
% %
% % plot(Data_TOTAL(:,6),'--g')
% % hold on
% % plot(Data_23_b204201(:,6))
% % plot(Data_01_b203202(:,6))
% % plot(Data_11_b203765(:,6))
% % plot(Data_21_b204171(:,6))
% % plot(Data_22_b204193(:,6))
% % plot(Data_24_b204216(:,6))
% % plot(Data_26_b204228(:,6))
% % plot(Data_35_b204871(:,6))
% % plot(Data_FIAT(:,6),'--k')
% % grid on
```

### A.0.3 Stability of the R-PHIL co-simulation: settings and practical aspects

This section describes more in detail the implemented model in Turin and used to try out the stability of the whole remote co-simulation. The model includes a 22 kV distribution feeder, a HV-MV transformer and a connexion in the loop with an R-L-C load bank located in Bari's facility. Firstly, the *SC\_Console* block is depicted in Figure A.15. Beside the monitored variable a couple of manual switches are placed. The first one is already known by the reader, as its function is to start and stop the saving process. The second one is instead included for the first time: in nominal conditions it will be placed on the 1 value and put on 0 when we want to reset the feedback signal received from Bari. Such situation may verify once the instability in the model is found. In this condition the reset switch is activated and after that the realtime simulation can be easily resetted from RT-LAB<sup>®</sup>.

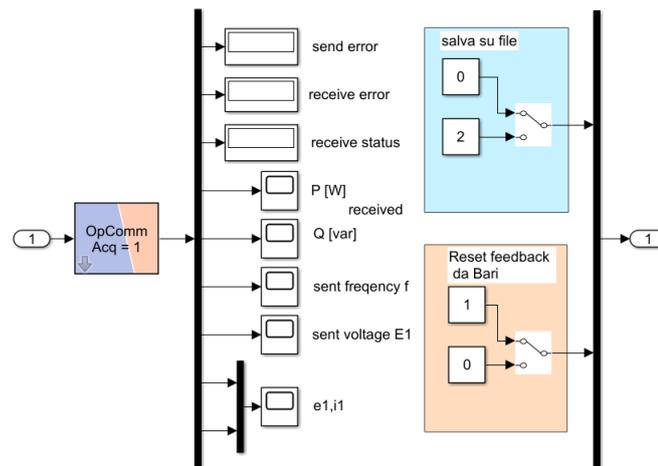


Figure A.15: *SC\_Console* subsystem of the model.

The command generated by the reset switch is then sent to the *SM\_Master* in which it is multiplied with the signal received from Bari as an output of the *OpSyncRec* block. This is shown in Figure A.16.

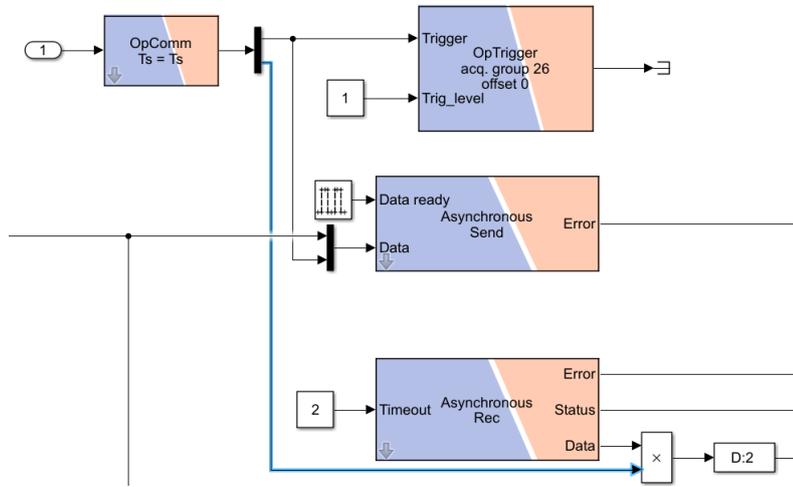


Figure A.16: *SM\_Master* subsystem: focus on the reset of the feedback from Bari.

As one of the task of these set of remote co-simulations is to verify the stability of the model as an added delay is included, it is necessary to add in the *SM\_Master* a *delay block*. To more easily monitor the added delay, the length of it is inserted by an input port. For this sake, the *Input port* option for the *Delay length* parameter has to be selected. Moreover, to avoid a saturation of the delay an extremely high value is inserted as the *Upper limit*. Those mentioned settings are highlighted in Figure A.17.

The delay length's value is imposed by a *Constant* block, as shown in Figure A.18. For data analysis's sake, this value is also saved on file. Please note that the value imposed by the *Constant* block represents the number of periods composing ore desired delay. By multiplying the *time step*  $T_s$  of the Turin model ( $150 \mu s$ ) for the delay length, we can obtain the added delay in *ms*.

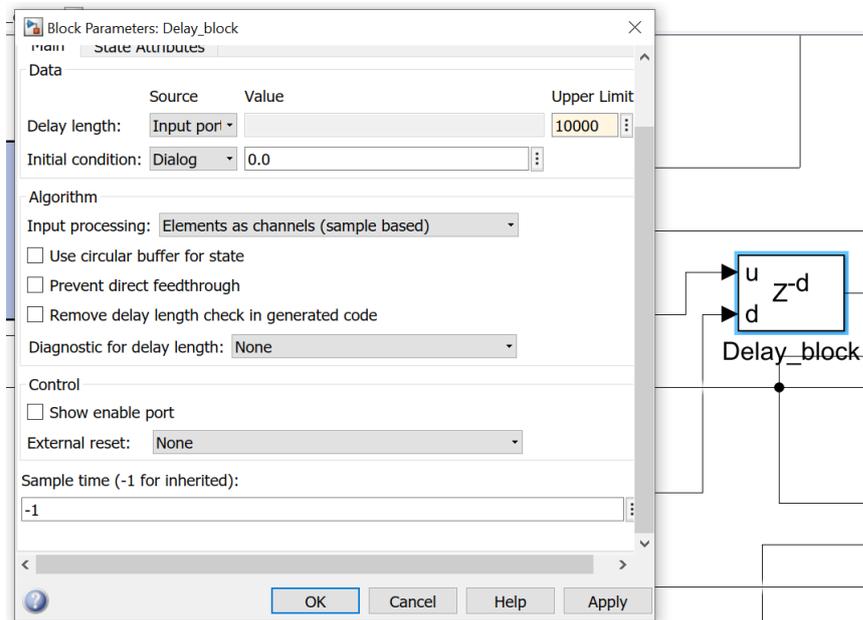


Figure A.17: Delay block's settings.

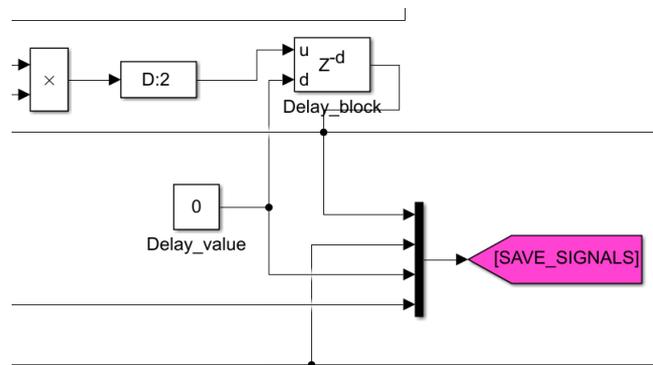


Figure A.18: *SM\_Master* subsystem: focus on the delay block.

A last aspect to discuss concerns the LV load model placed in the *SM\_Master*. The load, which emulates the microgrid located in Bari, is simulated with imposed power. As mentioned in Chapter 2, the  $(P, Q)$  signal is amplified by a scale factor in order to have a more significant impact by the microgrid. As shown in Figure A.19, the scale factor is imposed with a *Constant* block. For data analysis's sake, the constant value is saved on file as well.

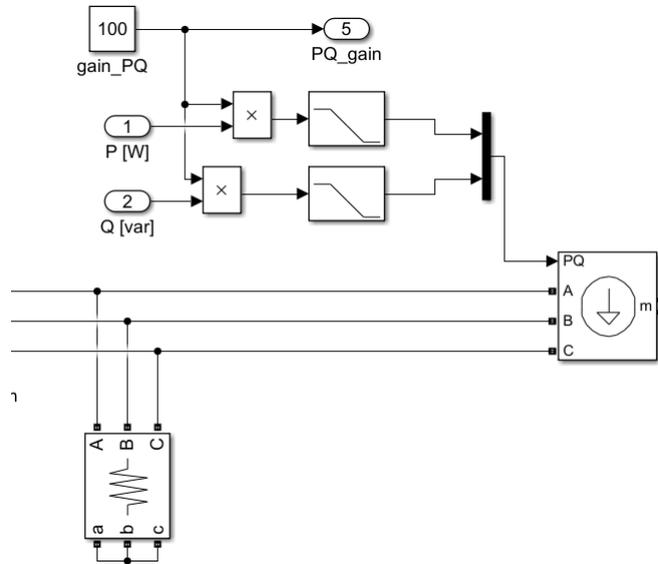


Figure A.19: *SM\_Master* subsystem: focus on load model and the  $(P, Q)$  scale factor.

Please note that for the sake of brevity, the Simulink model of the distribution feeder is not shown here, as it differs from the previous model only in the HV-MV transformer.

# Appendix B

This Appendix wants to further highlight the setups used in order to implement the SSN Method, which has been introduced in Chapter 3.

## B.0.1 Implementation of the SSN method: settings

Firstly, a deeper description of the Artemis Guide block (Figure B.1) is reported.

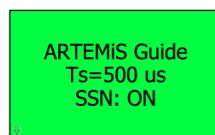


Figure B.1: Artemis Guide block.

The Artemis Guide block presents a series of tabs. The *General* tab, illustrated in Figure B.2. In this tab we impose the Time step  $T_s$  and we check the box to enable the SSN method.

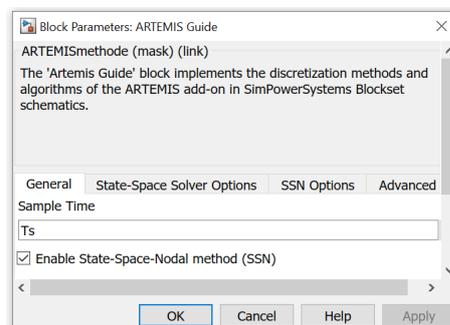


Figure B.2: Artemis Guide setup: General.

In the tab *SSN options* we select the desired SSN solver, the maximum number of iterations and we decide whether or not we want the SSN node type to be verified. This tab is shown in Figure B.3.

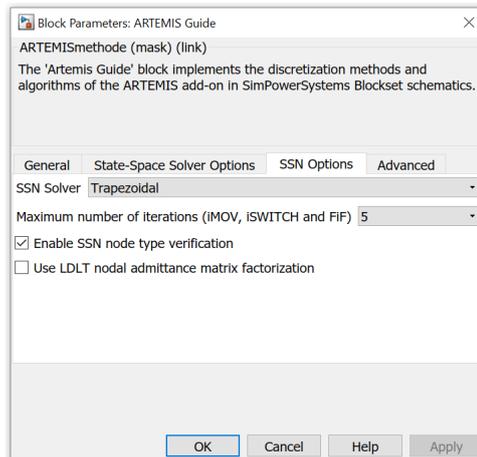


Figure B.3: Artemis Guide setup: SSN Options.

The last tab to be discussed is the *Advanced* tab. Here we define the SSN group maximum number of iterations, we choose whether or not we want to use the Artemis transformer saturation method and we select the type of Distributed Parameter Line. This tab is illustrated in Figure B.4.

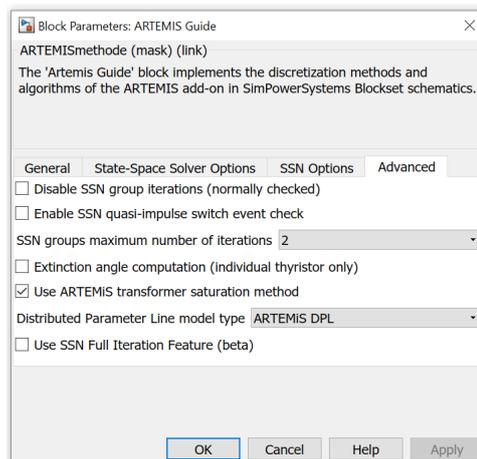


Figure B.4: Artemis Guide setup: Advanced.

A second aspect to be illustrated concerns the *SSN Interface block* and its settings. For each of these blocks, we are required to define the type of the

group interfacing with such SSN node. The types can be set as shown in Figure B.5.

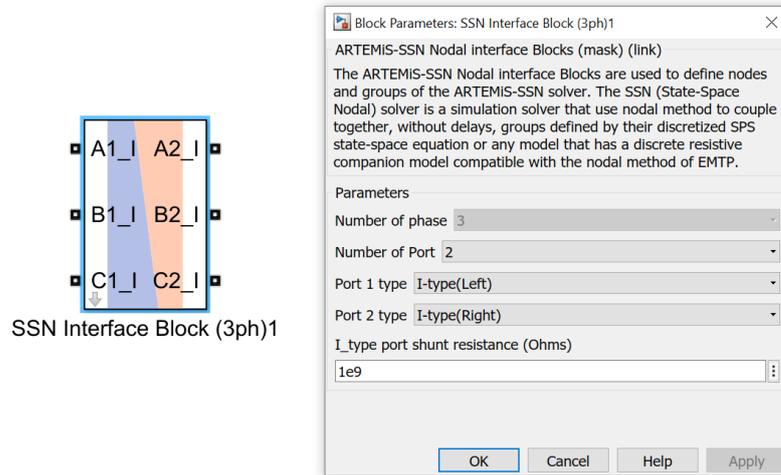


Figure B.5: SSN Interface Block and its settings: an example.

A last aspect to be discussed here is related to the model of the OLTC transformer. The OLTC is indeed composed by a certain number of switches (15 switches per each phase) in order to discretely vary the winding ratio. However, each SSN group cannot include more than 12 switches to avoid an excessive computational burden. For this reason, some SSN interface blocks were inserted in the OLTC model, as depicted in Figure B.6. The same principle has been applied for the subsystems *OLTCA*, *OLTCB* and *OLTCC*, shown in the *Left* side of the Figure B.6.

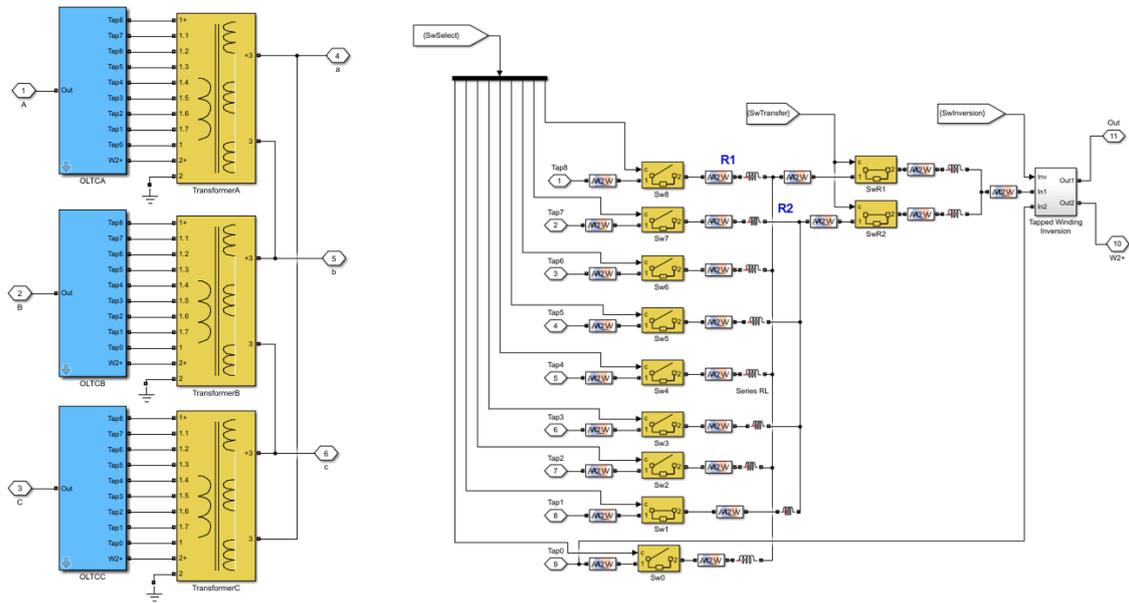


Figure B.6: OLTC Transformer model (*Left*); Model of the adjustable windings for a single phase (*Right*).

# Appendix C

The following Appendix aims to explain the imposed settings in the simulation implemented for Chapter 4. Secondly, the Matlab codes used to obtain the results of Chapter 4 are reported.

## C.0.1 Characterization of the distributed sources: Simulink settings

As the distribution network implemented in Chapter 4 is the same debated in Chapter 3, the upper-level Simulink model is shown in Figure C.1.

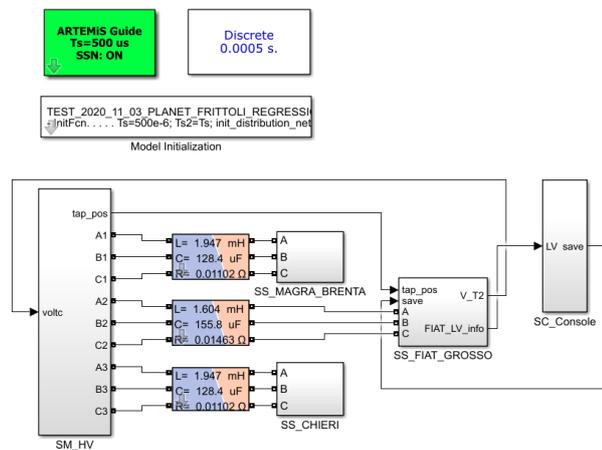


Figure C.1: Upper level of the implemented model.

For more details about the implementation of the decoupling of the network see Chapter 2 and Appendix B.

The information monitored in the *SC.Console* block are illustrated in Figure C.2. Beside the save-on-file trigger and the tap position, the monitored

variables below the secondary substation are the active and reactive power and the line-to-line voltage amplitude.

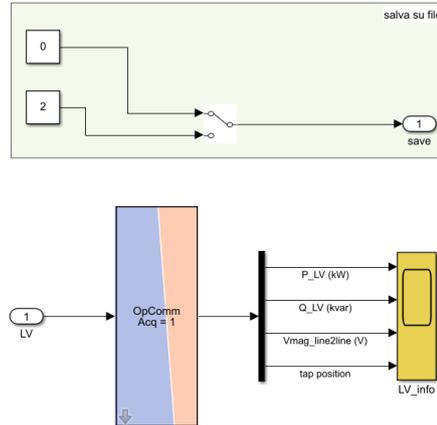


Figure C.2: Content of the *SC\_Console* block.

The test requires the manual variation of the tap position of the second HV-MV transformer. The tap variation is achieved by using a *Signal Builder* block and adding an input at the *OLTC transformer* model, as shown in Figure C.3.

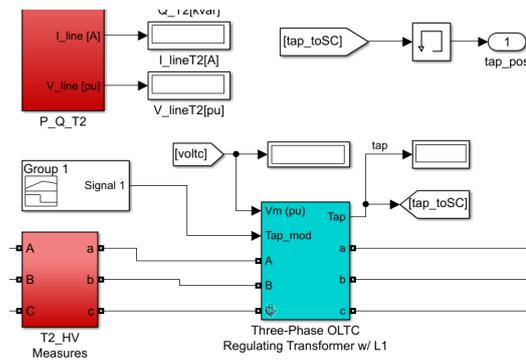


Figure C.3: Focus on the *OLTC transformer* model and the required adjustments.

In order to do so, the *OLTC transformer* model has been slightly modified, in order to control the *Tap* output regardless of the Automatic Voltage Regulator input. The required adjustments are illustrated in Figure C.4. The key adjustment, as it can be seen from Figure C.4 (lower), is to by-pass the



Regarding the measures, in the first case we only apply a *Moving Average* block to slightly clean-up the signals, while in the second case a *Band-limited white noise* is added with a *Sample and Hold* block for a proper sampling of the values.

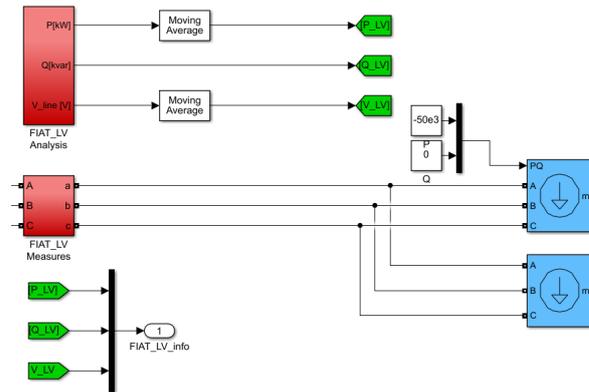


Figure C.6: Focus on the LV grid model of the FIAT feeder: first two implementations of the regression.

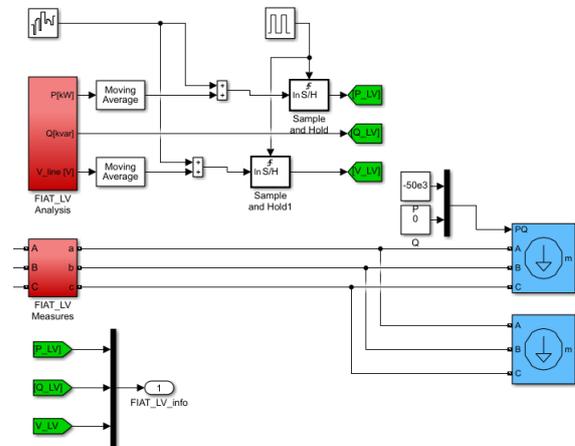


Figure C.7: Focus on the LV grid model of the FIAT feeder: last two implementations of the regression.

The settings of the Simulink blocks mentioned above are now explicitly illustrated. In the Figure C.8 the *Moving Average* block, the *Band-limited white noise* block and the *Pulse Generator* block's settings are illustrated respectively.

The defined value for the *Noise power* is specifically low in order to simulate what would be a high frequency noise affecting a measure.

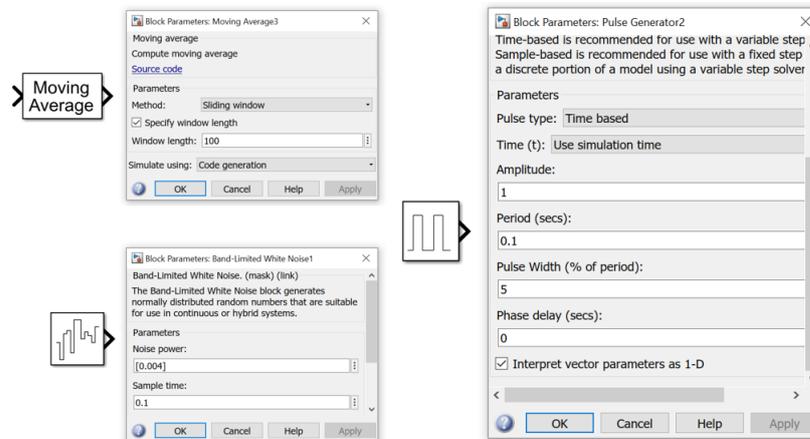


Figure C.8: *Moving Average*, *White Noise* and *Pulse Generator* Simulink blocks.

## C.0.2 Matlab scripts

It is now reported the Matlab code used to implement the non-linear regression previously discussed in Chapter 4. Two versions of the code are here illustrated: the first one is related to the estimation of the three parameters ( $P_{nom}$ ,  $k_p$  and  $P_{gen}$ ). On the other hand, the second version regards the estimation of only  $P_{nom}$  and  $P_{gen}$ , with  $k_p$  known.

To implement both the versions of the code, a Matlab function has been specifically created.

The inputs of the function `NonLinearRegressionV2` are the maximum number of iterations `iter_max`, the tolerance to be used in the convergence test `tol_max`, the line-to-line nominal voltage `V_n`, the set of measures of voltage `V` and active power below the secondary substation `y`, the set of parameters `x`, the real values of the parameters `x_true` and the exponential coefficient `kp`.

The outputs are the final set of the parameters `xfin`, the percentage errors along the iterative process `err_x` and the values of the parameters along the iterative process `xtemp`.

The first version of the code is here reported:

```
function [xfin,err_x,xtemp] = NonLinRegressionV2(iter_max,...
                                                tol_max,Vn,V,y,x,x_true)

    %number of measured values
```

```
Nobs = length(V);
%number of parameter to estimate
Nvar = length(x);
% definition of variable z
z = V/Vn;

%initialization of Jacobian matrix
J = zeros(Nobs, Nvar);
%initialization of percentage error vector
err_x = zeros(Nvar,iter_max+1);
%initialization of xtemp vector
xtemp = zeros(Nvar,iter_max+1);

for i=0:iter_max
    % computation of the vector y_est
    y_est = x(1)*z.^x(2) - x(3);

    % Jacobian matrix construction
    for k = 1:1:Nobs
        J(k,1) = z(k)^x(2);
        J(k,2) = x(1)*log(z(k))*z(k)^x(2);
        J(k,3) = -1;
    end

    % percentage error computation
    err_x(:,i+1) = abs((x_true-x)./x_true)*100;
    % output printing
    fprintf('\n %3i %10.3f %10.3f %10.3f %10.3f %10.3f %10.3f',...
        i,x(1),x(2),x(3),err_x(1,i+1),err_x(2,i+1), err_x(3,i+1));

    % computation of delta_x
    dx=inv(J'*J)*J'*(y-y_est);
    xtemp(:,i+1) = x;

    %Convergence test
    if max(abs(dx)) <= tol_max
        break
    end

    % update of the parameters
    x = dx + x;
```

```
        end
    xfin = x;
```

The second version of the code is here reported:

```
function [xfin,err_x,xtemp] = NonLinRegressionV2(iter_max,...
                                                tol_max,Vn,V,y,x,x_true,kp)

%number of measured values
Nobs = length(V);
%number of parameter to estimate
Nvar = length(x);
% definition of variable z
z = V/Vn;

%initialization of Jacobian matrix
J = zeros(Nobs, Nvar);
%initialization of percentage error vector
err_x = zeros(Nvar,iter_max+1);
%initialization of xtemp vector
xtemp = zeros(Nvar,iter_max+1);

for i=0:iter_max
    % computation of the vector y_est
    y_est = x(1)*z.^kp - x(2);

    % Jacobian matrix construction
    for k = 1:1:Nobs
        J(k,1) = z(k).^kp;
        J(k,2) = -1;
    end

    % percentage error computation
    err_x(:,i+1) = abs((x_true-x)./x_true)*100;

    % output printing
    fprintf('\n %3i %10.3f %10.3f %10.3f %10.3f %10.3f',...
            i,x(1),kp, x(2),err_x(1,i+1),err_x(2,i+1));

    % computation of delta_x
    dx=inv(J'*J)*J'*(y-y_est);
```

```
xtemp(:,i+1) = x;

%Convergence test
if max(abs(dx)) <= tol_max
    break
end

% update of the parameters
x = dx + x;
end
xfin = x;
```

# Appendix D

The Appendix D fulfills the necessity to clarify some technical aspects of the implementation of the electric faults. In the first part of the Appendix the required settings are explained. Secondly, the Matlab scripts to represent the results are shown.

## D.0.1 Faults implementation: required settings

Firstly, the upper level of the Simulink model is represented in Figure D.1. Beside the *SM\_Master* block there are the three *SS\_Slave* blocks and the *SC\_Console* block.

Please note that the only saved signals (also reported in the *Console* block) are taken from the *SS\_MAGRA\_BRENTA* block, beside the tap changer information coming from the *Master* block.

Please note that more details on how creating a decoupled network are reported in the Appendix B.

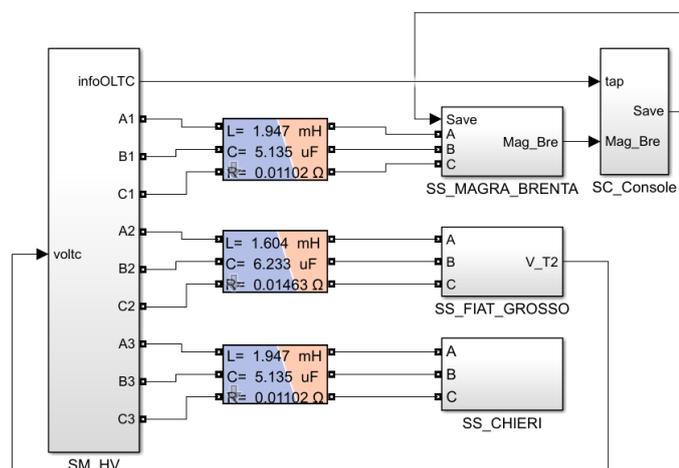


Figure D.1: Upper level of the Simulink model.

The content of the *SS\_Slave* block is illustrated in Figure D.2. Beside the *saveonfile* trigger, a series of signal related to the fault analysis are observed.

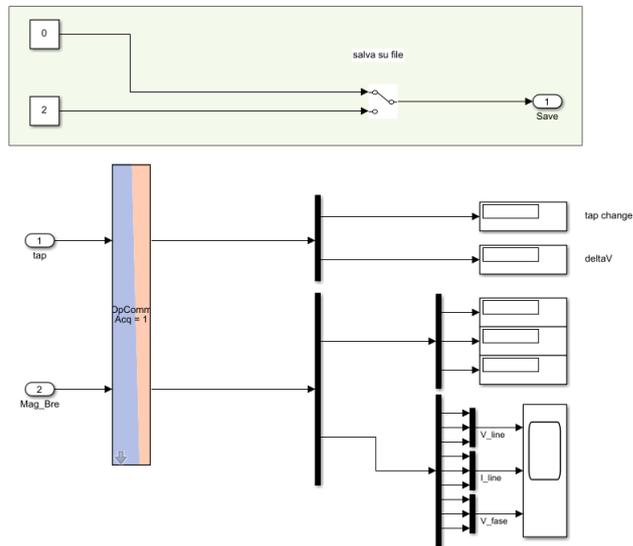


Figure D.2: Content of the *Slave* block.

To implement the electric faults, the *Three-phase fault* block is used. Such block is shown in Figure D.3, considering the location used for the simulations of Chapter 5. The block enables the implementation of three-phase faults, as well as asymmetrical faults.

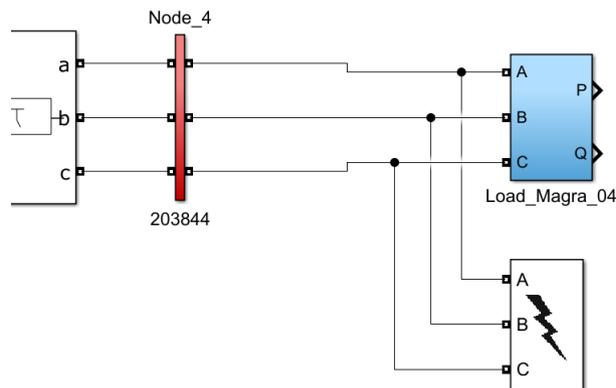


Figure D.3: Three-phase fault block in the chosen location.

Moreover, the block enables the user to decide whether to include the ground or not in the fault. The other parameters to be chosen are the *Switching*

times, the *Fault Resistance*  $R_{on}$ , the *Ground Resistance*  $R_g$ , the *Snubber Resistance*  $R_s$  and the *Snubber Capacitance*  $C_s$ . The interface panel of the block is illustrated in Figure D.4.

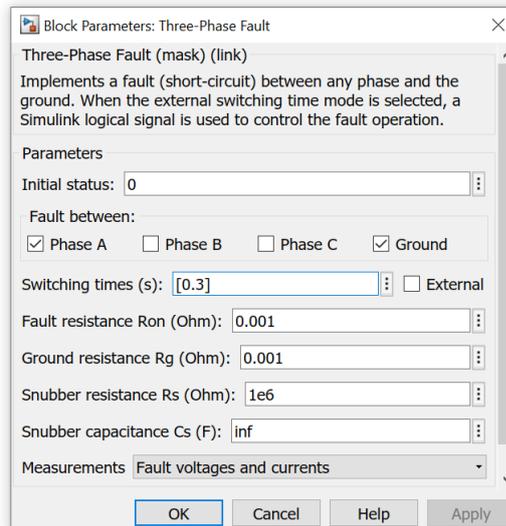


Figure D.4: Three-phase fault block: parameters.

The last aspect to be discussed is the status of the neutral connection of the HV-MV transformer. To study the faults with an isolated neutral as well as a compensated grounding connection we set the *Winding 2 connection* parameter to  $Yn$ . The required setup is shown in Figure D.5.

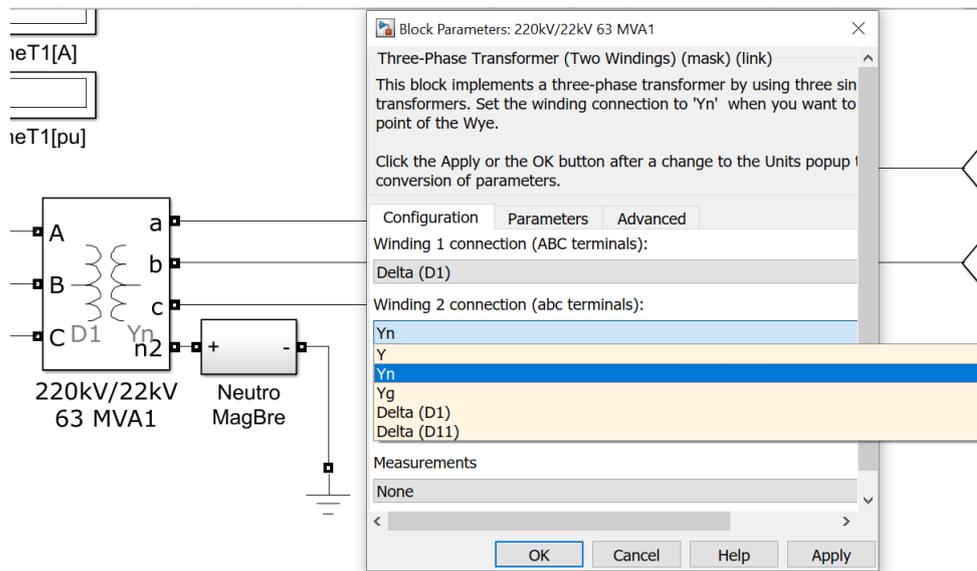


Figure D.5: Secondary winding connection setup.

## D.0.2 Faults implementation: Matlab code

It is here shown the Matlab code to represent the results related to the three-phase fault and the line-to-line fault implementations.

```

clc
clear
close all
%% Three-phase
data1 = load('MV_Mag3ph_01.mat');
data2 = load('MV_Mag3ph_Ron_02.mat');
data3 = load('MV_Mag3ph_Ron_03.mat');

%% % Ron = 0.001 Ohm
time = data1.data(1,:);

iph_a = data1.data(5,:);
iph_b = data1.data(6,:);
iph_c = data1.data(7,:);
iph_abc = [iph_a; iph_b; iph_c];

vph_a = data1.data(8,:);
vph_b = data1.data(9,:);

```

```

vph_c = data1.data(10,:);
vph_abc = [vph_a; vph_b; vph_c];
vph_seq = zeros(3,length(vph_abc(1,:)));

%
h1 = figure(1);
subplot 211
plot(time, vph_abc*1e-3, 'linewidth', 1.1);
xlim([0.08 0.12])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

subplot 212
plot(time, iph_abc, 'linewidth', 1.1);
xlim([0.08 0.12])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'northeast')
set(h1, 'color', [1 1 1])

% ++++++

h2 = figure(2);
subplot 211
plot(time, vph_abc*1e-3, 'linewidth', 1.1);
xlim([0.12 0.2])
ylim([-22 22])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northwest')

subplot 212
plot(time, iph_abc*1e-3, 'linewidth', 1.1);
xlim([0.12 0.2])
ylim(1e1*[-1.2 1.2])
xlabel('time (s)')
ylabel('current (kA)')

```

```
grid on
legend('i_a','i_b','i_c','location', 'northwest')
set(h2, 'color', [1 1 1])

%% Ron = 0.001*5*10e2 Ohm
time2 = data2.data(1,:);
iph_a2 = data2.data(5,:);
iph_b2 = data2.data(6,:);
iph_c2 = data2.data(7,:);
iph_abc2 = [iph_a2; iph_b2; iph_c2];

vph_a2 = data2.data(8,:);
vph_b2 = data2.data(9,:);
vph_c2 = data2.data(10,:);
vph_abc2 = [vph_a2; vph_b2; vph_c2];

%% Ron = 0.001*25*10e2 Ohm
time3 = data3.data(1,:);
iph_a3 = data3.data(5,:);
iph_b3 = data3.data(6,:);
iph_c3 = data3.data(7,:);
iph_abc3 = [iph_a3; iph_b3; iph_c3];

vph_a3 = data3.data(8,:);
vph_b3 = data3.data(9,:);
vph_c3 = data3.data(10,:);
vph_abc3 = [vph_a3; vph_b3; vph_c3];

%%
h5 = figure(5);

subplot 231
plot(time, vph_abc*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim([-22 22])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

subplot 232
```

```

plot(time2, vph_abc2*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim([-22 22])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

```

```

subplot 233
plot(time3, vph_abc3*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim([-22 22])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

```

```

subplot 234
plot(time, iph_abc*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim(1e1*[-1.2 1.2])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'southwest')
set(h1, 'color', [1 1 1])

```

```

subplot 235
plot(time2, iph_abc2*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim(1e1*[-1.2 1.2])
xlabel('time (s)')
ylabel('current (kA)')
grid on
legend('i_a','i_b','i_c','location', 'southwest')

```

```

subplot 236
plot(time3, iph_abc3*1e-3, 'linewidth', 1.1);
xlim([0.14 0.18])
ylim(1e1*[-1.2 1.2])
xlabel('time (s)')

```

```
ylabel('current (kA)')
grid on
legend('i_a','i_b','i_c','location', 'southwest')

set(h5, 'color', [1 1 1])

%% Line-To-Line fault

data5 = load('MV_Mag2ph_01.mat');

timeL2L = data5.data(1,:);
iaphL2L = data5.data(5,:);
ibphL2L = data5.data(6,:);
icphL2L = data5.data(7,:);
iabcphL2L = [iaphL2L; ibphL2L; icphL2L];
vaphL2L = data5.data(8,:);
vbphL2L = data5.data(9,:);
vcphL2L = data5.data(10,:);
vabcphL2L = [vaphL2L; vbphL2L; vcphL2L];
% ++++++

h6 = figure(6);
subplot 211
plot(timeL2L, vabcphL2L*1e-3, 'linewidth', 1.1);
xlim([0.12 0.2])
ylim([-22 22])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northwest')

subplot 212
plot(timeL2L, iabcphL2L*1e-3, 'linewidth', 1.1);
xlim([0.14 0.2])
ylim(1e1*[-1.2 1.2])
xlabel('time (s)')
ylabel('current (kA)')
grid on
legend('i_a','i_b','i_c','location', 'northwest')
set(h6, 'color', [1 1 1])
```

```
If_2ph = 8268.4;
If_3ph = 9497.4;
```

```
ratio2PH3PH = 100*If_2ph/If_3ph; %86percento da teoria
```

It is here shown the Matlab code to represent the results related to the line-to-ground fault, whether with isolated neutral or with compensated neutral grounding.

```
clc
clear
close all
%% Three-phase
data1 = load('MV_Mag1ph_01.mat');
data2 = load('MV_Mag1ph_02.mat');
%% % Ron = 0.001 Ohm
time = data1.data(1,:);

iph_a = data1.data(5,:);
iph_b = data1.data(6,:);
iph_c = data1.data(7,:);
iph_abc = [iph_a; iph_b; iph_c];

vph_a = data1.data(8,:);
vph_b = data1.data(9,:);
vph_c = data1.data(10,:);
vph_abc = [vph_a; vph_b; vph_c];
vph_seq = zeros(3,length(vph_abc(1,:)));

%
h1 = figure(1);
subplot 211
plot(time, vph_abc*1e-3, 'linewidth', 1.1);
xlim([0.29 0.37])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

subplot 212
plot(time, iph_abc, 'linewidth', 1.1);
```

```

xlim([0.29 0.37])
ylim([-300 450])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'northeast')
set(h1, 'color', [1 1 1])

h3 = figure(3);
plot(time, iph_abc, 'linewidth', 1.1);
xlim([0.33 0.4])
ylim([-80 80])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'northeast')
set(h3, 'color', [1 1 1])
%% Rf varia
Vconc = 22e3*sqrt(2);
invZf = [3.33E-06; 1.00E-05; 2.00E-05;
         5.00E-05; 1.00E-04; 1.25E-04;
         1.67E-04; 2.50E-04; 5.00E-04;
         1.00E-03; 1.00E-02; 1.00E+00; 1.00E+03];
Vb = [1.79E+04; 1.77E+04; 1.74E+04;
      1.63E+04; 1.51E+04; 1.49E+04;
      1.48E+04; 1.57E+04; 2.07E+04;
      2.60E+04; 3.07E+04; 3.11E+04;3.11E+04];
Vc = [1.82E+04; 1.85E+04; 1.88E+04;
      1.94E+04; 2.20E+04; 2.30E+04;
      2.46E+04; 2.73E+04; 3.31E+04;
      3.27E+04; 3.15E+04; 3.12E+04; 3.12E+04];

h4 = figure(4);
plot(Vb/Vconc,'linewidth', 1.1)
hold on
plot(Vc/Vconc,'linewidth', 1.1)
legend('V_b', 'V_c','location', 'best')
grid on
xlabel('inverse of Z_f (\Omega ^{-1})')
ylabel('V_{phase} over maximum V_{line}')
xticks([1 2 3 4 5 6 7 8 9 10 11 12 13])

```

```

xticklabels({'3.33e-6', '1e-5', '2e-5', '5e-5',
            '1e-4', '1.25e-4', '1.67e-4', '2.5e-4', '5e-4',
            '1e-3', '1e-2', '1', '1e3'})
xlim([1 13])
set(h4, 'color', [1 1 1])
%% Fasori di Vo e Io

Vomag = 1e-3*data1.data(11,end);
Vopha = (pi/180)*(data1.data(12,end));
Iofmag = data1.data(13,end);
Iofpha = (pi/180)*(data1.data(14,end));
Iosmag = data1.data(15,end);
Iospha = (pi/180)*(data1.data(16,end));

x = zeros(3,1);
y = zeros(3,1);
[x(1), y(1)] = pol2cart(Vopha, Vomag);
[x(2), y(2)] = pol2cart(Iofpha, Iofmag);
[x(3), y(3)] = pol2cart(Iospha, Iosmag);

fig1 = figure(8);
h11 = compass(x(1),y(1));
hold on
h12 = compass(x(2),y(2),'g');
hold on
h13 = compass(x(3),y(3),'b');
hold on
set(h11, 'LineWidth',2, 'color', [0 0 0]);
set(h12, 'LineWidth',2, 'color', [0.8500, 0.3250, 0.0980]);
set(h13, 'LineWidth',2, 'color', [0, 0.4470, 0.7410]);
set(fig1, 'color', [1 1 1])
legend('omopolar voltage (kV)', 'I_o faulted feeder (A)',
      'I_o healthy feeder (A)');
%% Neutro compensato

time2 = data2.data(1,:);

iph_a2 = data2.data(5,:);
iph_b2 = data2.data(6,:);
iph_c2 = data2.data(7,:);

```

```
iph_abc2 = [iph_a2; iph_b2; iph_c2];

vph_a2 = data2.data(8,:);
vph_b2 = data2.data(9,:);
vph_c2 = data2.data(10,:);
vph_abc2 = [vph_a2; vph_b2; vph_c2];
%
h6 = figure(6);
subplot 211
plot(time2, vph_abc2*1e-3, 'linewidth', 1.1);
xlim([0.29 0.37])
xlabel('time (s)')
ylabel('phase voltage (kV)')
grid on
legend('v_a','v_b','v_c','location', 'northeast')

subplot 212
plot(time2, iph_abc2, 'linewidth', 1.1);
xlim([0.29 0.37])
ylim([-300 450])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'northeast')
set(h6, 'color', [1 1 1])

h7 = figure(7);
plot(time2, iph_abc2, 'linewidth', 1.1);
xlim([0.37 0.44])
ylim([-80 80])
xlabel('time (s)')
ylabel('current (A)')
grid on
legend('i_a','i_b','i_c','location', 'northeast')
set(h7, 'color', [1 1 1])
%% Fasori compensato
Vomag2 = 1e-3*data2.data(11,end);
Vopha2 = (pi/180)*(data2.data(12,end));
Iofmag2 = data2.data(13,end);
Iofpha2 = (pi/180)*(data2.data(14,end));
Iosmag2 = data2.data(15,end);
```

```
Iospha2 = (pi/180)*(data2.data(16,end));

x2 = zeros(3,1);
y2 = zeros(3,1);
[x2(1), y2(1)] = pol2cart(Vopha2, Vomag2);
[x2(2), y2(2)] = pol2cart(Iofpha2, Iofmag2);
[x2(3), y2(3)] = pol2cart(Iospha2, Iosmag2);

fig2 = figure(8);
h21 = compass(x2(1),y2(1));
hold on
h22 = compass(x2(2),y2(2),'g');
hold on
h23 = compass(x2(3),y2(3),'b');
hold on
set(h21, 'LineWidth',2, 'color', [0 0 0]);
set(h22, 'LineWidth',2, 'color', [0.8500, 0.3250, 0.0980]);
set(h23, 'LineWidth',2, 'color', [0, 0.4470, 0.7410]);
set(fig2, 'color', [1 1 1])
legend('omopolar voltage (kV)', 'I_o faulted feeder (A)',
      'I_o healthy feeder (A)');
```

# Appendix E

This Appendix aims to clarify the more practical aspect of the simulations exposed in Chapter 6.

Firstly, the overall Simulink model is debated, digging deeper in the data acquisition mode.

Secondly, the PV model implemented in Simulink is described, including the Matlab script used to execute di Perturb & Observe MPPT algorithm.

## E.0.1 Overall model: an overview

Firstly, the overall Simulink model is shown in Figure E.1. The model is once again quite similar to the version previous used. The mere adaptations are in function of the simulation of Chapter 6. The voltage measurement necessary for the OLTC operation comes now from the *SS\_CHIERI* Subsystem, as the OLTC is used for this feeder for the voltage regulation.

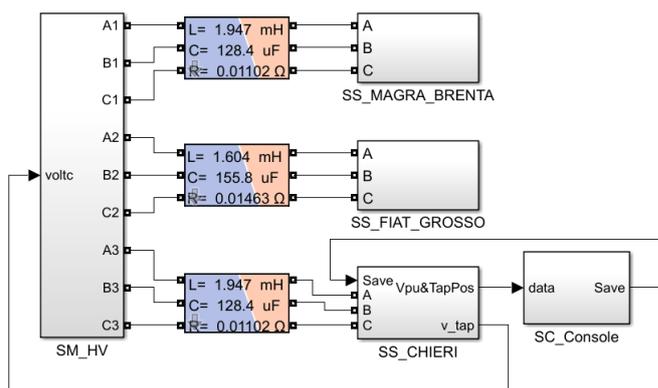


Figure E.1: Upper level of the Simulink model.

The content of the *SC\_Console* is now illustrated in Figure E.2. Beside the save-on-file trigger, the voltage (pu) measured on each CHIERI feeder's bus

are reported. Moreover, the tap position is observed.

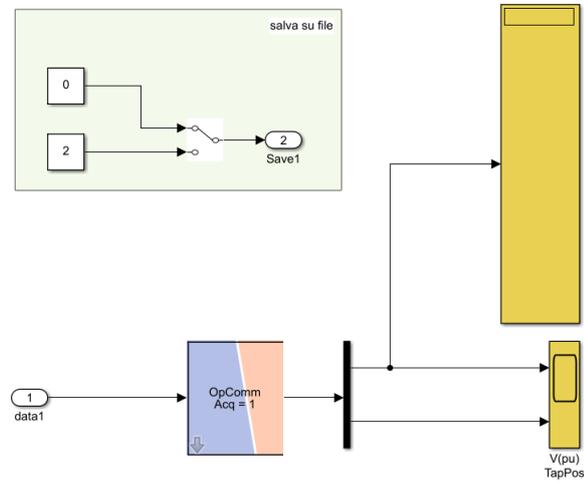


Figure E.2: Content of the *SC\_Console* subsystem.

The focus is now brought on the *SS\_CHIERI* Subsystem. The first sublayer of the subsystem is shown in Figure E.3. The CHIERI feeder's subsystem is located here. Moreover, the *OpWriteFile* is positioned.

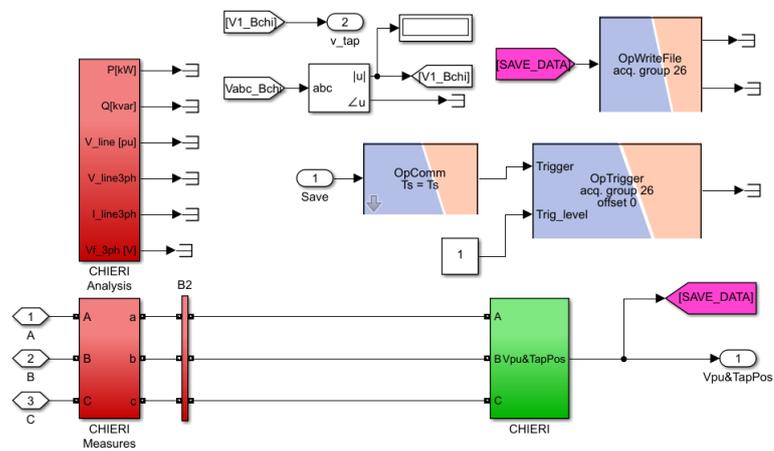


Figure E.3: Content of the *SS\_CHIERI* subsystem.

### E.0.2 PV system: Simulink model and settings

The photovoltaic system model is now described. The Simulink Subsystem is illustrated in Figure E.4.

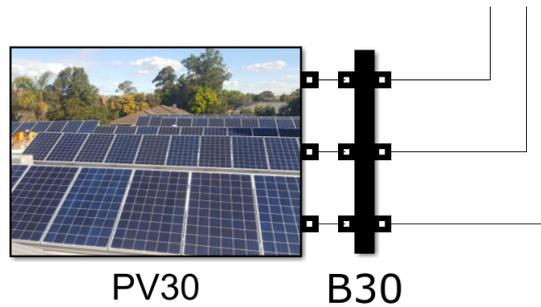


Figure E.4: Subsystem block of the PV system.

The overall scheme, with the different component, is shown in Figure E.5.

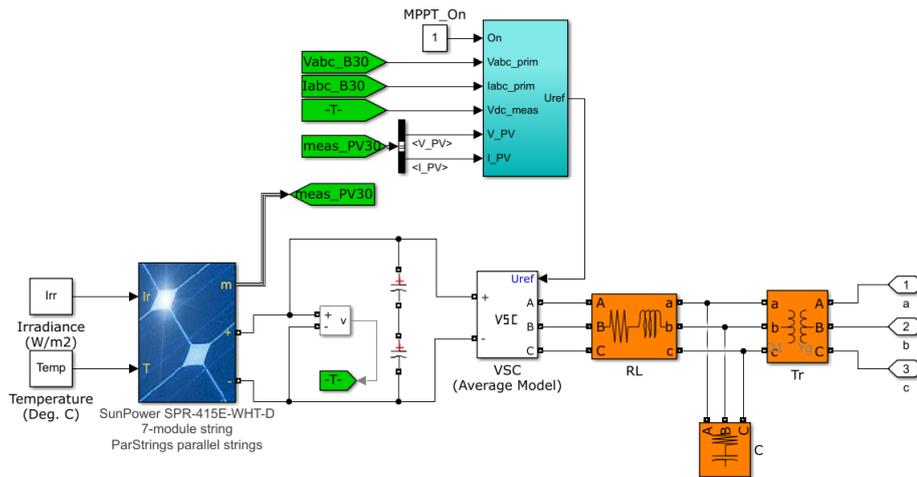


Figure E.5: Overall Simulink scheme of the PV system.

Firstly, the PV array model is discussed. To implement this component on Simulink the *PV Array* block is used. Such block, shown in Figure E.6, includes the dependence of the solar production from the irradiance and the environmental temperature. For this application an irradiance  $G = 1000W/m^2$  and a temperature  $T = 30^\circ C$  are used.

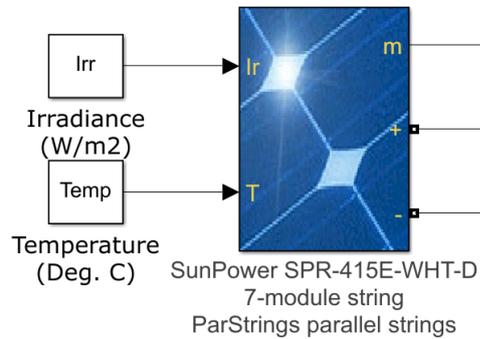


Figure E.6: Simulink block of the PV Array.

The SunPower SPR-415E-WHT-D PV panel is used with its specifications. The array is composed by 7 series-connected modules and a parallel-connected modules varying for the five Cases debated in Chapter 6.

The three-phase VSC inverter is implemented by the *VSC Average Model* Simulink block, shown in Figure E.7. By using the average model to implement the inverter we can execute the simulation with a high sample time, thus having a significantly lower risk of achieving an overrun in the real-time simulation. Moreover, as the aim of the study is not analyzing the high spectrum of the frequency domain, this approximation will not invalidate the results. As we use the average model, the Pulse Width Modulation is not implemented, thus the block receives the three-phase reference voltage instead of the legs' command.

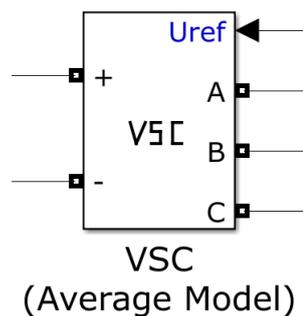


Figure E.7: Simulink block of the three-phase inverter.

The Control Subsystem is now described. The scheme is represented in Figure 6.3 and Figure 6.5.

The Subsystem receives as input  $v_{abc}$  and  $i_{abc}$  measured at the LV-MV transformer, the  $V_{PV}$  and the  $I_{PV}$  coming from the PV array and the DC link voltage. The output of the Subsystem is the three-phase reference voltage, which is sent to the inverter's block. The content of the Subsystem is reported in Figure E.8.

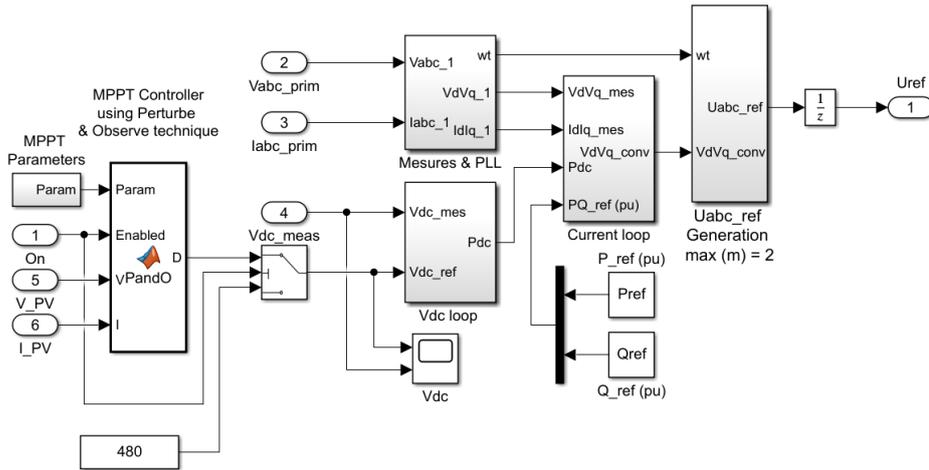


Figure E.8: Overall Simulink scheme for the control.

The Matlab script implementing the MPPT is now reported. The inputs are the enable trigger, the  $V_{PV}$  and the  $I_{PV}$ . The output is the updated reference value of the DC link voltage.

```
function D = Pand0(Param, Enabled, V, I)

% Perturb & Observe algorithm.

% D output = Reference for DC link voltage (Vdc_ref)
%
% Enabled input = 1 to enable the MPPT controller
% V input = PV array terminal voltage (V)
% I input = PV array current (A)
%
% Param input:
Dinit = Param(1); %Initial value for Vdc_ref
Dmax = Param(2); %Maximum value for Vdc_ref
Dmin = Param(3); %Minimum value for Vdc_ref
deltaD = Param(4); %Increment value used to increase/decrease Vdc_ref
```

```
%  
  
persistent Vold Pold Dold;  
  
% dataType = 'double';  
  
if isempty(Vold)  
    Vold=0;  
    Pold=0;  
    Dold=Dinit;  
end  
P= V*I;  
dV= V - Vold;  
dP= P - Pold;  
  
if dP ~= 0 && Enabled ~=0  
    if dP < 0  
        if dV < 0  
            D = Dold + deltaD;  
        else  
            D = Dold - deltaD;  
        end  
    else  
        if dV < 0  
            D = Dold - deltaD;  
        else  
            D = Dold + deltaD;  
        end  
    end  
end  
else  
    D=Dold;  
end  
  
if D >= Dmax || D<= Dmin  
    D=Dold;  
end  
  
Dold=D;  
Vold=V;  
Pold=P;
```

The *Measures & PLL* Subsystem is now discussed deeper. The scheme is shown in Figure E.9. Firstly, the measured values are converted in per units, as the whole control is developed using pu values. Secondly the PLL is implemented obtaining the  $\theta$  angle ( $wt$  output). In the PLL block' settings the PID gains are set to the suggested values. Finally, the  $abc-dq$  transformation is applied.

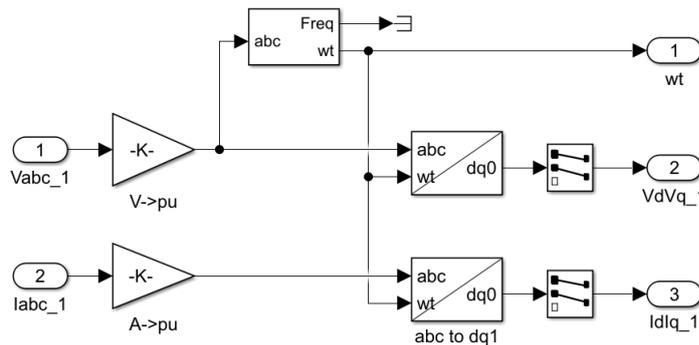


Figure E.9: Simulink scheme for the measures of voltage and current and the conversion to the  $(dq)$  reference frames.

The *Vdc loop* Subsystem is now debated. The content of the Subsystem is illustrated in Figure E.10. The measured value and the reference value of the DC link voltage are converted in per units. The error of the feedback is used as input of the PI regulator. The output of the subsystem is sent to the current loop.

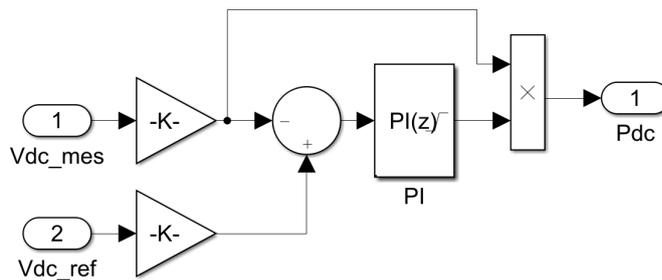


Figure E.10: Simulink scheme of the voltage DC link closed loop.

The next Subsystem to be discussed is the *Current loop* ones. The Simulink scheme of the current loop us shown in Figure E.11. The reference values of current are obtained by the reference values of active and reactive power. As

already discussed in Chapter 6, the PV system does not inject reactive power in the utility grid, thus we set  $Q^* = 0$ . On the other hand the active power  $P^*$  is set at 1 pu. The output of this subsystem are the reference values of  $v_d$  and  $v_q$ .

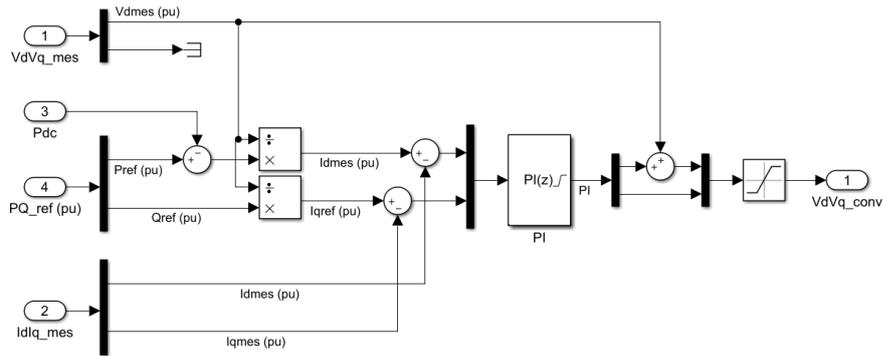


Figure E.11: Simulink scheme of the current closed loop.

Regarding the setup of the two PI regulators, an empirical process has been followed [4], which is now discussed.

Firstly, the  $k_p$  is set at a low value and  $k_i$  is set to zero. A square wave reference is imposed. The  $k_p$  value is now increased until the overshoot is reached. The  $k_i$  value is then increased in order to reduce the overshoot and cancel the steady state error. After this procedure, the obtained values for both the PI regulators are listed in Table E.1.

Table E.1: Parameters of the PI regulators.

Parameter	Value
$k_{p,vdc}(\frac{A}{V})$	14
$k_{i,vdc}(\frac{A \cdot s}{V})$	160
$k_{p,i}(\frac{V}{A})$	0.3
$k_{i,i}(\frac{V \cdot s}{A})$	200

### E.0.3 Matlab script with the required parameters

It is now reported the Matlab script listing all the necessary parameters for the distribution grid simulation, aside the PV system.

```
% Initialization code

% % MQTT subscribe
% Grid_subscribe

initial_step = 1;

% Time set-up
t_load_feed = 1;

% t_PV_feed = 4; % 1h
t_PV_feed = t_load_feed; % 15min

% t_ini_PV = 1040*t_PV_feed; %1h
t_ini_PV = initial_step*t_load_feed; %15min
t_ini_15min = initial_step*t_load_feed;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f = 50; %Hz, the nominal frequency of residential networks
w = 2*pi*f;

% Zone 1: High-Voltage network
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V_hi = 220e3;
% VLL_rms for the High-Voltage zone (equivalent transmission grid)
SC_current = 30e3;
% Rated short-circuit current of the equivalent network impedance
X_hi = V_hi/(sqrt(3)*SC_current);
% Equivalent reactive impedance value according to the SC-current
% (base impedance)
X_over_R = 20;
% X/R ratio
L_hi = X_hi/(2*pi*f);
% Equivalent inductance
R_hi = X_hi / X_over_R;
```

```

% Equivalent resistance value
C_hi1 = 1/(2*pi*f*X_hi*1e5);
% Small parallel capacitor of network impedance
% (High impedance at fundamental freq)
C_hi=[1,1,1;1,1,C_hi1;1,1,1];

%% HV/MV Trans Param
% 63MVA Trans:
MVA_1 = 63e+6;
Pcc_1 = 180.77e+3;
Vcc_1 = 15.93;%
Po_1t = 41.7e+3;
Io_1 = 0.07;%

Vbase = 22000;

Zcc_1 = Vcc_1/100;
Pcc_1 = Pcc_1 / MVA_1; % per unit
R_1 = Pcc_1;
X_1 = sqrt(Zcc_1^2-R_1^2);
R2_1 = R_1/2 * (Vbase^2/MVA_1);
R1_1 = R2_1/2 * (220000/22000)^2;
L2_1 = X_1/2 * (Vbase^2/MVA_1)/(2*pi*f);
L1_1 = L2_1 * (220000/22000)^2;

Zm_1 = Io_1/100;
Po_1 = Po_1t / MVA_1;
Gm_1 = Po_1;
Ym_1 = Io_1/100;
Bm_1 = sqrt(Ym_1^2-Gm_1^2);
% Rm_1 = Gm_1/Ym_1^2;
% Lm_1 = Bm_1/Ym_1^2;
Rm_1 = V_hi^2/Po_1t;
% I_in = MVA_1/(sqrt(3)*V_hi);
% S_o = sqrt(3)* V_hi*(Io_1/100)*I_in;
% Q_o = sqrt(S_o^2-Po_1t^2);
% X_o = V_hi^2/Q_o;
Lm_1 = 3.37284e+06/(2*pi*f); % X_o/(2*pi*f);

% 55MVA Trans:
MVA_2 = 55e+6;

```

```

Pcc_2 = 182.84e+3;
Vcc_2 = 11.46;%
Po_2t = 74.93e+3;
Io_2 = 0.63;%

Zcc_2 = Vcc_2/100;
Pcc_2 = Pcc_2 / MVA_2; % per unit
R_2 = Pcc_2;
X_2 = sqrt(Zcc_2^2-R_2^2);
R2_2 = R_2/2 * (Vbase^2/MVA_2);
R1_2 = R2_2*(220000/22000)^2;
L2_2 = X_2/2* (Vbase^2/MVA_2)/(2*pi*f);
L1_2= L2_2*(220000/22000)^2;

Zm_2 = Io_2/100;
Po_2 = Po_2t / MVA_2;
Gm_2 = Po_2;
Ym_2 = Io_2/100;
Bm_2 = sqrt(Ym_2^2-Gm_2^2);
Rm_2 = 645936; % V_hi^2/Po_2t;
% I_in = MVA_2/(sqrt(3)*V_hi);
% S_o = sqrt(3)* V_hi*(Io_2/100)*I_in;
% Q_o = sqrt(S_o^2-Po_2t^2);
% X_o = V_hi^2/Q_o;
Lm_2 = 143413/(2*pi*f); % X_o/(2*pi*f);

%%
dLine_hi = 20; % (km) Transmission line distance

%% 3-phase Distribution Network
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S_Xfo_1 = 63e6; % Rated power of the step-down transformer (VA)
S_Xfo_2 = 55e6; % Rated power of the step-down transformer (VA)
V_dis=22000; %V,The nominal voltage of the distribution network

% Neutral Impedance calculation
Rneutral_pu = 0.9; % percentage of neutral impedance
Rneutral_SI = Rneutral_pu * V_dis^2/S_Xfo_1; % Neutral impedance value

R_dl=0.313;
% ohms per kilometre, the resistance of 6.6kV distribution line

```

```

% according to Janpanese standard
X_dl=0.377;
% ohms per kilometre, the impedance of 6.6kV distribution line
% according to Janpanese standard
L_dl=X_dl/w; %6.6kv line inductance
dLine_dis = 10; % (km) Distribution line distance

% For HIL Interface
V_base=100; %Volts(V), the base rms value phase to ground voltage

%% Parameters for Distributed Line Parameters

load('MAGRA_param.mat')
load('BRENDA_param.mat')
load('GROSSO_param.mat')
load('FIAT_param.mat')
load('CHIERI_param.mat')
load('ReteRurale_profili.mat')
PV1 = PV(30:80);

% LINE FIAT
% StartNode,EndNode,R1[ohm/km],R0,L1[H/km],L0,C1,CO,Length[km], Imax[A]
PIS_Param_Fiat = [
    1551412 204871 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 2.801 340;
    204871 204228 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 0.233 340;
    204228 204171 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 0.357 340;
    204171 204201 0.201378848 0.302068272 0.000345027...
    0.000517541 3.67358E-07 3.67358E-07 0.409 265;
    204201 204216 0.201499848 0.302249772 0.000345067...
    0.000517601 3.6728E-07 3.6728E-07 0.416 265;
    204216 204193 0.18800012 0.28200018 0.00034059...
    0.000510885 3.75999E-07 3.75999E-07 0.287 315;
    204193 203202 0.18800012 0.28200018 0.00034059...
    0.000510885 3.75999E-07 3.75999E-07 0.280 315;
    203202 203765 0.144000164 0.216000246 0.000331933...
    0.0004979 3.72833E-07 3.72833E-07 0.414 340
];

```

```
% Node,Active power, Inductive Reactive Power
Load_Fiat = [
204871 0 0
204228 1347.00 426.87
204171 1347.00 426.87
204201 255.73 47.93
204216 698.25 381.85
204193 698.25 381.85
203202 637.65 225.00
203765 77.73 -6.60
];

MV_Load_Fiat = [
204871 33 13.2
204228 0 0
204171 0 0
204201 0 0
204216 0 0
204193 0 0
203202 198 79.2
203765 0 0
];

% LINE MAGRA
% StartNode,EndNode,R1,R0,L1,L0,C1,C0,Length,Imax[A]
PIS_Param_Magra = [
1551411 204874 0.172000048 0.258000072 0.000335961...
0.000503941 3.90908E-07 3.90908E-07 0.11 285;
204874 208221 0.144865072 0.217297608 0.000328108...
0.000492162 4.16193E-07 4.16193E-07 3.56 340;
208221 203845 0.31482022 0.47223033 0.000344268...
0.000516401 2.90787E-07 2.90787E-07 0.60 215;
203845 203844 0.325762008 0.488643012 0.000349344...
0.000524016 2.99887E-07 2.99887E-07 1.26 200
];

Profile = LP;
TimePoints = [0:size(Profile,1)-1];
TimePoints = TimePoints(30:80);
```

```
% Node,Active power, Inductive Reactive Power
```

```
Load_Magra = [
204874 0 0
208221 0 0
203845 155.00 56.81
203844 155.00 56.81
];
```

```
P=[ 155 200 230 140 100];
Q=[56 70 85 50 35 ];
```

```
MV_Load_Magra = [
204874 137.544 36.96
208221 0 0
203845 322.08 135.52
203844 0 0
];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% LINE BRENTA
```

```
% StartNode,EndNode,R1,RO,L1,L0,C1,C0,Length,Imax[A]
```

```
PIS_Param_Brenta = [
1551411 203294 0.144000164 0.216000246 0.000327859...
0.000491788 4.16999E-07 4.16999E-07 2.187 340;
203294 203548 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.382 280;
203548 2052801 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.171 280;
2052801 204514 0.21544534 0.32316801 0.000351183...
0.000526775 3.07083E-07 3.07083E-07 0.119 265;
204514 204621 0.205333128 0.307999692 0.000347882...
0.000521823 3.22641E-07 3.22641E-07 0.117 265;
204621 205428 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.171 265;
205428 205271 0.200168364 0.300252546 0.000347364...
0.000521046 3.22337E-07 3.22337E-07 2.646 265;
205271 204659 0 0 0.000327859...
0.000491788 4.16999E-07 3.22337E-07 0.265 315;
204659 203615 0.18800012 0.28200018 0.00034059...
0.000510885 3.75999E-07 3.75999E-07 0.045 315;
```

```

203615 208265 0.220289212 0.330433818 0.000345984...
0.000518976 3.01704E-07 3.01704E-07 0.146 265
];

```

```

% Node,Active power, Inductive Reactive Power
Load_Brenta = [
203294 1347.00 426.87
203548 637.65 225.00
2052801 0 0
204514 255.73 47.93
204621 180.60 -223.20
205428 182.00 111.48
205271 0 0
204659 0 0
203615 255.73 47.93
208265 0 0
];

```

```

MV_Load_Brenta = [
203294 0 0
203548 0 0
205280 0 0
204514 0 0
204621 0 0
205428 0 0
205271 0 0
204659 0 0
203615 0 0
208265 76.384 33.088

```

```
];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

% LINE CHIERI
% StartNode,EndNode,R1,R0,L1,L0,C1,C0,Length,Imax[A]
PIS_Param_Chieri = [
1551413 204946 0.166497452 0.249746178 0.000334856...
0.000502284 3.99339E-07 3.99339E-07 1.230 285;
204946 203890 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.872 280;

```

```

203890 204834 0.28133468 0.42200202 0.000789011...
0.001183517 1.89415E-07 1.89415E-07 1.055 225;
204834 203319 0.197080928 0.295621392 0.000343602...
0.000515403 3.70134E-07 3.70134E-07 0.370 265;
203319 203323 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.312 280;
203323 204249 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 1.178 280;
204249 203324 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.879 280;
203324 203333 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 1.011 280;
203333 203892 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.226 280;
203892 205351 0.197080928 0.295621392 0.000343602...
0.000515403 3.70134E-07 3.70134E-07 0.370 315;
205351 203874 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 0.872 280;
203874 203137 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 1.014 280;
203137 203923 0.204271232 0.306406848 0.000345985...
0.000518978 3.65491E-07 3.65491E-07 0.354 265;
203923 203921 0.264783816 0.397175724 0.00030662...
0.000459929 2.53565E-07 2.53565E-07 0.379 225;
203921 205304 0.188219372 0.282329058 0.000331413...
0.000497119 3.74209E-07 3.74209E-07 2.006 315;
205304 204220 0.188219372 0.282329058 0.000331413...
0.000497119 3.74209E-07 3.74209E-07 2.006 315;
204220 205638 0.235999852 0.353999778 0.000356506...
0.000534759 3.44999E-07 3.44999E-07 1.014 280;
205638 203937 0.17620746 0.26431119 0.00032584...
0.000488761 3.82143E-07 3.82143E-07 1.736 285
];

```

```

% Node,Active power, Inductive Reactive Power
Load_Chieri = [
204946 0 0
203890 155.00 56.81
204834 110.37 44.37
203319 637.65 225.00
203323 571.33 388.53

```

```

204249 182.00 111.48
203324 226.67 124.53
203333 0 0
203892 155.00 56.81
205351 110.37 44.37
203874 255.73 47.93
203137 255.73 47.93
203923 571.33 388.53
203921 571.33 388.53
205304 180.60 -223.20
204220 255.73 47.93
205638 255.73 47.93
203937 637.65 225.00
];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% LINE GROSSO

```

```

% StartNode,EndNode,R1,R0,L1,L0,C1,C0,Length,Imax[A]

```

```

PIS_Param_Grosso = [

```

```

    1551412 205358 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 1.000 340;
    205358 2052802 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 1.153 340;
    2052802 205024 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 0.929 340;
    205024 204730 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 1.000 340;
    204730 204748 0.14568158 0.21852237 0.000328345...
    0.000492518 4.15432E-07 4.15432E-07 0.314 340;
    204748 204813 0.146120568 0.219180852 0.000328472...
    0.000492708 4.15023E-07 4.15023E-07 0.249 340;
    204813 203975 0.175965944 0.263948916 0.000337812...
    0.000506719 3.91982E-07 3.91982E-07 0.354 315;
    203975 203974 0.235999852 0.353999778 0.000356506...
    0.000534759 3.44999E-07 3.44999E-07 0.133 280;
    203974 204716 0.229360824 0.344041236 0.000354439...
    0.000531658 3.50195E-07 3.50195E-07 0.194 225;
    204716 203385 0.144000164 0.216000246 0.000327859...
    0.000491788 4.16999E-07 4.16999E-07 0.151 340;
    203385 203558 0.144000164 0.216000246 0.000327859...

```

```

    0.000491788 4.16999E-07 4.16999E-07 0.205 340
];

% Node,Active power, Inductive Reactive Power
Load_Grosso = [
    205358 40.95 50.72
    2052802 0 0
    205024 1347.00 426.87
    204730 182.00 111.48
    204748 637.65 225.00
    204813 77.73 6.60
    203975 637.65 225.00
    203974 255.73 47.93
    204716 571.33 388.53
    203385 637.65 225.00
    203558 1347.00 426.87
];

%%
%Initialization of the ZIP load coefficients

% Matrix a used for active power
% Matrix b used for reactive power

% a = zeros(43,3);
% b = zeros(43,3);
a_ind = [1.21 -1.61 1.41];
a_comm = [0.76 -0.52 0.76];
a_res = [1.5 -2.31 1.81];

b_ind = [4.35 -7.08 3.72];
b_comm = [6.92 -11.75 5.83];
b_res = [7.41 -11.97 5.55];

a = [a_ind; a_ind; a_ind; a_comm; a_res; a_comm; a_comm;
    a_res; a_comm; a_res; a_ind; a_ind; a_comm; a_ind;
    a_comm; a_comm; a_res; a_comm; a_res; a_res; a_ind;
    a_ind; a_comm; a_comm; a_ind; a_res; a_comm; a_comm;
    a_ind; a_ind; a_comm; a_comm; a_comm; a_res; a_res;
    a_res; a_res; a_comm; a_comm; a_res; a_res; a_res; a_res;];

```

```

b = [bind; bind; bind; bcomm; bres; bcomm; bcomm;
     bres; bcomm; bres; bind; bind; bcomm; bind;
     bcomm; bcomm; bres; bcomm; bres; bres; bind;
     bind; bcomm; bcomm; bind; bres; bcomm; bcomm;
     bind; bind; bcomm; bcomm; bcomm; bres; bres;
     bres; bres; bcomm; bcomm; bres; bres; bres; bres;];

%%%%%% End of useful Initialization %%%%%%%%%%%
%%
% System frequency (Hz):
Fnom=50;
% Specialized Power Systems sample time (s):
% Ts_Power=1/(33*Fnom)/100;
Ts_Power = 100e-6;
% Inverter Control system sample time (s):
Ts_Control= 100e-6;

% *****
%                               POWER PARAMETERS
% *****

Pnom = 1600e3;      % Inverter nominal 3-phase power (VA)
Vnom_prim = 22e3;  % Nominal inverter primary line-to-line voltage(Vrms)
Vnom_dc = 480;     % Nominal DC link voltage (V)

% Pnom = 12.5e3;      % Inverter nominal 3-phase power (VA)
% Vnom_prim = 400;   % Nominal inverter primary line-to-line voltage(Vrms)
% Vnom_dc = 480;     % Nominal DC link voltage (V)

% Nominal inverter secondary line-to-line voltage (Vrms):
Vnom_sec= 0.85*Vnom_dc/2/sqrt(2)*sqrt(3);
% Vnom_sec= 400;

% Transformer parameters:
% Nominal voltage in Vrms, Resistance in pu and Leakage inductance in pu

Pnom_xfo=Pnom;      % Transformer nominal power (VA)
TotalLeakage=0.06;  % Transformer total leakage (pu)

W1_xfo= [Vnom_prim TotalLeakage/25/2  TotalLeakage/2];

```

```

% Winding 1 (Grid side)
W2_xfo= [Vnom_sec TotalLeakage/25/2 TotalLeakage/2];
% Winding 2 (DC link side)

Rm_xfo=200; % Magnetization resistance (pu)
Lm_xfo=200; % Magnetization inductance (pu)

% Solar Panel
Irr = 1000; %irradiance
Temp = 30; %temperature
ParStrings = 500;
%500 -> 1.6 MVA
%420 -> 1.25 MVA
%340 -> 1 MVA
%210 -> 0.63 MVA
%135 -> 0.40 MVA

% Inverter choke RL [Rpu Lpu]
RLchoke=[ 0.15/100 0.15 ]; % in pu
Pbase_sec=Vnom_sec^2/Pnom;
RL(1)=RLchoke(1)*Pbase_sec;
RL(2)=RLchoke(2)*Pbase_sec/(2*pi*Fnom);

% Filter C Parameters
Qc=0.1*Pnom; % Capacitive reactive power (var)
Pc=Qc/50; % Active power (W)

% DC link energy for 3/4 cycle of Pnom
Ceq= 3/4 * (Pnom/Fnom*2/Vnom_dc^2);
Clink=Ceq*2; % Cp & Cn (F)

% IGBT Bridge parameters

Rs=1e6; % IGBT Snubber (Ohm)
Cs=inf; % IGBT snubber (F)
Ron=1e-3; % IGBT conduction resistance
Vf=0; % IGBT Forward voltage
Vfd=0; % Diode Forward voltage

```

```

% *****
%
% CONTROL PARAMETERS
% *****

% MPPT Control (Perturb & Observe Algorithm)

Increment_MPPT= 0.01; %Increment value used to increase/decrease Vdc_ref
Limits_MPPT= [ 583 357 ]; % Upper & Lower limit for Vdc_ref (V)

% VDC regulator (VDCreg)

Kp_VDCreg=14;          % Proportional gain
Ki_VDCreg= 160;       % Integral gain
LimitU_VDCreg= 1.5;   % Output (Idref) Upper limit (pu)
LimitL_VDCreg= -1.5; % Output (Idref) Lower limit (pu)
%
% Current regulator (Ireg)

RLff(1)= W1_xfo(2) + W2_xfo(2) + RLchoke(1); % Feedforward values
RLff(2)= W1_xfo(3) + W2_xfo(3) + RLchoke(2); % Feedforward values

Kp_Ireg= 0.3;          % Proportional gain
Ki_Ireg= 200;         % Integral gain
LimitU_Ireg= 1.5;     % Output (Vdq_conv) Upper limit (pu)
LimitL_Ireg= -1.5;    % Output (Vdq_conv) Lower limit (pu)

% PWM Modulator Parameters

Fc= 100 * Fnom ; % Carrier frequency (Hz)

% Reference values

Pref = 1; % pu
Qref = 0; % pu

```

# Bibliography

- [1] Andrea Michele Aleo. Tesi Magistrale in Ingegneria Elettrica, 2021.
- [2] M. Banja and M. Jégard. Renewable technologies in the EU electricity sector: trends and projections, June 2020.
- [3] G. Belforte. Corso di controlli automatici - capitolo 3a - laurea in ingegneria elettrica. 2018.
- [4] Prof. R. Bojoi. Corso di fondamenti di azionamenti. 2018.
- [5] Prof. R. Bojoi. Corso di smart electricity systems. 2020.
- [6] A. Bokhari, A. Alkan, R. Dogan, M. Diaz-Aguiló, F. de León, D. Czarkowski, Z. Zabar, L. Birenbaum, A. Noel, and R. E. Uosef. Experimental determination of the zip coefficients for modern residential, commercial, and industrial loads. *IEEE Transactions on Power Delivery*, 29(3):1372–1381, 2014.
- [7] E. Bompard, S. Bruno, A. Cordoba-Pacheco, C. Diaz-Londono, G. Giannoccaro, M. La Scala, A. Mazza, and E. Pons. Connecting in real-time power system labs: an italian test-case. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, pages 1–6, 2020.
- [8] E. Bompard, S. Bruno, S. Frittoli, G. Giannoccaro, M. L. Scala, A. Mazza, E. Pons, and C. Rodio. Remote phil distributed co-simulation lab for tso-dso-customer coordination studies. In *2020 AEIT International Annual Conference (AEIT)*, pages 1–6, 2020.
- [9] Prof. E. Carpaneto. Corso di sistemi elettrici di potenza. 2019.
- [10] Center For Climate And Energy SOLUTIONS. Global Emissions. <https://www.c2es.org/content/international-emissions/>.

- [11] M. Chiampi, L. Zilberti, O. Bottaucio, D. Giordano, A. Manzin, and P.E. Roccato. In *Elettrotecnica: dai campi ai circuiti*. politeko, 2017.
- [12] Prof. P. Di Leo. Corso di progettazione di impianti elettrici. 2020.
- [13] C. Dufour, J. Mahseredjian, and J. Belanger. A combined state-space nodal method for the simulation of power system transients. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–1, 2011.
- [14] EEGGUIDE.COM. Tap Changing Transformer, 2021.  
<https://www.eeeguide.com/tap-changing-transformer/>.
- [15] Helena Gerard, Enrique Israel Rivero Puente, and Daan Six. Coordination between transmission and distribution system operators in the electricity sector: A conceptual framework. *Utilities Policy*, 50:40 – 48, 2018.
- [16] GSE. Rapporto Statistico Solare Fotovoltaico - 2019, June 2020.  
[https://www.gse.it/Dati-e-Scenari\\_site/statistiche\\_site](https://www.gse.it/Dati-e-Scenari_site/statistiche_site).
- [17] H. Hooshyar, L. Vanfretti, and C. Dufour. Delay-free parallelization for real-time simulation of a large active distribution grid model. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6278–6284, 2016.
- [18] IRENA. Off-grid renewable energy systems: status and methodological issues, February 2015.  
[https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2015/IRENA\\_Off-grid\\_Renewable\\_Systems\\_WP\\_2015.pdf](https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2015/IRENA_Off-grid_Renewable_Systems_WP_2015.pdf).
- [19] IRENA. Letting in the Light: How solar photovoltaics will revolutionise the electricity system, June 2016.  
[https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2016/IRENA\\_Letting\\_in\\_the\\_Light\\_2016.pdf](https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2016/IRENA_Letting_in_the_Light_2016.pdf).
- [20] ISGAN. TSO-DSO interaction, 2014.  
[https://www.iea-isgan.org/wp-content/uploads/2014/02/ISGAN\\_DiscussionPaper\\_TSODSOInteractionOverview\\_2014.pdf](https://www.iea-isgan.org/wp-content/uploads/2014/02/ISGAN_DiscussionPaper_TSODSOInteractionOverview_2014.pdf).
- [21] ISGAN. Ancillary services from distributed energy sources, 2019.  
<http://www.iea-isgan.org/smartnet/>.

- [22] R. Kabiri, D. G. Holmes, and B. P. McGrath. Voltage regulation of lv feeders with high penetration of pv distributed generation using electronic tap changing transformers. In *2014 Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6, 2014.
- [23] P. Kundur, N.J. Balu, and M.G. Lauby. *Power System Stability and Control*. EPRI power system engineering series. McGraw-Hill Education, 1994.  
<https://books.google.it/books?id=w0lSAAAAMAAJ>.
- [24] P. Zhai H.-O. Pörtner D. Roberts J. Skea P.R. Shukla A. Pirani W. Moufouma-Okia C. Péan R. Pidcock S. Connors J.B.R. Matthews Y. Chen X. Zhou M.I. Gomis E. Lonnoy T. Maycock M. Tignor Masson-Delmotte, V. and T. Waterfield (eds.). IPCC, 2018: Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, 2019.  
<https://www.ipcc.ch/sr15/>.
- [25] OPAL RT Technologies. RT Lab - Online Courses.  
[https://opal-rt.com/opal\\_tutorial/preparing-simulink-model-real-time-exe](https://opal-rt.com/opal_tutorial/preparing-simulink-model-real-time-exe)
- [26] G. Sybille and P. Giroux (Hydro-Quebec). Oltc regulating transformer (phasor model).
- [27] Thomas Kirk - OPAL RT Technologies. Real Time Simulation for Energy Storage Applications - including Battery Management System Testing, 2019.
- [28] B.M. Weedy, B.J. Cory, N. Jenkins, J.B. Ekanayake, and G. Strbac. *Electric Power Systems*. Wiley, 2012.