

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :

**COMMANDE ET STABILISATION DES SYSTEMES
EN RESEAUX**

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique & systèmes

Par : *Amine* BOUAZIZ

DEVANT Le JURY

Président : *Mohamed* LAFIFI.....M.C. A..... UBM Annaba

Directeur de mémoire : *Sofiane* GHERBI.....M.C. A.....UBM Annaba

Examination : *Brahim* BOULEBTATECHE ...M.C. BUBM Annaba

Examination: *Nasreddine* DEBBACHE.....Professeur.... UBM Annaba

Résumé :

Le travail présenté dans ce mémoire concerne la simulation sous logiciel TRUETIME de la commande en réseau de la puissance d'un système éolien et sa stabilisation par deux méthodes, à savoir la méthode de placement de pôles et la méthode de la commande par le régulateur quadratique linéaire (LQR). Premièrement dans ce travail on a fait référence aux systèmes commandés par réseaux où leurs structures, leurs protocoles de liaison et leurs influences sur la qualité du contrôle ont été présentés. Deuxièmement on a décrit en détail le logiciel de simulation de système en réseau TRUETIME 2.0 bêta qui a servi d'outil pour l'analyse du système éolien commandé en réseau. Troisièmement après avoir configuré le système éolien sous Simulink à l'aide des blocks du TRUETIME nous avons implémenté au système une boucle de commande incluant les paramètres du régulateur PI de la méthode de placement de pôles. Quatrièmement nous avons repris la simulation précédente en remplaçant les paramètres du PI par ceux du PID relatifs à la méthode de la régulation quadratique linéaire(LQR), en vue de la comparaison de la réponse indicielle du système dans les deux cas.

Les mots clés : Système commandé en réseau, Modélisation et simulation, Eolienne Régulation, Méthodes de régulation, Réseaux, Contrôle de puissance

Abstract :

The work presented in this thesis concerns the simulation under TRUETIME software of the grid control of the power of a wind system and its stabilization by two methods, namely the pole placement method and the linear quadratic controller (LQR) control method. First in this work reference was made to network-controlled systems where their structures, binding protocols and influences on the quality of control were presented. Second, the network system simulation software TRUETIME 2.0 beta was described in detail and used as a tool for the analysis of the grid controlled wind system. Thirdly, after configuring the wind system in Simulink using the TRUETIME blocks, we implemented a control loop including the PI controller parameters of the pole placement method. Fourthly, we repeated the previous simulation by replacing the PI parameters with those of the PID relating to the linear quadratic regulation method (LQR), in order to compare the index response of the system in both cases.

The keywords : Network controlled system, Modelling and simulation, Wind turbine Regulation, Regulation methods, Networks, Power control

نبذة مختصرة :

يتعلق العمل المقدم في هذه الأطروحة بالمحاكاة في إطار برنامج TRUETIME للتحكم في الشبكة لقدرة نظام الرياح وتثبيتته بواسطة طريقتين ، هما طريقة وضع القطب وطريقة التحكم التربيعية الخطية (LQR). تمت الإشارة أولاً في هذا العمل إلى الأنظمة التي تسيطر عليها الشبكة حيث تم عرض بنياتها وبروتوكولاتها الملزمة وتأثيراتها على جودة التحكم. ثانياً ، تم وصف برنامج محاكاة نظام الشبكة TRUETIME 2.0 بيتا بالتفصيل واستخدم كأداة لتحليل نظام الرياح الذي يتم التحكم فيه على الشبكة. ثالثاً ، بعد تكوين نظام الرياح في Simulink باستخدام كتل TRUETIME ، قمنا بتطبيق حلقة تحكم بما في ذلك معلمات جهاز التحكم PI لطريقة وضع القطب. رابعاً ، كررنا المحاكاة السابقة من خلال استبدال معلمات PI بمعلمات PID المتعلقة بطريقة التنظيم التربيعي الخطي (LQR)، من أجل مقارنة استجابة مؤشر النظام في كلتا الحالتين.

الكلمات المفتاحية: نظام يتحكم فيه الشبكة ، النمذجة والمحاكاة ، توربينات الرياح ، التنظيم ، طرق التنظيم ، الشبكات ، التحكم في الطاقة



A MES CHERS PARENTS

A MES CHERES TANTES

A MON FRERE ET SA FEMME

Remerciements

Je ne peux m'empêcher d'adresser toute ma reconnaissance à tous ceux qui, de près ou de loin, m'ont apporté leur concours de quelque nature que ce soit.

Je tiens à exprimer toute ma gratitude envers M. GHERBI SOFIANE, Maître de conférence classe A, qui a permis que ce travail s'accomplisse sous sa direction. Son encadrement, sa disponibilité et ses conseils m'ont été d'un grand apport. Qu'il en soit remercié.

Je tiens à remercier très vivement M. Mohammed Mourad LAFIFI, d'avoir bien voulu présidé ma séance de soutenance. Qu'il soit assuré de mon profond hommage.

Je tiens aussi à exprimer toute ma reconnaissance à M. B. BOULEBTATECHE et M. N DEBBACHE pour avoir accepté d'examiner et de juger ce travail.

J'aimerais également exprimer toute ma gratitude envers mes amis qui se reconnaissent, pour leur soutien tout au long de mes études et dans ma vie quotidienne.

Je ne terminerai pas cet avant - propos sans adresser un grand merci pour tous les enseignants du département d'Electronique qui ont contribué à ma formation tout au long de ce cursus, et qui m'ont permis de faire mes premiers pas dans la recherche. Qu'ils soient assurés de toute ma gratitude.

Je n'oublierai pas le soutien lointain de tous les membres de ma famille lointaine, qui se trouve à l'intérieur et à l'extérieur du pays, je les remercie de tout cœur.

Sommaire

Table des matières.....	i
Liste des figures	iii
Liste des tableaux.....	v
Liste des abréviations.....	vi
Introduction générale.....	1

Chapitre I : Les systèmes commandés en réseaux

I. Les systèmes commandés en réseau	4
II. Domaines d'application.....	12
III. Différents types de réseau et d'architectures.....	13
IV. Avantages et Inconvénients	20
V. Conclusion	22

Chapitre II. Présentation du logiciel TRUETIME

I. Introduction sur les logiciels de simulation temps réel.....	24
II. Présentation du logiciel TRUETIME 2.0 beta	25
III. Présentation des différents blocs de TRUETIME.....	26
IV. Exemple de simulation TRUETIME	38
V. Conclusion.....	40

Chapitre III. Applications. Application1 : Commande en réseau de la puissance d'un système éolien .

I. Introduction	42
II. Présentation du système éolien.....	47
III. Boucle de commande de puissance en réseau.....	48

IV.	Détermination des paramètres du contrôleur PI de la méthode de placement des pôles.....	51
V.	Description du système éolien sous SIMULINK /MATLAB	52
VI.	Test de différents scénarios de perturbations sur le réseau-retards et pertes de paquets.....	58
VII.	Conclusion.....	64

Chapitre IV- Application 2 : Commande LQR de la puissance d'un système éolien

I.	La commande LQR	66
II.	Détermination des paramètres du contrôleur PID-LQR.....	68
III.	Application pour la commande en réseau du système éolien	72
IV.	Tests de différents scénarios d'incertitudes sur le réseau –perte de paquets et partage de bande passante	74
V.	Conclusion.....	79
VI.	Comparaison des résultats avec ceux obtenus dans le chapitre III en termes de performances	80
Conclusions Générales		85
Références Bibliographiques		87

Liste des figures

Chapitre I : Chapitre I : Les systèmes commandés en réseaux

Figure I-1 : Architecture de système classique (point à point)	6
Figure I-2 : Installation typique d'un SCR.....	8
Figure I-3 : Architecture des systèmes commandés en réseaux (SCR).....	8
Figure I.4 : Structure directe	9
Figure I.5 : Structure hiérarchique 1 d'un SCR	11
Figure I.6 : Structure hiérarchique 2 d'un SCR	11
Figure I.7 : Structure hiérarchique 3 d'un SCR.....	11
Figure I.8 : Classification des réseaux selon leur taille.....	17

Chapitre II : Chapitre II. Présentation du logiciel TRUETIME

Figure II-1 : La bibliothèque de blocs TRUETIME bêta 2.0.....	27
Figure II-2 : La boîte de dialogue du bloc de noyau TRUETIME.....	28
Figure II-3 : La boîte de dialogue du bloc réseau filaire TRUETIME.....	31
Figure II.4 : La boîte de dialogue du bloc réseau sans fil TRUETIME.....	34
Figure II-5 : La boîte de dialogue des blocs autonomes TRUETIME.....	37
Figure II-6 : Un modèle de système de contrôle.....	39
Figure II-7 : Résultats de simulation d'un simple contrôle TRUETIME.....	40

Chapitre III. Applications. Application1 : Commande en réseau de la puissance d'un système éolien .

Figure III-1 : système de commande (a) indique le cas de la boucle fermée	42
Figure III-2 : représentation d'état d'un système lineaire.....	44
Figure III-3 : forme canonique de KALMAN.....	45
Figure III.4 : système avec bouclage de sorties.....	46
Figure III-5 : vue d'une éolienne éclatée.....	47
Figure III-6 : fonctionnement d'une éolienne	48
Figure III-7 : paramètres du système éolien.....	50
Figure III-8 : boucle e commande en puissance du système éolien de (6kw).....	51
Figure III-9 : configuration du système eolien sous simulink/TRUETIME	53

Figure III-10 : noyau du réseau (TRUETIME network).....	54
Figure III-11 : paramètres du block TRUETIME network	55
Figure III-12 : NCEUD 1 simulation de différentes perturbations du réseau	56
Figure III-13 : NCEUD 2 système éolien a commander	56
Figure III-14 : NCEUD 3 commande par régulateur PI.....	57
Figure III-15 : visualisation du système éolien sous simulink.....	57
Figure III-16 : Réponse indicielle du système scenario 1.....	59
Figure III-17 : Signal de commande scenario 1.....	59
Figure III-18 : Réponse indicielle du système scenario 2.....	59
Figure III-19 : Signal de commande scenario 2.....	60
Figure III-20 : Réponse indicielle du système scenario 3.....	60
Figure III-21 : Signal de commande scenario 3.....	60
Figure III-22 : Réponse indicielle du système scenario 4.....	61
Figure III-23 : Signal de commande scenario 4.....	61
Figure III-24 : Réponse indicielle du système scenario 5.....	62
Figure III-25 : Signal de commande du système scénario 5.....	62
Figure III-26 : réponse indicielle du système scenario 6	63
Figure III-27 : Signal de commande scenario 6.....	63
Figure III-28 : Réponse indicielle du système scenario 7.....	64
Figure III-29 : Signal de commande scenario 7.....	64

Chapitre IV- Application 2 : Commande LQR de la puissance d'un système éolien

Figure IV-1 : configuration du système éolien sous SIMULINK/TRUETIME par la méthode LQR.....	73
Figure IV-2 : NCEUD 3 commande par régulateur PID-LQR	74
Figure IV-3 : Réponse indicielle du système scenario 1	74
Figure IV.4 : Signal de commande scenario 1.....	75
Figure IV-5 : Réponse indicielle du système scenario 2	75
Figure IV-6 : Signal de commande scenario 2.....	75
Figure IV-7 : Réponse indicielle du système scenario 3.....	76

Figure IV-8 : Signal de commande scenario 3.....	76
Figure IV-9 : Réponse indicielle du système scenario 4.....	77
Figure IV-10 : Signal de commande scenario 4.....	77
Figure IV-11 : Réponse indicielle du système scenario 5.....	77
Figure IV-12 : Signal de commande scenario 5.....	78
Figure IV-13 : Réponse indicielle du système scenario 6.....	78
Figure IV-14 : Signal de commande scenario 6.....	78
Figure IV-15 : Réponse indicielle scenario 7.....	79
Figure IV-16 : Signal de commande scenario 7.....	79

Liste des tableaux

Tab IV-1 : comparaison des méthodes de placement de pôles et PID-LQR.....	83
---	----

Liste des abréviations

NCS :	Networked Control system
SCR :	Système Commandé en Réseau
CSD :	Control System Deadlin.,
AN:	Activity Network
CAN :	Controller Area Network
CRC :	Contrôle de redondance cyclique
CSMA/CD:	Carrier Sense Multiple Access with Collision Detection
CSMA/CA :	Carrier Sense Multiple Access with Collision Avoidance
PCSMAs :	Predictable Carrier Sense Multiple access
RA :	Réseau d'activité
Lan :	Réseaux locaux
Vlan :	Réseaux locaux virtuels
VPN :	Réseaux locaux privés
WAN :	(Wide Area Network) : Appelés aussi réseaux longue distance
Réseaux locaux sans fils (wireless, LAN ou WLAN) : ce sont des réseaux sans connexions physiques visibles.	
Man :	Les réseaux métropolitains (MAN) : les réseaux métropolitains permettent
SISO :	Entrée simple sortie simple (Single Input Single Output)
Moteur DC :	Moteur à courant continu
TCP :	Transfer Control Protocol
MIMO :	Entrée multiple sortie multiple (Multiple Input Multiple output)
CAN :	Controller Area Network
LAN :	Local Area Networks WAN: (Wide Area Network)
PID :	Régulateur à trois actions (proportionnelles, dérivée, intégrale).
U_{PID} :	La commande PID
LQR :	Linear aquadratique automatic
PI :	Régulateur à une action (intégrale)

INTRODUCTION GENERALE

Les systèmes de contrôle en réseau sont les boucles de contrôle de rétroaction fermées à travers un réseau en temps réel. C'est-à-dire que dans les systèmes de commande en réseau, des réseaux de communication sont utilisés pour échanger les signaux d'information et de commande (entrée de référence, sortie d'usine, entrée de commande, etc.) entre les composants du système de commande (capteurs, contrôleurs, actionneurs, etc.).

Du fait des grands avantages que présentent ces systèmes à savoir (le faible coût, le poids réduit, l'installation et la maintenance facile.... etc.) ; ils ont été largement appliqués à de nombreux systèmes de contrôle complexes, comme les usines de fabrication, les véhicules, les aéronefs et les engins spatiaux.

Malgré cela, les réseaux de communication dans les boucles de régulation rendent l'analyse et la conception d'un système de régulation en réseau compliquées.

En effet, la présence d'un réseau dans la boucle de commande induit de nombreuses problématiques spécifiques telles que la quantification du signal lors de l'émission/réception des paquets d'information, la compression/décompression des informations transitant par le réseau, la perte de paquets due à la congestion du réseau– la gestion du trafic par le routeur afin de garantir la stabilité ou la performance du système considéré lors de la congestion, et la stabilisation des systèmes en présence de retards. Ces derniers induits par le réseau (de capteur à contrôleur et de contrôleur à actionneur), se produisent lorsque les capteurs, actionneurs et contrôleurs échangent des données sur le réseau. Ces retards peuvent être constants, variables dans le temps et, dans la plupart des cas, aléatoires.

Lorsque le flux des données devient trop important, la capacité du réseau est dépassée et de l'information est perdue, de plus l'apparition de retard dégrade la stabilité et la performance de contrôle du système de boucle de régulation.

Dans ce contexte, de nombreux chercheurs se sont intéressés à l'étude de la stabilité, à la conception des contrôleurs pour la stabilisation et à l'atteinte des performances des systèmes de

contrôle en réseau. Toutes ces approches se sont mises en œuvre dans le but d'optimiser le fonctionnement des procédés industriels qui s'avère être une tâche difficile.

Les variations sont considérées comme des perturbations nuisibles. On observe alors une dégradation des performances des procédés, risquant même leur instabilité.

Il est donc nécessaire de recourir à des techniques de commandes avancées, capables de rendre en considération les perturbations et permettent un bon comportement des procédés.

Parmi les solutions existantes, l'utilisation de commandes ou de contrôleurs aptes à assurer un fonctionnement optimal ont été envisagé par plusieurs chercheurs.

Pour notre part, notre contribution va consister à mettre en évidence, via une simulation par l'exploitation d'un logiciel "TRUETIME" le comportement d'un système commandé en réseau représenté par le système éolien en adoptant deux méthodes en vue d'optimiser le choix de la commande en réseau.

Pour cela notre travail sera organisé comme suit, dans le :

- Chapitre I, nous allons introduire les notions relatives aux systèmes commandés en réseaux à savoir les retards induits, les pertes de paquets, le partage des bus de communication, les différents protocoles de relais, etc.
- Chapitre II : nous allons décrire le logiciel qui va servir à la simulation du système éolien à étudier avec ses boîtes de dialogue, ses blocks Simulink et les paramètres des systèmes pris en considération.
- Chapitre III : nous allons modéliser un système éolien à commander en réseau à l'aide Du simulateur TRUETIME sous Simulink. Ensuite, nous inclurons les paramètres PI de la méthode de placement de pôles, à la boucle de commande en réseau du système éolien en vue de visualiser son comportement dans différents scénarii.
- Chapitre IV. : nous reprendrons la même simulation en remplaçant les paramètres P.I par les P.I.D de la commande LQR afin de procéder à un comparatif du comportement du système sous les mêmes conditions avec un régulateur différent.

Cette modélisation sera réalisée pour optimiser les choix de la commande en réseaux.

CHAPITRE I

Les systèmes commandés en réseaux

Chapitre I – Les systèmes commandés en réseaux

I-1. Introduction

Une tendance importante dans les systèmes industriels et commerciaux modernes est d'intégrer les fonctions de communication, les fonctions de calcul et celles de contrôle aux différents niveaux du système. La connexion traditionnelle point à point, qui lie le contrôleur central aux différents composants a été mise en application avec succès dans l'industrie pendant des décennies. Cependant, les installations physiques et l'extension des fonctions ont poussé les limites de l'architecture point à point. Par conséquent, une relation point à point traditionnelle n'est plus appropriée pour répondre à de nouvelles exigences, telles que la modularité, la décentralisation de la commande et l'intégration du diagnostic.

La perte de paquets et la congestion du réseau apparaissent dans les réseaux à file d'attente (routeur ou switch) lorsque le flux de données est trop important.

La gestion de la file d'attente s'effectue par le protocole mis en place au niveau du routeur, qui détermine de quelle manière les paquets sont rejetés (pour l'AQM par exemple, le nombre de paquets rejetés est proportionnel à la longueur de la file lorsqu'une certaine longueur de référence est atteinte). Le contrôle de congestion fait l'objet de nombreux travaux dans le domaine de l'informatique [12] et le cas du contrôle de la file en prenant en compte l'état de congestion a été étudié en par plusieurs auteurs. [12]

Lorsqu'un paquet est perdu, il peut être réémis (par exemple, dans le cas du TCP) ou non (UDP...). Dans le second cas, l'analyse de stabilité du système commandé par réseau inclut la considération de perte d'information et ne peut être effectuée par les approches continues.

Ainsi selon [12], un principe de seuil d'incertitude est utilisé pour déterminer le taux de perte de paquets pour lequel le système bouclé non perturbé reste stable au sens des moindres carrés (l'espérance de la norme de l'état au carré est nulle en l'infini).

I-2. Systèmes commandés en réseau (SCR)

Les systèmes commandés en réseau (Networked Control System) [2] sont des systèmes automatiques traditionnels où les capteurs, contrôleurs, actionneurs et autres composants sont distribués autour d'un médium de communication (Figure I.1).

Le médium de communication sert à assurer les fonctions suivantes :

- Etablir un lien de communication pour envoyer l'information entre chaque producteur et les consommateurs concernés,
- Respecter les contraintes de temps réel imposées par l'application,
- Faciliter la maintenance (détection des fautes, simplification du câblage),
- Participer à la reconfiguration (réorganisation des échanges suite à la défaillance d'un composant ou suite à l'adjonction d'un nouveau composant ;

Le réseau de terrain répond à ces exigences, de plus son utilisation offre la possibilité d'implémenter des lois de commande sophistiquées [12].

Les systèmes commandés en réseau sont des systèmes où le réseau est utilisé comme moyen de communication dans la boucle de commande.

Ce type de système appelé "Networked Control Systems" (NCS) présente plusieurs avantages :

- La réduction des coûts en poids et en nombre de fils
- Une grande flexibilité
- Une maintenance facile.

Cependant, les performances de commande ou de supervision de ces systèmes sont directement liées à celles du réseau. En effet, celui-ci introduit entre autres inconvénients majeurs des retards dus à la transmission des informations mais aussi à la perte potentielle d'informations. Ces paramètres vont directement influencer voire dégrader les performances du système. Depuis plusieurs années, ces problèmes ont attiré l'attention de la communauté d'Automatique. Plusieurs travaux ont été menés sur le thème de la commande des systèmes en réseau. Avec ces systèmes, il est important de considérer le réseau comme une partie du système et non pas comme un simple moyen de communication.

Pour cela, deux problématiques doivent être considérées :

- La première consiste à adapter les lois de commande et de diagnostic aux performances des réseaux.
- La seconde doit permettre de garantir les performances du système en adaptant les performances du réseau.

I.2. 1. Systèmes de commande classique :

La commande classique a commencé dans les années 1930 par des boucles de régulation simple ou le système point à point. C'est un système où les différents éléments qui le constituent (capteur, actionneur, contrôleur... etc) sont reliés par des câbles dédiés. Ce type de système a été utilisé jusqu' aux années 60 (voir figureI.1).

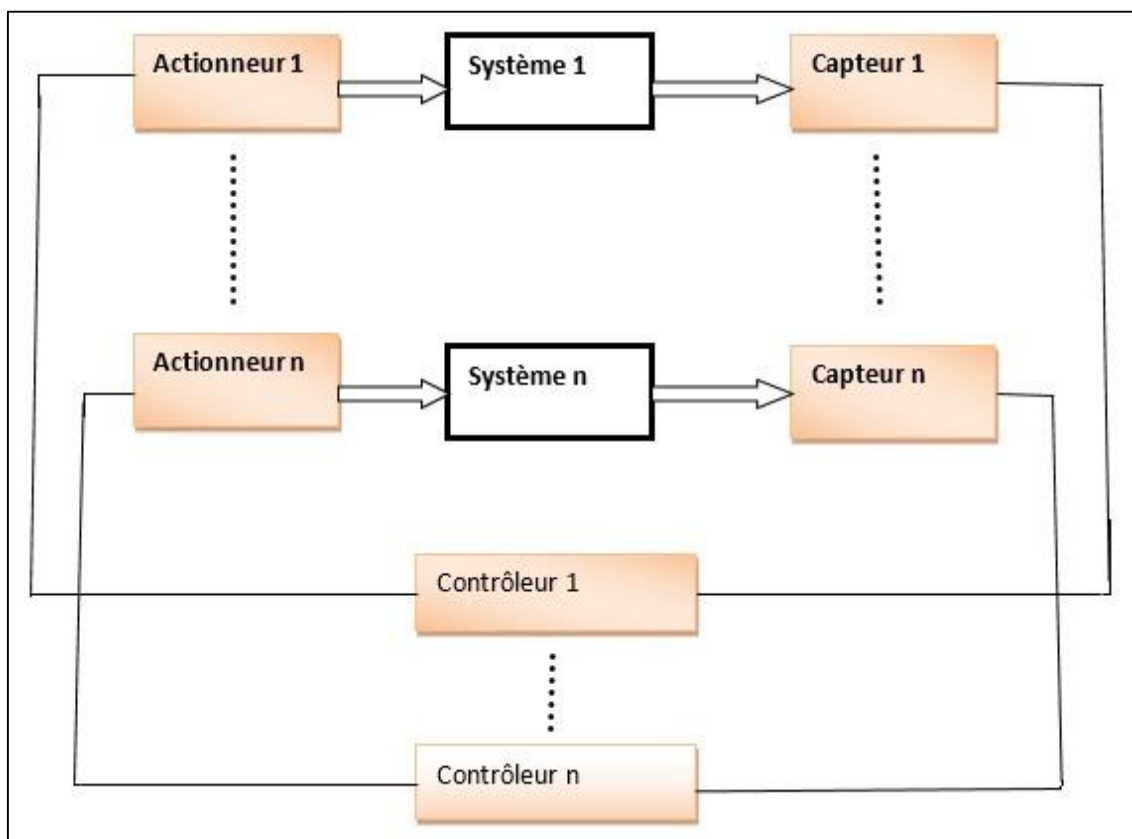


Fig. I. 1. Architecture de système classique (point à point)

Ce type de connexion est sujet à de nombreux inconvénients, qui rendent la tâche de conduite et de maintenance difficile. Parmi ces inconvénients on peut citer :

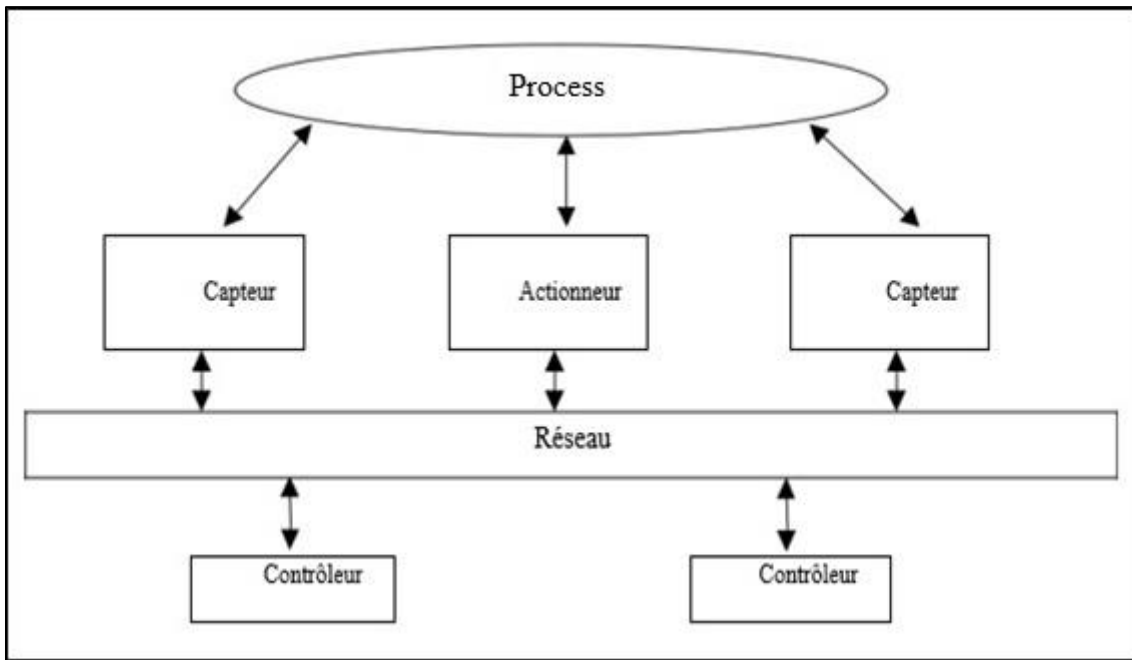
- Un câblage traditionnel : Chaque capteur ou actionneur requiert au moins deux fils électriques (alimentation et information) ;
- Une maintenance difficile ;
- Un encombrement ;
- Une durée de vie courte ;
- Des défauts fugitifs (sensibilité aux vibrations) ;

I-3. Systèmes de commande en réseau :

Les systèmes de commande en réseau (SCR) ou (Network Control System) sont des systèmes de commande traditionnels où les capteurs, les contrôleurs, les actionneurs et autres composants sont reliés par un moyen de communication (Figure I.2). Les capteurs, contrôleurs et actionneurs constituent avec les systèmes de communication les éléments d'un SCR. Les systèmes commandés en réseau se décomposent en trois parties distinctes :

- Le procédé : le terme procédé regroupe le système lui-même, les capteurs et les actionneurs.
- Le contrôleur : le terme contrôleur regroupe aussi bien la loi de commande que le Diagnostic.
- Le réseau : utilisé comme médium de communication entre le procédé et le contrôleur.

L'installation type d'un système en réseau ainsi que son architecture est illustrée respectivement par la figure I.2 et I.3.



FigI.2 Installation typique d'un SCR

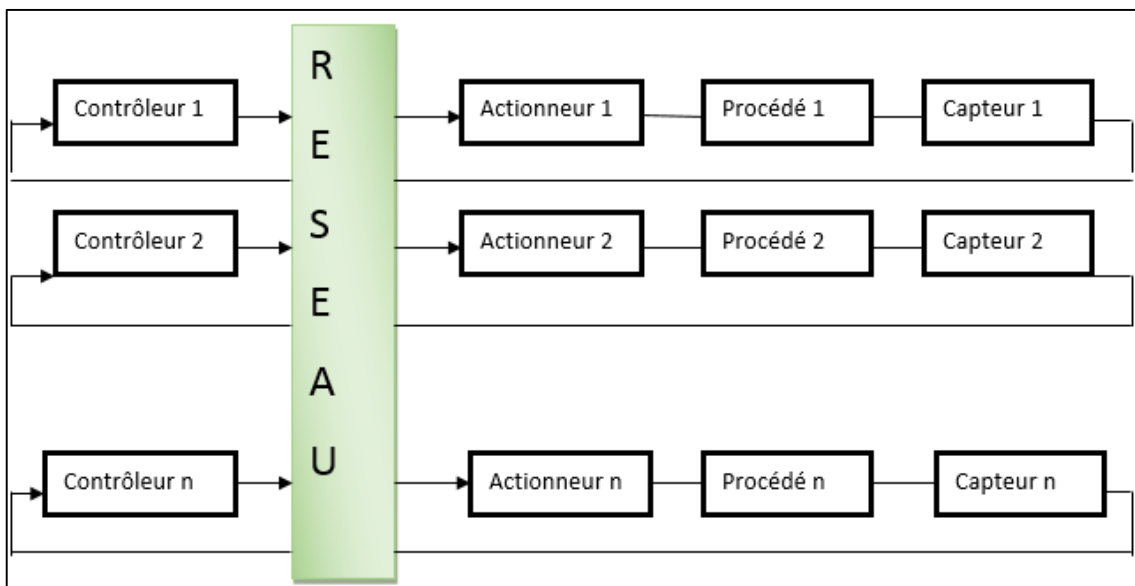


Fig. I.3. Architecture des systèmes commandés en réseaux (SCR)

Deux configurations générales typiques de SCR sont présentées dans les parties suivantes selon [2] et [12].

1.4.1. Structure Directe

Structure directe : Le SCR dans la structure directe se compose d'un contrôleur et d'un site à distance contenant un système physique, des capteurs, des actionneurs et d'autres composants. Le contrôleur et le système distant sont physiquement situés à différents endroits et sont directement liés par un réseau de communication afin d'effectuer la commande en boucle fermée à distance comme cela est illustré dans la figure I.4. Le signal de commande est encapsulé dans un message et envoyé au système distant par l'intermédiaire du réseau.

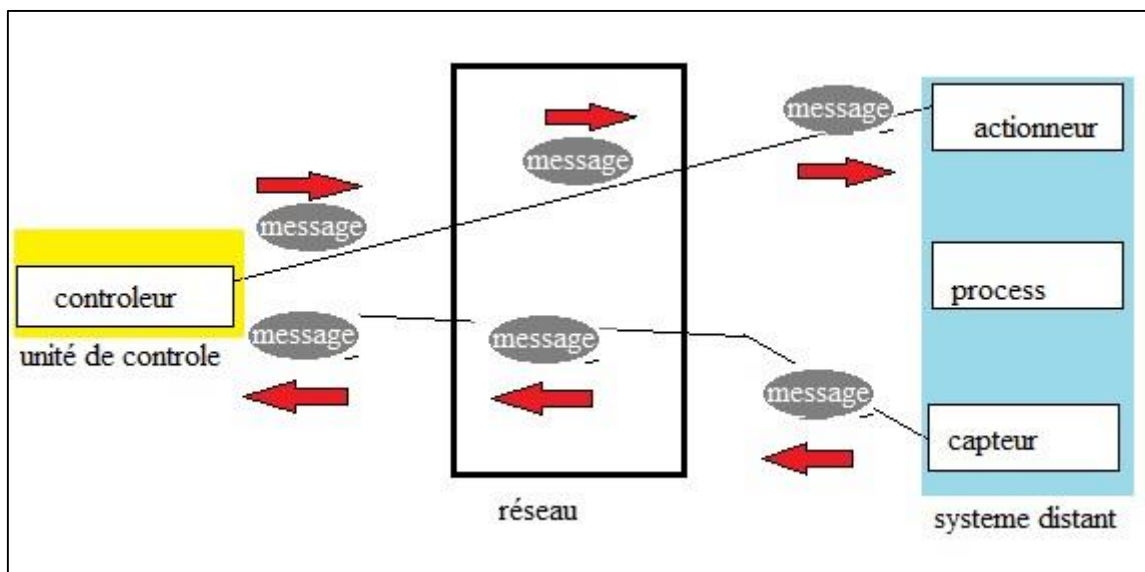


Fig. I.4 : Structure Directe

Le système distant renvoie alors la sortie du système au contrôleur mettant la mesure du capteur dans un message destiné au contrôleur. Plusieurs contrôleurs peuvent être implémentés dans une seule unité de commande pour commander plusieurs boucles fermées dans une structure directe.

1.4. 2. Structure hiérarchique

Structure hiérarchique : la structure hiérarchique de base se compose d'un contrôleur principal et d'un système à distance contenant un contrôleur (contrôleur distant), un capteur et un système physique à réguler (figure 1.3). Périodiquement, le contrôleur principal calcule et envoie la consigne dans une trame par l'intermédiaire d'un réseau.

Le système à distance traite la consigne pour effectuer localement la commande en boucle fermée et renvoie la mesure du capteur au contrôleur principal. Le contrôleur distant doit satisfaire les performances de la boucle fermée avant de recevoir le message du contrôleur principal. Cette structure est employée dans plusieurs applications comme celles des robots mobiles et de télé-opération [12].

L'utilisation de la structure directe ou hiérarchique dépend des conditions d'application et de préférences du concepteur. Par exemple, un robot manipulateur exige habituellement plusieurs moteurs qui tournent simultanément ensemble.

Il peut être plus commode et plus robuste d'employer un contrôleur existant et de formuler le problème dans une structure hiérarchique [2].

D'autre part, un concepteur peut avoir besoin d'un système de commande de vitesse de moteur pour avoir une réponse plus rapide de commande au-dessus du réseau. La structure directe peut être préférée dans ce cas-ci.

Selon le détail relatif à l'utilisation du réseau et le retard induit on distingue pour résumer, trois types de structure illustrées par les figures respectives :

- Le réseau est utilisé uniquement pour faire communiquer les capteurs et le contrôleur, dans ce cas-ci, le retard induit par le réseau influence uniquement le passage des informations des capteurs au contrôleur (figure I.5.)
- Le réseau est utilisé uniquement pour faire communiquer les actionneurs et le contrôleur. Dans ce cas-ci le retard induit par le réseau influence uniquement le passage des informations du contrôleur aux actionneurs ; (figure I.6.)
- Le réseau est utilisé dans la boucle fermée entre le procédé et le contrôleur ou le module de diagnostic. Cette structure est la plus utilisée et l'influence du réseau est dans ce cas maximale, en effet, le retard induit par le réseau intervient sur l'ensemble du système (Fig.I.7)

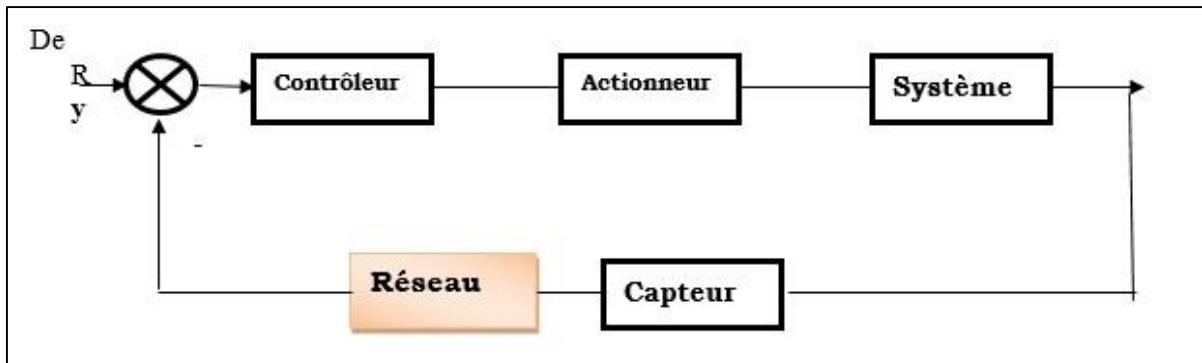


Fig. I.5. Structure hiérarchique 1 d'un SCR

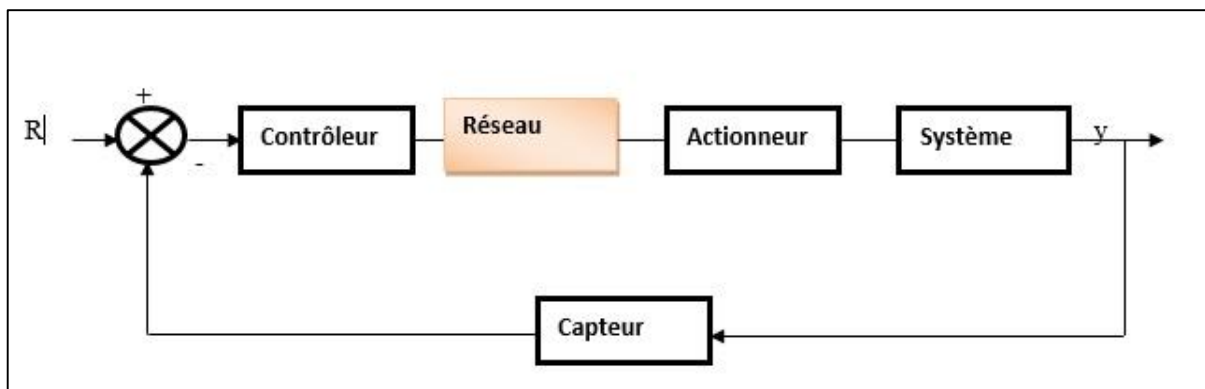


Fig.I.6. Structure hiérarchique 2 d'un SCR

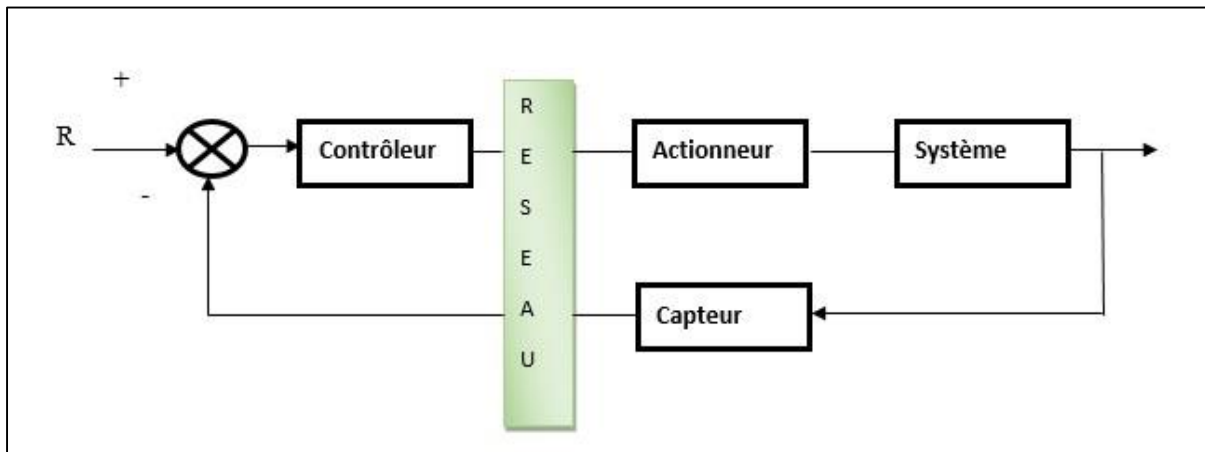


Fig. I.7. Structure hiérarchique 3 d'un SCR

I-5. La différence entre la commande classique et la commande en réseau :

La commande des systèmes traditionnellement connus est basée sur l'hypothèse que toutes les informations transmises sont reçues sans erreur, sans pertes et sans retard, ce qui signifie que le lien entre les différents composants du système est dit "parfait".

Par contre, dans les systèmes commandés en réseau, ce dernier qu'il soit filaire ou sans fil est dit « imparfait ». C'est donc le réseau qui fait la différence entre les deux systèmes, le classique et celui commandé à travers un réseau. Par conséquent, c'est la raison pour laquelle l'analyse et la modélisation des NCS devient incontournable.

Les systèmes commandés par réseau constituent une nouvelle classe de systèmes, introduisant des problèmes spécifiques liés à la présence de retards, à la perte d'information ou à la gestion du flux des données. Ces contraintes prennent une importance considérable lors de la commande de procédés rapides pour lesquels les caractéristiques du réseau ne peuvent plus être négligées.

II. Domaines d'application

Les systèmes commandés en réseau sont des systèmes automatiques traditionnels répartis où les capteurs, les contrôleurs, actionneurs et autres composants sont distribués autour d'un réseau de communication.

Les SCR ont plusieurs domaines d'application vu les principaux avantages qu'ils présentent à savoir le faible coût, le poids réduit, l'installation et la maintenance simple et la grande fiabilité. Ces avantages ont pour conséquence une application large à de nombreux systèmes de contrôle complexes tels que :

- Les usines de fabrication,
- Les avions et les engins spatiaux.
- Les réseaux de capteurs mobiles
- La commande des véhicules
- Les systèmes autoroutiers automatiques
- Les bâtiments intelligents
- La commande des drones
- La robotique

Le réseau permet aux équipements du système d'échanger les données, améliore la modularité et rend l'architecture plus flexible. Par ailleurs l'insertion du réseau dans la boucle de contrôle introduit les effets tels que les délais, les pertes de paquets et la prise en compte de la charge du réseau. Ces effets vont perturber la performance du système en terme de stabilité, adaptabilité, robustesse et précision [13]

III. Les différents types de réseau

Un réseau est un ensemble d'éléments reliés entre eux et réglés de manière qu'ils puissent communiquer. C'est aussi simple que ça. Et les réseaux informatiques n'échappent pas à cette règle. Afin de pouvoir communiquer, les êtres humains ont été dotés d'un langage auditif et/ou visuel. Dans le cas où notre interlocuteur ne comprend pas notre langue, alors nous nous dotons tout simplement d'interprètes.

En transposant cette définition globale des réseaux, nous pouvons conclure qu'un réseau informatique est un ensemble d'équipements informatiques reliés – disons plutôt interconnectés – entre eux et paramétrés de manière qu'ils puissent communiquer.

Il est de bon ton de souligner ici l'adage informatique qui dit que le réseau c'est l'ordinateur, c'est-à-dire que sans le réseau l'ordinateur est sous-exploité.

Voici quelques exemples de la réseautique :

- La mise en place d'un réseau informatique permet de faciliter et de sécuriser le stockage de l'information.
- Elle permet la standardisation des applications et le partage des données entre les postes de travail de manière efficace.
- La mise en réseau bien conçue facilite les opérations de gestion et de maintenance des applications et des équipements informatiques.
- La mise en réseau permet de réduire considérablement les coûts d'infrastructure.

Grâce au réseau, les ressources matérielles et logicielles sont partagées entre plusieurs utilisateurs. Par exemple, au lieu d'acheter plusieurs imprimantes pour chaque service, une imprimante peut être partagée par tous les services. Il en est de même pour les applications distribuées.

III.1. Les différents types de réseaux et leur architecture

Il existe différentes sortes de réseaux, en fonction de la taille (nombre de machines), du débit des informations autrement dit de la vitesse de transfert des données, des types de protocoles de communication, de leur localisation...etc.

Dans un premier temps, nous allons définir brièvement les modèles conceptuels des réseaux, les différents types de réseaux locaux (*LAN*), les réseaux locaux virtuels (*VLAN*), les réseaux métropolitains (*MAN*), les réseaux étendus (*WAN*), les réseaux privés (*VPN*) et les réseaux sans fil (*WIRELESS*).

On distingue plusieurs types de réseaux qui se différencient entre eux en fonction de la distance entre les systèmes informatiques, ou encore en fonction de la technologie qui permet de les mettre en œuvre.

III.1.1. Les réseaux locaux (*LAN*)

LAN signifie Local Area Network (en français Réseau Local). Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).

Ce sont des réseaux de taille plus ou moins modeste, complexes, qui permettent l'échange de données informatiques et le partage de ressources (données, disques durs, périphériques divers, etc.).

L'étendue géographique des réseaux locaux ne dépasse pas 10 km (ex. : pour un immeuble ou un campus).

Le débit, ou la vitesse de communication, varie de quelques Mbps à 100 Mbps. Le nombre de stations ne dépasse généralement pas 1 000.

Une variante du *LAN* est le LAN fédérateur ou réseau de base (backbone) qui est la voie principale empruntée par le trafic.

III.1.2. Les réseaux locaux virtuels (VLAN)

Un réseau local virtuel est un groupe logique d'unités ou d'utilisateurs qui peuvent être regroupés par fonction, service ou application peu importe l'emplacement de leur segment physique. La configuration d'un réseau local virtuel est effectuée dans le commutateur par un logiciel.

Les réseaux locaux virtuels ne sont pas uniformisés et nécessitent l'utilisation d'un logiciel propriétaire vendu par le fournisseur de commutateurs. Ce type de réseau est vu plus en détails à la section suivante.

III.1.3. Les réseaux locaux sans fils (WIRELESS, LAN ou WLAN)

Ce sont des réseaux sans connexions physiques visibles. Ces réseaux utilisent les ondes (radio, infrarouges, etc.) comme support de communication. Les ordinateurs mobiles ou les assistants personnels (Palm Pilot, etc.) constituent le secteur informatique en plus forte progression. Beaucoup de possesseurs de ce type d'ordinateurs ont également un ordinateur relié à des LAN ou des WAN, chez eux ou au bureau, auxquels ils sont reliés à tout instant.

III.1.3.1. Les réseaux métropolitains (MAN)

Les MAN (Métropolitain Area Network) interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme s'ils faisaient partie d'un même réseau local.

Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

Les MAN s'avèrent donc être des réseaux métropolitains permettant l'interconnexion de plusieurs réseaux locaux (LAN) répartis sur différents sites dans une zone urbaine dont l'étendue géographique n'excède pas 200 km. Ces réseaux peuvent être privés ou publics. Ils se distinguent aussi par leurs taux d'erreurs de communication. Le taux d'erreurs pour les réseaux

MAN reste faible bien que plus élevé que pour les réseaux locaux : de 1 bit erroné sur 10⁸ à 1 bit sur 10¹⁵. Le débit est élevé car supérieur à 100 Mbps (sur liens de fibre optique).

III.1.3.2. Les réseaux étendus (WAN)

Un WAN (Wide Area Network ou réseau étendu) interconnecte plusieurs LANs à travers de grandes distances géographiques.

Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles.

Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau. Le plus connu des WAN est Internet.

Un WAN (Wide Area Network ou réseau étendu) interconnecte LANs à travers de grandes distances géographiques.

Les WAN (Wide Area Network) appelés aussi réseaux longue distance se situent à l'échelle nationale et internationale. Ce sont généralement des réseaux de télécommunications gérés par des opérateurs, qui assurent la transmission des données entre les villes et les pays à l'échelle de la planète. Leurs supports de transmission sont variés (ligne téléphonique, ondes hertziennes, fibre optique, satellite, etc.).

La plupart de ces types de réseaux sont publics. Le taux d'erreurs de communication est plus élevé que celui des MAN : de 1 bit erroné sur 10⁶ à un bit erroné sur 10¹². Les débits généralement plus faibles que dans les réseaux locaux dépendent du support de transmission : ils varient de 56 kbps à plus de 625 Mbps pour les réseaux ATM (Asynchronous Transfer Mode).

III.1.3.3. Les réseaux privés virtuels (VPN)

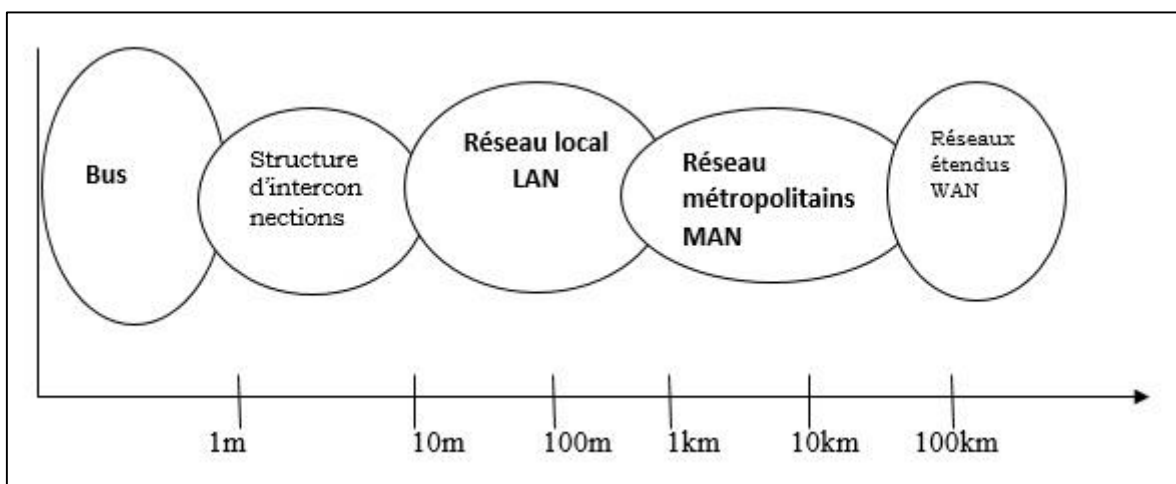
Les réseaux privés virtuels consistent en l'interconnexion de LAN à l'échelle nationale ou internationale. Ces réseaux restent privés et sont transparents pour l'utilisateur. Ils permettent en fait, par exemple pour une entreprise, de s'affranchir de certaines contraintes, telles que la

localisation géographique. Ils rendent possible une transmission plus sécuritaire des données sur un réseau, en particulier sur Internet.

III.2. Classification des réseaux selon leur taille

Selon leurs tailles les réseaux ne transmettent pas les données de la même manière voire (figure I.8.)

- Le bus s'étend entre quelques mètres (jusqu'à 1mètre)
- La structure d'interconnexion s'étend entre 1mètre et 10mètres
- Le réseau local LAN s'étend entre 10 mètres et 10 kilomètres
- Le réseau MAN s'étend entre 10 kilomètres et 100 kilomètres
- Le réseau étendu WAN plus de 100 kilomètres



FigI.8. Classification des réseaux selon leur taille

En résumé [2] les réseaux locaux : Ethernet souvent dénommés LAN, pour Local Area Networks) sont largement répandus et habituellement utilisés pour connecter des ordinateurs dans les réseaux privés. Leur taille est limitée, rendant le retard connu et borné, et ils ont un taux de transmission élevé. Ethernet constitue un exemple courant de réseau LAN en bus, sans contrôleur centralisé.

L'accès au bus peut être effectué par protocole à détection de porteuse (CSMA - Carrier Sense Multiple Access) avec détection de collision (CD - Collision Détection). Avec cette méthode

l'émission débute seulement si le canal est libre, mais deux (ou plus) émetteurs peuvent décider d'émettre en même temps, ce qui induit une collision. Cette dernière est automatiquement détectée et les deux émetteurs arrêtent leur émission, pour la reprendre après un temps aléatoire. Les caractéristiques du retard dépendent fortement de la charge du réseau, rendant Ethernet très difficile à modéliser.

Concernant les réseaux longue distance : Protocole de transfert - Les réseaux grande distance ou WAN (Wide Area Network) sont caractérisés par la présence d'un ou plusieurs routeurs et de commutateurs, qui gèrent et distribuent l'information.

La principale influence du routeur est d'induire une file d'attente où sont stockés les messages avant d'être retransmis. Notons que ce problème apparaît également dans les LAN, où les commutateurs sont de plus en plus utilisés. Lorsque les flux d'entrée dans le routeur sont trop importants, celui-ci devient congestionné et des paquets d'information sont perdus. L'émetteur peut - être informé des paquets reçus par l'envoi des acquittements (acknowledgements) par le destinataire.

Le protocole de contrôle de transfert TCP (Transfer Control Protocol) est un exemple classique de protocole utilisé dans les réseaux longue distance et LAN, mis en place aux deux extrémités du canal de communication afin de permettre aux utilisateurs d'émettre et de recevoir des données au travers d'internet. C'est un protocole dit sécurisé, dans le sens où tout paquet perdu est réémis. Son objectif principal est de maximiser le taux de transmission de l'utilisateur en ajustant la taille de la fenêtre d'émission de l'utilisateur

III.3 L'influence du réseau sur la qualité de Contrôle du système :

L'influence du réseau sur la qualité de contrôle du système est due à plusieurs éléments dont :

III.3.1. L'échantillonnage et le Retard :

La transmission d'un signal continu via un réseau nécessite l'échantillonnage du système, le codage en format binaire, l'encapsulation dans une trame de communication ensuite l'envoi et enfin la décapsulation et le décodage.

Toute cette procédure prend du temps et donc introduit un retard qui s'avère variable à cause du temps d'accès au médium et aussi du temps de transmission qui dépend essentiellement des conditions dans lesquelles le réseau fonctionne à savoir : la charge du réseau et la qualité du canal de transmission.

En général, les réseaux introduisent des retards dus au temps de transmission et de traitement de l'information entre deux ou plusieurs nœuds. Il est connu dans la théorie de la commande que le retard peut influencer la performance du système commandé et entraîner son instabilité [2].

En résumé, l'élément fondamental induit par la présence d'un réseau dans la boucle de commande est la présence de retards entre les capteurs et la commande et entre la commande et les actionneurs. La commande des systèmes à retard constitue un sujet très vaste et a fait l'objet d'un sujet très vaste., notamment avec les approches déterministes.

La plupart des analyses de stabilité classiques (en utilisant les fonctions de Lyapunov-Krasovkii et Lyapunov-Rasumikhin par exemple) considèrent un retard constant ou de borne connue. Ce retard constant peut être compensé à l'aide d'un prédicteur de Smith ou plus généralement un prédicteur d'état à horizon fixe.

III.3.2. La perte des paquets :

Un autre problème qui peut influencer un NCS est la perte de paquets.

En effet, envoyer les informations via un réseau peut entraîner leurs pertes. Ces pertes sont le résultat des erreurs au niveau de la couche physique du réseau (les pertes sont plus probables dans les réseaux sans fil que dans les réseaux filaires) ou bien de la charge des buffers (à cause de la charge du réseau). Un trop long retard peut être considéré comme perte de paquets surtout pour les applications temps réels [2].

III.3.3. Débit :

Chaque réseau de communication est limité par une bande passante (un nombre limité d'envoi d'informations par seconde). Dans certaines applications, cette limite de la bande passante pose quelques problèmes au niveau de la performance des NCSs [2]

La présence sur le réseau de multiples utilisateurs suppose un partage et une redistribution des ressources, ainsi qu'une limite de la bande passante disponible pour chacun des émetteurs.

La bande passante minimum garantie détermine la qualité de service (QoS) assurée par le réseau. Celle-ci dépend de la gestion de la file d'attente au niveau du routeur ainsi que de celle des émissions, effectuées par le contrôleur centralisé (par exemple au niveau du routeur).

Le problème d'allocation de ressources est classiquement abordé en mathématiques discrètes afin d'illustrer les solutions par programmation linéaire.

Pour une qualité de service (QoS) garantie, la contrainte de bande passante peut être prise en compte lors de la synthèse du contrôleur dédié au système commandé à travers le réseau

IV. Avantages et inconvénients

Les principaux réseaux de données tels que ARPA- NET qui ont été spécialement développés il y a environ 30 à 40 ans, ont évolué vers des protocoles de réseau modernes largement utilisés comme Ethernet et Internet pour des usages généraux.

Les technologies de réseau de données fournissent plusieurs avantages sur la liaison de points de données comme les ordinateurs. Les réseaux permettent des transferts de données à distance et des échanges de données entre utilisateurs, réduisent la complexité des connexions de câblage et les coûts des médias, et facilitent la maintenance.

En raison de leur attractivité, de nombreuses entreprises et instituts industriels ont manifesté leur intérêt pour l'application des réseaux à des fins de contrôle industriel à distance et d'automatisation industrielle.

À la suite d'importants travaux de recherche et de développement, plusieurs protocoles de réseau pour le contrôle des essais industriels ont été mis au point. Par exemple, le Controller Area Network (CAN) a été développé à l'origine dans les pays européens tels que l'Allemagne.

Les systèmes commandés en réseau sont des systèmes où le réseau est utilisé comme moyen de communication dans la boucle de commande. Ce type de système appelé "Networked Control Systems" (NCS) présente plusieurs avantages, parmi ceux-ci, on peut notamment citer :

- L'avantage majeur et que on peut envoyer ou transmettre plusieurs information travers un seul câble au lieu de plusieurs câble ou d'envoyer en série des flux de zéros et de uns dans le cas analogique ;
- Une meilleure disponibilité des systèmes de commande, autrement dit dans le cas où un contrôleur est en panne, ceci n'implique pas nécessairement l'effondrement de toute l'architecture.
- La communication à travers un réseau permet aussi une multitude d'échanges d'informations entre les composants, ce qui se traduit par la disparition des frontières ou les diverses couches de l'entreprise. Par conséquent les données destinées à diverses applications comme la commande, le diagnostic, la maintenance, la supervision, se côtoient sur le même médium de communication ;
- Une maintenance plus facile
- La possibilité de relier plusieurs composants sur la même paire de câbles pour former un réseau avec un medium partagé, permettant alors une meilleure flexibilité ;
- Un moindre coût d'utilisation ;

IV.2 : les inconvénients

Le système commandé en réseau ne présente pas que des avantages. Parmi les inconvénients majeurs on retrouve essentiellement ce qui suit :

- Des retards dus à la transmission des informations ;
- Une perte potentielle d'information.
- Une influence directe, négative conséquente à ces paramètres sur le système, voire sa dégradation.

V. Conclusion

Dans ce chapitre, nous avons défini de manière sommaire respectivement le système classique et les systèmes commandés en réseau.

La description détaillée a permis de mettre en relief les différences entre les deux systèmes. Il est également fait mention des principaux avantages et inconvénients des systèmes commandés en réseau.

Nous avons décrit les différentes structures de Systèmes commandés en réseau (SCR), avec particulièrement l'influence détaillée de ces derniers sur la qualité du contrôle de système.

CHAPITRE II :
Présentation du logiciel
TRUETIME

Chapitre II. Présentation du logiciel TRUETIME

II.1. Introduction

Les logiciels de simulation temps réel fonctionnent avec une carte d'acquisition en entrées et sorties, en temps réel.

On peut classer ces logiciels en 4 catégories :

- * Enregistrement
- * Asservissements
- * Identification de processus
- * Traitement du signal

1. Enregistrement ; on peut :

- Faire des enregistrements $Y = f(t)$, de 1 à 4 voies simultanément. Les tensions visualisées peuvent être référencées à la masse ou être différentielles.
- Faire des enregistrements $Y = f(X)$. Les tensions peuvent être également référencées à la masse ou différentielles.

2. Asservissements

Cette catégorie regroupe 3 logiciels :

- Correcteurs analogiques programmés (PID, PIR, prédicteur de Smith, R(p))
- Correction tachymétrique quelconque
- Correcteurs numériques
- Reconstruction du flux d'une machine asynchrone : il est possible d'enregistrer la réponse du système bouclé et d'imprimer cette réponse, sans interrompre le fonctionnement du correcteur.

3. Identification

Cette catégorie regroupe 3 logiciels :

- Identification à partir de la réponse indicielle par exemple : processus d'ordre n avec ou sans retard
- Réponse impulsionnelle (corrélation avec un bruit blanc ou une séquence binaire pseudo-aléatoire)
- Modèle : comparaison processus et modèle (Simulink par exemple)

4. Traitement du signal

Cette catégorie regroupe 5 logiciels :

- Filtres numériques (FIR ou IIR)
- FFT (d'un signal acquis ou simulé ou de 2 signaux)
- FFT inverse (reconstruction d'un signal visible à l'oscilloscope à partir de son spectre bilatéral)
- Harmoniques (reconstruction d'un signal visible à l'oscilloscope en ajoutant petit à petit les harmoniques)
- Synthèse de filtres actifs (ne fonctionne pas en temps réel)

II.2. Présentation du logiciel TRUETIME 2.0 beta

L'insertion d'une boucle de commande dans un réseau est un défi pour la modélisation, l'analyse, la conception et la simulation des systèmes commandés en réseau, en temps réel [5].

Les techniques pour la modélisation d'un procédé, de l'échantillonneur, du contrôleur et de l'actionneur sont bien connues mais les techniques pour modéliser les principales caractéristiques d'un réseau de communication et les interactions entre le réseau et le système à commander sont très complexes et ne permettent pas de traiter le problème dans sa globalité. Il est donc nécessaire de faire appel à des outils de simulation.

Parmi ces outils, le TrueTime [5][8] [14] logiciel orienté co-conception (commande, ordonnancement de tâches, réseau) , est particulièrement bien adapté à nos objectifs et c'est cet outil qui a été utilisé pour notre travail. De plus, cet outil est de loin le plus utilisé [5]

Le simulateur TrueTime est une boîte à outils basée sur Matlab/Simulink, permettant la simulation de systèmes de contrôle distribués temps réel. Il est développé par un groupe de chercheurs suédois à LUND Technology depuis 1999.

Il permet d'estimer le comportement temps réel des systèmes multi- tâches exécutant des tâches de contrôle- commande modélisées par des blocs Simulink.

Ces différentes tâches peuvent être distribuées et inter- connectées à travers un réseau. TrueTime permet de simuler des modèles simples de réseaux de communication et donc d'étudier leurs influences sur les performances des applications réalisées à travers ce réseau..

Selon [8], les blocs TrueTime sont reliés avec des blocs Simulink ordinaires pour former un système de contrôle en temps réel. du monde réel et l'architecture de l'ordinateur sous la forme d'exécution de la tâche et de la communication en réseau.

II.2.1. Présentation des différents blocs de TRUETIME 2.0 beta

Toutes les étapes fondamentales de la création d'une simulation TRUETIME sont décrites dans le manuel [14]. Les fonctions de code pour les tâches et les commandes d'initialisation peuvent être écrites soit sous forme de fonctions C++, soit sous forme de M-files Matlab. Les rouages internes du simulateur tels le modèle des tâches, les détails de la mise en œuvre et les détails du calendrier sont mentionnés.

Les différents blocs de réseau permettent (blocs de noyau) de communiquer sur des réseaux filaires ou sans fil simulés.

Les différentes politiques d'ordonnancement implantées dans TrueTime sont : **FP** (Fixed Priority) – **RM** (Rate monotonic), **DM**(Deadline Monotonic) et **EDF** (Earliest Dead line First)

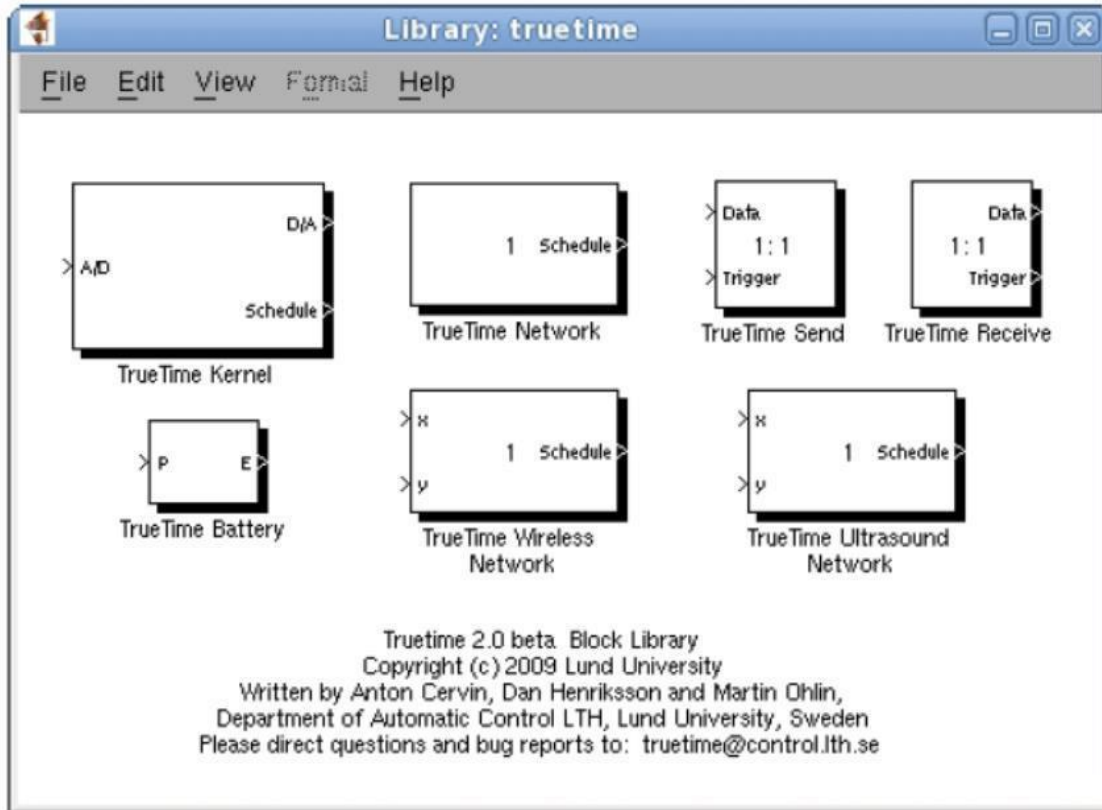


Fig.II.1. La bibliothèque de blocs TRUETIME bêta 2.0

Le simulateur TRUETIME se compose de plusieurs blocs :

II.2.1.1. Le bloc noyau (TRUETIME Kernel block) :

Le bloc de noyau est une fonction S-Simulink qui simule un ordinateur doté d'un noyau en temps réel, A / D et de convertisseurs D / A, une interface de réseau, et les canaux d'interruptions externes. Le noyau exécute des tâches définies par l'utilisateur et interrompt gestionnaires. En interne, le noyau conserve plusieurs structures de données que l'on trouve couramment dans un noyau temps réel : une file d'attente prête, une file d'attente de temps, et les dossiers de tâches, gestionnaires d'interruption, des moniteurs et des minuteries qui ont été créés pour la simulation.

Le bloc est configuré par le dialogue masque de blocage (Figure II.2.). Le principal paramètre est le nom de la fonction d'initialisation, parce que chaque bloc de noyau doit être initialisé au début de la simulation. Un argument facultatif pour le script d'initialisation qui peut être réglé. Il y'a également la possibilité de régler la batterie et l'horloge. Cette dernière peut dériver et son décalage peut être réglé.

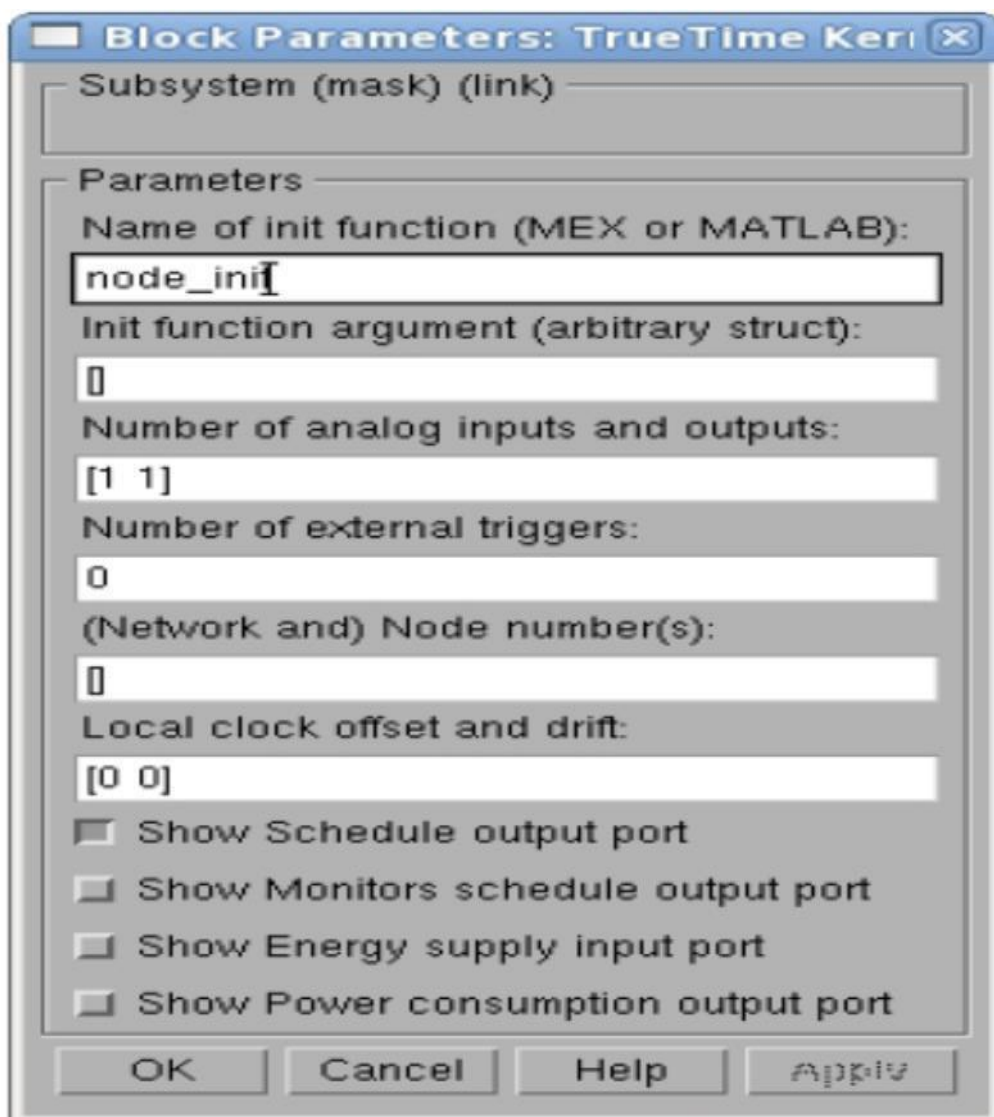


Fig. II.2. La boîte de dialogue du bloc de noyau TRUETIME

Le bloc de noyau est configuré à l'aide de la boîte de dialogue de masque de bloc, voir Figure II.2.), avec les paramètres de base suivants, dont certains peuvent également être définis au moment de l'exécution avec la commande `ttSetKernelParameter` :

- **Fonction Init** : Le nom du script d'initialisation, voir Section 5.
- **Init function argument** : un argument optionnel du script d'initialisation. Cela peut être n'importe quelle structure Matlab.
- **Battery(batterie)**: Activation de cette case , si le noyau doit dépendre d'une source d'alimentation.
- **Clock drift (Dérive horaire)** : La dérive horaire égale à 0.01 si l'heure locale doit être 1% plus rapide que l'heure nominale (le temps réel de simulation).
- **Clock offset** : une constante de temps à partir du temps nominal.

Avec l'utilisation de `ttSetKernelParameter`, la vitesse d'exécution actuelle du noyau peut être réglée ainsi que la consommation d'énergie actuelle. Ceci permet de simuler une mise à l'échelle dynamique de la tension. Cette fonctionnalité peut être utile avec le bloc batterie.

II.2.1.2. Le Bloc réseau Filare/ sans fil (TRUETIME Wired/Wireless Network Blocks)

Le bloc réseau TRUETIME simule l'accès moyen et la transmission de paquets dans un réseau local.

Lorsqu'un nœud essaie de transmettre un message (en utilisant la primitive `ttSendMsg`), un signal de déclenchement est envoyé au bloc réseau sur le canal d'entrée correspondant. Lorsque la transmission simulée du message est terminée, le bloc réseau envoie un nouveau signal de déclenchement sur le canal de sortie correspondant au Nœud de réception.

Le message transmis est placé dans une mémoire tampon au niveau du nœud de l'ordinateur de réception. Un message contient des informations sur l'envoi et le nœud de réception de l'ordinateur, des données utilisateur arbitraires (généralement des signaux de mesure ou des signaux de commande), la longueur du message et des hommages en temps réel facultatifs tels qu'une priorité ou une échéance.

Six modèles simples de réseaux sont supportés :

- **Pour le réseau filaire** : il s'agit de CSMA/CD (par ex. Ethernet), CSMA/AMP (par ex. CAN), Round Robin (par ex. Token Bus), FDMA, TDMA (par ex. TTP) et Switched Ethernet.
- **Pour les réseaux non filaires** ; les protocoles MAC implantés dans TRUETIME sont : IEEE802.11b/g (*WiFi*) et IEEE802.15.4 (*ZigBee*). [5]

Dans un réseau local : le délai de propagation est ignoré, car il est généralement très faible. Seule la simulation au niveau paquet est prise en charge - il est supposé que les niveaux de protocole supérieurs dans les nœuds du noyau ont divisé les longs messages en paquets, etc.

Il est également un support de réseau en temps réel échographie dans la version 2.0 Bêta TRUETIME.

Les paramètres réseau suivants sont communs à tous les modèles :

- **Network number (numéro de réseau)** : le numéro du bloc de réseau. Les réseaux doivent être numérotés à partir de 1. Les Réseaux câblés et sans fil ne sont pas autorisés à utiliser le même numéro ;
- **Number of nodes (nombre de nœuds)** : le nombre de nœuds qui sont connectés au réseau. Ce numéro déterminera la taille des entrées SND (émission), RCV (réception) et Schedule (ordonnancement), ainsi que les sorties de bloc ;
- **Data rate (Débit de données) ((bits/s)** : la vitesse du réseau ;
- **Minimum frame size(taille minimale de l'image) (bits)** : ce paramètre indique la taille minimale de données, y compris toute surcharge introduite par le protocole. Par exemple, la taille minimale de la trame Ethernet, comprenant un en-tête de 14 octets et d'un contrôle CRC sur 4 octets, est de 512 bits ;
- **Pre-processing delay (délai(s) de prétraitement) (s)** : le temps de retard causé par l'interface de réseau sur l'extrémité d'émission. Ceci peut être utilisé pour modéliser, par exemple, une connexion série lente entre l'ordinateur et l'interface de réseau ;

- **Post-processing delay (s) Délai(s) de post-traitement (post-traitement)** Temps pendant lequel un message est retardé par l'interface réseau à l'extrémité de réception.
- **Probabilité de perte (0-1)** Probabilité qu'un message réseau soit perdu pendant la transmission. Les messages perdus consomment la bande passante du réseau, mais n'arrivent jamais à destination.

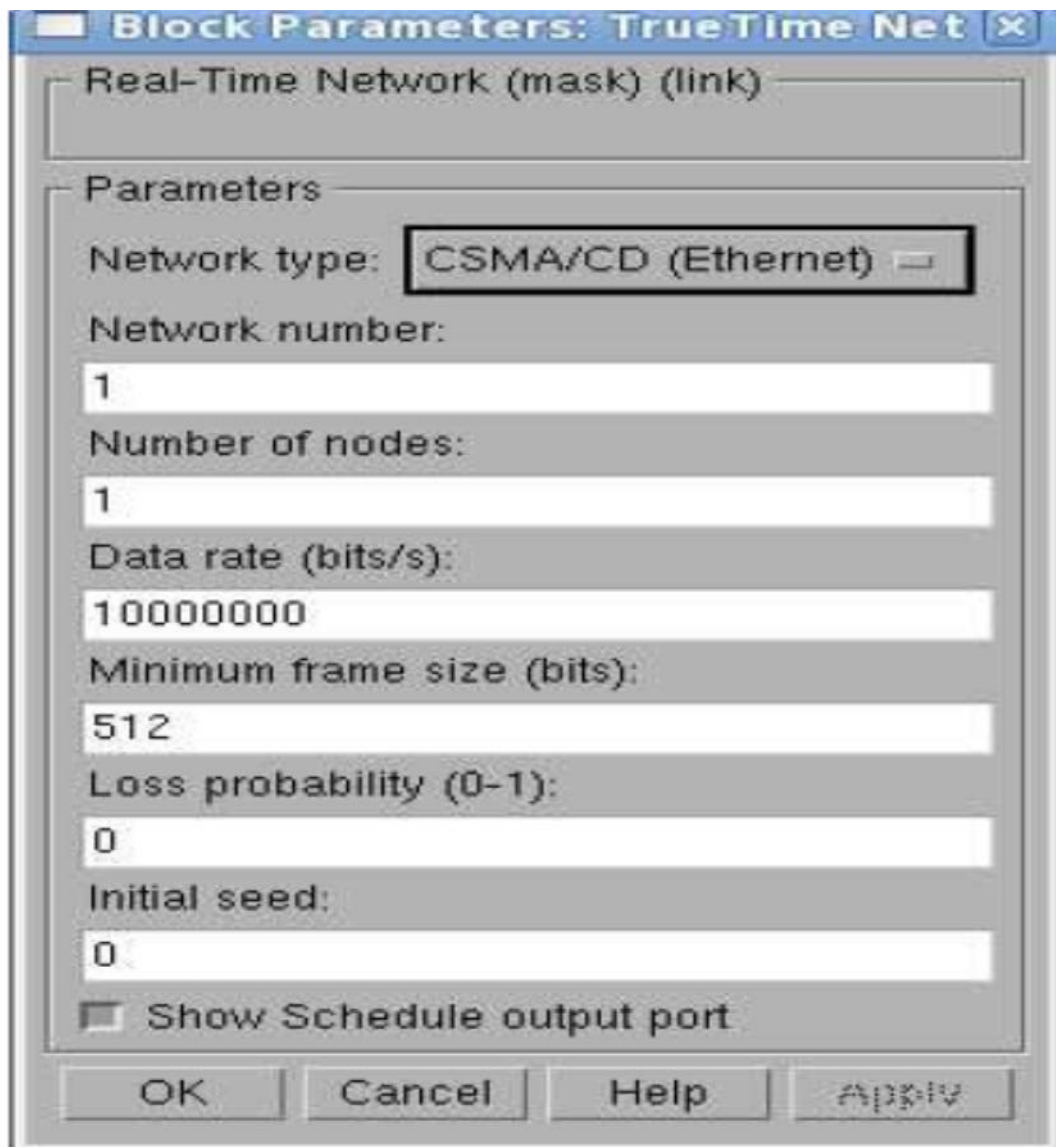


Fig.II.3 La boîte de dialogue du bloc **réseau filaire** TRUETIME

II.2.1.3. Le bloc réseau sans fil TRUETIME

L'utilisation du bloc réseau sans fil est similaire et fonctionne de la même manière que celle du bloc réseau câblé. Pour tenir compte également de la perte de chemin des signaux radio, il dispose d'entrées x et y pour spécifier l'emplacement réel des nœuds. Deux protocoles réseau sont actuellement pris en charge : IEEE 802.11b/g (WLAN) et IEEE 802.15.4 (ZigBee).

Le modèle de radio utilisé comprend un support pour :

- Réseaux sans fil ad hoc.
- Antenne isotrope.
- Incapacité d'envoyer et de recevoir des messages en même temps.
- Perte de chemin des signaux radio modélisés comme d/a où d est la distance en mètres et a est un paramètre choisi pour modéliser l'environnement.
- Interférences provenant d'autres terminaux.

Le bloc réseau sans fil est configuré via la boîte de dialogue de masque de bloc (voir Figure II.4) certains paramètres peuvent également être définis par nœud avec la commande `ttSetNetNetworkParameter`.

Le type de réseau détermine le protocole MAC à utiliser : il peut être 802.11b/g (WLAN) ou 802.15.4 (ZigBee).

Les paramètres suivants sont communs à tous les modèles :

- **Numéro de réseau** Le numéro du bloc réseau. Les réseaux doivent être numérotés à partir de 1. Les réseaux câblés et sans fil ne sont pas autorisés à utiliser le même numéro.
- **Nombre de nœuds** Nombre de nœuds connectés au réseau. Ce nombre déterminera la taille des entrées et sorties Snd, Rcv et Schedule du bloc.

- **Débit de données (bits/s)** La vitesse du réseau.
 - **Taille minimale de l'image (bits)** Un message ou une image plus courte que celle-ci sera rembourrée pour donner la longueur minimale. Indique la taille minimale de la trame, y compris toute surcharge introduite par le protocole. Par exemple, la plupart des protocoles réseau ont un nombre fixe de bits d'en-tête et de queue, le cadre doit donc être au moins de $taille\ f(en-tête) + f(queue)$.
 - **Transmit power (Puissance d'émission)** : Détermine la puissance du signal radio et, par conséquent, sa portée.
 - **Receiver signal threshold (seuil du signal du récepteur)** : Si l'énergie reçue est supérieure à ce seuil, le support est considéré comme occupé.
-
- **Path-loss exponent (Exposant de perte de trajet)** L'affaiblissement de trajet du signal radio est modélisé comme $\frac{1}{d^a}$, où d est la distance en mètres et a est un paramètre convenablement choisi pour modéliser l'environnement. Typiquement choisi dans l'intervalle [2-4].
 - **ACK timeout (Délai d'attente ACK) Délai pendant** lequel un nœud émetteur attendra un message ACK avant de conclure que le message a été perdu et de le retransmettre.
 - **Retry limit (Limite de réessayage)** : Le nombre maximum de fois qu'un nœud essaiera de retransmettre un message avant d'abandonner.

- **Error coding threshold (Seuil de codage d'erreur)** : Un nombre dans l'intervalle[0, 1] qui définit le pourcentage d'erreurs de bloc dans un message que le codage peut traiter. Par exemple, certains systèmes de codage peuvent reconstruire complètement un message s'il contient moins de 3 % d'erreurs de bloc. Le nombre d'erreurs de bloc est calculé à l'aide du rapport signal/bruit, où le bruit est celui de toutes les autres transmissions en cours.

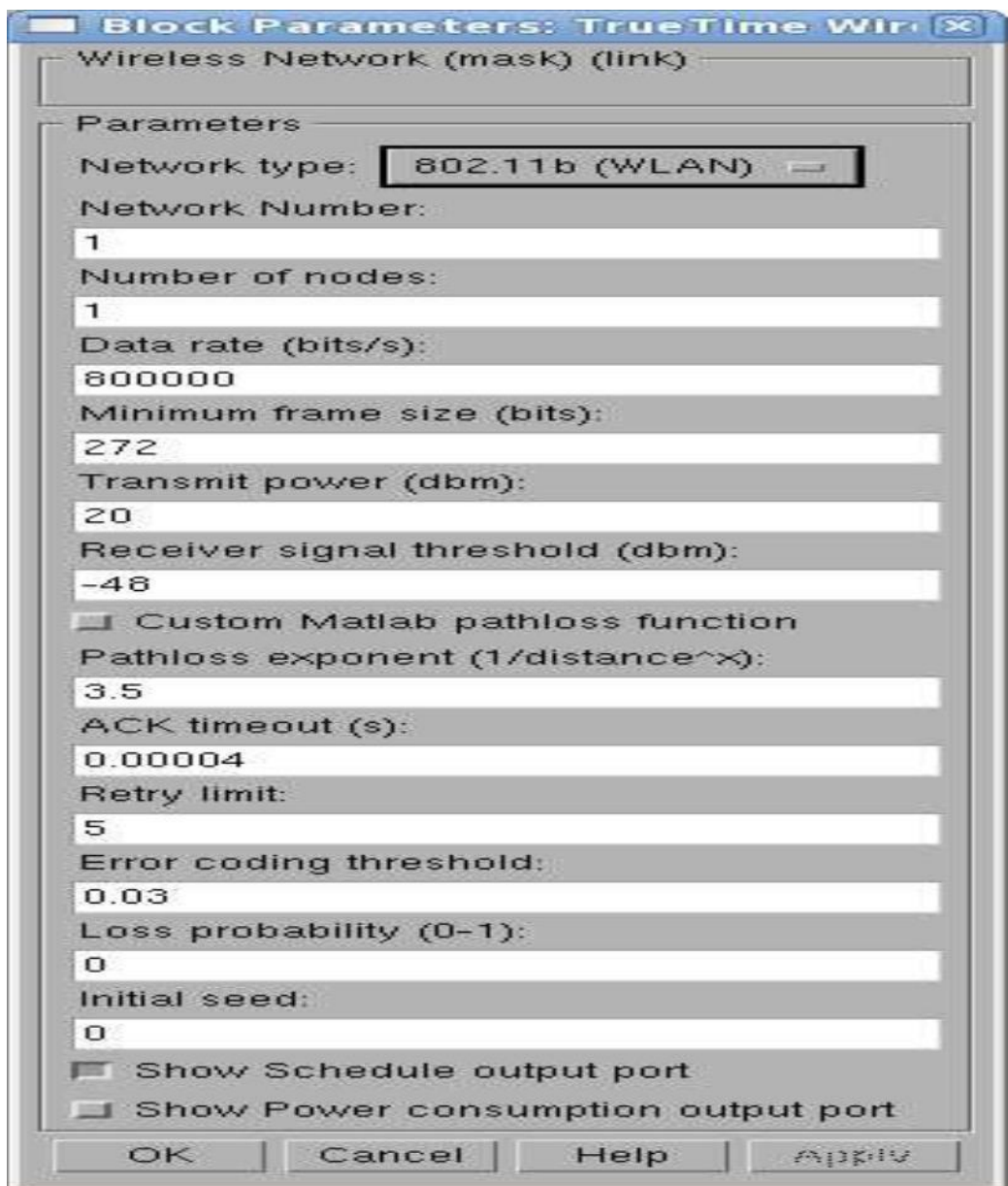


Fig.II.4. La boîte de dialogue du bloc réseau sans fil TRUETIME

II.2.1.4. La batterie TRUETIME:

Le bloc batterie possède un paramètre, la puissance initiale, qui peut être réglé à l'aide du masque de configuration. Pour utiliser la batterie, activez la case à cocher dans le masque de configuration du noyau et connectez la sortie de la batterie à l'entrée E du bloc noyau. Connectez chaque drain d'alimentation tel que la sortie P du bloc noyau, les modèles Simulink ordinaires et le bloc réseau sans fil à l'entrée P de la batterie. La batterie utilise un modèle d'intégrateur simple, de sorte qu'elle peut être chargée et rechargée à la fois.

Notez que le noyau n'exécutera aucun code s'il est configuré pour utiliser des batteries et que l'entrée P du bloc noyau est nulle.

Pour utiliser la batterie, il est nécessaire d'activer la case située dans le masque de configuration et de connecter la sortie de la batterie à l'entrée E de bloc de noyau. La batterie utilise un modèle intégrateur simple, de sorte qu'il peut être à la fois chargé et rechargé.

II.2.1.5. Les blocs d'interface réseau autonomes :

Les blocs réseau autonomes, nommés TrueTime Send et TrueTime Receive, peuvent être utilisés pour envoyer des messages en utilisant les blocs réseau sans utiliser les blocs du noyau. Cela signifie qu'aucun code d'initialisation ou de code de la tâche ne doivent être écrits.

Ceci permet de créer des simulations réseau TrueTime sans avoir à initialiser les noyaux, créer et installer des handlers d'interruption, etc.

En d'autres termes, il est possible de créer une simulation réseau complète dans Simulink sans aucun M-files. Il est également possible de mélanger ces blocs avec des blocs du noyau, de sorte que certaines stations utilisent les blocs réseau autonomes, tandis que d'autres utilisent les primitives standard ttSend et ttGetMsg depuis une fonction de code exécutée dans un bloc du noyau.

Les blocs réseau autonomes sont configurés à l'aide des boîtes de dialogue de masque de bloc, comme illustré à la Figure II.5 Les paramètres sont les mêmes que ceux utilisés dans les primitives TRUETIME send (envoyer) et TRUETIME Receive (recevoir).

Le bloc TrueTime Send dispose d'un port d'entrée de déclenchement Simulink, qui peut être configuré pour se déclencher en montée, en descente ou sur l'un des flancs.

Le bloc ttGetMsg possède un port de sortie de déclenchement optionnel. La valeur de la sortie de déclenchement commute entre 0 et 1 au fur et à mesure que les messages sont reçus.

Ce port peut être utilisé pour déclencher quelque chose qui doit être exécuté après la réception d'un nouveau message.

Selon [5] [14], la simulation de l'ensemble du réseau peut être créée dans Simulink en utilisant soit les fichiers m ou le code C ++.

Il est possible de mélanger les blocs autonomes avec des blocs de noyau dans une simulation. Cela signifie que certaines stations peuvent envoyer des messages sans codes de tâches m-fichier (par exemple, capteurs) et certaines stations utilisent des blocs de noyau (par exemple, contrôleurs).

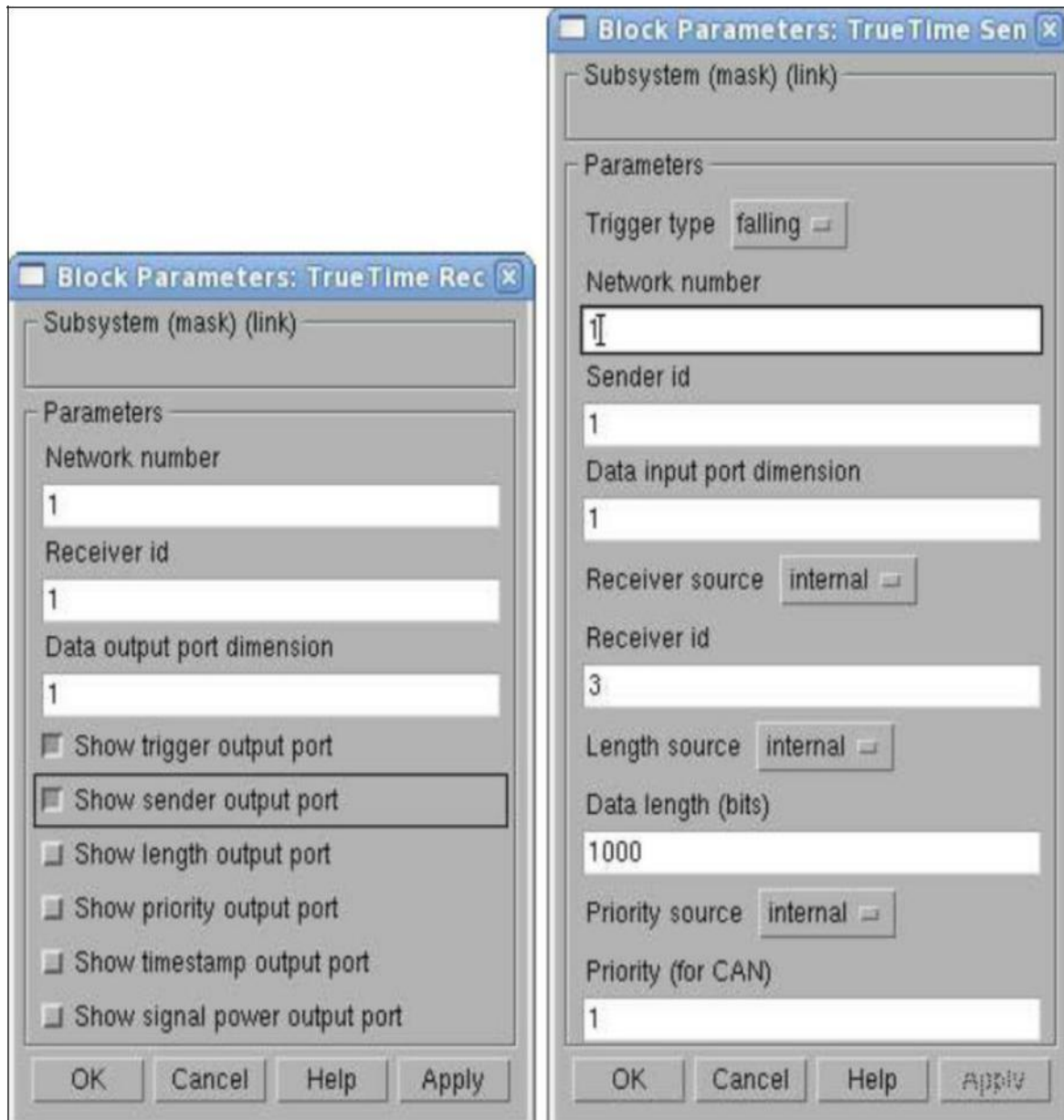


Fig.II.5. La boîte de dialogue des blocs autonomes TRUETIME

II.2.1.6. Quelques commandes utilisées en TRUETIME

- **ttInitKernel**: Initialiser le noyau du TRUETIME
- **ttInitNetwork**: Initialiser l'interface réseau du TRUETIME
- **ttCreatePeriodicTask**: Créer une tâche périodique TRUETIME.
- **ttCreateTask**: Créer une tâche TRUETIME

- **ttCreateEvent**: Créer un événement TRUETIME
- **ttAttachPrioFcn**: (utilisé que dans C++) Attribuer une fonction de priorité arbitraire à fin d'être utilisé par le noyau.
- **ttSetPriority**: Attribuer une priorité à la tâche.
- **ttSetPeriod**: Attribuer une période pour les tâches périodiques.
- **ttGetPriority**: Récupérer la priorité de la tâche.
- **ttGetPeriod** : Récupérer la priorité de la tâche périodique.
- **ttWait**: Attendre un événement.
- **ttNotifyAll**: Notifier toutes les tâches qui attendent des événements.
- **ttSleepUntil** : Mettre la tâche en attente pendant un certain moment.
- **ttSendMsg**: Envoyer un message sur le réseau.
- **ttGetMsg**: Récupérer le message qui est reçu à partir du réseau

II.3. Exemple de simulation d'un système commandé en réseau par TRUETIME (asservissement en vitesse moteur DC) : test de différents réseaux ...etc.

La simulation TRUETIME a été réalisée sur le système « servo moteur DC » où l'on remarque clairement les compartiments en nœuds du système.

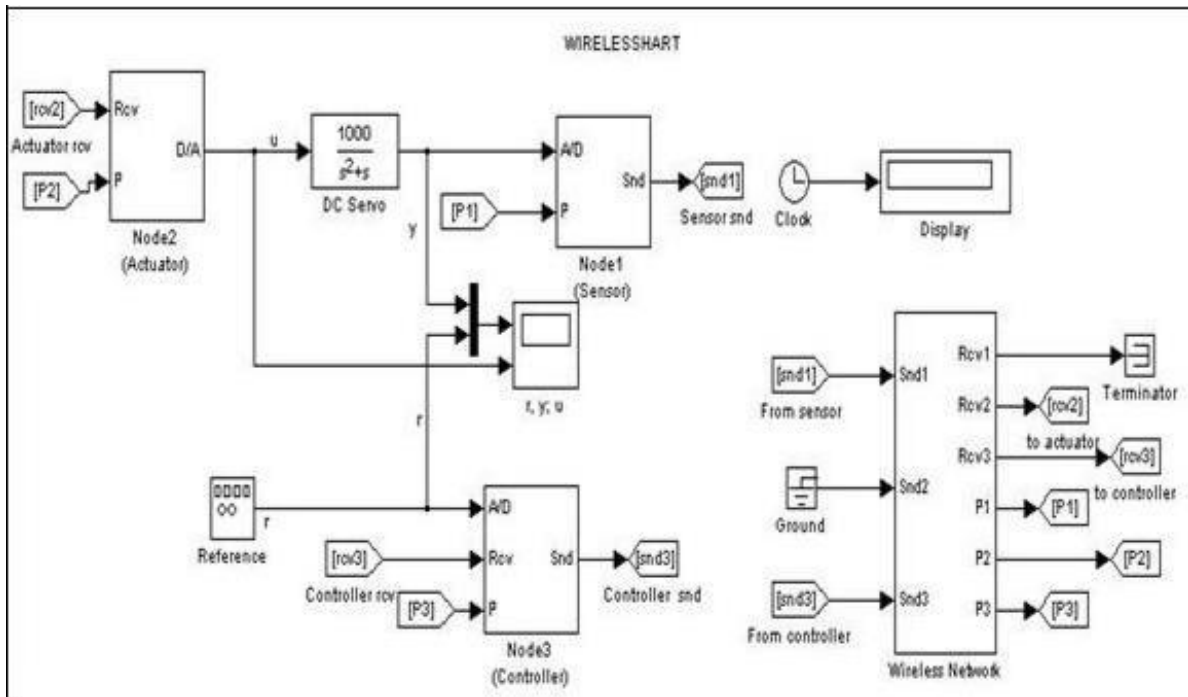


Fig.II.6. Modèle TRUETIME du système de contrôle d'un servo moteur DC.

L'exemple de simulation du système commandé par TRUETIME du « servo moteur » montre clairement les 3 nœuds :

- Nœud 1 : bloc du noyau TRUETIME du capteur
- Nœud 2 : bloc du noyau TRUETIME pour actionneur
- Nœud 3 : bloc du noyau TRUETIME pour contrôleur

Les résultats de la simulation du modèle sont portés sur la figure II.7 où les oscillations montre le transfert des données à l'aide du protocole WIRELESSHART implémenté sur le modèle de système de commande pour la commande des servomoteurs CC.

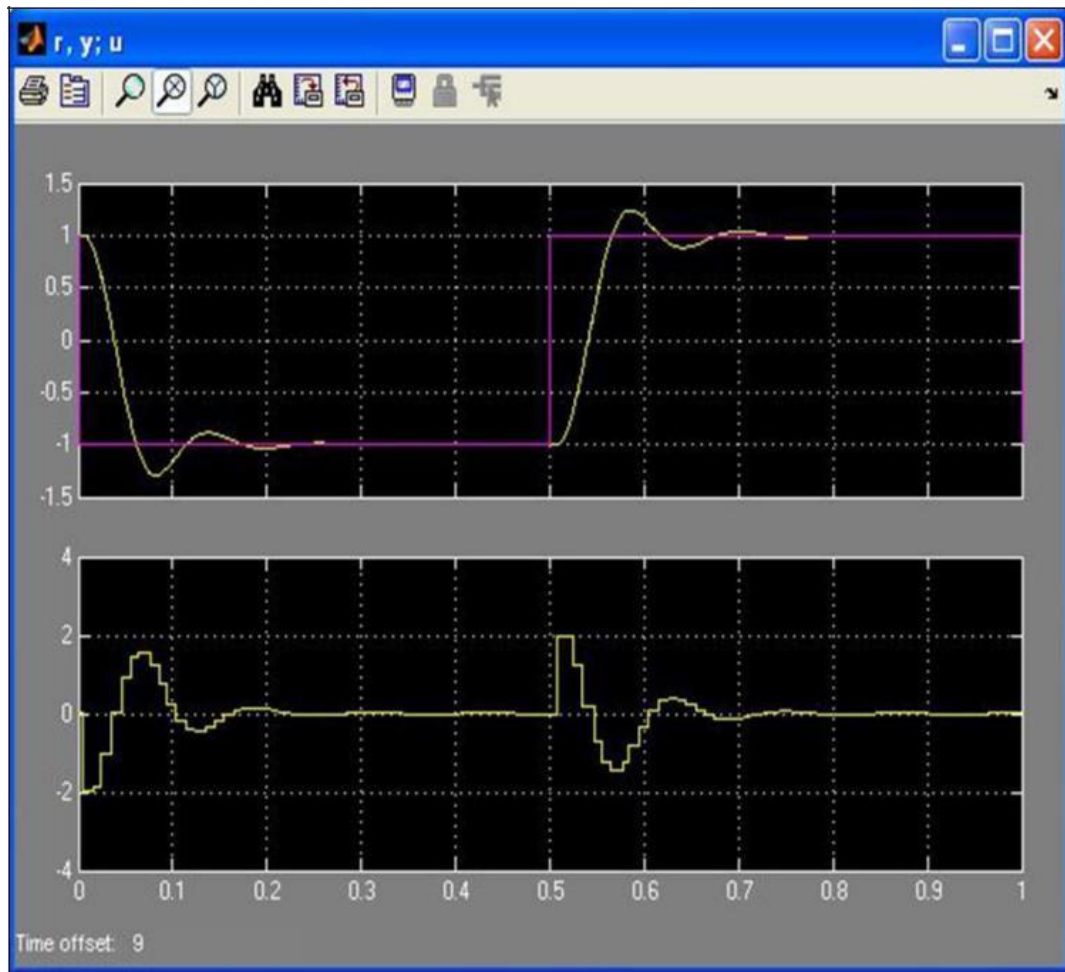


Fig. II.7. Résultats de simulation d'un simple contrôle TRUETIME

II.4. Conclusion :

Dans ce chapitre, nous avons présenté le simulateur TRUETIME qui fait partie de la plateforme Simulink de Matlab, simulateur très utilisé actuellement.

Ce simulateur offre l'avantage principal de simuler le comportement de systèmes commandés en réseau et d'aboutir à des résultats très fiables.

Un exemple de simulation TRUETIME sur un exemple de système « servo moteur DC » a été illustré où l'implémentation du protocole WIRELESSHART a permis une communication sûre et fiable avec peu de retard, sans influence sur la plage de débit et les performances du processus.

Chapitre III.

APPLICATION 1 :

Commande en réseau de la puissance

D'un système éolien par

La méthode de placement de pôles

Chapitre III. APPLICATION 1 : Commande en réseau de la puissance d'un système éolien par la méthode de placement de pôles

III.1. Introduction

III.1.1. Commande des systèmes dynamiques linéaires invariants

Un système de commande automatique consiste à “faire calculer” les entrées d'un système par un organe de commande en vue de satisfaire une tâche donnée. On peut le schématiser de la façon suivante :

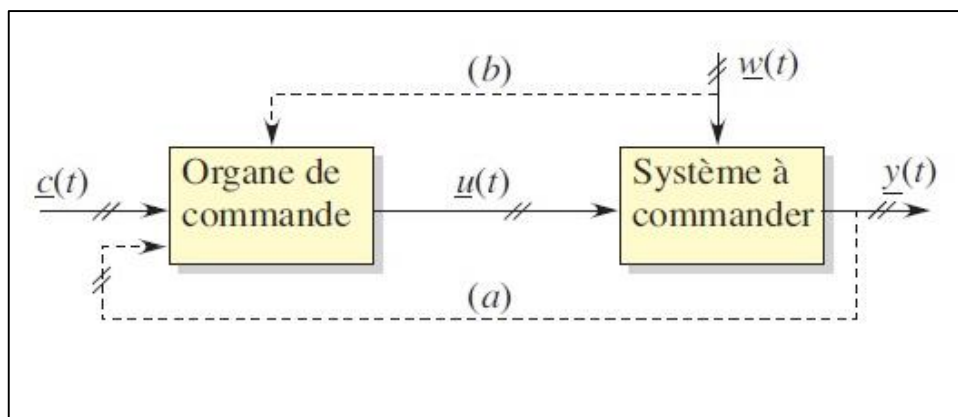


Figure III-1. Système de commande : (a) indique le cas de la boucle fermée

La mise en œuvre peut être effectuée en boucle ouverte ou fermée, cette dernière situation (figure III.1) s'avérant indispensable dans la très grande majorité des cas, ne serait-ce qu'à cause des perturbations $w(t)$ qui peuvent affecter le système à commander. On distingue aussi deux types de problèmes : *le suivi des consignes et la régulation*.

Dans le premier cas le système doit répondre avec précision et rapidité à toute variation des signaux d'entrée tandis que dans le second les sorties doivent conserver leurs valeurs en dépit des perturbations qui affectent le système.

Un certain nombre d'éléments sont cause d'écart entre les consignes et les sorties :

- Les perturbations affectant le système ;
- Les limites sur les commandes (saturations) ;
- L'utilisation incorrecte et/ou retardée par l'organe de commande des signaux qu'il reçoit

Lorsque les caractéristiques du système varient, le problème de la commande s'avère délicat et il peut être utile de prendre en compte les caractéristiques des perturbations dans l'algorithme de commande.

Lorsqu'on veut prendre en compte un critère dans l'algorithme de commande, on parle alors de *Commande optimale*, un certain nombre d'éléments doivent être pris en compte :

- Le critère d'optimalité de type min (Q) ou max (Q) qui peut constituer le but de la commande ou un ensemble de contraintes sur les commandes et/ou l'état du système ;
- Les informations sur les consignes et l'état du système ;
- Les caractéristiques du système à commander $y(t) = F(u(t), w(t))$, des connaissances sur $w(t)$ et sur les domaines de fonctionnement et de variation des diverses variables et fonctionnelles utilisées ;
- Et la part des informations sur le système qui sont prises en compte dans l'organe de commande.

Normalement l'information détenue, en particulier sur l'état du système et les caractéristiques des perturbations, est incomplète ou entachée d'erreurs (dans le cas contraire il n'y aurait pas vraiment de problème.) et l'optimalité dépend de ce que l'on connaît.

Il faut remarquer qu'en dépit du nom "*commande optimale*", ce que l'on cherche est une stratégie optimale (l'algorithme mis en œuvre dans l'organe de commande) et non une *commande optimale* satisfaisant un critère donné, par exemple une commande assurant une réponse en temps minimal.

D'un point de vue générale, on ne considérera que les systèmes dynamiques [11]) de Dimension finie, gouvernés par un ensemble d'équations de la forme :

$$\begin{aligned}\dot{x}(t) &= F(x(t), U(t)) \\ y(t) &= H(x(t), u(t))\end{aligned}$$

- où $x(t)(n \times 1)$ est le vecteur d'état (ensemble de variables d'état),
- $u(t)(m \times 1)$ le vecteur des entrées et
- $y(t)(r \times 1)$ le vecteur des sorties.

Les systèmes linéaires, ou linéarisés autour d'un point de fonctionnement (théorème de Hartman- Grobman), peuvent être décrits par un système différentiel vectoriel du premier ordre de la forme :

$$\dot{X}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t)$$

- où $A(t)(n \times n)$ est la matrice d'état du système et $B(t)$, $C(t)$ et $D(t)$ sont de dimensions respectives $(n \times m)$, $(r \times n)$, $(r \times m)$ (figure III.1.).

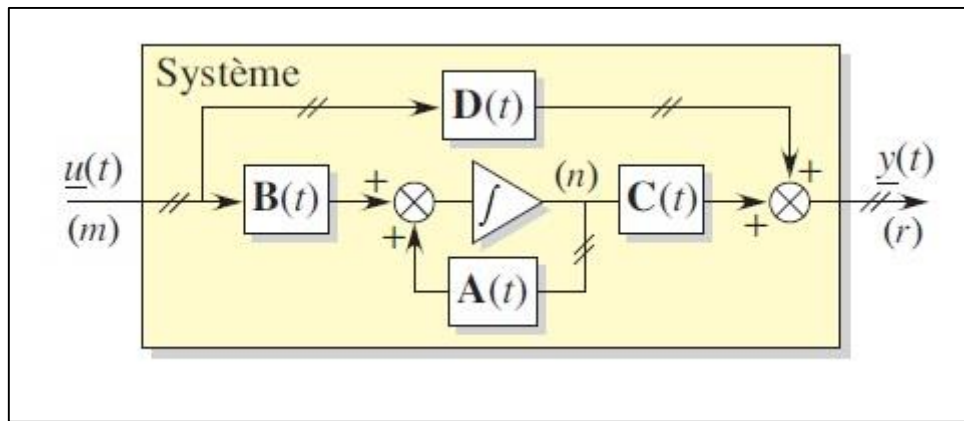


Figure III.2: Représentation d'état d'un système linéaire

Dans le cas des systèmes dynamiques linéaires invariants, les paramètres du système ne dépendent pas du temps. On a alors

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

Deux notions particulièrement importantes apparaissent dans l'étude de tels systèmes : la Commandabilité et l'observabilité .

Dans le premier cas on se demande s'il est possible de trouver une commande qui amène l'état d'un point à un autre de l'espace d'état en un temps fini.

Dans le second, on s'interroge sur la possibilité connaître l'état par observation des sorties et des entrées pendant un temps fini.

De nombreux critères d'observabilité et gouvernabilité peuvent être utilisés pour en vérifier l'existence [11]. Un résultat important indique qu'il est toujours possible de trouver une représentation d'état qui mette le système sous la forme dite forme canonique de Kalman suivante :

$$\left\{ \begin{array}{l} \frac{d}{dt} \begin{bmatrix} \underline{x}_{cno}(t) \\ \underline{x}_{co}(t) \\ \underline{x}_{ncno}(t) \\ \underline{x}_{nco}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{A}_{14} \\ 0 & \mathbf{A}_{22} & 0 & \mathbf{A}_{24} \\ 0 & 0 & \mathbf{A}_{33} & \mathbf{A}_{34} \\ 0 & 0 & 0 & \mathbf{A}_{44} \end{bmatrix} \begin{bmatrix} \underline{x}_{cno}(t) \\ \underline{x}_{co}(t) \\ \underline{x}_{ncno}(t) \\ \underline{x}_{nco}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ 0 \\ 0 \end{bmatrix} \underline{u}(t) \\ \\ \underline{y}(t) = [0 \quad \mathbf{C}_2 \quad 0 \quad \mathbf{C}_4] \begin{bmatrix} \underline{x}_{cno}(t) \\ \underline{x}_{co}(t) \\ \underline{x}_{ncno}(t) \\ \underline{x}_{nco}(t) \end{bmatrix} \end{array} \right.$$

- Où on a noté par l'indice "co" ce qui correspond au sous-espace *commandable* et *observable*, par l'indice "nco" ce qui correspond au sous-espace non *commandable* et *observable*, etc.

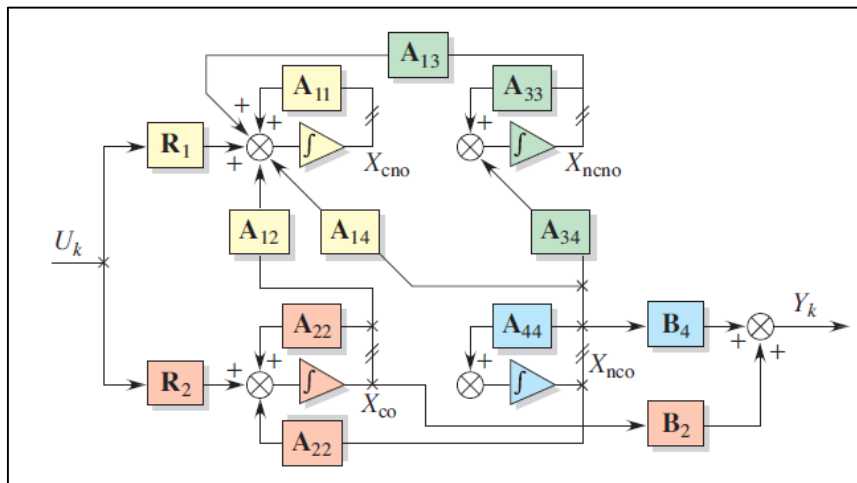


Fig. III.3. Forme canonique de Kalman

I.2. Bouclage du système

Dans le domaine de la commande des systèmes la notion de "bouclage" est essentielle. Tous les efforts se concentrent en effet sur la façon d'effectuer cette opération. Les avantages que l'on en retire concernent à la fois la maîtrise des perturbations qui affectent le système, la qualité du suivi de consigne et/ou de la régulation, l'ergonomie de la commande. . .

Le simple "retour de sortie" accompagné de la présence d'un compensateur, ou correcteur, dans une forme connue (le compensateur PID est le plus commun) comme organe de commande du procédé constitue dans beaucoup de cas une solution satisfaisante (figure III.4.).

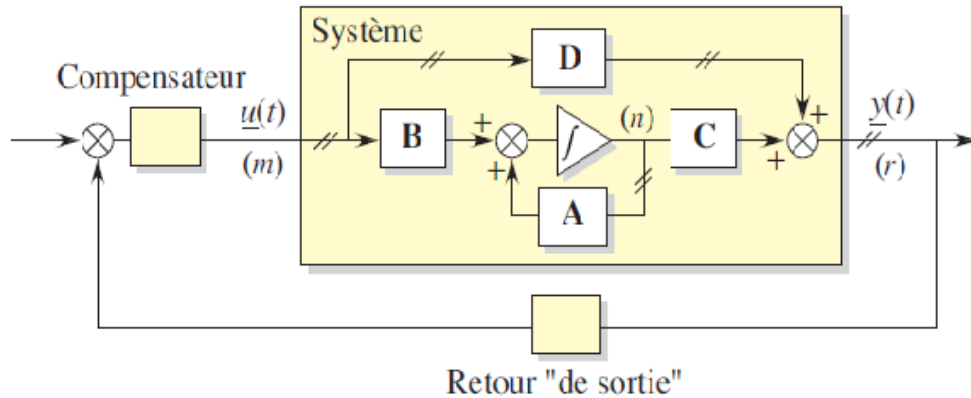


Fig.III.4. Système avec bouclage de sorties

Cette solution ne tient cependant pas compte de toutes les “variables” que l’on pourrait utiliser pour décrire le fonctionnement du système. On peut donc facilement imaginer améliorer le comportement du système bouclé en prenant en compte toutes ces informations.

III.1.2. Introduction au système éolien

Le développement et l’exploitation des énergies renouvelables ont connu une forte croissance au cours de ces dernières années. Naturellement décentralisées, il est intéressant de les exploiter sur le lieu de consommation, en les transformant directement soit en chaleur, soit en électricité selon les besoins.

La production d’électricité décentralisée par sources d’énergies renouvelables offre une plus grande sûreté d’approvisionnement des consommateurs tout en respectant l’environnement.

Aujourd'hui, l'énergie éolienne est devenue une réalité mondiale incontournable. L'évolution de la capacité de production d'électricité par les éoliennes n'a pas cessé d'augmenter depuis 1980. C'est la source qui progresse le plus rapidement.

L'industrie éolienne est en mesure de devenir une industrie énergétique mondiale si on se base sur l'accélération des installations dans la dernière décennie. Un marché important et compétitif s'est établi à un niveau international imposant à l'industrie le besoin d'améliorer la conception des éoliennes dans le but de réduire les coûts.

Il est même prévu que le vent sera la source de production d'électricité la moins chère. Il est clair qu'actuellement le coût de l'énergie éolienne concurrence celui de l'énergie

conventionnelle. Grâce à la progression de la productivité et à l'amélioration des rapports qualité/prix des matériaux, les coûts d'installations des éoliennes diminuent d'années en années [14].

III.2. Présentation du système éolien à commander

III.2.1. Les composants de l'éolienne

La figure III.1 montre la vue éclatée d'une éolienne avec ses différents composants [14].

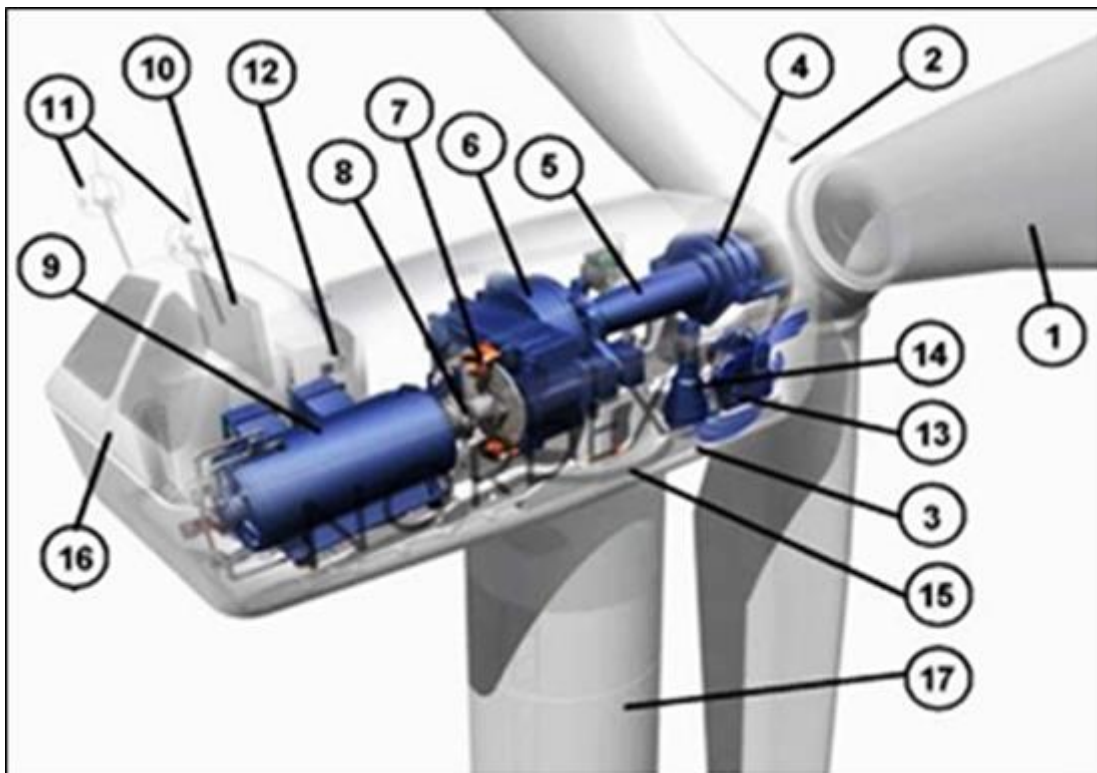


Fig. III.5. Vue d'une éolienne éclatée

1 : Pales ; 2 : Moyeu rotor ; 3 : Nacelle ; 4 : Cardan ; 5 : Transmission ; 6 : Multiplicateur De vitesse ; 7 : frein à disque ; 8 : Accouplement ; 9 : Génératrice ; 10 : Radiateur de refroidissement ; 11 : Centrale de mesures ; du vent ; 12 ; Contrôle ; 13 : Centrale hydraulique ; 14 : Mécanisme d'orientation face au vent ; 15 : Paliers du système d'orientation équipés d'un frein à disque ; 16 : Capot ; 17 : Mât.

III.2.2. Principe de fonctionnement d'un système éolien

Les systèmes de conversion de l'énergie éolienne, sont des mécanismes qui transforment l'énergie cinétique du vent en énergie électrique.

La figure suivante montre le principe de fonctionnement d'un système éolien [14]

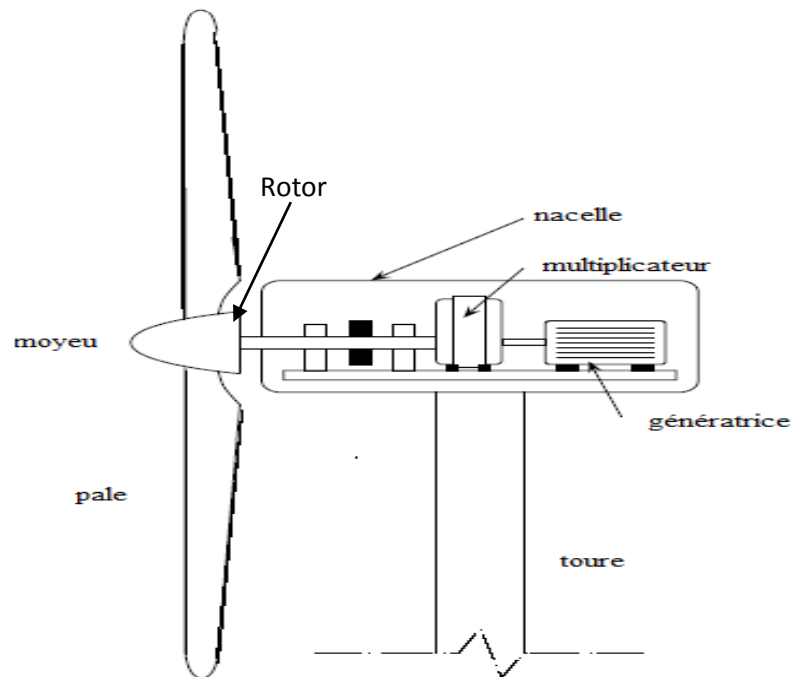


Fig. III.6. Fonctionnement d'une éolienne

L'énergie cinétique du vent est capturée par les pales et transmise au rotor, ce dernier transmet à son tour cette énergie à la machine électrique grâce au multiplicateur. Ce dernier augmente la vitesse du rotor à des valeurs appropriées à conduire la génératrice pour produire l'énergie électrique. Il existe plusieurs types d'éolienne (à vitesse fixe, variable, ...etc.) avec différentes génératrices synchrones et asynchrones. Enfin les bornes de la génératrice électrique sont reliées au réseau.

III.3. Boucle de commande de puissance en réseau d'un système éolien

Il existe plusieurs stratégies de commande des systèmes éoliens parmi lesquelles : les boucles de commande en vitesse, couple et puissance.

III.3.1. Modélisation de la chaîne de conversion éolienne

Les éoliennes permettent de convertir l'énergie du vent en énergie électrique, cette conversion se fait en deux étapes :

- Au niveau de la turbine (rotor), qui extrait une partie de l'énergie cinétique du vent disponible pour la convertir en énergie mécanique, en utilisant des profils

aérodynamiques. Le flux d'air crée autour du profil une poussée qui entraîne le rotor et constitue par la suite une force parasite.

- Au niveau de la génératrice, qui reçoit l'énergie mécanique et la convertit en énergie électrique, transmise ensuite au réseau électrique.

La production d'électricité à partir d'aérogénérateur ou éolienne dans un site isolé, nécessite l'utilisation de l'ensemble machine électrique-convertisseur statique. Nous savons que l'éolienne procure son énergie du vent, par conséquent, il existe une relation entre la vitesse du vent et la vitesse de rotation, le couple et la puissance sur le rotor d'une éolienne [15]. Nous définissons alors la puissance éolienne par la relation suivante :

$$P = C_p \frac{\rho \cdot S \cdot v^3}{2}$$

C_p : Coefficient qui dépend de la forme du rotor et de la vitesse du vent, ainsi qu'il existe une relation linéaire entre la vitesse du vent et la vitesse angulaire du rotor

ρ : Masse volumique de l'air en kg/m^3 .

S : Surface traversée par l'air en $\text{m}^2 \rightarrow S = \pi R^2$

R : Rayon du rotor de l'éolienne

V : Vitesse du vent en m/s .

III.3.2. Boucle de commande de puissance en réseau

Dans ce mémoire, nous allons présenter la boucle de commande en puissance dont le principe est le suivant : la puissance extraite par le rotor (appelée puissance aérodynamique) P_a est proportionnelle au facteur de puissance $C_p(\lambda, \beta)$, elle est donnée par:

$$P_a = \frac{1}{2} \cdot \rho S C_p(\lambda, \beta) v^3$$

Avec : $S = \pi R^2$, l'aire de la surface balayée par le rotor, ρ c'est la masse volumique de l'air, $C_p(\lambda, \beta)$ est le coefficient de puissance, il exprime la capacité de la turbine à extraire de l'énergie du vent, ce paramètre caractéristique de l'éolienne est fonction de la vitesse spécifique λ et du calage β , avec:

$$\lambda = \frac{\Omega_h R}{v}$$

Avec : Ω_h désigne la vitesse tangentielle du bout de la pale, R est le rayon de l'aire balayée par le Rotor.

On peut commander les systèmes éoliens en leur imposant une puissance de référence maximale extraite, selon la vitesse du vent disponible. Pour cela, on utilise la boucle de commande en puissance par la figure suivante :

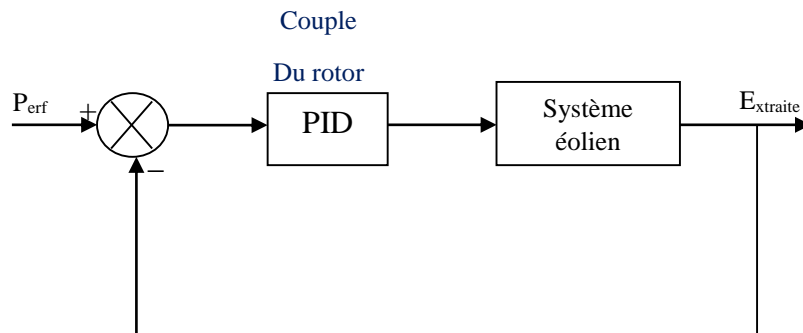


Fig. III.7. Boucle de commande en puissance

Dans ce chapitre, nous allons utiliser un modèle d'un système éolien de haute puissance (6-kw), à vitesse variable, avec pales fixes.

La génératrice est asynchrone à cage d'écureuil SCIG (Squirrel-cage Induction Génération).

Le tableau suivant résume les paramètres techniques du système :

Rotor de turbine	Boite de vitesse (drive train)	Génératrice SCIG
Longueur de pale : R=2.5 m	Le rapport de multiplication : $i=6.25$ LSS Inertie : $J_l = 3.6 \text{ kg.m}^2$ Efficacité : $\eta=0.95$	$P=2$, $R_s=1.265\Omega$, $R_r=1.43 \text{ m}\Omega$, $L_m=0.1397 \text{ H}$, $L_s = 0.1452 \text{ H}$, $L_r = 0.1452 \text{ H}$, $w_s = 100\pi \text{ rad/s}$, $\Gamma_{Gmax} = 40 \text{ Nm}$, $v_s = 220\text{V}$

Tableau III.1 Paramètres du système éolien

La fonction de transfert entre le couple du rotor et la puissance extraite est donnée par :

$$G(S) = \frac{35}{0.0036S^2 + 0.37S + 1}$$

III.3. Détermination des paramètres du contrôleur PI par la méthode de placement de pôles

Cette méthode consiste à choisir les pôles désirés du système corrigé en boucle fermée, car ces derniers déterminent la dynamique du système. Les paramètres du contrôleur Proportionnel intégral PI seront alors choisis de telle façon que le système corrigé aura les pôles désirés en boucle fermée. La boucle de commande est présentée par la figure suivante :

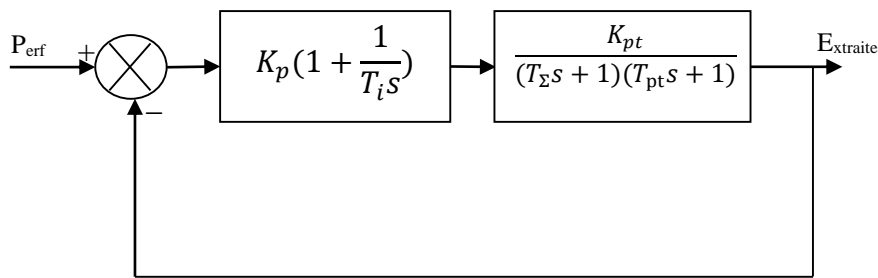


Figure III.8. Boucle de commande en puissance du système éolien de (6 KW)

Avec : $K_{pt} = 35$, $T_{\Sigma} = 0.01$, $T_{pt} = .36$

➤ **Application de la méthode de placement de pôles :**

La fonction de transfert en boucle fermée du système corrigé présente deux pôles et un zéro, il s'agit de dynamique ben connue d'un système du deuxième ordre :

$$\frac{1}{s^2 + 2zW_n \cdot s + W_n^2}$$

On pourra donc imposer les performances désirée par choix judicieux de la fréquence naturelle ω_n et du facteur d'amortissement z . Les paramètres du régulateur PI seront donc donnée en fonction de ω_n et z comme suite [8] :

$$\begin{cases} T_i = \frac{2z}{\omega_n} - \frac{1}{\omega_n^2 \cdot T_{pt}} \\ K_p = \frac{T_i \cdot T_{pt}}{K_{pt}} \cdot \omega_n^2 \end{cases}$$

➤ **Application numérique :**

Pour $\omega_n = 0.66$ rad/s et $z = 0.7$ et en utilisant l'équation de la fonction transfert en boucle fermée du système corrigé on trouve les paramètres du régulateur PI :

$$K_p = 0.10 \qquad T_i = 0.36$$

III.4. Description du système éolien sous Simulink/Matlab :

Cette partie explique en détail les phases d'ajustement de la simulation TRUETIME nécessaire à l'analyse du comportement du protocole CSMA/CD (Ethernet) implémenté sur l'exemple du système de contrôle avec trois nœuds pour le contrôle de puissance d'une éolienne.

La figure III.9 représente la simulation du contrôleur PI de la méthode de placement des pôles appliquée au système de l'éolienne représenté par sa fonction de transfert en boucle fermée ; les deux paramètres K_p et T_i précédemment calculés seront implémentés au block du régulateur PI.

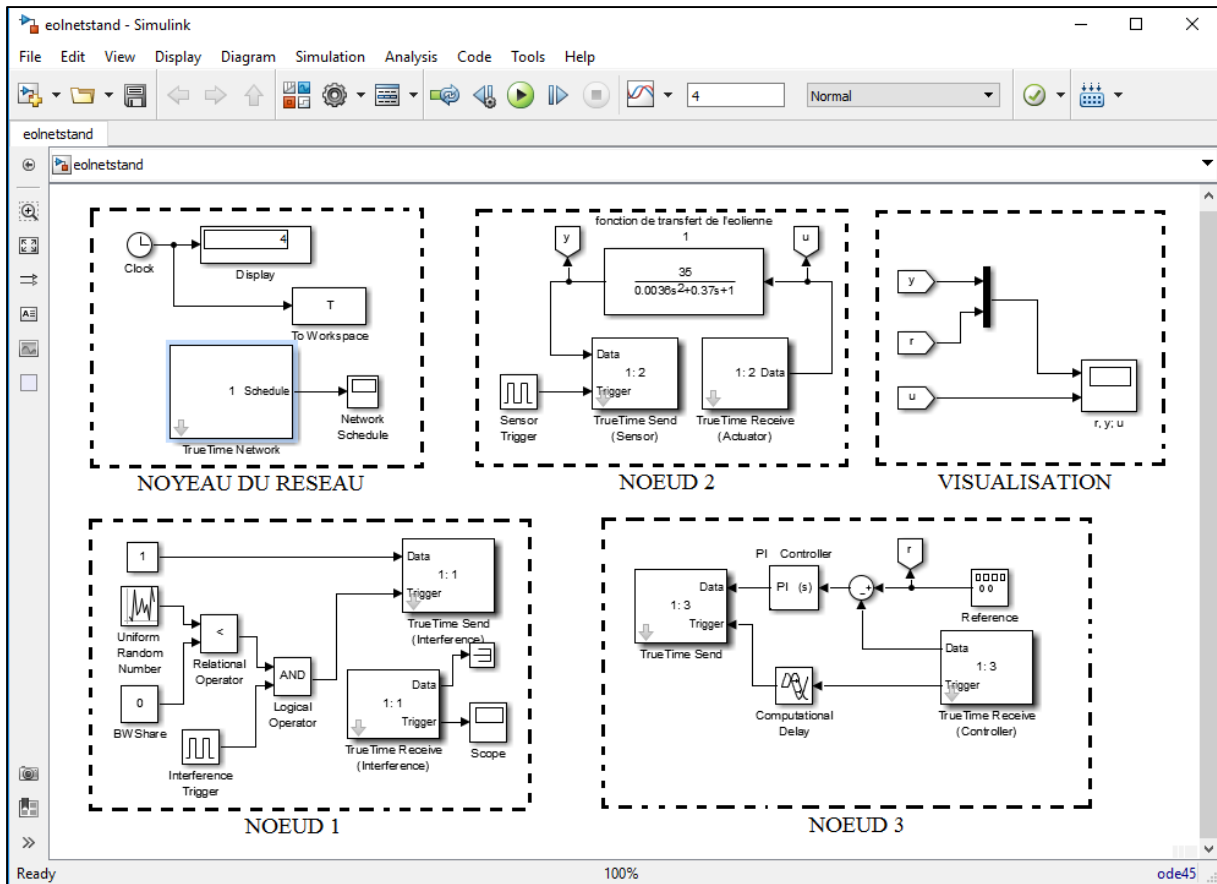


Fig. III.9. Configuration du système éolien sous Simulink/TRUETIME

Sous Simulink en utilisant les blocks du simulateur TRUETIME intégré à MATLAB, la commande en réseau du système éolien sera constituer d'un réseau TRUETIME Network et de trois nœuds comme suit :

- Noyau du réseau (TRUETIME Network)
- Nœud 1 simulation de perturbations
- Nœud 2 système éolien à commander
- Nœud 3 commande par régulateur PI

La figure III.10 représente le schéma Simulink détaillé du noyau du réseau . Dans cette partie du simulateur nous avons utilisé le block TRUETIME Network relié au block network Schedule et un TIMER afin de paramétrer le réseau de base ainsi que le nombre de nœuds reliés à ce dernier. De plus, nous avons configuré la graduation du temps de réponse du système global.

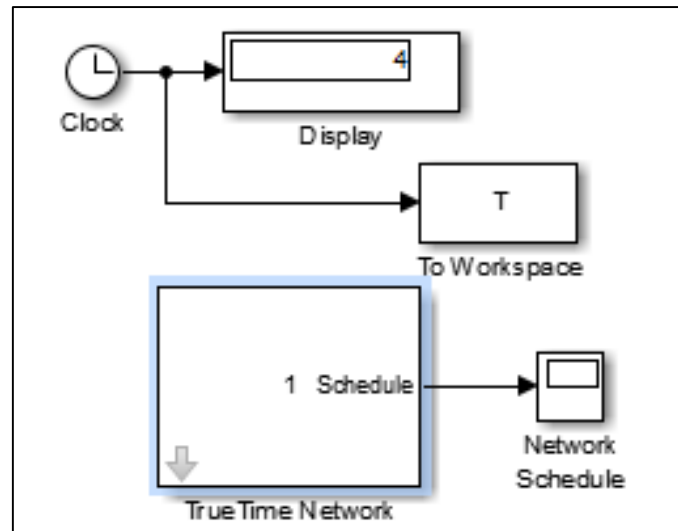


Fig. III.10. Noyau du réseau (TRUETIME Network)

Notons que les paramètres du block TRUETIME Network nous offrent la possibilité de changer le protocole de communication entre les différents éléments du réseau dans notre cas le protocole choisi est CSMA/CD Ethernet

Le protocole Ethernet a une vitesse de transfert de données relativement élevée par rapport à la norme IEEE 802.3 pour les réseaux informatiques filaires, bien que d'autres protocoles peuvent être mis en œuvre. Dans le cas de la commande en réseau d'une éolienne nous avons choisi ce protocole pour sa vitesse de transfert et ses pertes de données relativement moindres dans une distance importante, pour nous offrir un espace de simulation de perturbations plus larges (pertes de données-partage de la bande passante).

La figure III.11 représente la boîte de dialogue du block TRUETIME Network dans laquelle on définit les paramètres suivants :

Nombres de réseaux =1

Nombres de nœuds = 3

Probabilités de pertes de paquets =initialement a 0%

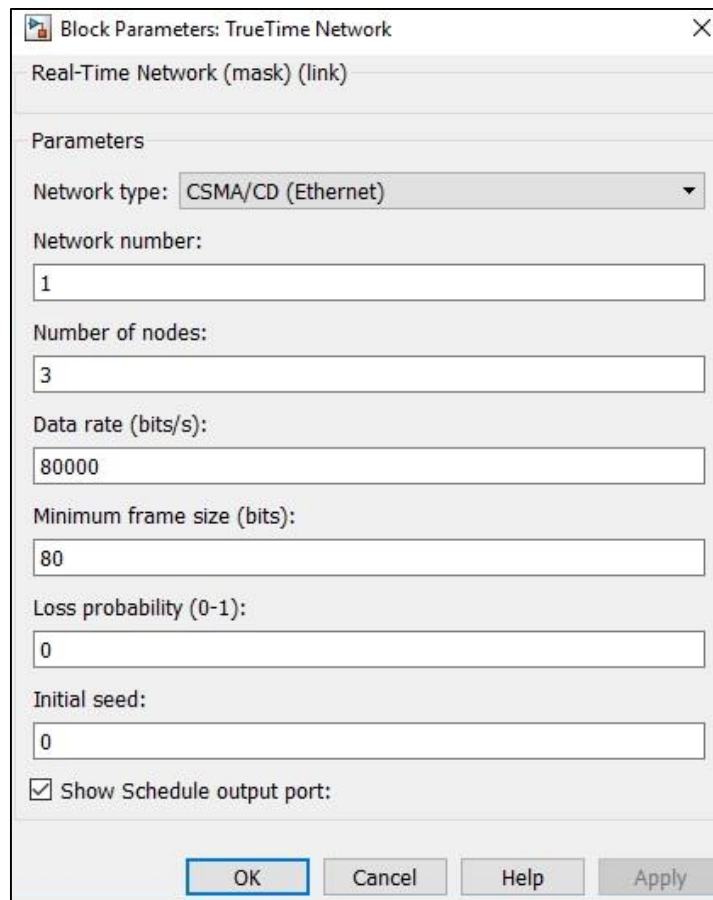
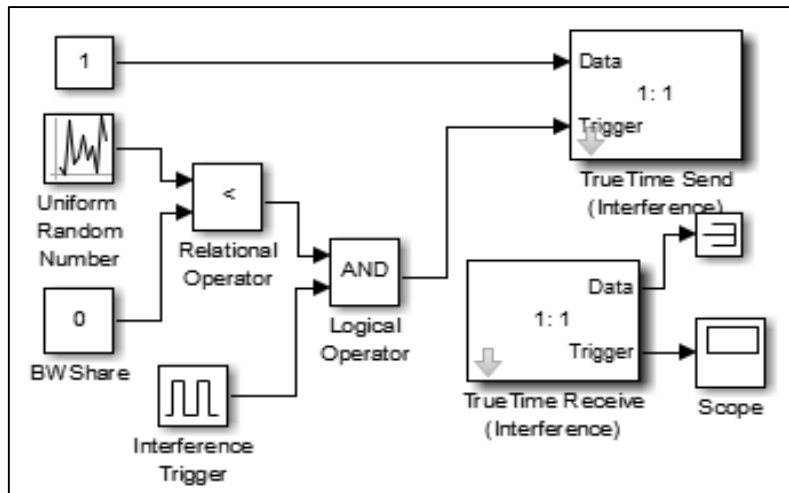


Fig. III.11. Paramètres du block TRUETIME Network

Le nœuds 1 est un nœud qui simule différentes perturbations relatives aux réseaux, la figure III.12 représente le nœuds 1 dans son intégralité.

Il inclut le block d'une porte logique AND qui additionne les différentes interférences simulées plus un paramètre (BW. SHARE) qui représente le pourcentage de partage de la bande passante.

Le tout sera redirigé vers la partie TRUETIME TRIGGER qui renvoie les interférences additionnées en sortie du nœud 1 vers l'entrée du système relié au réseau.



III.12. NEUD 1 simulation de différentes perturbations du réseau

La figure III.13 représente le nœud 2 qui constitue le système éolien à commander, la fonction de transfert du système a été intégrée au block système relié en sortie (y) au block TRUETIME SEND qui constitue la partie capteur du block réseau TRUETIME, en entrée (u) la boucle a été reliée à la partie actionneur du TRUETIME. Notons que ces parties capteurs- capteur actionneur du trentième ne sont pas réellement les capteurs actionneur du système mais sont à la base de la simulation du réseau auquel est relié la partie système éolien.

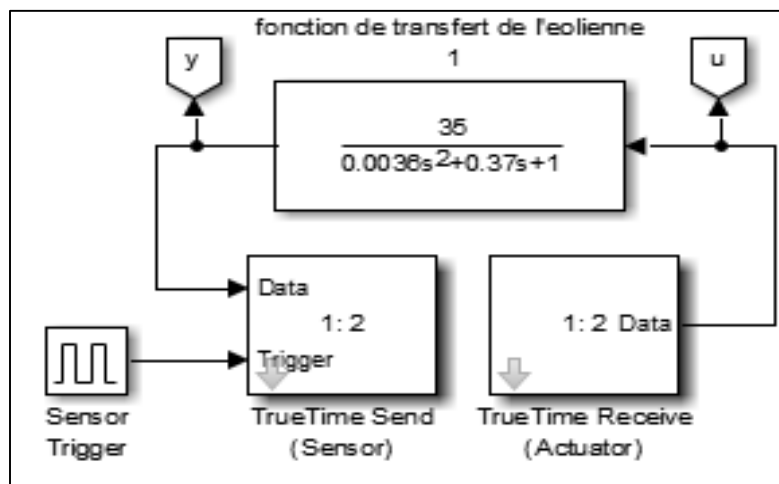


Fig.III.13. NEUD 2 système éolien à commander

Le nœud 3 représenté dans la figure III.14 montre la boucle de commande en réseau du système avec l'intégration au nœud du contrôleur PI relative à la méthode de placement de pôles. Le contrôleur du TRUETIME est relié à un intégrateur en incluant la notion du réseau a la boucle en sortie la réponse du contrôleur PI est intégrée au block TRUETIME SEND qui revoie la réponse au block réseau trentième pour faire la liaison avec le nœud 2

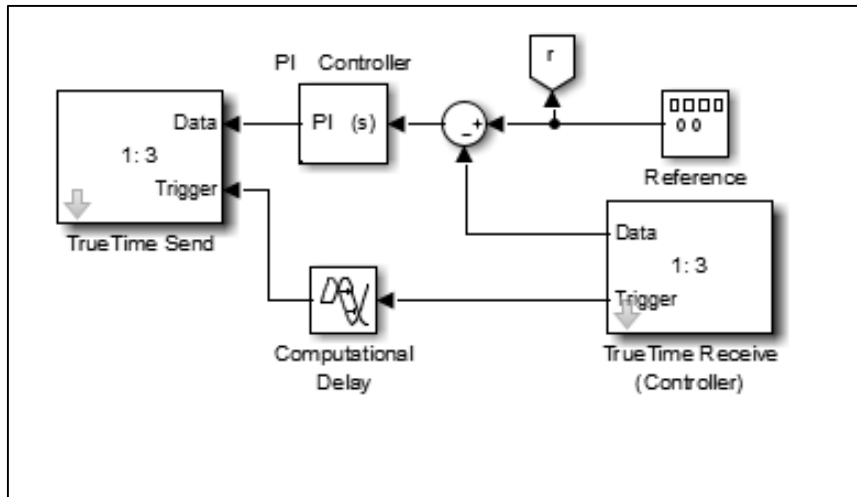


Fig.III.14. Nœud 3 commande par régulateur PI

La partie visualisation représentée dans la figure III.15 représente l'interfaçage du système, un multiplexeur a servi à relier les deux signaux relatifs à la sortie du système (y) et le signal de référence à atteindre (r) pour pouvoir faire plus aisément la comparaison entre les deux. Nous nous sommes choisis d'afficher séparément le signal de commande afin de faciliter la lecture de la réponse indicielle du système par rapport à la référence ; le tout étant relié à un scope qui affichera les résultats de la simulation du TRUETIME.

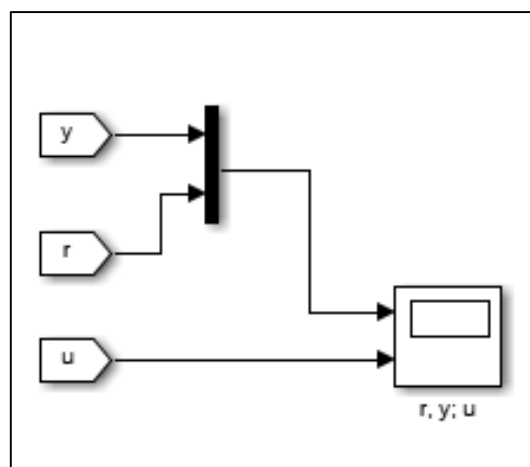


Fig.III.15. Visualisation du système éolien sous Simulink

III.4. Test de différents scénarios de perturbations sur le réseau (pertes de paquets, partage de la bande passante)

Dans cette partie du mémoire nous allons faire subir au réseau deux types de perturbations qui peuvent survenir dans le cas réel à savoir la perte de l'information et le partage de la bande passante.

Notons que la perte de l'information qui peut être due à des interférences, à l'endommagement partiel des relais, ou à une mauvaise connexion entre les différents éléments du réseau a été dans notre cas caractérisée par une perte de paquet dites (PACKAGE LOSS)

Le partage de la bande passante dit (BW. SHARE) quant à lui peut être dû à l'utilisation simultanée d'une même ressource ou juste l'exécution de plusieurs tâches en simultané dans un même réseau.

Les deux perturbations PACKAGE LOSS et BW SHARE sont données en pourcentage intervenant sur le réseau, dans les conditions idéales nous prendrons les deux paramètres à 0%.

La réponse indicielle du système (y) est représentée en rouge sur les graphes comme dans la figure III.12, la référence du système (r) est représentée en bleu et constitue la réponse idéale.

Remarque

La commande a été affichée séparément, les commentaires porteront uniquement sur la réponse du système de laquelle on déduira quelle commande en réseau est la plus adaptée à notre système éolien.

III.4.1. Scenario 1 : PACKAGE LOSS=0% BW SHARE=0% (condition idéale)

Sur la figure III.12 on peut voir un bon suivi de la consigne avec un temps de réponse de 0.26s

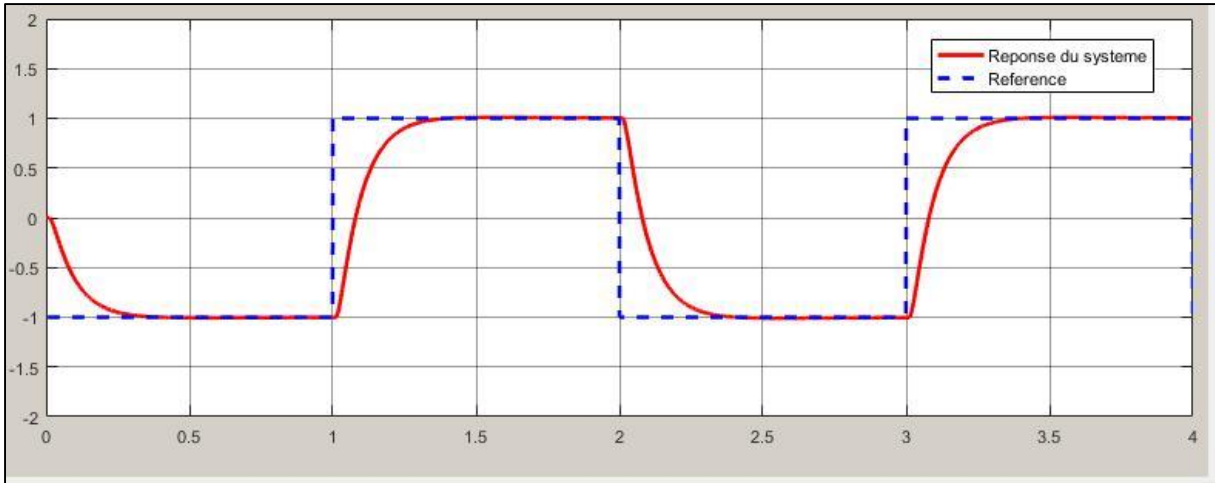


Fig.III.16. Réponse indicielle du système scenario 1

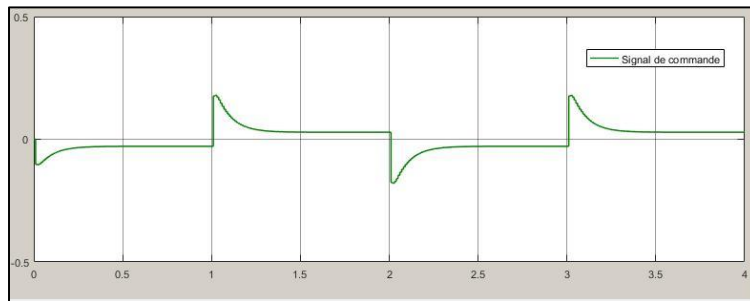


Fig.III.17. Signal de commande scenario 1

III.4.2. Scenario 2 : PACKAGE LOSS =0% BW SHARE = 30%

Dans la figure III.18 la réponse du système est toujours stable et avec un bon suivi de la consigne de commande et un temps de réponse de 0.26 seconde on note l'apparition d'un léger dépassement a 3.05 s ainsi qu'un infime retard.

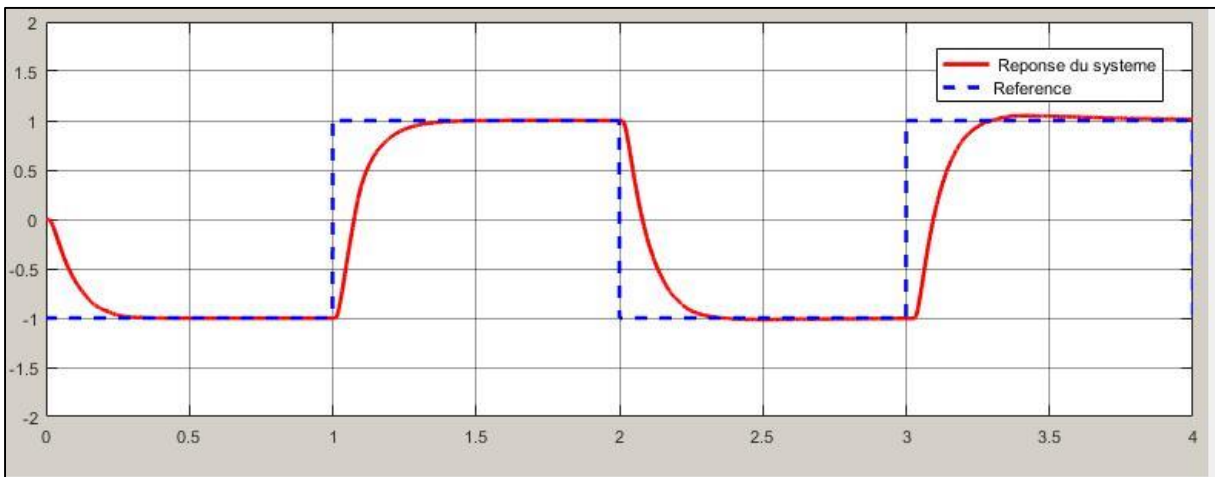


Fig.III.18. Réponse indicielle du système scenario 2

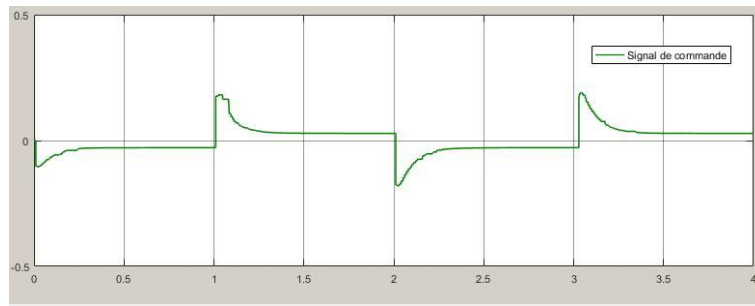


Fig.III.19. Signal de commande scenario 2

III.4.3. Scenario 3 : PACKAGE LOSS=0% BW SHARE=40%

Sous ces conditions on observe sur la figure III.20 une détérioration totale des performances de suivi et l'instabilité du système.

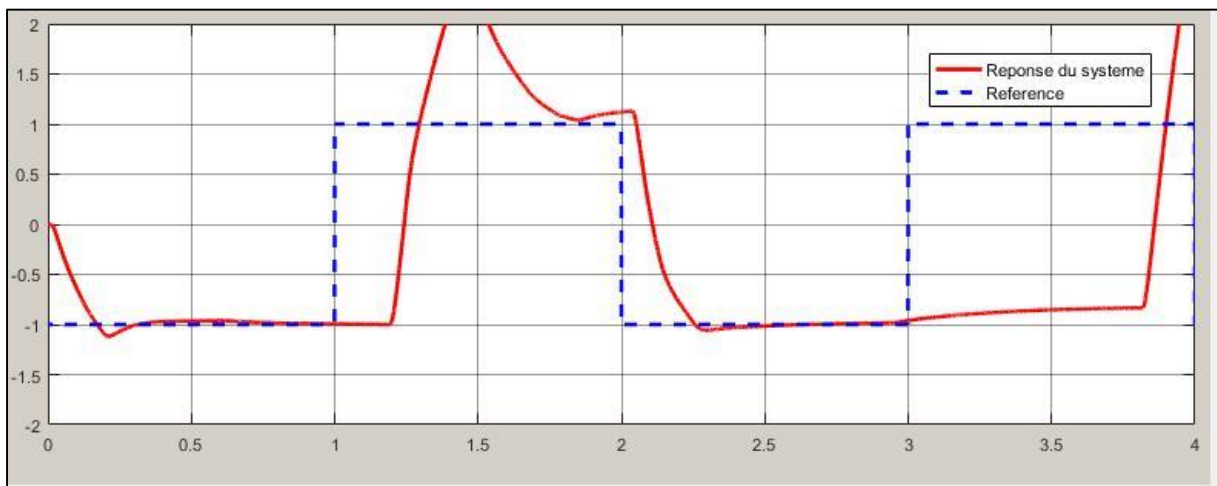


Fig.III.20. Réponse indicielle du système scenario 3

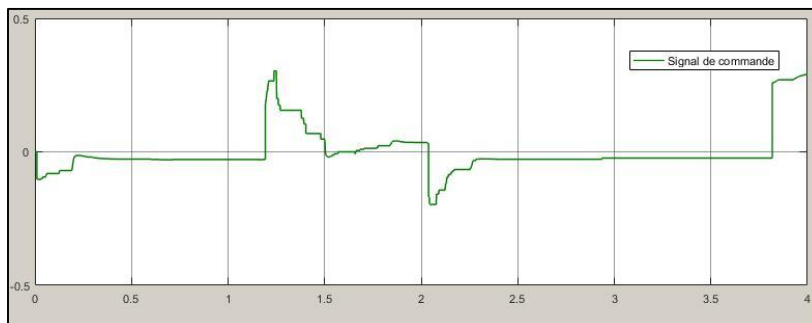


Fig.III.21. Signal de commande scenario 3

III.4.4. Scenario 4 : PACKAGE LOSS=30% BW SHARE =0%

Dans la figure III.22 Le système est stable avec un bon suivi de la consigne et un temps de réponse de 0.25s

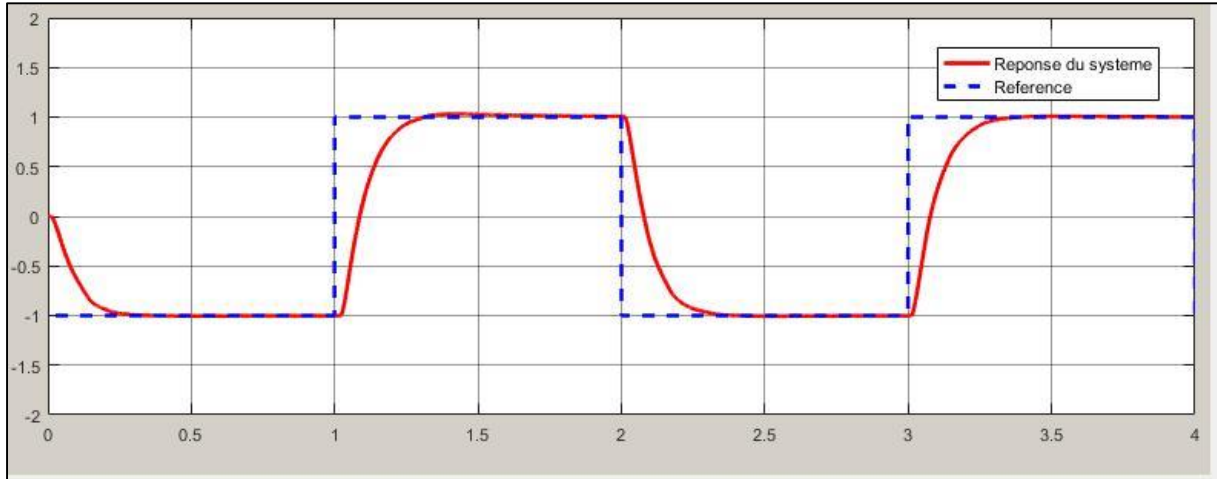


Fig.III.22. Réponse indicielle du système scenario 4

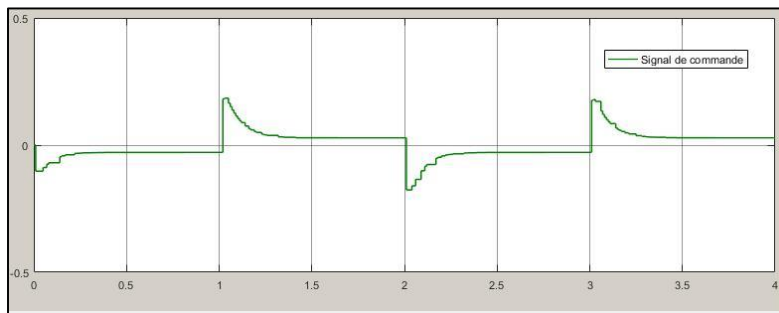


Fig.III.23. Signal de commande scenario 4

III.4.5. Scenario 5 : PACKAGE LOSS=40% BW SHARE=0%

Dans la figure III.24 on note l'apparition d'un dépassement à la seconde 2.15 ainsi que l'apparition d'un léger retard.

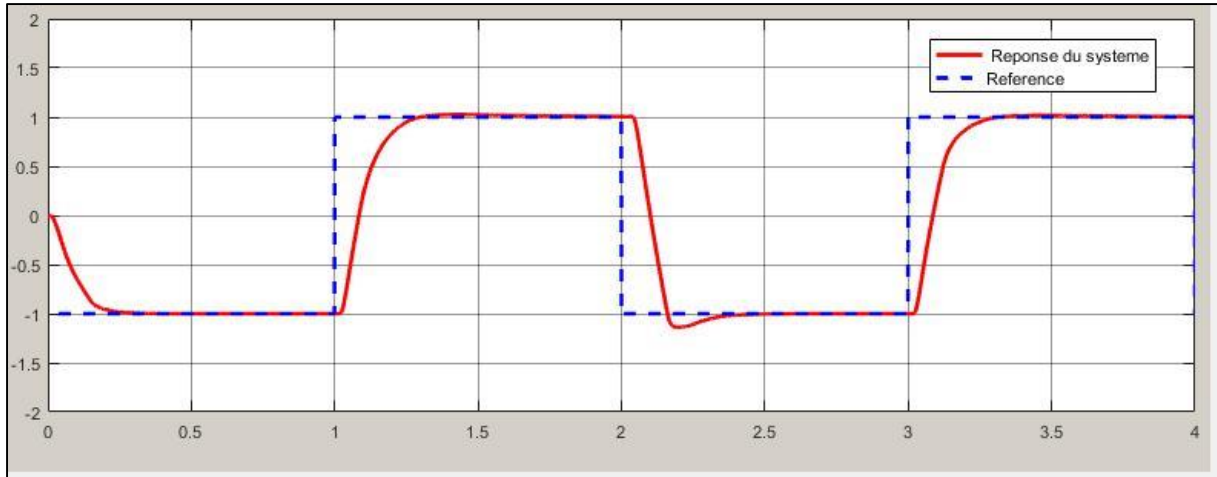


Fig.III.24. Réponse indicielle du système scenario 5

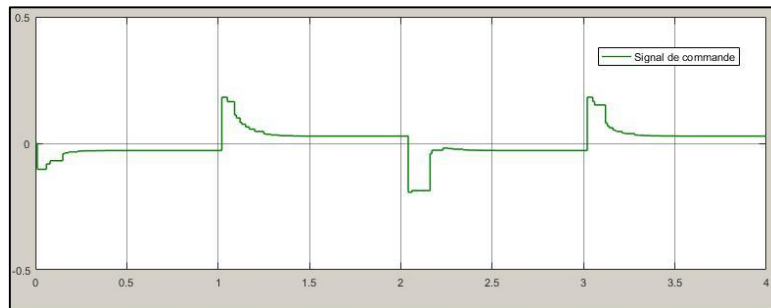


Fig.III.25. Signal de commande scenario 5

III.4.6. Scenario 6 : PACKAGE LOSS=50% BW SHARE =0%

Sur la figure III.26 on peut observer l'apparition de retards a 0.1s 1.1s et 2.1s ainsi que de nombreux dépassements même si le système tend à revenir vers la stabilité à partir de la 3eme seconde.

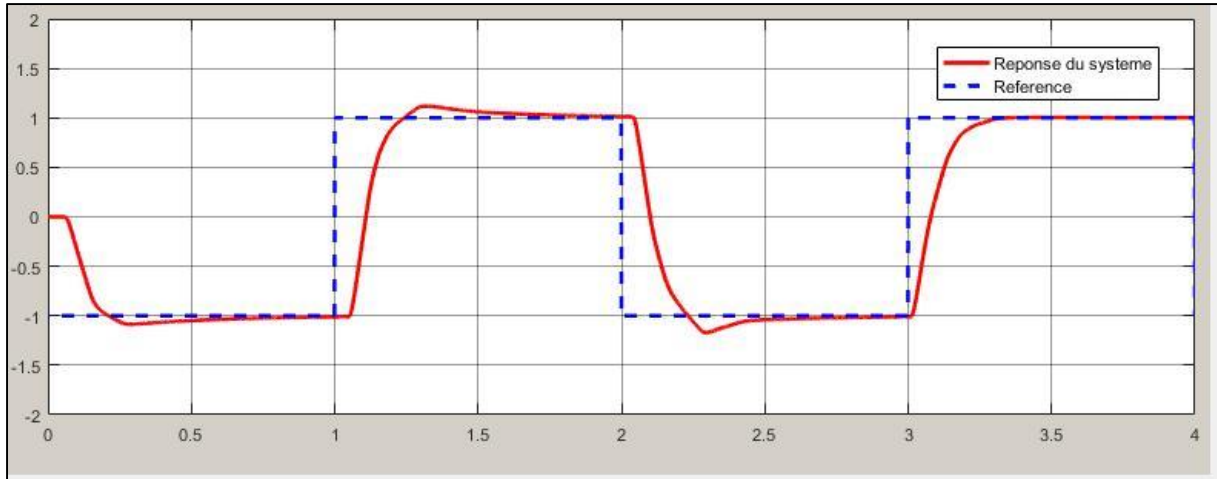


Fig.III.26. Réponse indicielle du système scenario 6

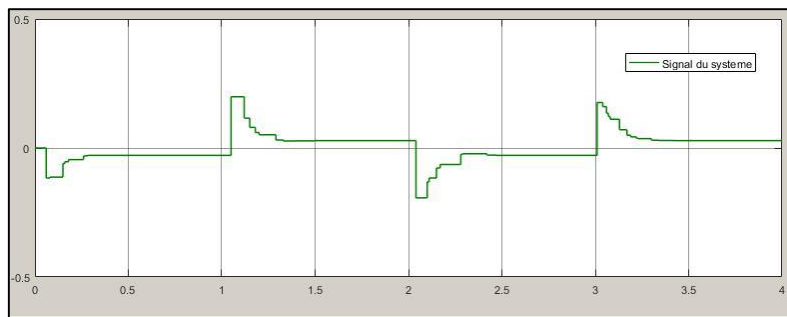


Fig.III.27. Signal de commande scenario 6

Scenario 7 : PACKAGE LOSS=60% BW SHARE=0%

Sur la figure III.28 on note l'apparition de plusieurs retards a 0s 2s et 3.1s les dépassements sont omni-présents.



Fig.III.28. Réponse indicielle du système scenario 7

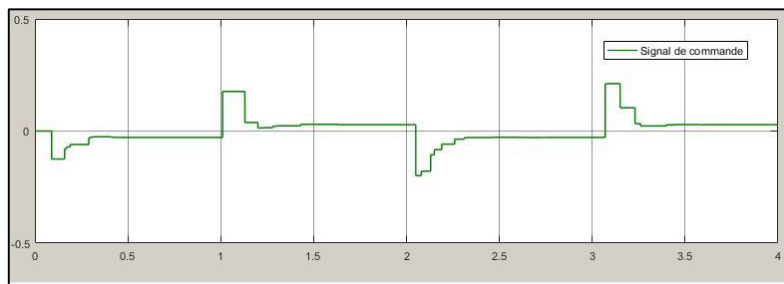


Fig.III.29. Signal de commande scenario 7

III.5. Conclusion

Dans ce chapitre nous avons simulé différents scénarios de perturbations sur un système éolien commandé en réseau.

Les résultats obtenus montrent que le système commandé en réseau stabilisé par la méthode de placement de pôle est performant et garde un bon suivi de la consigne jusqu'à un niveau de partage de bande passante de 30%. Il conserve également ces performances jusqu'à 40% de perte de paquet. On constate qu'au-delà de ce seuil on note l'apparition de dépassements et de retards à répétition, même si le système reste quasiment stable et tends à revenir vers ses performances initiales.

Chapitre IV. Application 2 :
Commande LQR de la puissance
D'un système éolien

IV. Application 2 : Commande LQR de la puissance d'un système éolien

IV.1.la commande LQR

Les solutions de contrôle optimales quadratique fournissent une procédure de conception automatisée - nous n'avons qu'à décider quel chiffre de mérite utiliser [10].

Le régulateur quadratique linéaire (LQR) est une technique de conception bien connue qui fournit des gains de rétroaction pratiques.

Pour éviter les coûts élevés et la complexité de la réalisation de régulateurs optimaux simples par le biais d'un système complet de rétroaction d'état, une méthode indirecte de réalisation de la même régulation est suggérée ici. Cette méthode simplifie celle développée par CALOVIC et CUK et est meilleure en raison du besoin d'un nombre plus restreint de sorties mesurables du système [10].

Considérons le système linéaire décrit comme suit :

$$\dot{\mathbf{X}} = \mathbf{A}_x + \mathbf{B}\mathbf{U} \text{ avec } \mathbf{x}(0) = \mathbf{x}_0 \dots\dots(1)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \dots\dots\dots(2)$$

Où \mathbf{x} est un vecteur d'état n du système, \mathbf{u} est un vecteur m du système en entrée, \mathbf{y} est un vecteur L du système en sortie, et \mathbf{A}, \mathbf{B} , et \mathbf{C} , sont des matrices à coefficient constant de dimensions appropriée.

L'entrée contrôlée \mathbf{u} a été supposée être la sortie d'un contrôleur à 3 termes avec $\mathbf{y}, \int \mathbf{y} dt, \dot{\mathbf{y}}$ Comme entrées.

Leur relation dans la formation de \mathbf{u} peut s'exprimer comme suit :

$$\mathbf{U} = -\mathbf{K}_p \mathbf{y} - \mathbf{K}_i \int_0^{\infty} \mathbf{y} dt - \mathbf{K}_d \dot{\mathbf{y}} \text{ pour } \mathbf{u}(0) = \mathbf{0} \dots\dots\dots(3)$$

où \mathbf{K}_p , \mathbf{K}_i et \mathbf{K}_d sont des matrices de gain de rétroaction proportionnelles, intégrales et dérivées de taille $M \times L$

d'après l'équation (1) et (2) on peut écrire l'équation (3) de la forme suivante :

$$\mathbf{U} = -\mathbf{K}_p \mathbf{C}\mathbf{X} - \mathbf{K}_i \int_0^t \mathbf{y} dt - \mathbf{K}_d \mathbf{C}(\mathbf{A}_x + \mathbf{B}\mathbf{u})$$

Ou

$$U = \bar{k}_p x - \bar{k}_i \int_0^t y dt \quad \dots\dots(4)$$

ou le nouveau gain \bar{k}_p et \bar{k}_i sont définies comme suit :

$$\bar{k}_p = (\mathbf{I}_m + \mathbf{K}_a \mathbf{C} \mathbf{B})^{-1} (\mathbf{K}_p \mathbf{C} + \mathbf{K}_d \mathbf{C} \mathbf{A}) \quad \dots\dots(5)$$

$$\bar{k}_i = (\mathbf{I}_m + \mathbf{K}_d \mathbf{C} \mathbf{B})^{-1} \mathbf{K}_i \quad \dots\dots(6)$$

avec \mathbf{I}_m une matrice d'unité.

En exprimant $\int y dt$ en termes d'un nouvel ensemble de variables z, on peut écrire :

$$\mathbf{Z} = \int y dt \text{ pour } \mathbf{Z}(0) = \mathbf{0} \quad \dots\dots\dots(7)$$

c'est pourquoi $\dot{\mathbf{Z}} = \mathbf{y} = \mathbf{C} \mathbf{x} \dots\dots\dots(8)$

Définissons les états augmentés du système comme suit :

$$\bar{\mathbf{x}} = [\mathbf{x} \mathbf{Z}]^t \dots\dots\dots(9)$$

Le système peut maintenant être décrit par l'équation suivante :

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{A}} \bar{\mathbf{x}} + \bar{\mathbf{B}} u \quad \dots\dots\dots(10)$$

dans laquelle la matrice A du système augmenté et la matrice de commande B peuvent être exprimées sous la forme

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}, \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

avec Q,R comme état augmenté et matrices de pondération d'entrée pour minimiser l'index de performance qui s'écrit de la forme suivante

$$J = \frac{1}{2} \int_0^{\infty} (\bar{\mathbf{x}}^t \bar{\mathbf{Q}} \bar{\mathbf{x}} + u^t \mathbf{R} u) dt \quad \dots\dots\dots(11)$$

Si les conditions pertinentes de contrôlabilité et d'observabilité soient remplies le témoin optimal souhaité u^* peut être évalué comme suit :

$$U^* = -R^{-1} \tilde{B}^T \bar{P} \bar{x} \quad \dots(12)$$

où la matrice P est la solution positive -définie de l'équation de Riccati de la matrice algébrique non linéaire de la forme suivante :

$$\bar{P} \bar{A} + \bar{A}^T \bar{P} - \bar{P} \bar{B} R^{-1} \bar{B}^T + \bar{Q} = 0$$

La comparaison des équations 4 et 12 donne :

$$\bar{k}_p = R^{-1} B^T P_{11} \quad \text{et} \quad \bar{k}_i = R^{-1} B^T P_{12} \quad \dots(13)$$

Où P₁₁, P₁₂ sont les éléments de la première ligne de la matrice \bar{P} exprimer comme suit :

$$\bar{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$$

Après évaluation de ces nouvelles matrices de gain Kp et Kd, les gains proportionnels, intégraux et dérivés Kp, Ki et Kd peuvent être évalués à partir des équations 5 et 6 qui peuvent être réécrits dans les formulaires qui suivent :

$$[K_p \ K_d] = \bar{k}_p \bar{C}^{-1} \quad \text{ou} \quad \bar{C} = \begin{bmatrix} c \\ CA - CB \bar{k}_p \end{bmatrix} \quad \dots\dots\dots(14)$$

$$\text{Avec } K_i = (I_m + K_d CB) \bar{k}_i \quad \dots\dots\dots(15)$$

VI.2. Détermination des paramètres du contrôleur PID-LQR

Considérons un système linéaire sous forme d'équation d'état suivant :

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ y(t) = Cx(t) \end{cases}$$

Où on va supposer que La paire (A, B) est stabilisable, c'est-à-dire qu'il n'y a pas de Mode instable et ingouvernable dans le système.

Soit un régulateur par retour d'états dont le processus a pour équation d'état l'équation (3-19). Le problème simplifié du régulateur linéaire quadratique consiste à trouver la Matrice du correcteur K qui minimise la fonction du coût (ou le critère de performance)

Suivante :

$$J(x_0, u) = \frac{1}{2} x^T(t_f) S x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^T(t) Q(t) x(t) + u^T(t) R(t) u(t)) dt \quad (3-20)$$

Les matrices de pondération Q et R sont définies positives et symétriques. Et S est la Matrice de solution de l'équation de Riccati (est définie positives et symétrique).

Le Lagrangien s'écrit alors :

$$L(x, u, p, t) = p^T A(t)x + p^T B(t)u + \frac{1}{2}(x^T Q(t)x + u^T R(t)u) \quad (3-21)$$

La loi de commande optimale est obtenue si le dérivé de lagrangien par rapport à la loi

De commande est nul :

$$\frac{\partial L}{\partial u} = B^T(t)p + R(t)u = 0 \quad (3-22)$$

Donc, on peut tirer l'optimale à partir d'équation (3.22)

$$u_{opt} = -R^{-1}(t)B^T p(t) \quad (3-23)$$

Ou

$$p(t_f) = Sx(t_f) \quad (3-24)$$

Le principe du maximum donne la condition suivante :

$$\dot{p} = -\frac{\partial L}{\partial x} = -A^T(t)p - Q(t)x \quad (3-25)$$

Alors l'équation dynamique du système en boucle fermée s'écrit :

$$\dot{x} = A^T(t)x(t) - B(t)R^{-1}(t)B^T(t)p(t) \quad (3-26)$$

Les équations (3-24) et (3.26) peuvent se mettre sous la forme d'un système matriciel

Appelé système Hamiltonien :

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \begin{pmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{pmatrix} \begin{pmatrix} x(t) \\ p(t) \end{pmatrix} \quad (3-27)$$

Ecrivons $p(t) = P(t)x(t)$, comme nous y incite (4-8), avec comme condition finale

$P(t_f) = S$. L'équation (3-26) s'écrit alors :

$$\dot{p}(t) = -(A^T(t)P(t) + Q(t))x(t) \quad (3-28)$$

L'équation d'état du système peut s'écrire de la manière suivante :

$$(\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q)x = 0 \quad (2-29)$$

La solution est alors obtenue en résolvant l'équation (différentielle) de Riccati suivante :

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (3-30)$$

Avec la condition finale $P(t_f) = S$. Remarquons que la condition :

$$x^T(P + PA + A^T \dot{P} - PBR^{-1}P + Q)x = 0 \quad (3-31)$$

S'écrit aussi :

$$\frac{d}{dt}(x^T P x) + x^T Q x + u^T R u \quad (3-32)$$

Il est intéressant de noter que la commande optimale obtenue s'écrit comme un retour

D'état $u = -k(t)x$ avec :

$$K = -R^{-1}B^T P \quad (3-33)$$

• Résolution du problème de poursuite

On considère les systèmes linéaires décrits par

$$\begin{cases} \dot{\underline{x}}(t) = \mathbf{A}(t)\underline{x}(t) + \mathbf{B}(t)\underline{u}(t) \\ \underline{y}(t) = \mathbf{C}(t)\underline{x}(t) \end{cases}$$

En posant $e(t) = z(t) - y(t)$, on introduit le critère :

$$J(\underline{u}(t)) = \frac{1}{2} \underline{e}^T(t_f) \mathbf{F}(t_f) \underline{e}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \underline{e}^T(t_f) \mathbf{Q}(t) \underline{e}(t_f) + \underline{u}^T(t) \mathbf{R}(t) \underline{u}(t) dt$$

• Construction de la solution

1. L'Hamiltonien s'écrit :

$$\mathcal{H} = \frac{1}{2}[\underline{z}(t) - \mathbf{C}(t)\underline{x}(t)]^T \mathbf{Q}(t)[\underline{z}(t) - \mathbf{C}(t)\underline{x}(t)] \\ + \underline{u}^T(t)\mathbf{R}(t)\underline{u}(t) + \underline{\lambda}^T(\mathbf{A}(t)\underline{x}(t) + \mathbf{B}(t)\underline{u}(t))$$

2.Calcul de $\partial\mathcal{H}/\partial\mathbf{u}$:

$$\frac{\partial\mathcal{H}}{\partial\underline{u}} = \underline{0} \Rightarrow \underline{u}^*(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{\lambda}^*(t)$$

3.

$$\dot{\underline{x}}(t) = \frac{\partial\mathcal{H}}{\partial\underline{\lambda}} = \mathbf{A}(t)\underline{x}(t) + \mathbf{B}(t)\underline{u}(t)$$

$$\Rightarrow \dot{\underline{x}}^*(t) = \mathbf{A}(t)\underline{x}^*(t) - \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{\lambda}^*(t)$$

$$\dot{\underline{\lambda}}^*(t) = -\frac{\partial\mathcal{H}}{\partial\underline{x}} = -\mathbf{C}^T(t)\mathbf{Q}(t)\mathbf{C}(t)\underline{x}^*(t) - \mathbf{A}^T(t)\underline{\lambda}^*(t) + \mathbf{C}^T(t)\mathbf{Q}(t)\underline{z}(t)$$

On a déduit le système hamiltonien :

$$\begin{bmatrix} \dot{\underline{x}}^*(t) \\ \dot{\underline{\lambda}}^*(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(t) & -\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t) \\ -\mathbf{C}^T(t)\mathbf{Q}(t)\mathbf{C}(t) & -\mathbf{A}^T(t) \end{bmatrix} \begin{bmatrix} \underline{x}^*(t) \\ \underline{\lambda}^*(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^T(t)\mathbf{Q}(t) \end{bmatrix} \underline{z}(t)$$

Les conditions au limites sont donnée par

$$\underline{x}(t_0) = \underline{x}_0$$

$$\underline{\lambda}(t_f) = \frac{1}{2} \frac{\partial}{\partial \underline{x}(t_f)} [\underline{e}^T(t_f)\mathbf{F}(t_f)\underline{e}(t_f)] \\ = \mathbf{C}^T(t_f)\mathbf{F}(t_f)\mathbf{C}(t_f) - \mathbf{C}^T(t_f)\mathbf{F}(t_f)\underline{z}(t_f)$$

on introduit $\mathbf{P}(t)$ et $\mathbf{g}(t)$ tels que $\underline{\lambda}^*(t) = \mathbf{P}(t)\underline{x}^*(t) - \mathbf{g}(t)$ en reportant ce changement de variable dans le système hamiltonien ,on arrive a :

$$\dot{\underline{\lambda}}^*(t) = \dot{\mathbf{P}}(t)\underline{x}^*(t) + \mathbf{P}(t)\dot{\underline{x}}^*(t) - \underline{\dot{g}}(t)$$

$$\begin{aligned}\dot{\mathbf{P}}(t) &= -\mathbf{P}(t)\mathbf{A}(t) - \mathbf{A}^T(t)\mathbf{P}(t) \\ &+ \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{P}(t) - \mathbf{C}^T(t)\mathbf{Q}(t)\mathbf{C}(t) \\ \underline{\dot{g}}(t) &= [\mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t) - \mathbf{A}^T(t)]\underline{g}(t) - \mathbf{W}(t)\underline{z}(t)\end{aligned}$$

Les conditions aux limites deviennent :

$$\begin{aligned}\mathbf{P}(t_f) &= \mathbf{C}^T(t_f)\mathbf{F}(t_f)\mathbf{C}(t_f) \\ \underline{g}(t_f) &= \mathbf{C}^T(t_f)\mathbf{F}(t_f)\underline{z}(t_f)\end{aligned}$$

5.le contrôle optimal est donné par :

$$\begin{aligned}\underline{u}^*(t) &= -\mathbf{K}(t)\underline{x}^*(t) + \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{g}(t) \\ &= -\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{P}(t)\underline{x}^*(t) + \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{g}(t)\end{aligned}$$

6.la trajectoire optimale est donnée par :

$$\dot{\underline{x}}^*(t) = [\mathbf{A}(t) - \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{P}(t)]\underline{x}^*(t) + \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{g}(t)$$

7.la valeur du critère a l'optimum peut être obtenue par :

$$J^*(t) = \frac{1}{2}\underline{x}^{*T}(t)\mathbf{P}(t)\underline{x}^*(t) - \underline{x}^{*T}(t)\underline{g}(t) + h(t)$$

Ou h(t) satisfait :

$$\begin{aligned}\dot{h}(t) &= -\frac{1}{2}\underline{g}^T(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\underline{g}(t) - \frac{1}{2}\underline{z}^T(t)\mathbf{Q}(t)\underline{z}(t) \\ \text{et } h(t_f) &= -\underline{z}^T(t_f)\mathbf{Q}(t_f)\underline{z}(t_f)\end{aligned}$$

VI.3. Application pour la commande en réseau du système éolien

Dans cette partie nous allons reprendre le même schéma Simulink précédent, fig.3.9 en apportant une légère modification afin d'intégrer le régulateur PID-LQR au système de l'éolienne.

Pour ce faire nous remplacerons le bloc du contrôleur PI de ma méthode de placement de pole par un bloc PID intégrant les paramètres K_p , K_i , et K_d de la méthode de la régulation quadratique linéaire (LQR)

La figure VI.1 représente le schéma de la nouvelle simulation obtenu.

Le système reste le même, seul la boucle de commande en réseau du nœud 3 change de la manière décrite dans la figure VI.2.

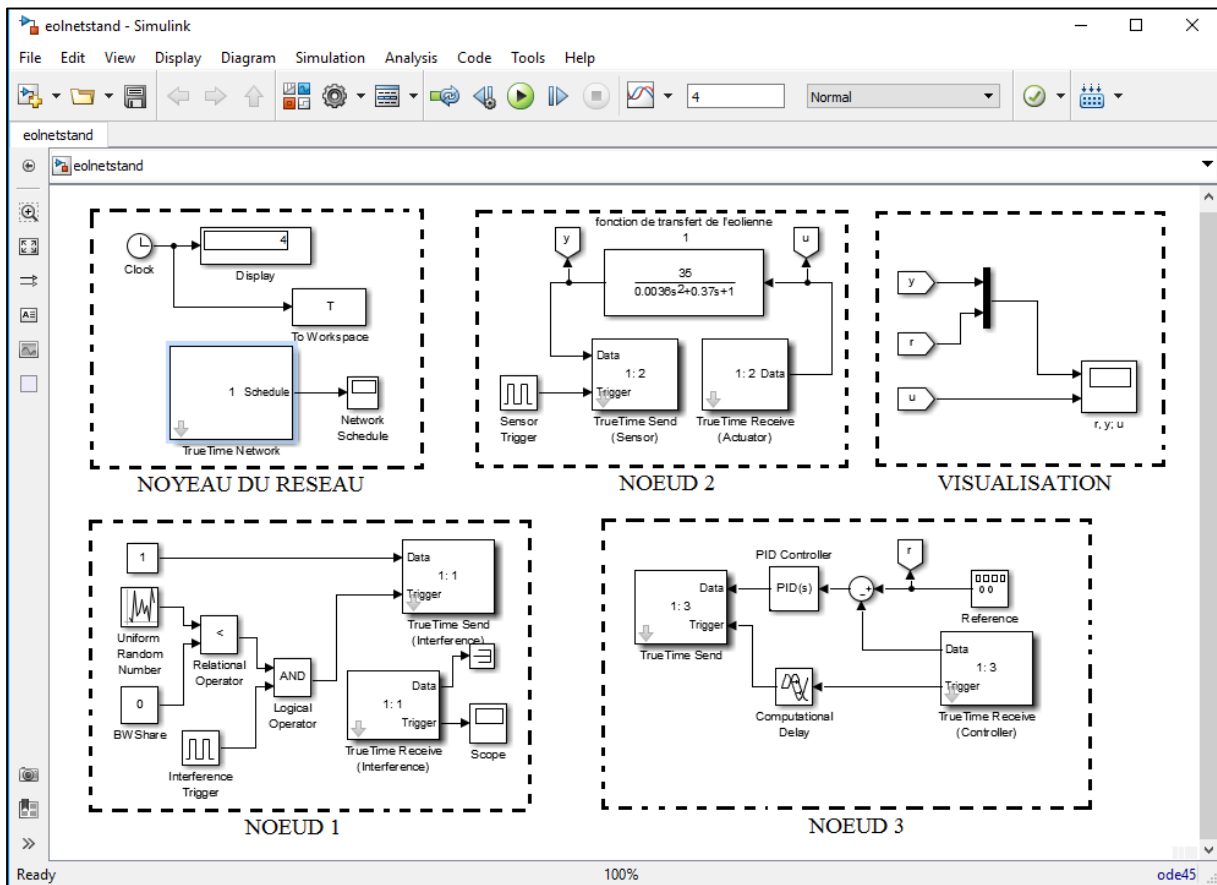


Fig.VI.1. Configuration du système éolien sous Simulink /TRUETIME pour la méthode LQR

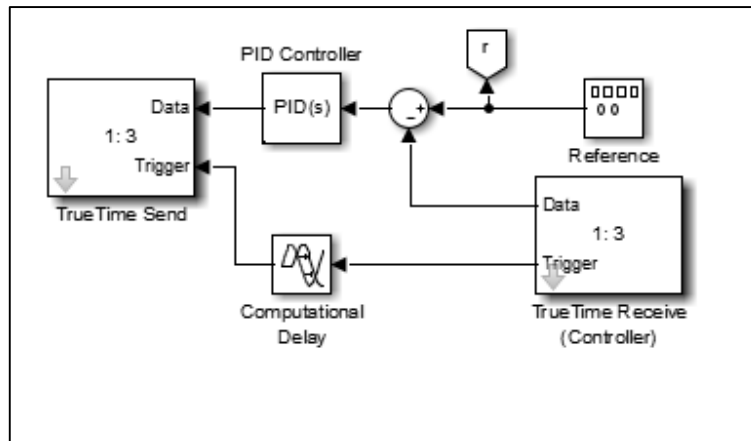


Fig.VI.2. NCEUD 3 commande par régulateur PID-LQR

VI.3. Test de différents scénarios de perturbations sur le réseau (pertes de paquets, partage de la bande passante)

Dans cette partie nous allons faire subir au système les mêmes perturbations que lors de la simulation du chapitre 3 avec les mêmes valeurs utiliser pour la simulation précédente

Voir chapitre III partie 4.

VI.3.1. Scenario 1 : PACKAGE LOSS=0% BW SHARE=0% (condition

Idéale)

Dans la figure VI.3 on peut voir un bon suivi de la consigne avec un temps de réponse de 0.5s

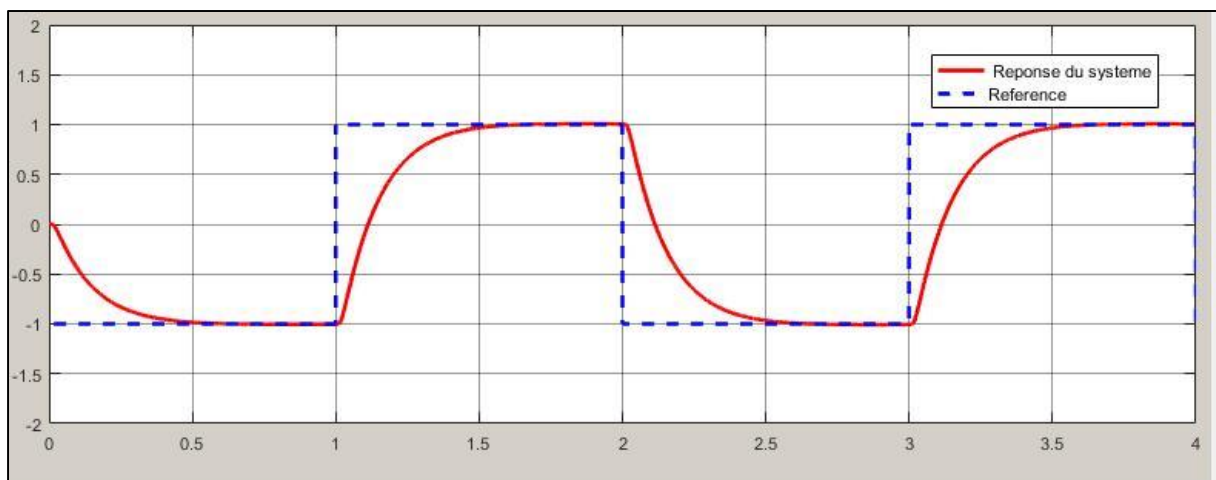


Fig.VI.3. Réponse indicielle du système scenario 1

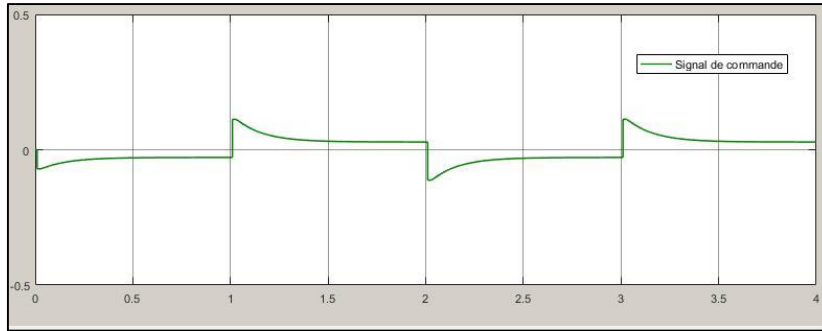


Fig.VI.4. Signal de commande scenario 1

VI.3.2. Scenario 2 : PACKAGE LOSS=0% BW SHARE=30%

Dans la figure VI.5. le système est stable et a un bon suivi e la consigne mais nous notons l'apparition d'un léger dépassement a 3.01s a peine perceptible .

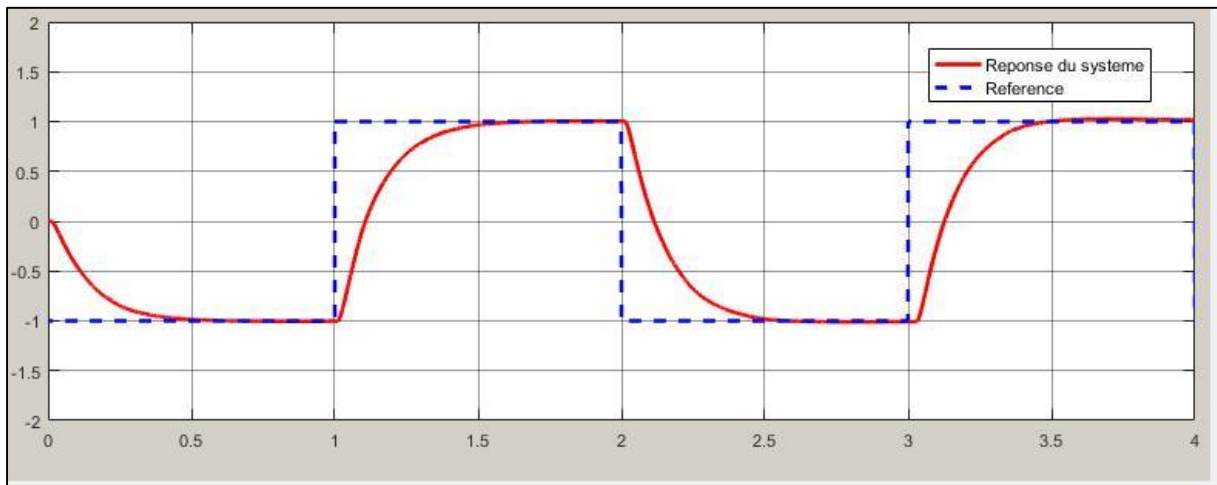


Fig.VI.5. Réponse indicielle du système scenario 2

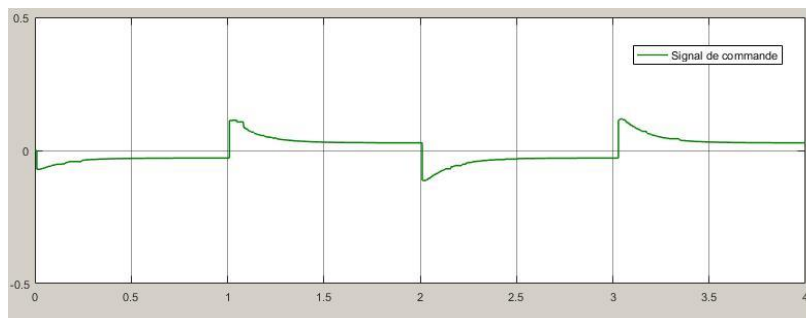


Fig.VI.6. Signal de commande scenario 2

VI.3.3. Scenario 3 : PACKAGE LOSS=0% BW SHARE=40%

La figure VI.7 montre une détérioration totale des performances.

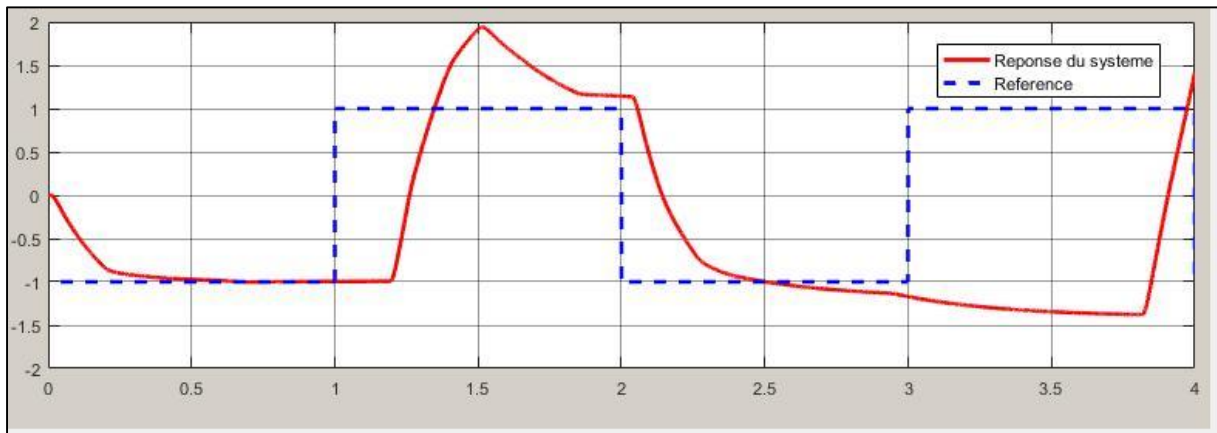


Fig.VI.7. Réponse indicielle du système scenario 3



Fig.VI.8. Signal de commande scenario 3

VI.3.4. Scenario 4 : PACKAGE LOSS=30% BW SHARE=0%

Sous ces conditions le système est stable, la réponse a un bon suivi de la consigne et un temps de réponse de 0.5s

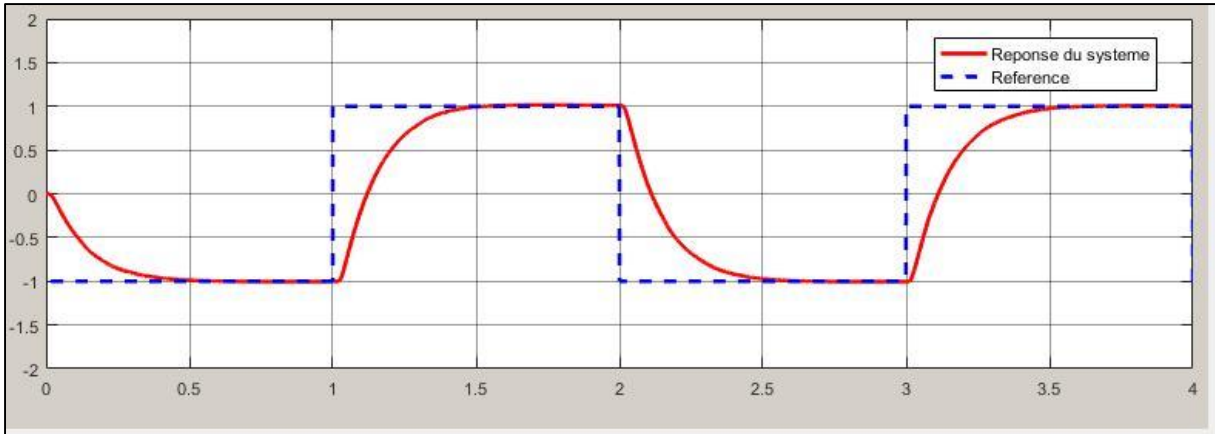


Fig.VI.9. Réponse indicielle du système scenario 4

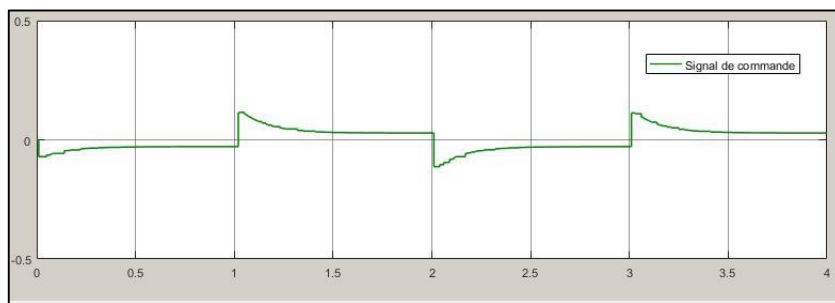


Fig.VI.10. Signal de commande scenario 4

VI.3.5. Scenario 5 : PACKAGE LOSS=40% BW SHARE=0%

Bon suivi de la consigne et temps de réponse a 0.5s pendant apparition d'un retard léger a 2.1s

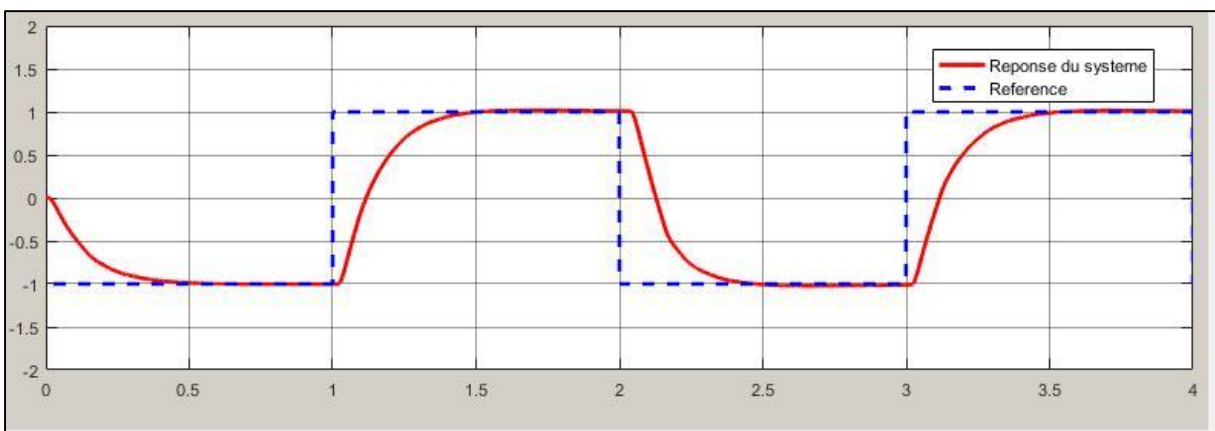


Fig.VI.11. Réponse indicielle du système scenario 5

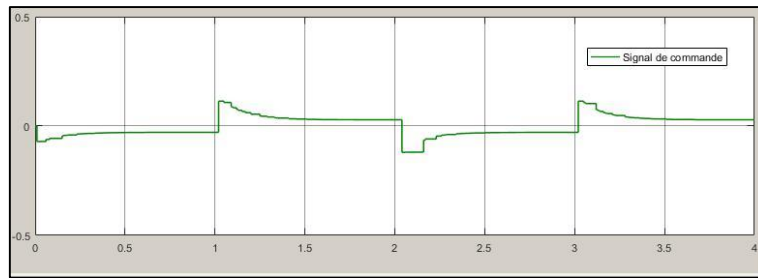


Fig.VI.12. Signal de commande scenario 5

VI.3.6. Scenario 6 : PACKAGE LOSS=50% BW SHARE=0%

Apparition de retards a 1.1s et 2.1s avec plusieurs zones de dépassement léger.

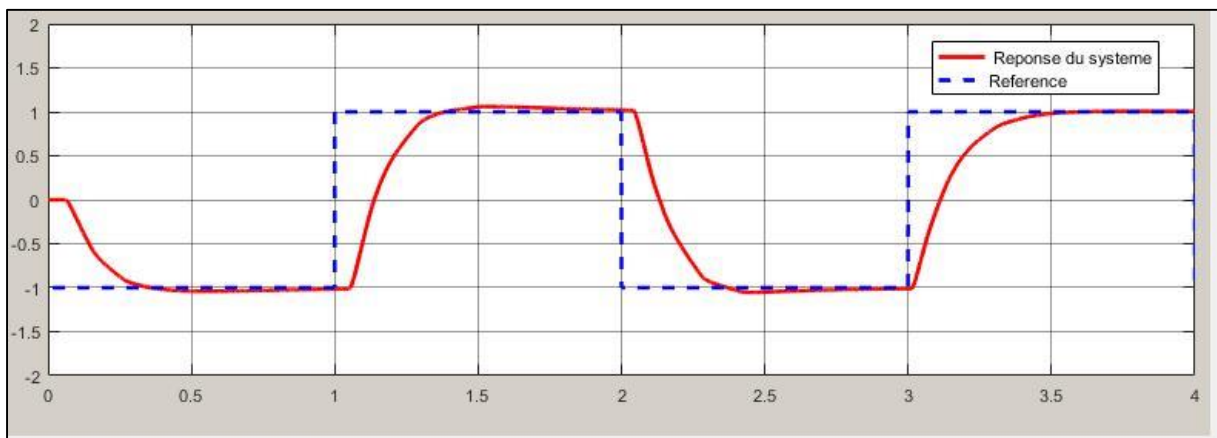


Fig.VI.13. Réponse indicielle du système scenario 6

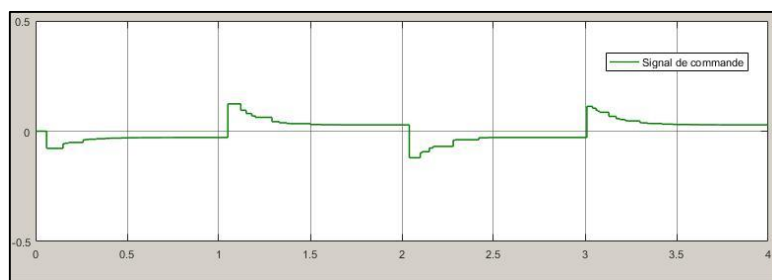


Fig.VI.14. Signal de commande scenario 6

VI.3.7. Scenario 7 : PACKAGE LOSS=60% BW SHARE=0%

Retard de plus en plus important à partir de 2.1s avec des dépassements a 0.5s,2.5 et 3.5 et un retard à 3.2s

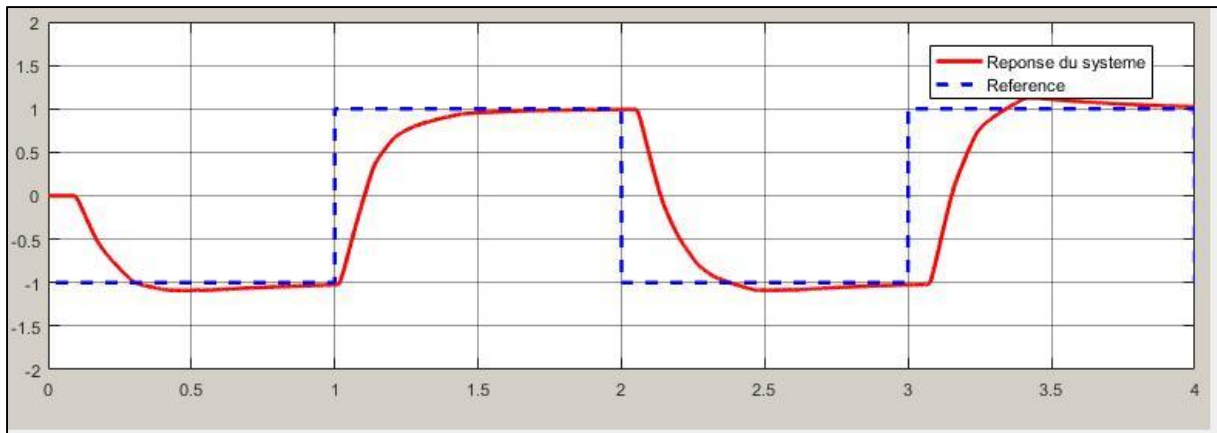


Fig.VI.15. Réponse indicielle du système scenario 7

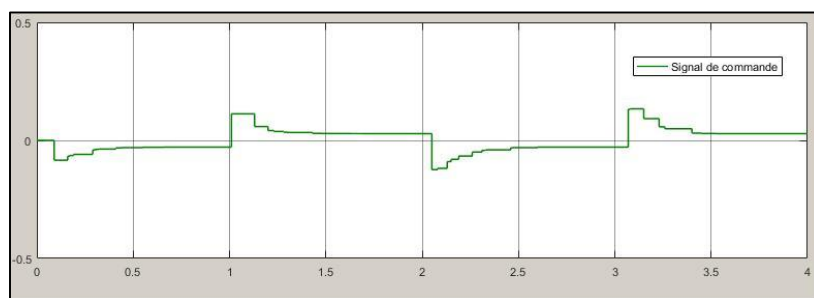


Fig.VI.16. Signal de commande scenario 7

VI.4. Conclusion

Dans ce chapitre nous avons simulé différents scénarios de perturbations sur un système éolien commandé en réseau par la méthode quadratique optimal (LQR)

Les résultats obtenus montrent que le système commandé en réseau stabilisé par la méthode du PID-LQR est performant et garde un bon suivi de la consigne jusqu'à un niveau de partage de bande passante de 30%. Il conserve également ces performances jusqu'à 40% de perte de paquet. On constate qu'au-delà de ce seuil on note l'apparition de dépassements et de retards à répétition, pas très décelable au début puis avec de plus en plus d'ampleur.

VI.5. Comparaison des deux méthodes (placement de pole et PID-LQR)

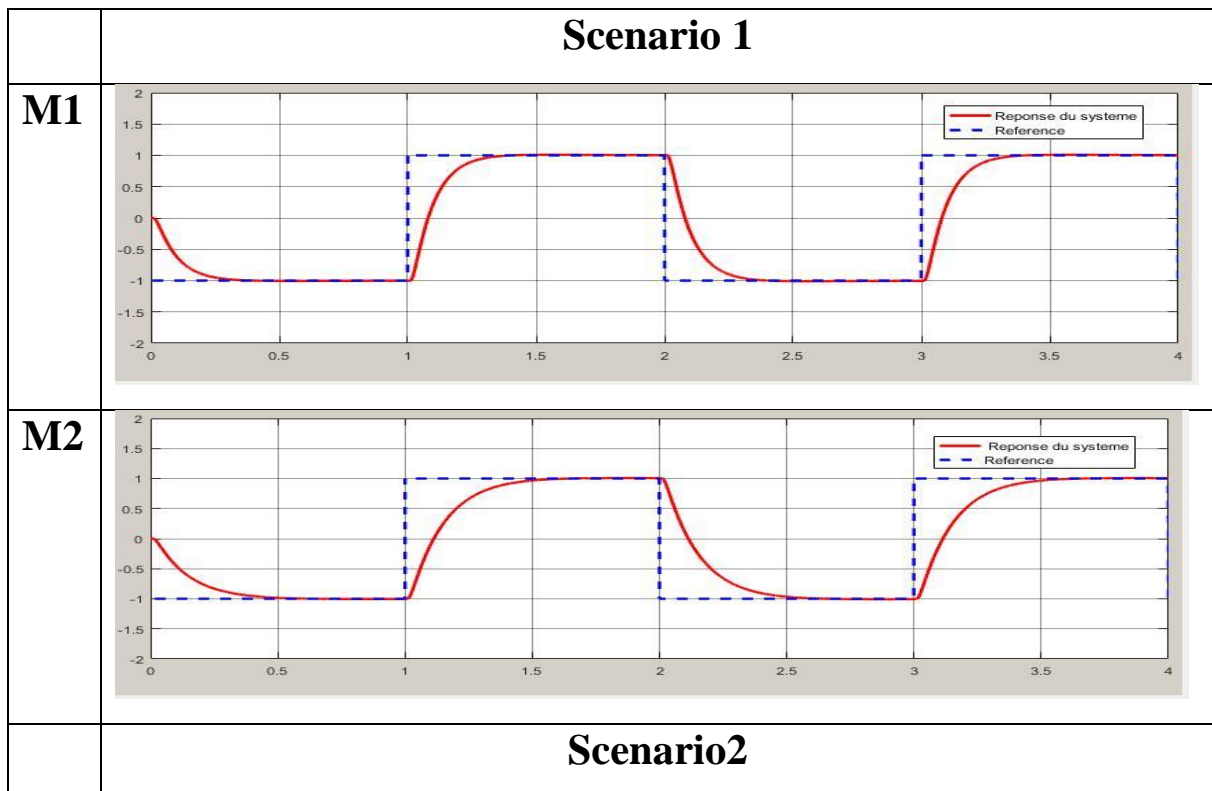
Le tableau VI.1 est un tableau de synthèse qui a pour but de faire la comparaison entre toutes les réponses indicielles du system obtenu durant les différents scénarios sur les deux méthodes

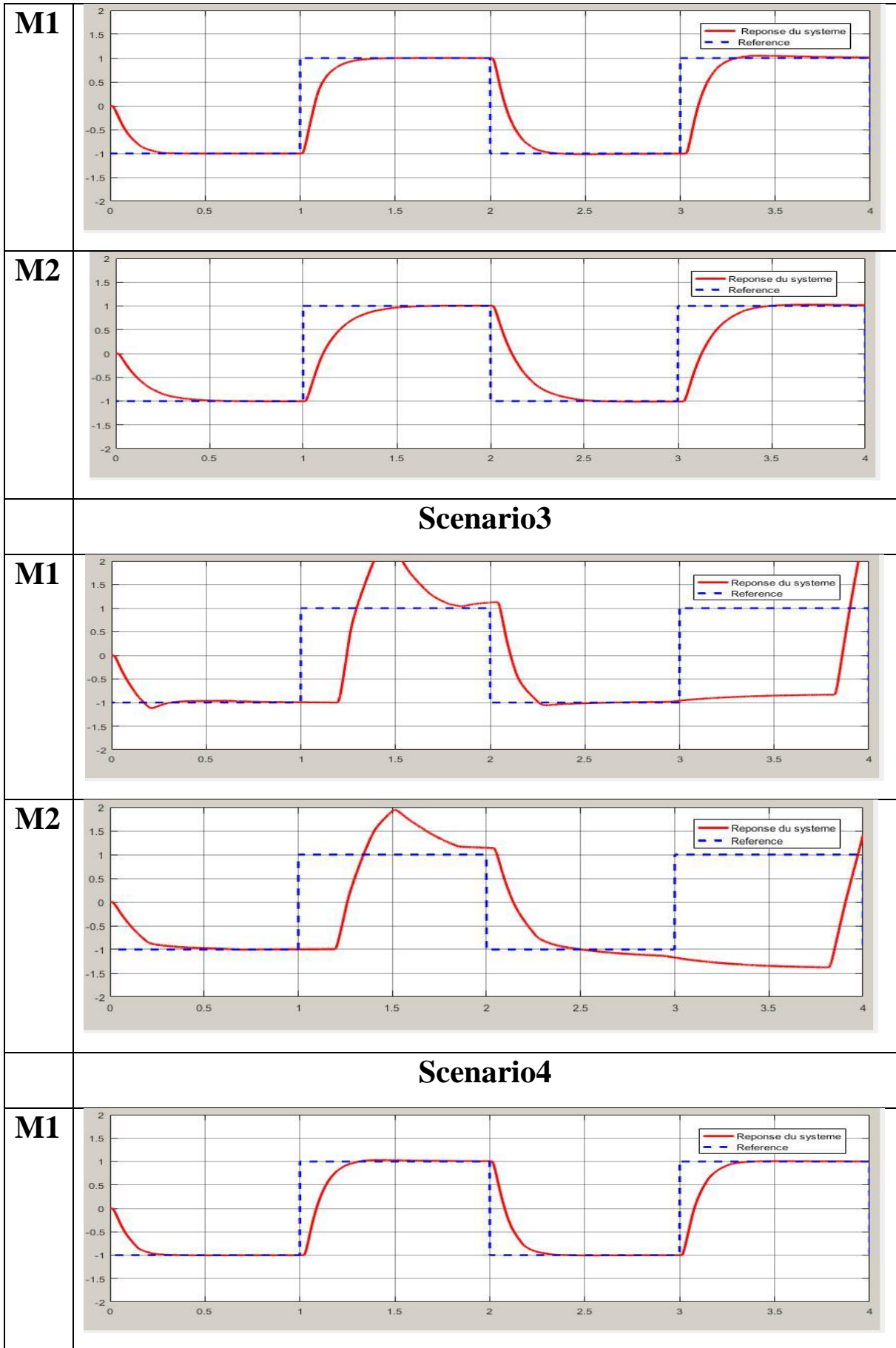
M1 : Méthode de placement de pôles

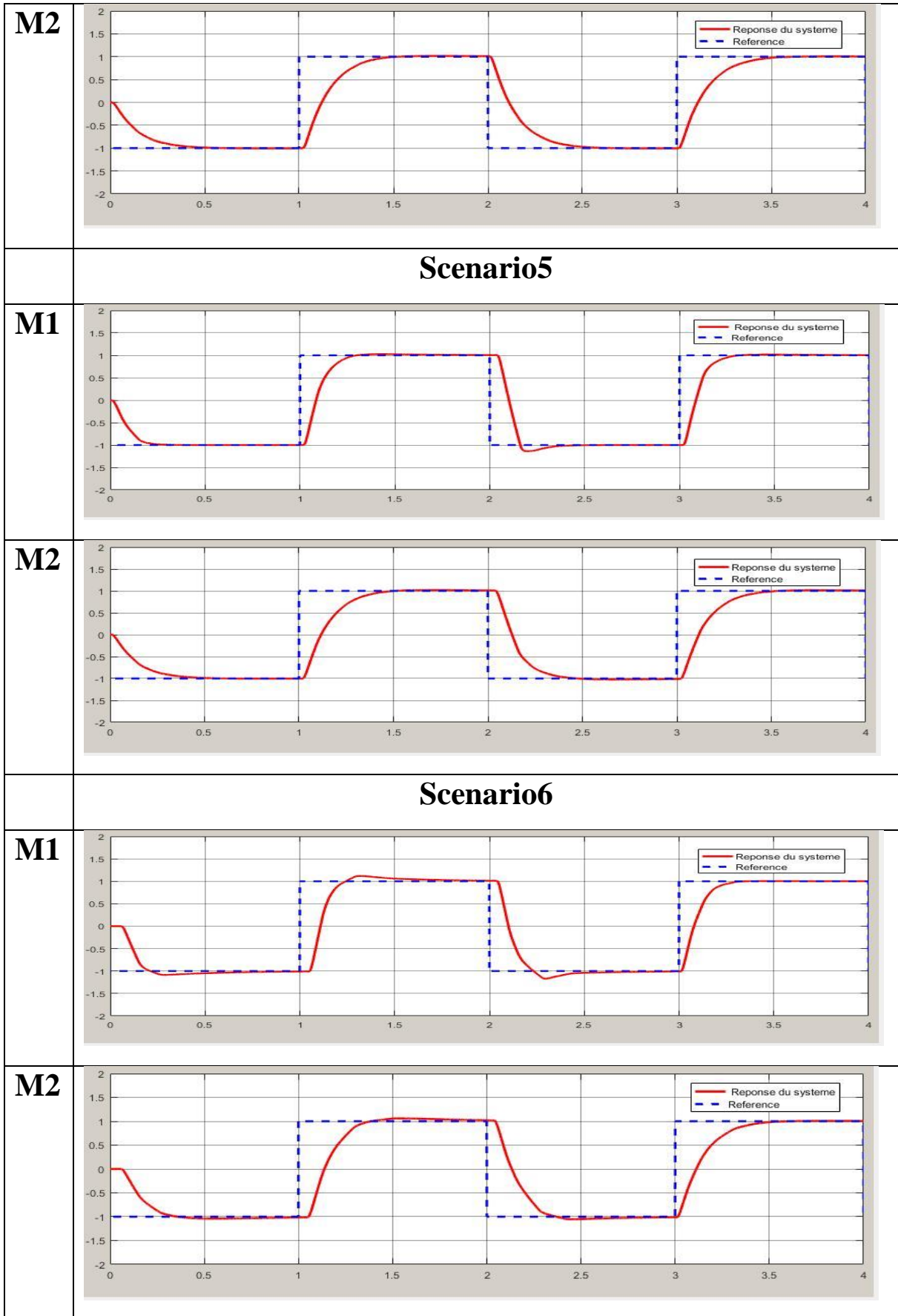
M2 : Méthode du PID-LQR

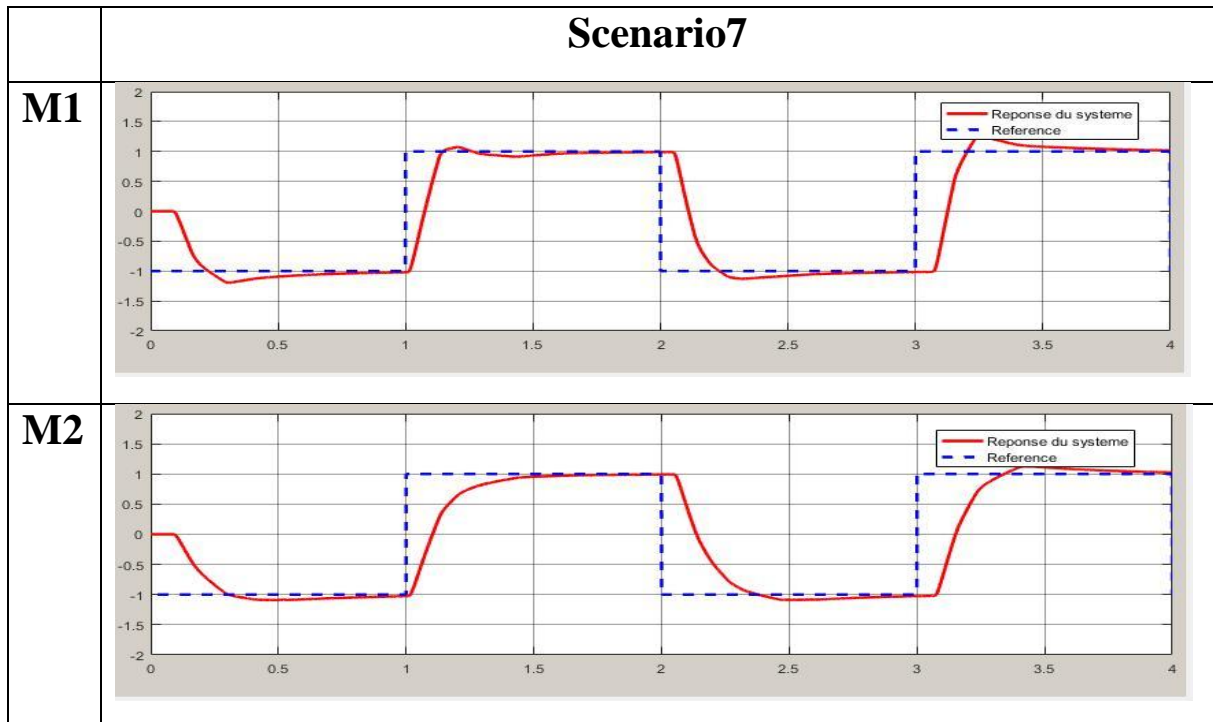
On note d'après le tableau que la méthode de placement de pôles a une réponse du system plus rapide comparée à la méthode de la commande LQR, et que la méthode LQR quand elle résiste un peu plus à la perturbation due à la perte de données.

Le tableau démontre aussi que lors du scénario 3 les fluctuations de la réponse du système défaillant sont plus importantes dans le cas de la méthode de placements des pôles.









Tab.VI.1. Comparaison des méthodes de placement de pôles et PID-LQR

***CONCLUSION
GENERALE***

Conclusion générale

L'objectif de ce travail consiste à mettre en évidence, via une simulation par l'exploitation d'un logiciel "TRUETIME " le comportement d'un système commandé en réseau représenté par le système éolien en adoptant deux méthodes en vue d'optimiser le choix de la commande en réseau.

La première partie introduit les notions générales relatives aux systèmes commandés en réseaux tels les différentes structures des SCR, leurs problématiques (retards induits, pertes de paquets, perturbations, instabilité ; etc.), les divers réseaux de communication existants ainsi l'influence de ces réseaux sur la qualité du contrôle de ces réseaux ;

Dans la deuxième partie, nous avons présenté le logiciel « TRUETIME » qui a été utilisé pour la simulation du système éolien, de manière détaillée ou tous les composants constitutifs de ce logiciel ont été décrits ;

Dans la troisième partie, nous avons modélisé un système éolien moyennant le simulateur TRUETIME sous simulink avec introduction de paramètres P.I de la méthode de placement des pôles, à la commande en réseau du système éolien en vue de visualiser son comportement sous divers scénarii. Les résultats montrent que la commande selon la méthode placement de pôles est rapide et performante..

Dans la quatrième partie, nous avons repris la même simulation en remplaçant les paramètres P.I par les P.I.D de la commande LQR afin de procéder à un comparatif du comportement du système sous les mêmes conditions avec un régulateur différent. Les résultats obtenus montrent que la commande LQR est plus lente mais plus résistante .

Cette modélisation réalisée pour optimiser les choix de la commande en réseaux, nous a permis de conclure ce qui suit :

Les deux méthodes de contrôles à savoir la méthode de placement de pôles et la méthode de la commande LQR ont chacune des avantages et des inconvénients .

Dans les conditions idéales de la commande par placement de pôles : elle est plus rapide et plus performante.

La commande LQR quant à elle, elle est plus lente dans les conditions idéales de partage de bande passante et de pertes de paquets. Mais résiste mieux aux perturbations dues à la perte de donnée et au partage de la bande passante. Elle conserve une réponse indicielle du système plus proche de la référence que la méthode par placement de pôles dans les mêmes conditions.

Par conséquent, on choisira donc une commande en réseau adaptée aux types de réseaux selon ses données ; la commande LQR pour les réseaux à commander de longues distances avec des retards et des pertes importantes et la commande par placement de pôles dans les réseaux commandés à plus courte distance et avec un réseau plus chargé .

Références Bibliographiques

- [1] EMMANUEL WITRANT: Stabilisation des systèmes commandés par réseaux; (2005)
- [2] LIQIAN ZHANG (1) YANG SHI (2) TONGWEN CHEN (3) BIAO HUANG (4): A New Method for Stabilization of Networked Control Systems with Random Delays ; (2003)
- [3] RACHNA DHAND, GARETH LEE, GRAEME COLE : Communication Delay Modelling and its Impact on Real-Time Distributed Control Systems-School of Engineering and Energy Murdoch University Murdoch, WA, Australia): Communication Delay Modelling and its Impact on Real-Time Distributed Control Systems
- [4] YODYIUM TIPSUWAN, MO-YUEN CHOW* (2003) : Control methodologies in networked control systems. Advanced Diagnosis and Control Lab, Department of Electrical and Computer Engineering, North Carolina State University, Daniels Hall, Raleigh, NC 27695-7911, USA .
- [5] NGUYEN XUAN HUNG. "Réseaux de Communication et Applications de Contrôle-Commande", Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse (INSA Toulouse), décembre 2000.
- [6] IMEN DILANEH, " étude de la commande et de l'observation à distance de systèmes", thèse Doctorat, université de Cergy Pontoise. Novembre 2009 -résumé
- [7] DANIELS HALL, RALEIGH Engineering, North Carolina State University, , NC 27695-7911, USA
- [8] MARTIN OHLIN, DAN HENRIKSSON ET ANTON CERVIN. "TrueTime 1.5 – Référence Manual". Lund Institute of Technology, Sweden, January 2007.
- [9] BENOÎT BERGEON. Prof.2013-2014.Commande linéaire des systèmes multivariables.

[10] SUBRATA MUKHOPADHYAY P.I.D. EQUIVALENT OF OPTIMAL REGULATOR- Article in electronics letters february 1978

[11] GÉRARD BLANCHET (29 Novembre 2007) : Optimisation Quadratique en Automatique ;

[12] RONY GHOSTINE (2008) : Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau.. Doctorat de l'Institut National Polytechnique de Lorraine - Spécialité Automatique, Traitement du signal et Génie Informatique.

[13] RACHDI N.D, MECHRAOUI Amine, THIRIET jean mar (2012) : Pronostic de défaillances d'un robot mobile commandé au travers d'un réseau sans fil. Grenoble.

[14] M. BELATEL et A. OUAZETA Modélisation et identification par les réseaux de neurones d'un système énergétique éolien à base d'une génératrice asynchrone-, LEC Faculté des Sciences et de la Technologie, Université Constantine 1(2017)

[15] Dr. BOULOUIHA H. MERABET : Tehniques de commandes avancées. Université de Relizane – 2014-2015.