
Étude des régulations en tension des réseaux de distribution

RAPPORT D'ACTIVITÉ DE STAGE 2A

Rafael Accácio NOGUEIRA

Orienté par
M. Hervé GUÉGUEN

25 septembre 2017

Table des matières

Liste des Acronymes	4
Glossaire	5
1 Objectif	6
2 Introduction Générale	6
2.1 Sur le lieu de travail	6
2.1.1 L'IETR	6
2.1.2 Division des Équipes de Recherche	6
2.1.3 L'équipe AUT	7
2.2 Le Projet	7
3 Division du travail	8
3.1 Première Partie - Lecture	8
3.2 Deuxième Partie - Mise en Main	8
3.3 Troisième Partie - Programmation	8
3.4 Quatrième Partie - Intégration	8
3.5 Cinquième Partie - Rédaction	9
4 Méthodologie	10
4.1 Deuxième Partie - Mise en Main	10
4.1.1 À propos du DIgSILENT PowerFactory	10
4.2 À propos du réseau	11
4.3 Troisième Partie - Programmation	13
4.4 Quatrième Partie - Intégration	17
4.4.1 À propos des temps de simulation	19
5 Résultats	19
5.1 Scripts	19
5.2 Simulations avec le régulateur	22
6 Difficultés	25
7 Prochains Travaux	25
8 Conclusions	26
Liste des tableaux	27
Table des figures	28

Liste des Acronymes

AUT AUTomatic control. 6

CSV Comma Separated Values. 15, 16

DPL DIgSILENT Programming Language. 8

DSL DIgSILENT Simulation Language. 17

EDF Électricité de France. 10

EMT Electro Magnetic Transient. 11

Enedis l'ancienne ERDF. 10

GUI Interface Graphique d'utilisateur. 10, 11, 24

IETR Institut d'Électronique et de Télécommunications de Rennes. 6, 10

JPL NASA Jet Propulsion Lab - Caltech - USA. 6

POLIMI Politecnico Milano - Italia. 6, 10

RMS Root Mean Square - Moyenne quadratique. 11

USP Universidade de São Paulo - Brasil. 6

Glossaire

Q_{Cx-xx} Puissance Réactive de la charge $Cx - xx$. 14

Q_{GDx} Puissance Réactive du générateur GDx . 15

V_{Nxx} Tension du bus Nxx . 14

1 Objectif

L'objectif de ce document est faire un rapport du stage, qui détaille le lieu de travail, toutes les choses produits pendant le stage en expliquant les méthodes utilisés et qui montre les résultats obtenus, les commente et dit les implications des conclusions.

2 Introduction Générale

Afin de compléter la formation du 2A CentraleSupélec, un stage a été réalisé entre les mois de juillet et septembre de 2017, au sein du laboratoire de l'équipe AUTomatic control (AUT) du Institut d'Électronique et de Télécommunications de Rennes (IETR) en travaillant au cadre du projet d'Étude des régulations en tension des réseaux de distribution, avec l'orientation de M. Hervé GUÉGUEN.

2.1 Sur le lieu de travail

2.1.1 L'IETR

L'IETR est un institut de recherche français, spécialisé en électronique et télécommunication, localisé à Rennes, comptant avec plus que 300 enseignants-chercheurs, ingénieurs, doctorants et administratifs, il est formé par équipes de recherche des écoles et instituts de recherche de la région comme le CNRS, l'Université Rennes 1, INSA de Rennes, CentraleSupélec et Université de Nantes.

En relation aux partenariats avec autres instituts et entreprises, L'IETR a une liste considérable de partenaires, incluent des centres publiques comme CEA, CNES et Club Automatique et Automatisation industrielle de la SEE, petites et moyennes entreprises privés comme A&P Lithos, Adlightec et Advansee, et grandes groupes comme Alstom, EDF et Mitsubishi.

Son partenariat International compte sur plus de 70 Universités, Instituts et Agences de recherche parmi tout le monde, incluent NASA Jet Propulsion Lab - Caltech - USA (JPL), Politecnico Milano - Italia (POLIMI) et Universidade de São Paulo - Brasil (USP).

2.1.2 Division des Équipes de Recherche

Afin de meilleur catégoriser les thématiques des projets de recherche L'IETR est divisé en 6 départements/équipes :

- Antennes & Dispositifs Hyperfréquences (ADH)
- Signal & Communications (SC)
- Ondes & Signaux (OS)
- Image
- Microélectronique & Microcapteurs (MM)
- Automatique (AUT)

L'organigramme structurel avec tant les parties de recherche quant les parties administratifs du IETR peut être vu dans la figure 2.1.

2.1.3 L'équipe AUT

L'équipe de Automatique est **basé** a CentraleSupélec et travaille dans diverses thématiques utilisant les connaissances des domaines d'analyse et commande des systèmes hybrides.

Ses projets ont des applications **que** couvrent diverses métiers, on peut voir quelques exemples de projets : comme dans le métier des bâtiments intelligents (régulation de chauffage pour augmenter l'efficacité énergétique), de la santé (models des pancréas artificiels pour contrôler le métabolisme de glucose), du transport (stratégies de régulation de vitesse des automobiles) , de la distribution d'énergie (Études de stabilisation en tension de générateurs distribués dans un réseau de distribution) entre autres.

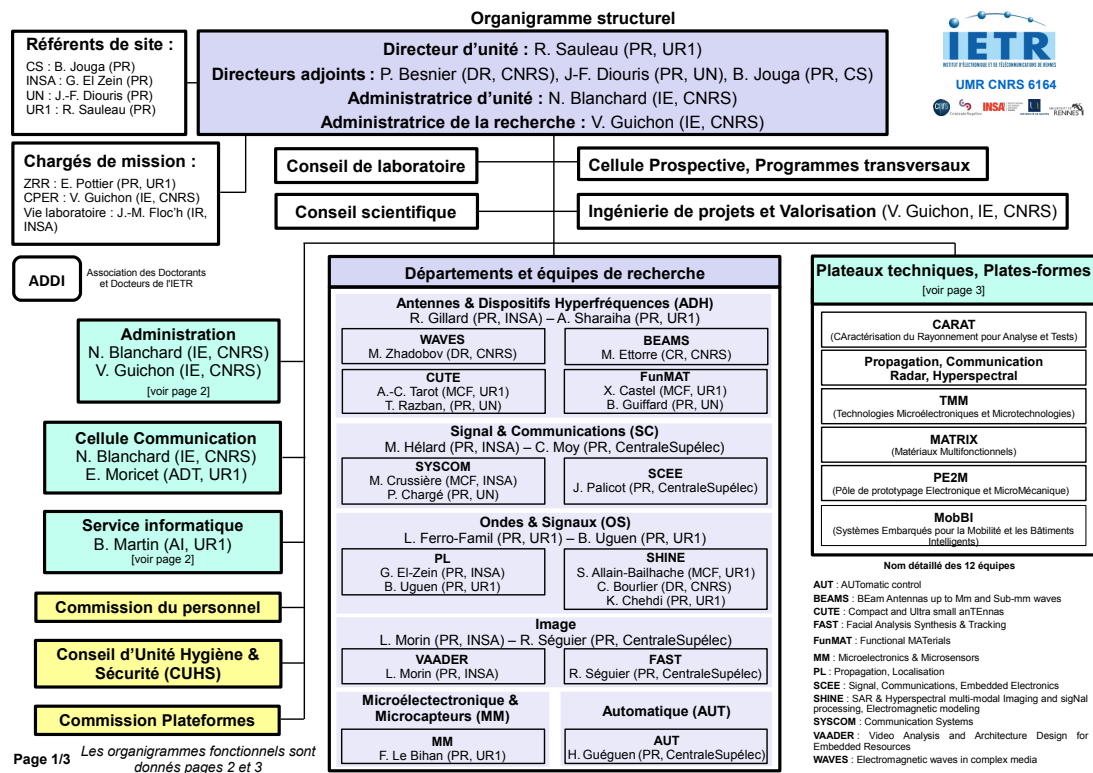


Figure 2.1 – Organigramme de l'IETR.

2.2 Le Projet

L'objectif du projet réalisé pendant le stage était faire des simulations d'un réseaux de distribution électrique et **développer** des régulateur **des** tension des bus du circuit **tout ça** utilisant les logiciels MATLAB et DigSILENT PowerFactory.

Les résultats **obtenues** a partir de ces simulations ont été interprété et comparé avec les résultats provenant du travaille de Marjorie Cosson [1] **a fin** d'obtenir une revalidation de ses conclusions.

3 Division du travail

Pour faire le travail un peu plus simple en concentrant les efforts en petites tâches chacune par soi, optimisant le temps. Le travail était divisé en plusieurs *Work Packages*, chacun composé par petites tâches. La division et ses respectives tâches peuvent être vu dans les sections 3.1 à 3.5 suivantes.

3.1 Première Partie - Lecture

La première partie consistait en lire le article de WAN Yidong [4], qu'explique d'une façon un peu simplifiée le problème et montre des façons de calculer les gains entre la tension des bus et la puissance réactive des charges, en utilisant scripts écrits en DlgSILENT Programming Language (DPL) dans le logiciel PowerFactory et la création d'une matrice de gain.

Après la lecture du article, a fin de comprendre le problème un peu mieux, la lecture de quelques parties de la thèse de Marjorie Cosson [1] a été faite.

Autres lectures supplémentaires ont été faites, [2] et [3]. Ces articles utilisent le même réseaux que [1] et quelques données ont aidé pour la reconstruction du réseaux dans le PowerFactory.

3.2 Deuxième Partie - Mise en Main

Après la lecture des documents commençait l'étude et pris en main du logiciel DlgSILENT PowerFactory, en lisant et regardant les tutoriels a l'internet, en faisant quelques petits exemples du logiciel a fin d'apprendre a utiliser les outils nécessaires pour faire les tests proposés et après, faire la montage de la modèle du réseaux dans le PowerFactory, le diagramme montré dans la figure 3.1.

3.3 Troisième Partie - Programmation

Pendant ce partie diverses scripts ont été créés en utilisant les langages MATLAB et Python :

- Pour charger les valeurs de puissance des charges.
- Pour charger les valeurs de puissance des générateurs.
- Pour calculer les gains entre les bus et les générateurs.
- Pour faire des matrices de gains.
- Pour créer des événements de charges et générateurs, faire des simulations et prendre les résultats en graphiques.

3.4 Quatrième Partie - Intégration

Pendant la quatrième partie, l'interface entre le PowerFactory et MATLAB a été requise a fin de créer une modèle de régulateur au simulink et utiliser dans le PowerFactory.

3.5 Cinquième Partie - Rédaction

La cinquième partie consistait en élaborer des rapports et autres documents de description du projet, comme ce document par exemple.

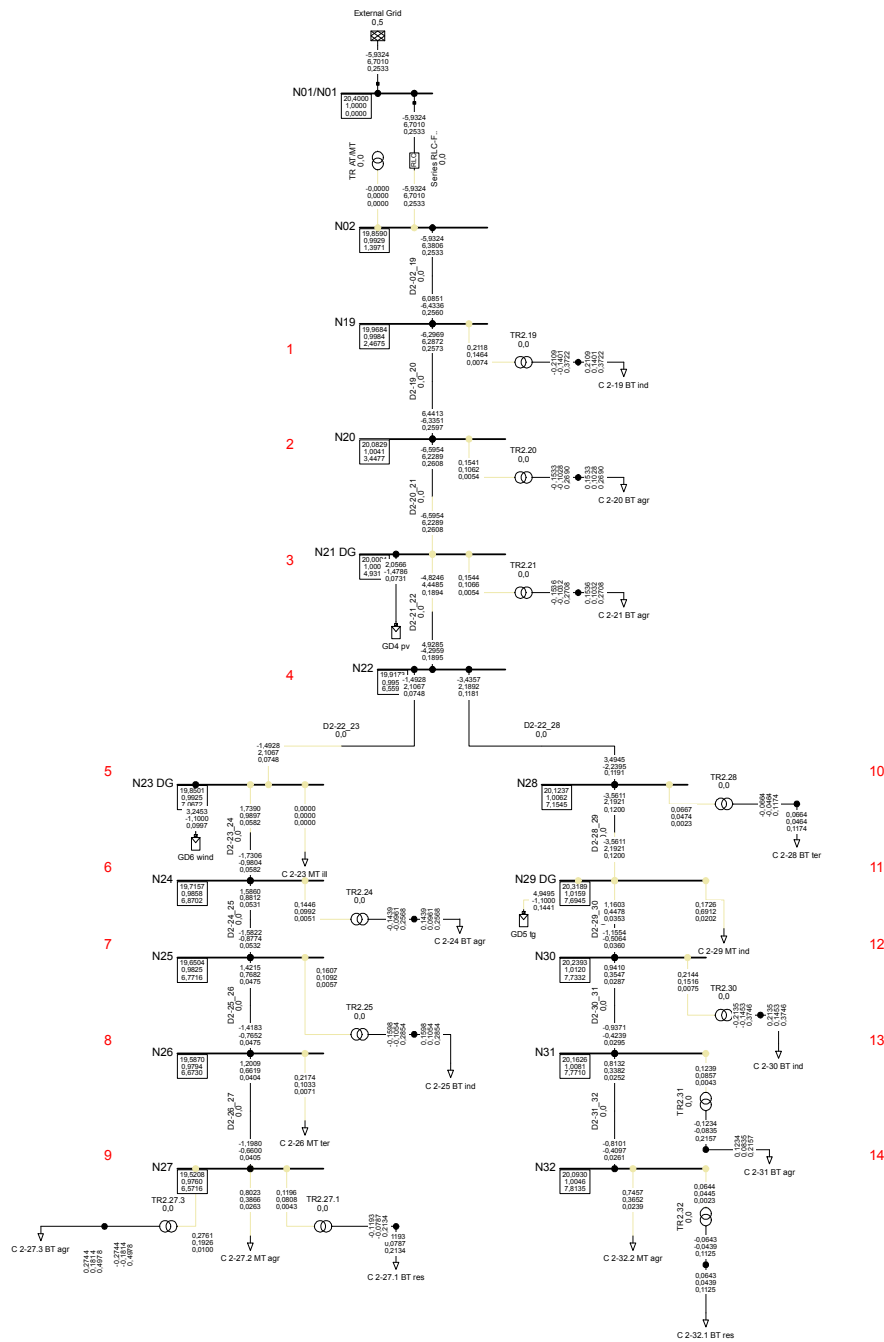


Figure 3.1 – Diagramme du réseau

4 Méthodologie

Cette section a comme but expliquer les méthodes utilisées pour faire chaque tâche des parties 2 à 4 du travail, décrits dans les sections 4.1 à 4.4.

4.1 Deuxième Partie - Mise en Main

Comme dit, le logiciel DlgSILENT PowerFactory a été utilisé pour faire des simulations du réseau, donc une petite explication du réseau et du logiciel sera faite dans cette section.

4.1.1 À propos du DlgSILENT PowerFactory

Le DlgSILENT PowerFactory est un logiciel beaucoup utilisé dans le métier d'Énergie, par entreprises comme Électricité de France (EDF), l'ancienne ERDF (Enedis) pour faire des simulations des réseaux électriques, que les permettent de vérifier la stabilité en cas de panne et surcharge ou sous-charge de parties du réseau, calculer les coûts d'opération et même programmer d'autres changements futurs du réseau. Quelques autres instituts comme L'IEETR et POLIMI, par exemple, l'utilisent pour ses thèmes de thèses et autres recherches.

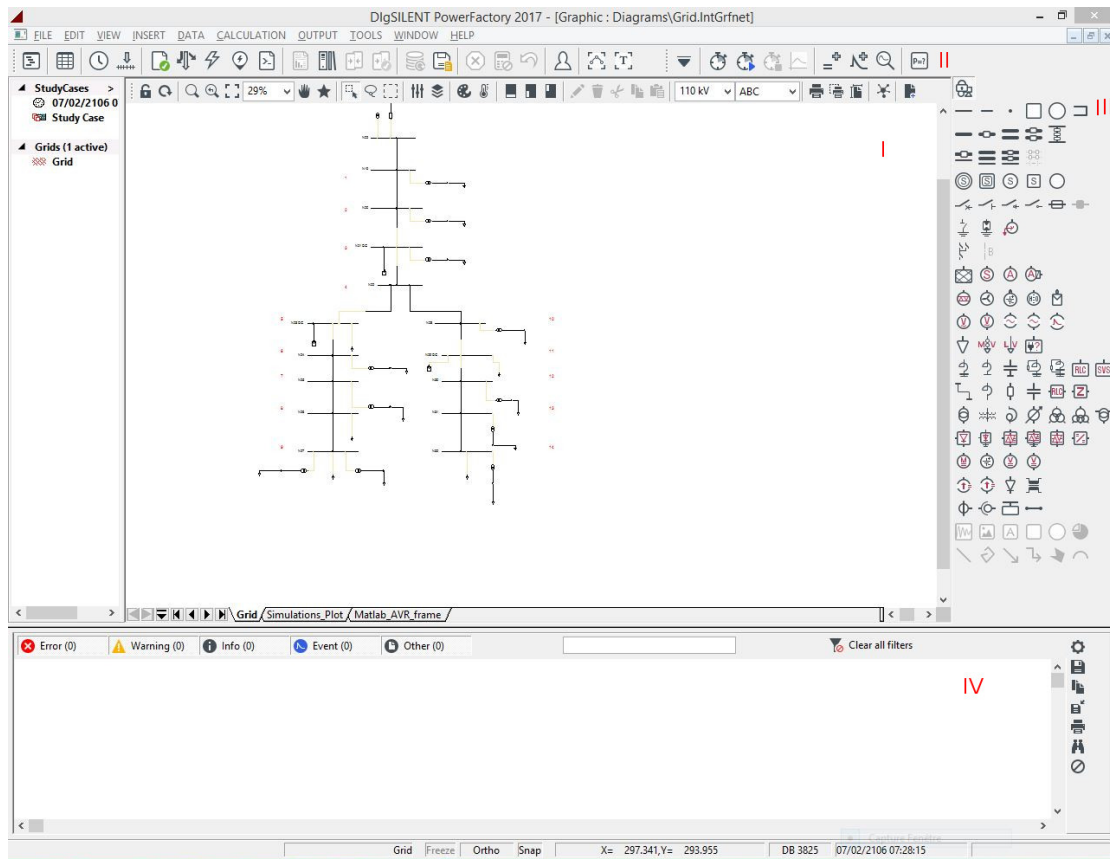


Figure 4.1 – Interface Graphique d'utilisateur (GUI) du DlgSILENT PowerFactory.

En peut voir dans la figure 4.1 quelques panneaux basiques importants du GUI du DIgSILENT PowerFactory

I Panneau Graphique

Où les diagrammes sont affichés, tant réseaux **quant** graphiques.

II Panneau des Outil

Où sont concentrés les outils du programme, pour modifier l’affichage, donnant une couleur différente par chaque bus par exemple ; faire des diverses types de simulation, de Court-circuit, calculs de flux de charge simulation Root Mean Square - Moyenne quadratique (RMS) et Electro Magnetic Transient (EMT), etc.

III Panneau de Dessin

Outils pour dessiner des éléments du réseau, comme bus, transformateur, charge etc

IV Panneau de Sortie

Où sont montrés les résultats des calculs et simulations, les avertissement et les erreurs.

4.2 À propos du réseau

Comme était dit, la figure 3.1 démontre le réseau utilisé. Si on compare avec le réseaux utilisé en [1] et [3], il est possible de voir que le réseaux de la figure 3.1 représente juste la moitié **du** original. **Cette** choix de prendre une partie du réseaux a comme raison la diminution des éléments et conséquemment la complexité des résolutions des simulations.

Le réseau est formé **pour** 16 charges **et** 3 générateurs distribués, 12 transformateurs. Dans le réseau original deux des trois générateurs étaient des machines synchrones mais elles ont été remplacées par des panneaux photovoltaïques, **a fin** de faire les réponses des tests plus vite, en vue de la dynamique des panneaux considérablement plus vite que des machines synchrones, qui dépendent des **constants** électromécaniques.

A fin de faire une meilleur description du réseaux **aux** les tableaux 1 à 7 **ont** les caractéristiques des éléments qui **la** composent.

Tableau 1 – Générateurs Distribués du Réseau.

GD	P[MW] nominal	P[MW] 1p.m.
GD4	3.2	2.056124
GD5	5.5	4.94595
GD6	5.5	3.245381

Tableau 2 – Transformateurs HV/MV.

Model	40 MVA132/20
Puissance	50MVA
Pertes Cuivre	176kW
Tension de court-circuit Relative	15.5%
Taps	12
Tension per Tap	1.5%

Tableau 3 – Transformateurs MV/LV.

Modèle	0.25MVA 20kV/0.4	0.4MVA 20kV/0.4	0.63MVA 20kV/0.4
Puissance	250kVA	400kVA	630kVA
Pertes Cuivre	2.6kW	3.7 kW	5.6kW
Tension de court-circuit Relative	4%	4%	4%
Nombres de Transformateurs	1	6	4

Tableau 4 – Transformateurs.

Nom	Modèle
TR AT/MT	40 MVA132/20
TR 2.19	0.63MVA 20kV/0.4
TR 2.20	0.4MVA 20kV/0.4
TR 2.21	0.4MVA 20kV/0.4
TR 2.24	0.4MVA 20kV/0.4
TR 2.25	0.4MVA 20kV/0.4
TR 2.27.1	0.63MVA 20kV/0.4
TR 2.27.3	0.63MVA 20kV/0.4
TR 2.28	0.25MVA 20kV/0.4
TR 2.30	0.63MVA 20kV/0.4
TR 2.31	0.4MVA 20kV/0.4
TR 2.32	0.4MVA 20kV/0.4

Tableau 5 – Caractéristiques des Lignes.

Nom	Genre	Longueur[km]
D2-02_19	ARG7H1RX 185mmq	3.6
D2-19_20	ARG7H1RX 185mmq	3.304
D2-20_21	Aerea Cu 70mmq	2.4
D2-21_22	Aerea Cu 70mmq	3.6
D2-22_23	Aerea Cu 70mmq	3
D2-22_28	ARG7H1RX 70mmq	2.4
D2-23_24	Aerea Cu 70mmq	3.08
D2-24_25	Aerea Cu 70mmq	1.65
D2-25_26	Aerea Cu 70mmq	1.8
D2-26_27	Aerea Cu 70mmq	2.2
D2-28_29	ARG7H1RX 70mmq	2.2
D2-29_30	ARG7H1RX 70mmq	2.4
D2-30_31	ARG7H1RX 70mmq	2.6
D2-31_32	ARG7H1RX 70mmq	2.7

Tableau 6 – Lignes.

Nom	Genre	Section[mm ²]	R[Ω/km]	L[mH/km]	C[μF/km]
ARG7H1RX 185mmq	Câble	185	0.2180	0.350	0.2900
ARG7H1RX 70mmq	Câble	70	0.5800	0.41	0.2100
Aerea Cu 70mmq	Aérien	70	0.2681	1.286	0.0090

Tableau 7 – Charges.

Nom	Genre	P[<i>MW</i>]1p.m.	Q[<i>MVAR</i>]1p.m.
C 2-19	LV	0.1894	0.1265088
C 2-20	LV	0.1147	0.0774413
C 2-21	LV	0.1155	0.0782289
C 2-23	MV	0	0.1
C 2-24	LV	0.1094	0.0741473
C 2-25	LV	0.1450	0.0984401
C 2-26	MV	0.3993	0.2049369
C 2-27.1	LV	0.2471	0.1656134
C 2-27.2	MV	.6083	0.2971269
C 2-27.3	LV	0.2094	0.1407233
C 2-28	LV	0.1205	0.08741
C 2-29	MV	0.1561	0.0798601
C 2-30	LV	0.1934	0.1347733
C 2-31	LV	0.0934	0.0640347
C 2-32.1	LV	0.1333	0.0923274
C 2-32.2	MV	0.5634	0.2791258

4.3 Troisième Partie - Programmation

Pendant **ce** partie, **diverses** scripts ont été **crées** en utilisant les langages MATLAB et Python, en intégration des bibliothèques Python du DIgSILENT PowerFactory, qui **pourvoit** quelques fonctions pour accéder aux éléments du réseau, changer ses caractéristiques et faire des mesures de courant, tension etc.

Dans cette section chaque script est décrit par son fonctionnement, pour une description minutieuse, les codes sont disponibles dans le site https://github.com/Accacio/scripts_powerfactory.

4.3.1 gain_calc – load2bus.py

Ce script change les valeurs de puissance réactive demandé par les charges pendant un intervalle de temps et prend les valeurs de tension du bus auquel la charge est connecté pendant ce temps et calcule le gain entre chaque charge et bus, formant une matrice de dimension 16×16 , avec le format suivant :

Tableau 8 – Matrice des Gain de sortie du script *gain_calc – load2bus.py*.

$\frac{V_{N01}}{Q_{C2-19}}$	$\frac{V_{N02}}{Q_{C2-19}}$	$\frac{V_{N19}}{Q_{C2-19}}$	$\frac{V_{N20}}{Q_{C2-19}}$	\dots	$\frac{V_{N32}}{Q_{C2-19}}$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
$\frac{V_{N01}}{Q_{C2-27.1}}$	$\frac{V_{N02}}{Q_{C2-27.1}}$	$\frac{V_{N19}}{Q_{C2-27.1}}$	$\frac{V_{N20}}{Q_{C2-27.1}}$	\dots	$\frac{V_{N32}}{Q_{C2-27.1}}$
$\frac{V_{N01}}{Q_{C2-27.2}}$	$\frac{V_{N02}}{Q_{C2-27.2}}$	$\frac{V_{N19}}{Q_{C2-27.2}}$	$\frac{V_{N20}}{Q_{C2-27.2}}$	\dots	$\frac{V_{N32}}{Q_{C2-27.2}}$
$\frac{V_{N01}}{Q_{C2-27.3}}$	$\frac{V_{N02}}{Q_{C2-27.3}}$	$\frac{V_{N19}}{Q_{C2-27.3}}$	$\frac{V_{N20}}{Q_{C2-27.3}}$	\dots	$\frac{V_{N32}}{Q_{C2-27.3}}$
$\frac{V_{N01}}{Q_{C2-28}}$	$\frac{V_{N02}}{Q_{C2-28}}$	$\frac{V_{N19}}{Q_{C2-28}}$	$\frac{V_{N20}}{Q_{C2-28}}$	\dots	$\frac{V_{N32}}{Q_{C2-28}}$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
$\frac{V_{N01}}{Q_{C2-32.1}}$	$\frac{V_{N02}}{Q_{C2-32.1}}$	$\frac{V_{N19}}{Q_{C2-32.1}}$	$\frac{V_{N20}}{Q_{C2-32.1}}$	\dots	$\frac{V_{N32}}{Q_{C2-32.1}}$
$\frac{V_{N01}}{Q_{C2-32.2}}$	$\frac{V_{N02}}{Q_{C2-32.2}}$	$\frac{V_{N19}}{Q_{C2-32.2}}$	$\frac{V_{N20}}{Q_{C2-32.2}}$	\dots	$\frac{V_{N32}}{Q_{C2-32.2}}$

Où V_{Nxx} et Q_{Cx-xx} sont la tension du bus Nxx e la puissance réactive de la charge $Cx-xx$.

4.3.2 gain_calc – generator2bus – test_1.py

Ce script change les valeurs de puissance réactive des générateurs pendant un intervalle de temps **e prends** les valeurs de tension du bus auquel le générateur est connecté pendant ce temps et calcule le gain entre chaque générateur et bus, formant une matrice de dimension 3×3 . L'allure de la courbe utilisé peut être vu dans la figure 4.2. Le format de la matrice est montré dans le tableau 9.

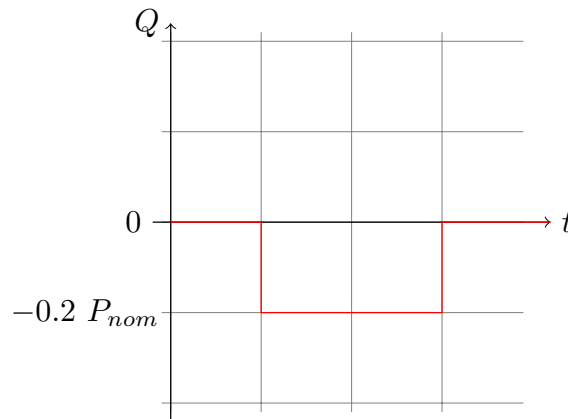


Figure 4.2 – Allure de la courbe utilisé pendant le Test 1.

Tableau 9 – Matrice des Gain de sortie du script
gain_calc – generator2bus – test_1.py.

$\frac{V_{N21}}{Q_{GD4}}$	$\frac{V_{N29}}{Q_{GD4}}$	$\frac{V_{N23}}{Q_{GD4}}$
$\frac{V_{N21}}{Q_{GD5}}$	$\frac{V_{N29}}{Q_{GD5}}$	$\frac{V_{N23}}{Q_{GD5}}$
$\frac{V_{N21}}{Q_{GD6}}$	$\frac{V_{N29}}{Q_{GD6}}$	$\frac{V_{N23}}{Q_{GD6}}$

Où Q_{GDx} est la puissance réactive du générateur GDx .

4.3.3 gain_calc – generator2bus – test_2.py

Fondamentalement le même que *gain_calc – generator2bus – test_1.py*, mais la courbe de changement de puissance **suivre** l'allure de la figure 4.3. Le format de la matrice est le même montré dans le tableau 9.

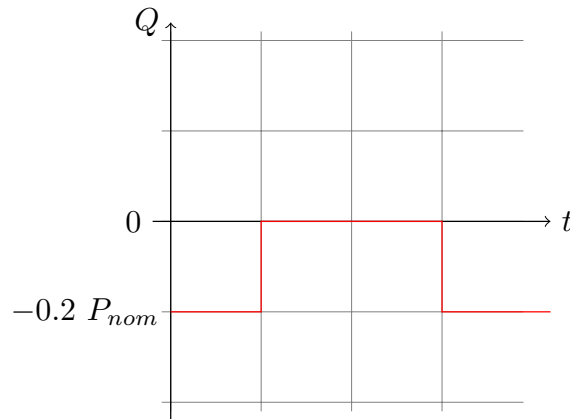


Figure 4.3 – Allure de la courbe utilisé pendant le Test 2.

4.3.4 import_generator_csv.py

Ce script prend des fichiers Comma Separated Values (CSV) qui a données de puissance nominale, active et réactive des générateurs, et les charge en chaque générateur.

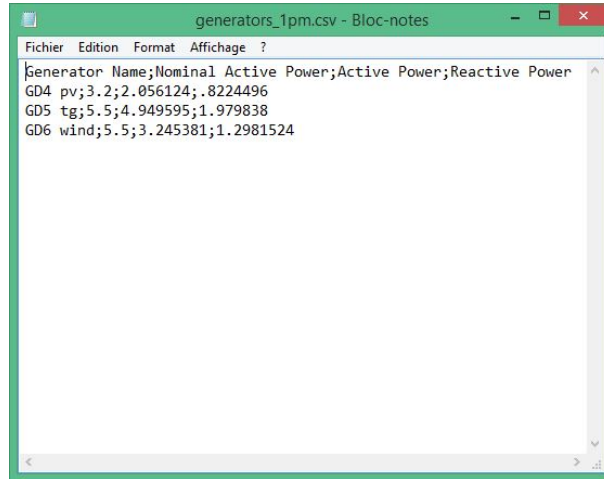


Figure 4.4 – Exemple d'un fichier CSV, séparé par point-virgule.

La choix d'utiliser fichiers CSV a été fait **en raison de** ce type de fichier **est** texte **sans** brut sans formatage, sa taille est très réduite. Et ce type de fichier est très simple pour faire des modifications et création a partir d'un tableau `.xls`, ainsi que la conversion entre ces deux extensions.

4.3.5 import_load_csv.py

Ce script prend des fichiers CSV qui a données de puissance active et réactive des charges, et les charge en chaque respective charge.

4.3.6 csv2mat.m

Ce script fait en MATLAB prend un fichier CSV sorti d'une simulation du DlgSILENT PowerFactory et le transforme en un fichier `.mat`. Il crée une structure composés par **autres** structures qui sont les éléments du circuit et composées par ses données pendant le temps, **a fin** de les utiliser pour faire des graphiques.

4.3.7 printpdf.m

Ce script a été créé pour faire les plots des graphiques et les sauvegarder en fichier `.pdf` pour **un futur use** dans les rapports (ce rapport par exemple).

4.3.8 teste_simul.py

Ce script crée un événement des changement de valeur de puissance active de la charge C 2-29 MT ind en augmentant sa valeur en 100% par un période de temps et après prend les valeurs de puissance active et réactive de cette même charge et la tension des bus N21 N23 et N29 (où les générateurs sont connectés) pendant le temps de la simulation et en exportant en un fichier .csv.

4.4 Quatrième Partie - Intégration

Le but de cette partie était reproduire les simulations du réseau (figure 3.1) en utilisant le régulateur vu en [1], montré dans la figure 4.5.

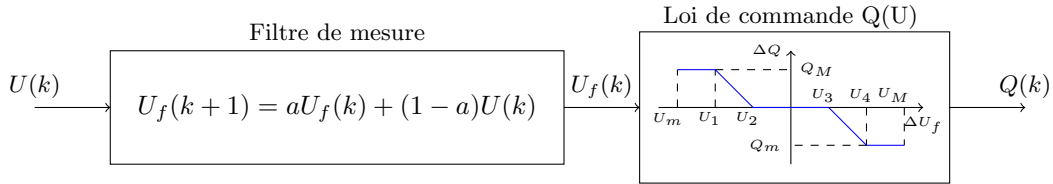


Figure 4.5 – Diagramme représentative du régulateur.

Les valeurs utilisés pour la loi de commande sont dans le tableau suivante :

Tableau 10 – Matrice des Gain de sortie du script

Variable	Valeur
U_m	18 kV
U_1	19 kV
U_2	19,25 kV
U_3	20,75 kV
U_4	21 kV
U_M	21 kV
Q_m	-0,4 P_{max}
Q_M	0,4 P_{max}

Pour construire le modèle du régulateur dans le DIgSILENT PowerFactory, il y a deux façons, faire toute la construction dedans PowerFactory en utilisant sa langage propriétaire de description des systèmes, DIgSILENT Simulation Language (DSL), ou faire une intégration avec MATLAB et utiliser les outils des *ToolBox* déjà créés.

A fin de faire l'intégration entre MATLAB et DIgSILENT PowerFactory, au minimum deux choses sont nécessaires :

1. Un Bloc générique défini au PowerFactory qui fait l'appel a un fichier .m
2. Un fichier .m qui décrit le fonctionnement du bloc

Comme ont voulait faire un régulateur dans le Simulink, un autre fichier, .mdl, était nécessaire. Ce fichier est appelé par le fichier .m et le donne les sorties de simulation.

La période d'échantillonnage du filtre numérique est de 1s et le pas de la simulation au PowerFactory est de 10ms. Comme une façon de régler la fréquence

d'appel au MATLAB n'était pas trouvé, la solution plus simple était l'implémentation du filtre dehors le MATLAB, donc un échantillonneur et le propre filtre ont été construits dans l'interface du PowerFactory, comme on peut voir dans les figures 4.6 à 4.8 et dans le simulink la loi de commande 4.9.

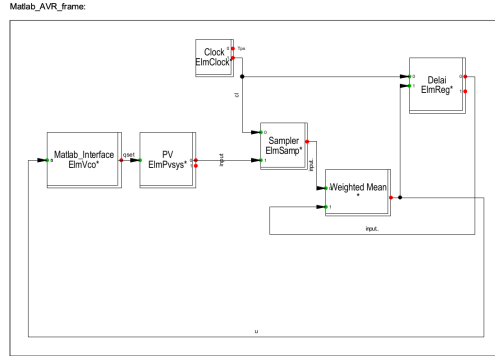


Figure 4.6 – Diagramme du régulateur assemblé dans PowerFactory.

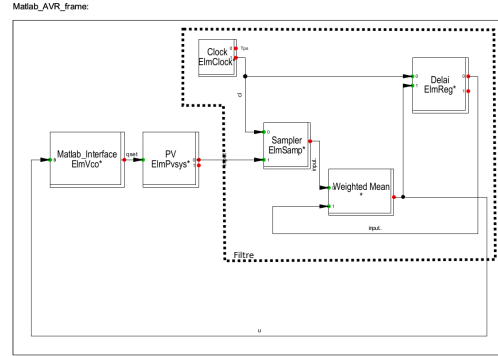


Figure 4.7 – Filtre et Échantillonneur implémentés dans PowerFactory.

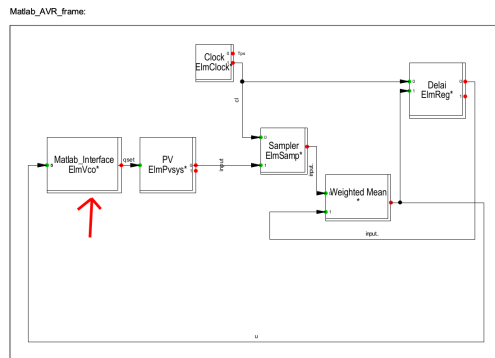


Figure 4.8 – Bloc Générique Matlab.

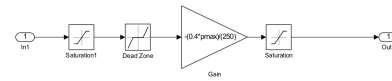


Figure 4.9 – Loi de commande en Simulink.

Le diagramme du modèle est dans la figure 4.10, et comme on peut voir, c'est essentiellement la même chose, le code dedans le bloc générique qui a été changé.

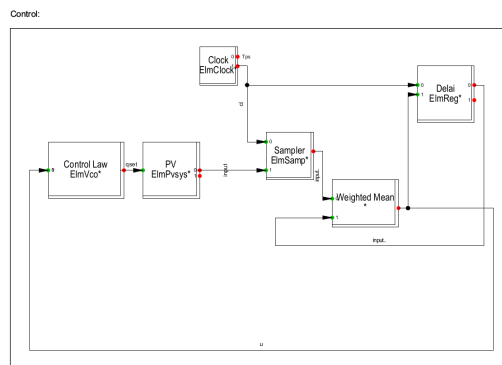


Figure 4.10 – Régulateur fait seulement en PowerFactory.

4.4.1 À propos des temps de simulation

Après faites toutes connexions et quelques tests avec le régulateur, ont a vu qu'une simulation de 50s a duré approximativement **5min**. Donc deux autres modèles ont été créés a fin de comparaison. Un avec **toute** le régulateur dans le PowerFactory et un autre avec la loi de commande fait dans le MATLAB (le seule changement était le code `.m`). Avec ces deux modèles une simulation de 50s a duré approximativement 50s. On peut voir dans le tableau 11 la comparaison entre les temps de simulation :

Tableau 11 – Temps réel de simulation en secondes
pour une simulation de 10s

	Cond. Ini.	Simulation
Sans Contrôle	1	10
Powerfactory	1	10
Powerfactory \Leftrightarrow MATLAB	20	12
Powerfactory \Leftrightarrow MATLAB \Leftrightarrow Simulink	57	140

On peut vérifier que il ne vaut pas la peine utiliser le Simulink pour faire ce genre de simulation **cas il soit appelé a chaque 0.01s**. Nouveaux tests doivent être faits **a fin de confirmer si même faisant appels a chaque seconde si les temps se maintiennent.**

5 Résultats

5.1 Scripts

En dessous les résultats **provenus** des scripts créés dans la troisième partie du projet **et après une** bref discussion.

— **gain_calc – generator2bus**

— **gain_calc – generator2bus – test_1 – 7am**

Tableau 12 – Matrice des Gain entre Puissance Réactive des
générateurs et la tension des Bus (Valeurs numériques)

1.7e-4	1.7e-4	1.7e-4
1.7e-4	2.5e-4	2.7e-4
1.7e-4	3.1e-4	2.4e-4

— **gain_calc – generator2bus – test_1 – 1pm**

Tableau 13 – Matrice des Gain entre Puissance Réactive des
générateurs et la tension des Bus (Valeurs numériques)

1.7e-4	1.7e-4	1.6e-4
1.7e-4	2.4e-4	2.7e-4
1.7e-4	3.0e-4	2.4e-4

— gain_calc – generator2bus – test_2

Tableau 14 – Matrice des Gain entre Puissance Réactive des générateurs et la tension des Bus (Valeurs numériques)

1.8e-4	1.8e-4	1.7e-4
1.8e-4	2.5e-4	2.8e-4
1.8e-4	3.1e-4	2.5e-4

— gain_calc – load2bus

A cause de sa taille, le tableau des résultats 15 sont dans une autre page.

— teste_simul

Testé sans régulateur on peut voir le gain négative du système dans les figures 5.1 et 5.2.

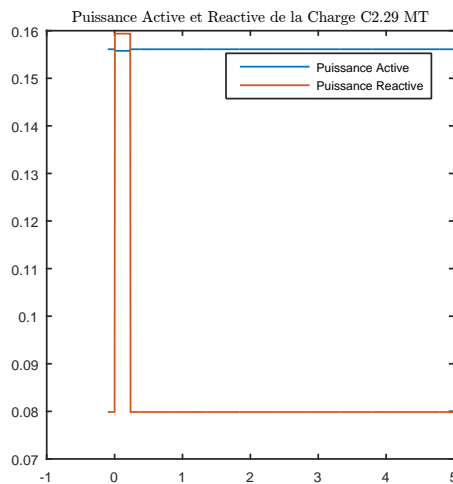


Figure 5.1 – Puissance Active et Réactive de la Charge C2 29 MT

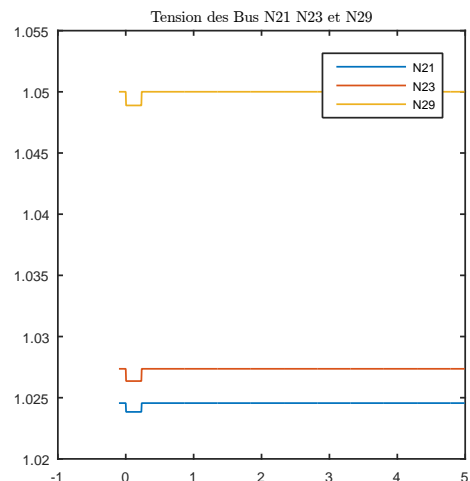


Figure 5.2 – Tension des Bus N21 N23 et N29 en p.u.

On peut voir a partir de ces données que le réseau est caractérisée pour les équations démontrés en [1], la tension du bus a un gain positif par rapport a puissance réactive du générateur, correspondant au qu'on espère pour la bonne réponse du régulateur.

Tableau 15 – Matrice des Gain entre Puissance Réactive des charges et la tension des Bus

[illegible]

5.2 Simulations avec le régulateur

Comme les réponses étaient les mêmes entre les modèles Powerfactory, Powerfactory \leftrightarrow MATLAB et Powerfactory \leftrightarrow MATLAB \leftrightarrow Simulink, on a choisi d'afficher les résultats du modèle Powerfactory \leftrightarrow MATLAB, afin d'avoir encore la communication entre les deux logiciels.

Ici se présentent les réponses des simulations et après une bref discussion sur les résultats.

Les premiers tests étaient pendant un intervalle de 50s et sans perturbations juste pour voir se le système reste avec sa tension **dedans** la zone permise, ça veut dire entre $\pm 5\%$ du valeur de tension nominal, $20kV$, donc l'intervalle est entre $19kV$ et $21kV$. Et on varie le paramètre a **fu** filtre a fin de voir la différence **qu'il** **pose au** système.

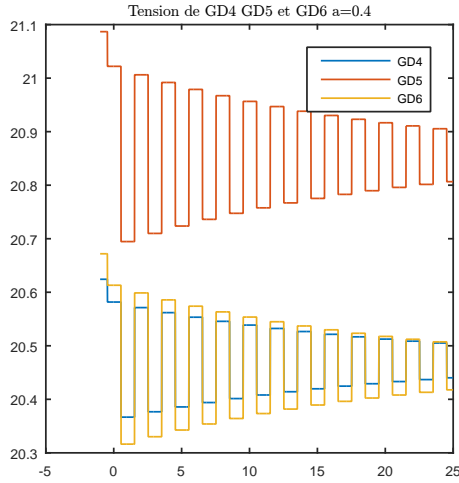


Figure 5.3 – Tension de GD4 sans perturbation $a = 0.4$.

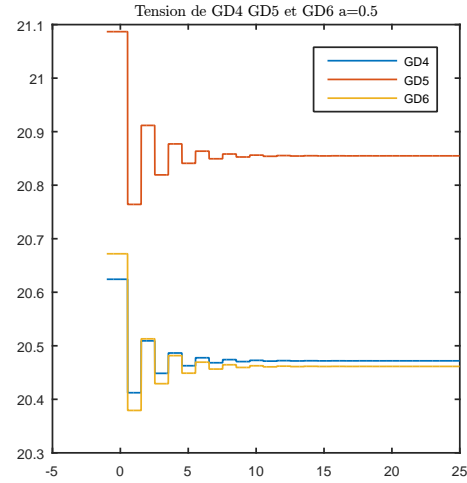


Figure 5.4 – Tension de GD4 sans perturbation $a = 0.5$.

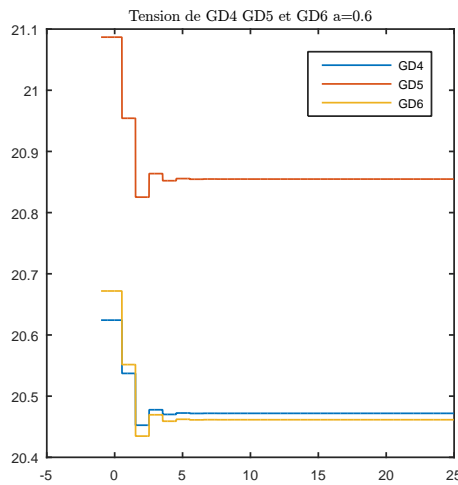


Figure 5.5 – Tension de GD4 sans perturbation $a = 0.6$.

On peut voir que le régulateur fonctionne, parce que les valeurs de tension restent dans l'intervalle proposé, et on peut percevoir que quand le paramètre a est plus proche de 0 le comportement du système commence à être oscillatoire.

Après ces tests, deux autres tests ont été faits, les deux avec perturbations a une charge (C2-29), le premier avec un échelon de puissance réactive et le seconde avec un échelon de puissance active. Les deux **les** augmentent en 50%. Comme les réponses des trois générateurs sont semblables, juste les réponses d'un générateur sont montrés dans les figures suivantes **a fin** de faciliter la visualisation.

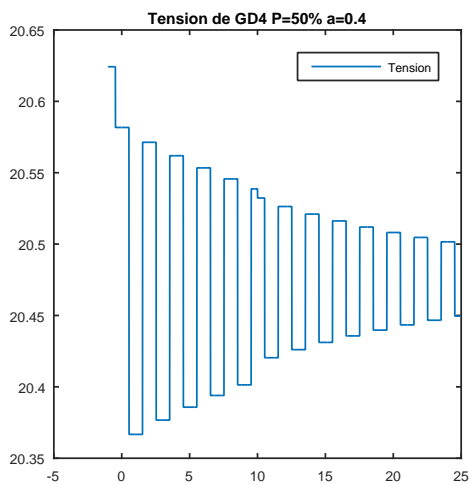


Figure 5.6 – Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.4$.

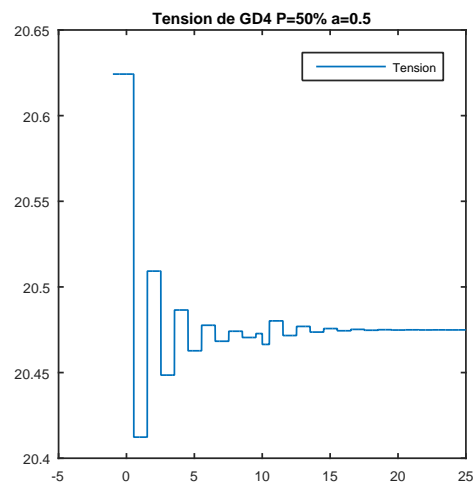


Figure 5.7 – Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.5$.

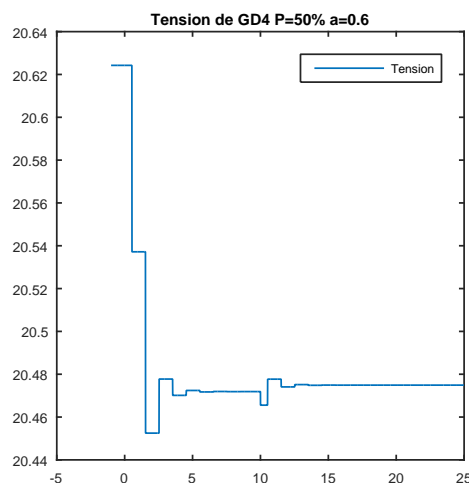


Figure 5.8 – Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.6$.

Pour le perturbation de puissance réactive :

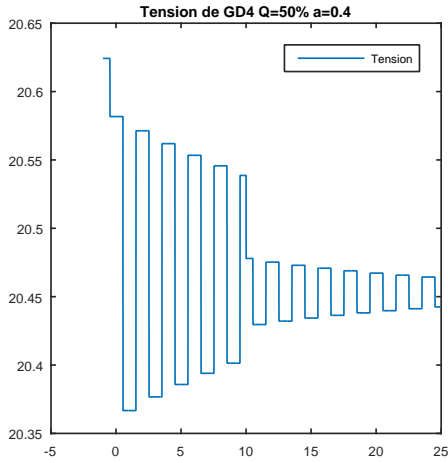


Figure 5.9 – Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.4$.

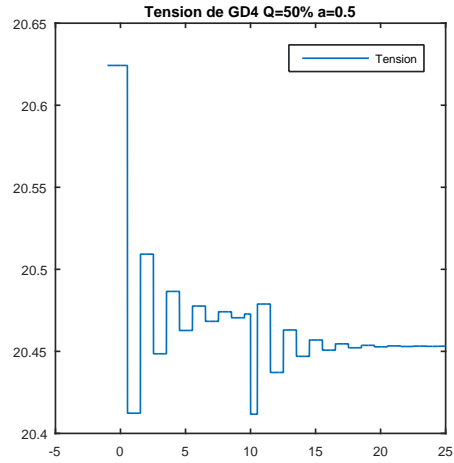


Figure 5.10 – Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.5$.

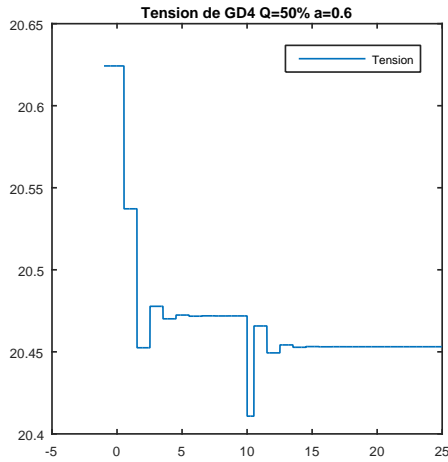


Figure 5.11 – Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.6$.

On peut voir, le même comportement. Quand a est plus proche de 0 la réponse commence à avoir oscillations. Mais même ainsi la tension reste dans l'intervalle, pour ces valeurs de a .

6 Difficultés

Pendant le projet quelques difficultés ont été trouvées et les principales sont les suivantes :

- Long temps de calcul
Le modèle avec le régulateur implémenté dans simulink prend 5 min à peu près, ce qui a difficile le *debugging* du système comme la génération des résultats.
- Tous blocs en PowerFactory sont synchrones
Dû à ça, le filtre a été implémenté dehors simulink, augmentant la complexité du modèle, 4 blocs plus ont été créés.
- Conditions Initiales
A cause des différents blocs utilisés chaque bloc avait besoin d'avoir ses conditions initiales cohérentes entre elles, ce qui causait de problèmes car elles ne fussent pas cohérents.
- Documentation du DlgSILENT PowerFactory
Documentation du logiciel n'est pas bien détaillée, causant quelquefois ambiguïté, créant le besoin de chercher l'information en autres lieux.
- Communauté PowerFactory presque inexistant
Pour trouver des informations il fallait chercher à l'internet, mais la principale source était le faq du PowerFactory, qu'est si réticent quant la documentation.

7 Prochains Travaux

Pour prochains travaux quelques choses peuvent être suggérées :

- Implémenter un bloc dans PowerFactory que soit appelé en temps différent du pas de la simulation.
- Réduire temps de communication PowerFactory \leftrightarrow Matlab
- Automatiser les diverses simulations et la génération des figures.

8 Conclusions

On peut arriver à quelques conclusions après avoir vu toute la méthodologie utilisée et les résultats en graphiques et tableaux, et elles sont :

- **Scripting en PowerFactory**

Il est possible d'utiliser Python et la combinaison entre l'API PowerFactory avec les bibliothèques Python déjà créées permet une gamme de possibilités.

- **Intégration Matlab \Leftrightarrow PowerFactory**

Assez facile si toutes les variables du système sont préalablement connues.

- **Stabilité **du** système dépend de la valeur de a**

Régulateur fonctionne mais dépend de a pour rester stable, valeurs plus proches de 1 font le système stabiliser plus vite.

Liste des tableaux

1	Générateurs Distribués du Réseau.	11
2	Transformateurs HV/MV.	11
3	Transformateurs MV/LV.	12
4	Transformateurs.	12
5	Caractéristiques des Lignes.	12
6	Lignes.	12
7	Charges.	13
8	Matrice des Gain de sortie du script <i>gain_calc – load2bus.py</i>	14
9	Matrice des Gain de sortie du script <i>gain_calc – generator2bus – test_1.py</i>	15
10	Matrice des Gain de sortie du script	17
11	Temps réel de simulation en secondes pour une simulation de 10s .	19
12	Matrice des Gain entre Puissance Réactive des générateurs et la tension des Bus (Valeurs numériques)	19
13	Matrice des Gain entre Puissance Réactive des générateurs et la tension des Bus (Valeurs numériques)	19
14	Matrice des Gain entre Puissance Réactive des générateurs et la tension des Bus (Valeurs numériques)	20
15	Matrice des Gain entre Puissance Réactive des charges et la tension des Bus	21

Table des figures

2.1	Organigramme du IETR.	7
3.1	Diagramme du reseau	9
4.1	GUI du DIgSILENT PowerFactory.	10
4.2	Allure de la courbe utilisé pendant le Test 1.	14
4.3	Allure de la courbe utilisé pendant le Test 2.	15
4.4	Exemple d'un fichier CSV, séparé par point-virgule.	16
4.5	Diagramme représentative du régulateur.	17
4.6	Diagramme du régulateur assemblé dans PowerFactory.	18
4.7	Filtre et Échantillonneur implémentées dans PowerFactory.	18
4.8	Bloc Générique Matlab.	18
4.9	Loi de commande en Simulink.	18
4.10	Régulateur fait seulement en PowerFactory.	18
5.1	Puissance Active et Réactive de la Charge C2 29 MT	20
5.2	Tension des Bus N21 N23 et N29 en p.u.	20
5.3	Tension de GD4 sans perturbation $a = 0.4$	22
5.4	Tension de GD4 sans perturbation $a = 0.5$	22
5.5	Tension de GD4 sans perturbation $a = 0.6$	22
5.6	Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.4$	23
5.7	Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.5$	23
5.8	Tension de GD4 avec perturbation en puissance active de 50% et $a = 0.6$	23
5.9	Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.4$	24
5.10	Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.5$	24
5.11	Tension de GD4 avec perturbation en puissance réactive de 50% et $a = 0.6$	24

Références

- [1] Marjorie Cosson. *Stability of a distribution electrical network. Analysis from a complex system point of view*. Theses, Université Paris-Saclay, September 2016.
- [2] Marcello Farina, Antonio Guagliardi, Federico Mariani, Carlo Sandroni, and Riccardo Scattolini. Model predictive control of voltage profiles in mv networks with distributed generation. *Control Engineering Practice*, 34 :18–29, 2015.
- [3] Federico Mariani. Controllo predittivo con vincoli soft e logica di gestione del tap changer di una rete elettrica con generazione distribuita. 2013.
- [4] Yidong Wang. Voltage stabilisation of distribution grid. 2017.