République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Béjaïa
Faculté de Technologie
Département de Génie Electrique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme de Master en Automatique et informatique industrielle

Thème

Modélisation et commande d'un simulateur d'hélicoptère (TRMS)

Réalisé par

M. BOUZENBOUA Sid Ali M. ALLAOUA Fawzi

Devant le jury composé de

Examinateur : Mr H.HADDAR
Examinateur : Mr H.LEHOUCHE

Encadré par : Mr O.GUENOUNOU Mr M.KACIMI

Promotion 2020 - 2021

Remerciements

En premier lieu, nous remercions le bon Dieu tout puissant de nous avoir donné la force et la volonté durant nos études et pendant la réalisation de ce projet.

Nous voudrions aussi remercier nos encadreurs M. GUENOUNOU et M. KACIMI pour leur aide et leur précieux conseils.

Comme nous tenons à remercier les membres du jury d'avoir accepté d'examiner et de juger notre modeste travail.

Un remerciement particulier à nos familles pour nous avoir soutenus, accompagnées et nous avoir permis d'en arriver là.

Enfin, nous remercions tous nos amis pour leur aide et leurs encouragements ainsi que tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

Dédicaces

A mes parents et mon frére qui m'ont soutenu, encouragé et surtout supporté durant mes études.

A mes amis qui ont toujours su être présents.

BOIUZENBOUA Sid-Ali

A mes parents et mon frère et mes sœurs qui m'ont toujours soutenue et encourage durant mes études et mes amis qui ont toujours été présent (k.L).

A mes amis qui ont toujours été présents.

ALLAOUA Fawzi

Table des matières

Table des figures

Liste des abréviations

DC	Direct courant
MIMO	Multi input Multi output
PID	Proportionnelle intégrale et dérivé
PSO	Particles swarm optimization
TRMS	Twin rotor MIMO system

Introduction générale

Le monde d'aujourd'hui a connu plusieurs développements dans différents domaines, particulièrement dans l'aéronautique. L'automatique est une solution qui à permis de résoudre de nombreuses difficultés rencontrées lors de la commande des systèmes développés. Les systèmes développés sont de plus en plus complexes et difficiles à commander, pour cela il existe plusieurs lois de commande qui ont connu aussi de nombreux travaux de recherches. L'application de ses commandes nécessite un modèle mathématique proche du modèle réel.

Quelle que soit la méthode utilisée pour obtenir un modèle mathématique d'un système physique, il existe toujours un compromis entre la simplicité du modèle et son aptitude à décrire l'ensemble des phénomènes qui le caractérise. Ces écarts entre le modèle et le système réel sont souvent modélisés par des grandeurs physiques.

Un hélicoptère de type TRMS (Twin Rotor Mimo System) est un système multivariable avec une entrée et une sortie pour chacun des deux sous-systèmes horizontal et vertical qui sont fortement couplés et non linéaires. Afin de les stabiliser et diminuer les interactions présentes entre eux, notre travail consiste à modéliser ce système et à lui appliquer une commande, afin d'assurer son bon fonctionnement et améliorer ses performances malgré la présence de plusieurs perturbations dues au changement des différentes conditions. Nous avons choisis de montrer ces performances à travers un simulateur de vol du TRMS.

Dans l'industrie les régulateurs PID sont beaucoup plus utilisés, pour résoudre les problèmes de contrôle. Sauf qu'il est difficile de déterminer le paramètre PID optimale ou presque optimale avec les méthodes de réglages classique. Pour toutes ces raisons, il est très souhaitable d'augmenter les capacités du contrôleur PID en ajoutant de nouvelles caractéristiques. Plusieurs approches ont été documentées dans la littérature pour déterminer les paramètres du PID, d'abord trouvés par Ziegler Nichols (ZN). Aussi nous avons les réseaux de neurones, l'approche floue, les techniques d'optimisation d'essaim de particules (PSO) et les algorithmes génétiques qui ne sont que quelques-

uns parmi de nombreuses œuvres.

Nous avons choisi l'algorithme d'optimisation par l'essaim de particules (en anglais : Particule Swarm Optimisation PSO) que nous allons utiliser pour l'auto-ajustement des paramètres du régulateur PID.

L'objet de ce mémoire est la commande d'un simulateur d'hélicoptère TRMS par un contrôleur PID optimisé par l'algorithme PSO, Pour cela, Le présent mémoire est organisé en trois chapitres qui sont résumés comme suit :

Le premier chapitre est consacré à l'historique du domaine de l'aéronautique ainsi qu'à la description du TRMS et aussi l'élaboration de son modèle mathématique. Par la suite nous allons donner les résultats de simulation obtenus à partir du modèle établi.

Dans le deuxième chapitre nous allons donnés une vue générale sur les metaheuristiques et une description de l'optimisation par essaim de particules ainsi que son principe de fonctionnement ; avant de finir sur un exemple de benchmark pour tester sa fiabilité.

Le troisième chapitre quant à lui commence sur une présentation générale du contrôleur PID et son principe de fonctionnement, ainsi que la méthode de synthèse des contrôleurs PID. Avant de finir sur les résultats de simulation et leurs interprétations.

Chapitre 1

Généralités sur le systéme d'hélicoptère TRMS et Modélisation

1.1 Introduction

Le domaine de l'ingénierie aéronautique est toujours en constante évolution et les systèmes deviennent de plus en plus complexes. Ainsi, pour ne pas mettre en danger la sécurité et la vie des pilotes et des usagés, les chercheurs proposent des simulateurs qui ont le même principe de fonctionnement que les systèmes réels mais plus accessibles à l'application de nouvelles lois de commande avant de les implémenter sur des systèmes réels.

Dans notre cas, nous allons étudié le simulateur de vol d'hélicoptère TRMS (Twin Rotor Mimo System). Dans ce chapitre, nous allons en premier lieu décrire le principe de vol d'un hélicoptère, ensuite nous présenterons les différents constituants et le principe de fonctionnement du simulateur de vol. Ensuite nous passerons à l'étude des différentes forces qui régissent notre système afin de réaliser un modèle de notre système TRMS.

1.2 Histoire des hélicoptères :

L'histoire de l'hélicoptère et autres voilures tournantes commence au début du XXe siècle, comme pour l'avion. Mais l'insuffisance de la puissance des moteurs et les problèmes de stabilité rendent les développements beaucoup plus longs et aléatoires. En 1486 le savant et ingénieur italien Léonard de Vinci [01] est le premier à s'être intéressé au concept d'hélicoptère, son dessin montre une machine volante à aile tournante comme sur la Figure 1.1.

En 1754 le Russe Lemonosov [02] présente à un aréopage scientifique un petit appareil à

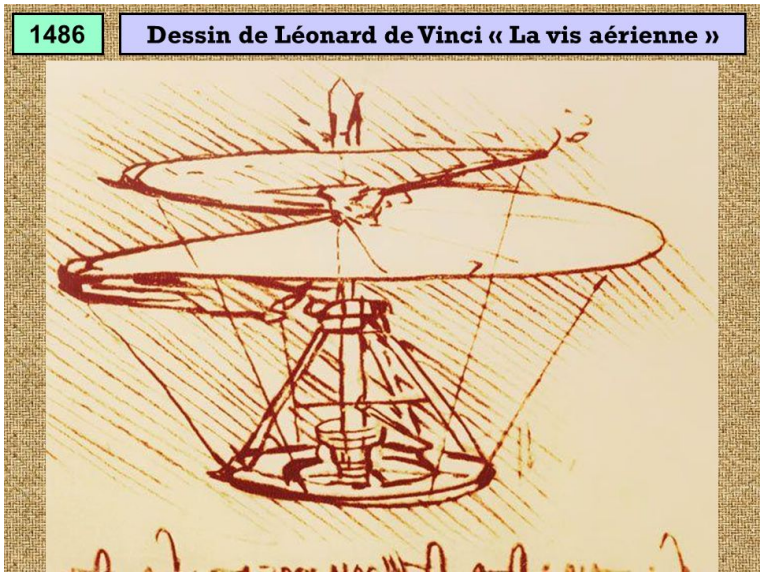

FIGURE 1.1 – Dessin de Léonard de Vinci [1]

deux rotors qui fonction par un mécanisme d'horlogerie, il démontre ainsi l'existence d'une force de sustentation.

En 1863 le français Gustave de Ponton d'Amécourt [02] construit un appareil qui fonctionne par un petit moteur à vapeur, il a aussi inventé le mot hélicoptère (du grec hélix, hélice et pteron, aile). En 1907 le mécanicien Paul Cornu [03] réalise le premier vol libre d'un homme en hélicoptère (Figure 1.2). L'appareil, lourd de 260kg en charge s'élève d'une trentaine de centimètres, puis de 1,50m avec le premier passager.

FIGURE 1.2 – L'hélicoptère de Paul Cornu [3]

En juillet 1917 le pilote Max Boucher [01] fait voler un avion sans l'intervention de l'homme sur 1 km. Et au début de l'année 1918, Georges Clemenceau lance un projet d'avions sans pilotes.

En 1939, l'américain Igor Sikorsky [01] a réalisé le Vought-Sikorsky 300, c'est le premier hélicoptère de configuration classique, avec un rotor principal et un rotor anti-couple.

En 1991, Les premiers drones miniatures firent leur apparition lors de la 1ère guerre du Golfe.

1.3 Principe de vol de l'hélicoptère :

Le rôle du rotor principal est d'assuré la traction et la portance d'hélicoptère. Pour lui permettre de voler, il ne suffit pas de tourner le rotor mais on doit changer l'angle d'attaque sur les pales (voir Figure 1.3).

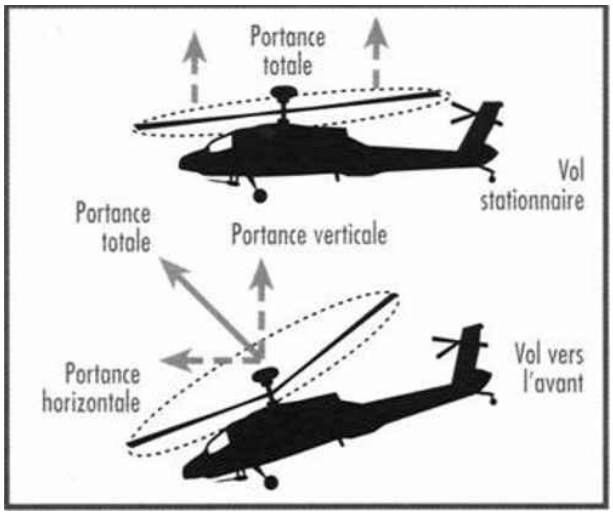

FIGURE 1.3 – Principe de vol [4]

Pour faire reculer ou avancer l'hélicoptère (ou tourner à droite ou à gauche), il va falloir incliner le disque rotor de sorte que l'hélicoptère doit être aspiré vers l'avant ou vers l'arrière (ou vers la droite ou vers la gauche). Pour cela il faut augmenter la portance du côté opposé de la direction voulu en changent l'angle d'attaque sur la pale (si on veut avancer l'hélicoptère il faut augmenter la portance sur la pale arrière). Dans cette même situation de mouvement l'effet gyroscopique entre en compte afin de maintenir l'hélicoptère en équilibre peu importe l'angle d'inclinaison de l'hélice principale (nous donnerons plus de détails de cette effet dans la suite du chapitre).

Pour faire tourner le rotor, il faut des moteurs assez puissants, ces turbomoteurs ne sont pas des turboréacteurs car ils ne servent pas à propulser l'hélicoptère vers l'avant, leur seule fonction

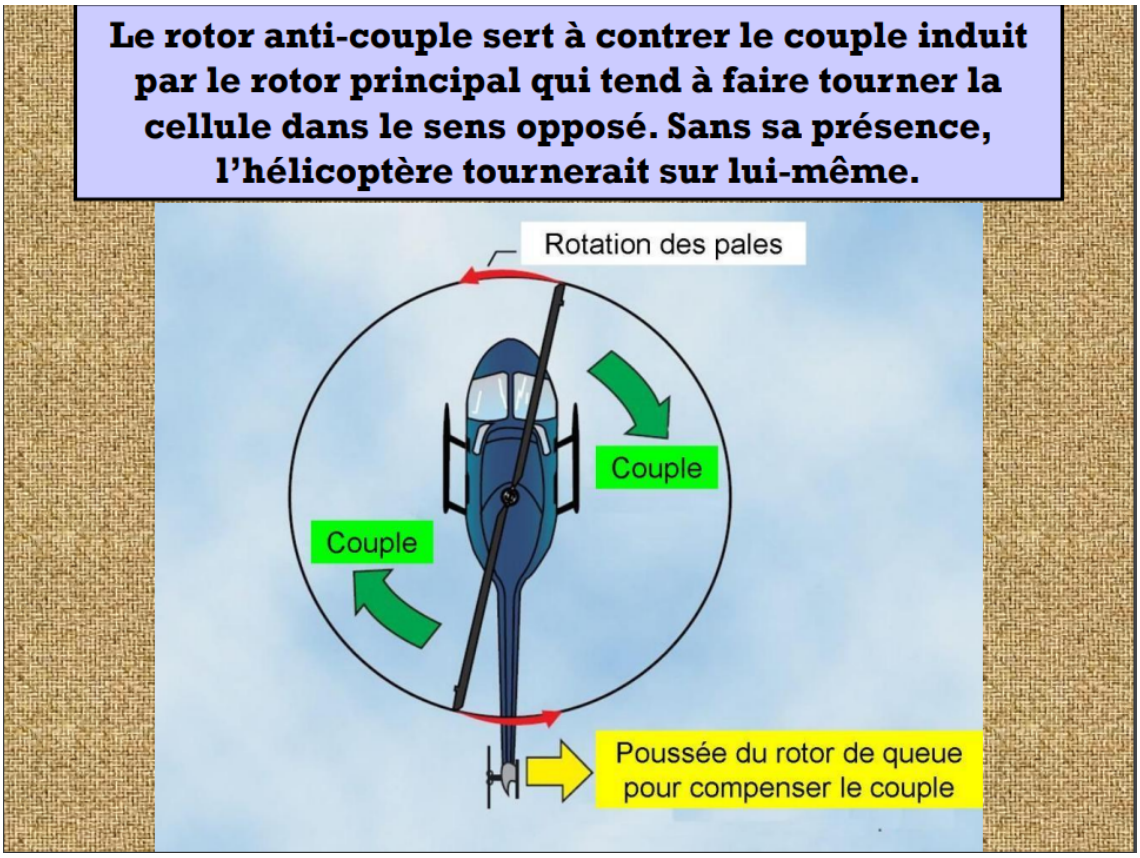

FIGURE 1.4 – Effet gyroscopique [4]

est de faire tourner le rotor. La rotation du rotor exerce sur l'air un couple dû à son mouvement. L'air en retour exerce un autre couple de même intensité mais dans la direction opposée. Pour contrer cet effet de couple, et éviter que l'hélicoptère ne tourne sur lui-même, on dispose à l'arrière un rotor de queue son rôle est de contrer le couple dû à la rotation du rotor principal appeler aussi rotot anti-couple comme on peut le voir sur la figure 1.4.

1.4 Description du simulateur :

Une photographie du système MIMO à deux rotors est présentée à la figure 1.5 réaliser par la compagnie Feedback spécialisée dans la conception des équipements d'ingénierie. Le système est utilisé pour démontrer les principes d'un système MIMO non linéaire, avec un couplage croisé important. Son comportement ressemble à celui d'un hélicoptère mais contrairement à la plupart des hélicoptères volants, l'angle d'attaque des rotors est fixe et les forces aérodynamiques sont contrôlées en faisant varier la vitesse des moteurs. Un couplage croisé significatif est observé entre les actions des rotors, chaque rotor influençant les deux positions angulaires . Il y a deux hélices

entraînées par des moteurs à courant continu aux deux extrémités d'une poutre, qui pivote sur sa base. L'articulation permet à la poutre de tourner de telle manière que ses extrémités se déplacent sur des surfaces sphériques. Un contrepoids est fixé à la poutre et il détermine une position d'équilibre stable. Les commandes du système sont les tensions d'alimentation des moteurs et les signaux mesurés sont la position de la poutre dans l'espace, c'est-à-dire deux angles de position.

Le mouvement de la tige autour de l'axe vertical est réalisé par l'hélice de queue qui permet un déplacement angulaire (angle d'orientation). Cependant, le mouvement de la tige autour de l'axe horizontal est assuré par l'hélice principale qui permet un déplacement angulaire (angle d'élévation).

Fig. 1. Twin Rotor MIMO System.

FIGURE 1.5 – Simulateur d'hélicoptère de type TRMS réaliser par Feedback [5]

1.5 Différentes configurations des rotors d'un hélicoptère :

Il existe des configurations diverses de rotors [06] tel que :

— **Configuration à rotor principal avec ou sans rotor de queue**. Par la suite, nous appellerons hélicoptère standard ou simplement hélicoptère, la configuration correspondant à un rotor principal et un rotor de queue.

— **Configurations à deux rotors principaux.** On peut distinguer les cinq configurations suivantes :

1. Configuration à deux rotors principaux en tandem (birotor en tandem).

2. Configuration à deux rotors principaux côte à côte.
3. Configuration à deux rotors principaux pivotants.

— **Un hélicoptère à trois rotors** : Moins performant en vol que le quad rotor, le tri-rotor est constitué de deux rotors à l'avant qui tournent dans des sens opposés pour modifier le tangage et d'un rotor en arrière pour régler le roulis.

— **Un hélicoptère à quatre rotors** : Un quad rotor est un engin volant doté de quatre rotors placés aux extrémités d'une armature en croix. Ces quatre rotors lui fournissent la force verticale (portance) qui lui permet de décoller.

1.6 Modélisation du TRMS

Nous commencerons par une modélisation détaillée du simulateur d'hélicoptère présenté dans le chapitre précédent, on présentera ensuite le modèle non-linéaire du TRMS. Nous continuons par la description des modèles découplés et on terminera par quelques résultats de simulations en boucle ouverte.

Un modèle non linéaire est dérivé dans cette section. Le modèle est basé sur la modélisation du principe fondamental de la dynamique. Les sorties de l'installation sont les suivantes : angle de position dans le plan vertical et l'angle de position dans le plan horizontal . Nous examinerons d'abord le plan vertical, puis nous nous intéresserons à l'horizontal. Une vue avant schématique des parties connectées du système MIMO à double rotor est représentée sur la Figure 1.6 avec les forces de gravitation liées.

1.6.1 Modélisation du sous-système vertical

Dans cette partie, nous allons étudier les forces et les moments des forces qui agissent sur le système couplé seulement dans le plan vertical (rotation autour de l'axe horizontal). La deuxième loi de Newton qui décrit le mouvement dans le plan vertical est donnée sous la forme suivante : [6-8]

$$M_v = J_v \frac{d^2\alpha_v}{dt^2} \tag{1.1}$$

Avec :

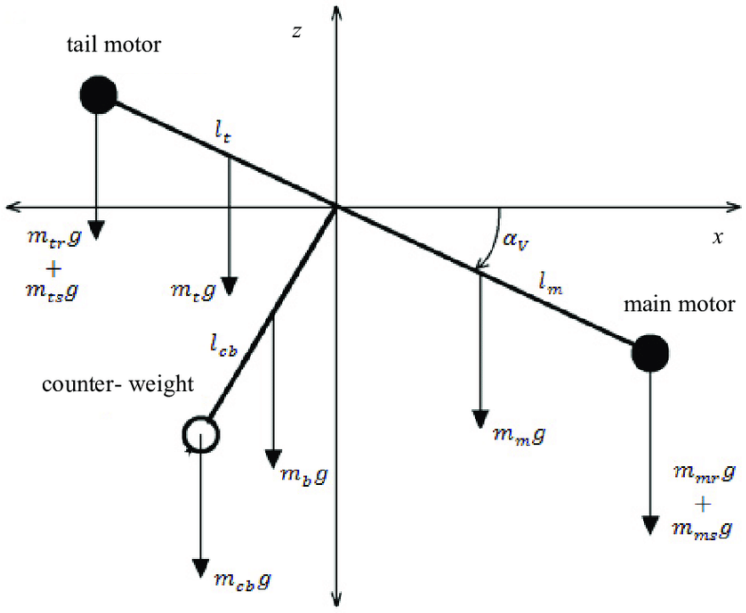

FIGURE 1.6 – Forces agissantes sur l'hélicoptères

- M_v : La somme des moments dans le plan vertical.
- J_v : La somme des moments d'inertie par rapport à l'axe horizontal.
- α_v : l'angle d'élévation.

— Les deux moments sont décrits par les équations suivantes :

$$M_v = \sum_{i=1}^{4} M_{vi} \tag{1.2}$$

$$J_v = \sum_{i=1}^{4} J_{vi} \tag{1.3}$$

Nous avons 4 moments cinétiques chaque moment représente une force qui s'applique a notre système.

a) Moment de la force aérodynamique M_{v1} :

La force aérodynamique aussi appeler force de portance est ce qui permet à l'hélicoptère de s'élever ou se poser et elle est liée à la vitesse angulaire appliquée par le moteur du rotor principal et la forme des hélices. La figure 1.7. nous montre le principe de la force de portance. Son équation est donnée sous la forme :

$$M_{v1} = a_1\tau_1^2 + b_1\tau_1 \tag{1.4}$$

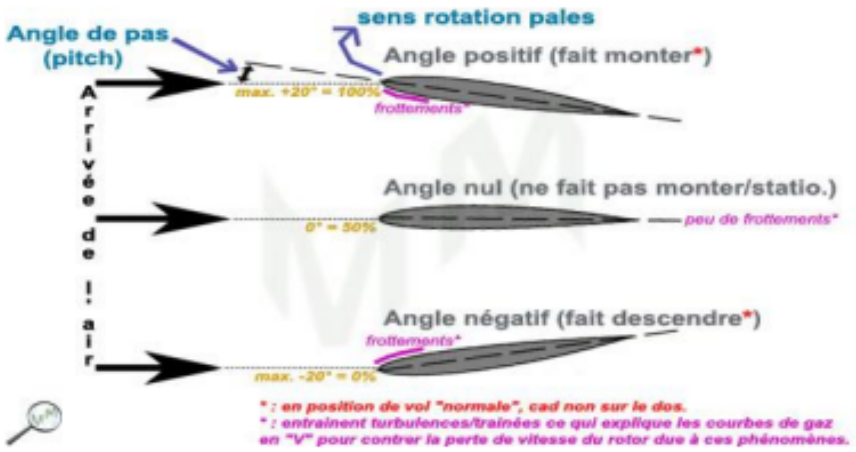

FIGURE 1.7 – Principe de portance [4]

Avec :

τ_1 : est le couple délivré par le rotor principal (vertical).

a_1 b_1 : sont des paramètres caractérisant le moment de la force aérodynamique

b) Moment de la force de gravitation M_{v2} :

Notre système est lié à la force de gravité et donc à un moment gravitationnel décrit par l'équation suivante :

$$M_{v2} = M_g sin(\alpha_v) \tag{1.5}$$

M_g : Moment de gravité

c) Moment de friction M_{v3} :

Le moment de friction est du frottement de l'air lors du mouvement et son équation est de la forme :

$$M_{v3} = f_1 \alpha_v - f_2 sin(\alpha_v) \tag{1.6}$$

avec : f_1 f_2 sont deux paramètres liés à la fonction de friction.

d) Moment de l'effet gyroscopique M_{v4} :

Le moment gyroscopique est causé par la force de Coriolis, ce moment est le résultat d'un changement dans la position de déplacement du rotor principal dans le sens de l'azimut, il est décrit comme suit :

$$M_{v4} = KM_{v1}\dot{\alpha_h}cos(\alpha_v) \tag{1.7}$$

avec : α_h : est l'angle d'azimut (horizontal)

K_v : est le paramètre de l'inertie gyroscopique

1.6.2 Modélisation du sous-système horizontal

De la même façon, on peut décrire le mouvement de la tige autour de l'axe vertical. Le mouvement horizontal peut être décrit comme étant un mouvement de rotation d'un solide.

a) Moment de la force aérodynamique M_{h1} La force aérodynamique dans le plan horizontal est la force de propulsion engendrée par la rotation des hélices du rotor secondaire. Le principe du moment aérodynamique dans le plan horizontal reste le même que celui sur le plan vertical, il est donc exprimé comme suit :

$$M_{h1} = a_2\tau_2^2 + b_2\tau_2 \tag{1.8}$$

avec :

— τ_2 est le couple délivré par le rotor secondaire.

— a_2 b_2 : sont des paramètres caractérisant le moment de la force aérodynamique.

b) Moment de la force de friction M_{h2} Le moment de friction est le frottement de l'air avec la partie secondaire de la poutre en fonction de la vitesse , elle s'exprime comme suit :

$$M_{h2} = f_3\alpha_v + f_4sin(\alpha_v) \tag{1.9}$$

Avec f_3 f_4 sont les paramètres de la fonction de friction.

1.6.3 Modélisation des propulseurs

Le TRMS contient deux propulseurs qui sont constitués chacun d'un moteur DC et une hélice, les deux moteurs sont identiques avec des charges mécaniques différentes.

Le moteur M_1 est modélisé par une fonction de transfert de premier ordre :

$$\tau_1(s) = \frac{K_1}{1.1s+1} u_1(s) \tag{1.10}$$

avec :

— K_1 : est le gain du moteur

— T_1 et T_2 :sont des constantes liées au moteur

— u_1 : est l'entrée de commande du moteur

La dynamique de couplage est décrit par la fonction de transfert de premier ordre suivante :

$$M_R(s) = \frac{K_c(T_0 s+1)}{T_p s+1} \tau 1 \tag{1.11}$$

avec :

K_c : est le gain de la réaction de l'élan

T_0 T_p :sont deux paramètres caractérisant la réaction de l'élan

Le moteur M_2 est modélisé par une fonction de transfert du premier ordre :

$$\tau_2(s) = \frac{K_2}{s+1} u_2(s) \tag{1.12}$$

K_2 :le gain du moteur

T_3 T_4 :sont des constantes liées au moteur

u_2 :l'entrée de commande du moteur

1.6.4 Tableaux des paramètres :

Les paramètres de ce modèle ont été choisis plus ou moins expérimentalement, le tableau 1.1 donne les valeurs des différents paramètres [5].

paramètres	Définition	valeurs
J_v	Moment d'inertie du rotor vertical	$6.8.10^{-2}kg.m^2$
J_h	Moment d'inertie du rotor horizontal	$2.10^{-2}kg.m^2$
a_1	Paramètre caractéristique statique	0.135
b_1	Paramètre caractéristique statique	0.0924
a_2	Paramètre caractéristique statique	0.02
b_2	Paramètre caractéristique statique	0.09
M_g	moment de gravité	0.32N.m
f_1	Paramètre de la fonction de friction dynamique	0.006N.m.s/rad
f_2	Paramètre de la fonction de friction dynamique	0.001N.m.s/rad
f_3	Paramètre de la fonction de friction dynamique	0.1N.m.s/rad
f_4	Paramètre de la fonction de friction dynamique	0.01N.m.s/rad
K_v	Paramètre d'inertie gyroscopique	0.05 s/rad
K_1	Le gain du Moteur 1	1.1
K_2	Le gain du Moteur 2	0.8
T_1	constante liée au moteur 1	1.1
T_2	constante liée au moteur 1	1
T_3	constante liée au moteur 2	1
T_4	constante liée au moteur 2	1

Tableau 1.1 : Valeurs numériques des paramètres du simulateur

1.7 Modèle d'état non linaire du simulateur

$$\begin{cases} \frac{d\alpha_v}{dt} = \dot{\alpha_v} \\ \frac{d\dot{\alpha_v}}{dt} = \frac{a_1}{J_v}\tau_1^2 + \frac{b_1}{J_v}\tau_1 - \frac{M_g}{J_v}sin(\alpha_v) - \frac{f_1}{J_v}\dot{\alpha_v} - \frac{f_2}{J_v}sin(\dot{\alpha_v}) - \frac{K}{J_v}a_1\dot{\alpha_h}cos(\alpha_v)\tau_1^2 - \frac{K}{J_v}b_1\dot{\alpha_h}cos(\alpha_v)\tau_1 \\ \frac{d\tau_1}{dt} = \frac{K_1}{T_1}u_1 - \frac{T_2}{T_1}\tau_1 \\ \frac{d\alpha_h}{dt} = \dot{\alpha_h} \\ \frac{d\dot{\alpha_h}}{dt} = \frac{a_2}{J_v}\tau_2^2 + \frac{b_2}{J_v}\tau_2 + \frac{f_3}{J_v}\dot{\alpha_v} - \frac{f_4}{J_h}sin(\dot{\alpha_h}) - \frac{a_1}{J_h}\frac{K_c(T_0s+1)}{T_ps+1}\tau 1^2 - \frac{b_1}{J_h}\frac{K_c(T_0s+1)}{T_ps+1}\tau 1 \\ \frac{d\tau_2}{dt} = \frac{K_2}{T_3}u_2 - \frac{T_4}{T_3}\tau_2 \end{cases}$$

1.8 Simulation en boucle ouvert du modèle du simulateur

Le modèle établi (équation 1.) du simulateur est utilisé, dans cette section, pour des fins de simulations. Ces dernières sont réalisées sous Matlab-Simulink afin d'étudier le comportement et les caractéristiques du simulateur d'une part et d'autre part pour la validation du modèle obtenu. Le Figure 1.8 donne la réponse du système après application d'une impulsion de tension de 1V et de durée 1 s à l'entrée de chaque moteur avec les conditions initiales suivantes : $\alpha_{v0} = [0.4; 0; 0; 0; 0; 0]$

$\alpha_{h0} = [0; 0; 0; 0; 0; 0]$

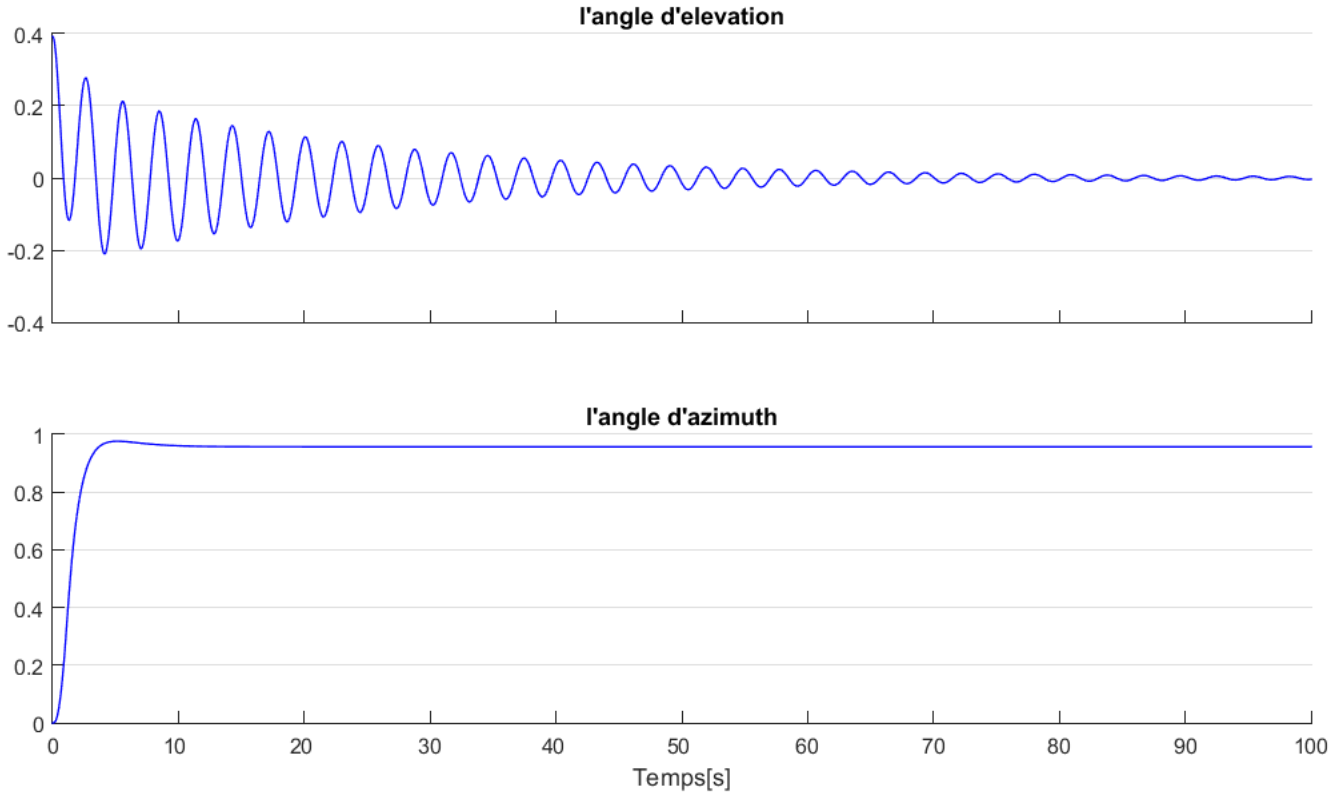

FIGURE 1.8 – Angle d'élévation et d'azimuth

1.8.0.1 Interprétation des résultants

La Figure ci-dessus montre que la réponse du sous-système d'élévation est oscillatoire amortie, ce dernier est dû aux forces gravitationnelles qui agissent sur le plan vertical uniquement.

Par contre le sous-système d'azimut augmente rapidement jusqu'à une valeur donnée et il se stabilise.

Les figures obtenues lors de la simulation de notre modèle sous Simulink de Matlab sont identique au figures donnée dans le manuel « Twin Rotor Mimo System Contrôle expriments 33-

949S » page 11 figure 08 [5]. Pour cela on peut valider et continuer à travailler avec ce modèle du TRMS

1.9 Conclusion

Dans ce chapitre, nous avons donné une brève description du simulateur TRMS, ainsi que son principe de fonctionnement. Pour passer à la modélisation de ce modèle d'hélicoptère de laboratoire, ou le schéma bloc du système a été réalisé, pour finir avec des tests de ce dernier en boucle ouvert avec le Simulink de Matlab, et le comparer avec le manuel du constructeur du TRMS.

Chapitre 2

Optimisation par essaim de particules (PSO)

2.1 Introduction

Dans ce chapitre, nous allons aborder les métaheuristiques, méthode d'optimisation globale et développer en détail l'algorithme PSO ainsi que son mode de fonctionnement. Pour ne pas rester dans un cadre descriptif, un exemple d'optimisation d'une fonction à deux variables par le PSO est étudié dans la dernière section du présent chapitre.

2.2 Métaheuristiques :

Une métaheuristique est un algorithme visant à résoudre des problèmes d'optimisation difficiles pour lesquels on ne connaît pas de méthode classique plus efficace. Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global.

Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin de trouver une approximation de la meilleure solution. Il existe un grand nombre de métaheuristiques différentes, allant de la simple Recherche locale à des algorithmes complexes de recherche globale, par contre ces méthodes utilisent un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

En d'autres termes, ces algorithmes se veulent des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds[12].

Il existe d'autres définitions et on peut citer entre autres, celle de **I.H. Osman et G. Laporte** qui considèrent une métaheuristique comme un processus de génération itératif qui guide une

heuristique subordonnée en combinant les deux processus, c'est à dire qui combine intelligemment différents concepts pour explorer et exploiter l'espace de recherche qui avec des stratégies d'apprentissage permet de structurer l'information afin de trouver efficacement des solutions optimales[13] .

Pour **T. Stutzle** , les métaheuristiques sont typiquement des stratégies de haut niveau qui guident une heuristique sous-jacente plus spécifique au problème, afin d'augmenter leur efficacité. L'objectif principal est d'éviter les inconvénients de l'amélioration itérative et, en particulier de la descente multiple en permettant à la recherche locale d'échapper aux optimums locaux. Cet objectif est atteint soit en autorisant des mouvements aggravants, soit en générant de nouvelles solutions de départ pour la recherche locale de manière plus "intelligente" qu'en fournissant simplement des solutions initiales aléatoires. De nombreuses méthodes peuvent être interprétées comme l'introduction d'un biais tel que des solutions de haute qualité soient produites rapidement[14] .

Il n'existe pas de définition qui fasse l'unanimité, mais toutes s'accordent sur les points suivants :

- Les métaheuristiques sont des stratégies permettant de guider la recherche d'une solution optimale.
- Le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.
- Les techniques qui constituent des algorithmes de type métaheuristique vont de la simple recherche locale à des processus d'apprentissage complexes.
- Les métaheuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
- Les métaheuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.
- Les concepts de base des métaheuristiques peuvent être décrit de manière abstraite, sans faire appel à un problème spécifique.
- Les métaheuristiques peuvent faire appel à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur.
- Les métaheuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche.

2.3 Classes des métaheuristiques :

On peut faire la différence entre les métaheuristiques qui s'inspirent de phénomènes naturels et celles qui ne s'en inspirent pas. Par exemple, les algorithmes génétiques et les algorithmes des fourmis s'inspirent respectivement de la théorie de l'évolution et du comportement de fourmis à la recherche de nourriture mais une telle classification ne semble cependant pas très utile et est parfois difficile à réaliser, car il existe de nombreuses métaheuristiques récentes qu'il est difficile de les classées dans l'une des deux catégories.[13]

Certains se demanderont par exemple si l'utilisation d'une mémoire dans la méthode tabou n'est pas directement inspirée de la nature mais encore on peut également distinguer les métaheuristiques qui travaillent avec une population de solutions de celles qui ne manipulent qu'une solution à la fois.

Les méthodes qui tentent itérativemment d'améliorer une solution sont appelées méthodes de recherche locale telles que La méthode tabou et la recherche à voisinages variables qui sont des exemples typiques de méthodes de recherche locale. Ces méthodes construisent une trajectoire dans un espace S de solutions en tentant de se diriger vers des solutions optimales[15].Alors que par exemple la plus connue des méthodes qui travaille avec une population de solutions est l'algorithme génétique.

Les métaheuristiques peuvent également être classées selon leur manière d'utiliser la fonction objectif, comme celles dites statiques qui travaillent directement sur la fonction objectif alors que d'autres dites dynamiques, font usage d'une fonction obtenue à partir de la fonction objectif en ajoutant quelques composantes qui permettent de modifier la topologie de l'espace des solutions. Ces composantes additionnelles pouvant varier durant le processus de recherche[16].

Certaines métaheuristiques font usage de l'historique de la recherche, alors que d'autres n'ont aucune mémoire du passée que l'on appel aussi processus Markoviens puisque l'action à réaliser est totalement déterminée par la situation courante.

L'usage de l'historique de la recherche peut être fait de diverses manières, généralement on différencie les méthodes ayant une mémoire à court terme de celles à long terme. Mentionnons également que certaines métaheuristiques utilisent les concepts additionnels tels que la diversification et l'intensification,et par diversification on entend généralement une exploration très large de l'espace de recherche, alors que le terme intensification vient mettre l'accent sur l'exploitation de l'information accumulée durant la recherche,il est important de bien doser l'usage de ces deux

ingrédients afin que l'exploration puisse rapidement identifier des régions de l'espace de recherche qui contiennent des solutions de bonne qualité, sans perdre trop de temps à exploiter des régions moins prometteuses[14].

2.3.1 L'algorithme génétique

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle (croisements, mutations, sélection, etc).Ils ont déjà une histoire relativement ancienne, puisque les premiers travaux publiés de John Holland sur les systèmes adaptatifs remontent à 1975.[15] Un algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données, bien que pour l'utiliser on doit disposer des cinq éléments suivants :

- **Un principe de codage de l'élément de population** : cette étape associe à chacun des points de l'espace d'état une structure de données et elle se place généralement après une phase de modélisation mathématique du problème traité, ainsi le choix du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très employés à l'origine mais les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs, pour l'optimisation de problèmes à variables continues.
- **Un mécanisme de génération de la population initiale** : ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Ainsi le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.
- **Une fonction à optimiser** qui prend ses valeurs dans R^+ et est appelée fitness ou fonction d'évaluation de l'individu. Elle est utilisée pour sélectionner et reproduire les meilleurs individus de la population.
- **Des opérateurs** permettant de diversifier la population au cours des générations et d'explorer l'espace d'état, ainsi l'opérateur de croisement recompose les gènes d'individus existant dans la population et l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état.
- **Des paramètres de dimensionnemment** tel que la taille de la population, le nombre

total de générations ou critère d'arrêt et les probabilités d'application des opérateurs de croisement et de mutation.

2.4 Algorithme d'essaim de particules :

C'est une métaheuristique visant à résoudre des problèmes d'optimisation difficile pour lesquels on ne connaît pas de méthode classique plus efficace.Il est de type stochastique itératif , qui progresse vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif.[6]

Cet algorithme s'inspire à l'origine du monde du vivant. Il s'appuie notamment sur un modèle développé par Craig Reynolds à la fin de l'années 1980, permettant de simuler le déplacement d'un groupe d'oiseaux. Ainsi, grâce à des règles de déplacement très simples (dans l'espace des solutions), les particules peuvent converger progressivement vers un minimum global, alors le PSO définit chaque individu (particule) par sa vitesse, sa position et la meilleure position qu'elle connait. Chaque particule **i** corrige sa vitesse **Vi** et sa position **Xi** selon les équations (2.1) et (2.2) respectivement.

$$V_i^{k+1} = wV_i^k + C_1R_1(X_i^p - X_i^k) + C_2R_2(X_g - X_i^k) \tag{2.1}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{2.2}$$

Où **w** est le poids de chaque particule , X_{pi} désigne le meilleur individu dans l'entourage de la particule **i** (personnel best ou la meilleure position qu'elle connait), X_g désigne la position du meilleur individu de tout l'essaim (on l'appelle global best), **C1** et **C2** sont des coefficients d'accélération, **R1** et **R2** sont des valeurs aléatoires.

2.5 Principe de fonctionnement du PSO :

Le principe du PSO est très simple, après initialisation de la population, de sa vitesse et l'initialisation des paramètres de l'algorithme (w, C1,C2, taille de la population N, limite des positions et des vitesses), il consiste à faire varier la vitesse et la position des particules suivant

les deux équations (2.1) et (2.2). À chaque itération k, pour chaque individu i, si la fitness des particules X_i^{k+1} est meilleure que le meilleur individu X_g, alors$X_g = X_i^{k+1}$.[6] Et si la fitness des particules X_i^{k+1} est meilleure que la fitness de la meilleure position connue par la particule i , c'est-à-dire si la fitness de X_i^{k+1} est meilleure que celle deX_i^p alors $X_i^p = X_i^{k+1}$.

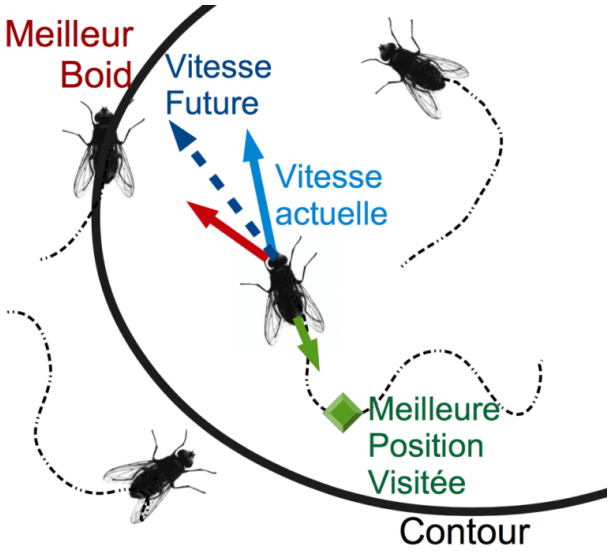

FIGURE 2.1 – Principe du PSO

2.6 Topologies du PSO :

La topologie détermine les voisins de chaque particule au cours du processus évolutif de PSO, chaque particule partage sa meilleure expérience avec ses voisines et apprend de leurs meilleures positions. Par conséquent, la topologie est importante pour la performance globale du PSO.

La topologie du meilleur global et celle du meilleur local sont deux topologies sociales courantes dans le PSO. Dans la topologie meilleur global, chaque particule est connectée à toutes les autres particules or dans la topologie meilleur local, chaque particule n'est connectée qu'avec ses voisins les plus proches.

Nous avons la topologie en étoile , en annaux , en arbre et pleins d'autres mais encore les différentes études ont confirmé que le PSO est très affecté par la topologie appliquée et que chaque topologie est optimale selon le problème auquel elle est associée .

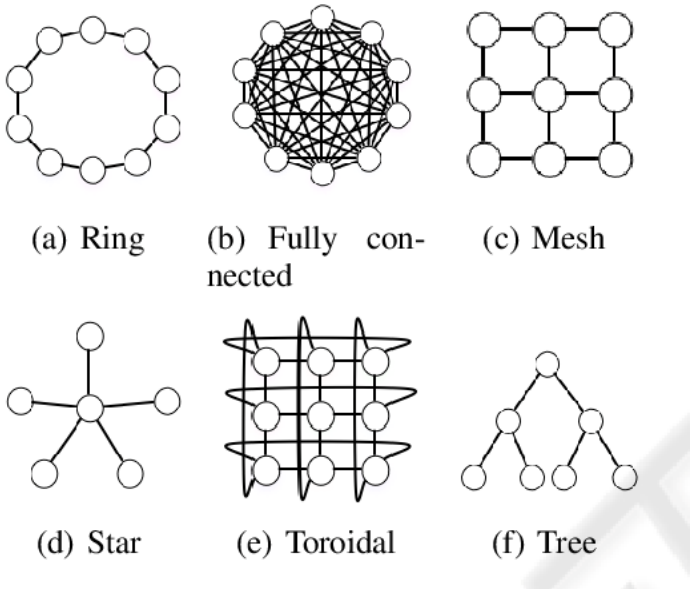

FIGURE 2.2 – Les différentes topologies

2.7 Paramètres du PSO :

Le PSO de base est influencé par un certain nombre de paramètres de contrôle, à savoir la dimension du problème, le nombre de particules, les coefficients d'accélération, le poids d'inertie, la taille du voisinage, le nombre d'itérations et les valeurs aléatoires qui échelonnent la contribution des composantes cognitive et sociale[9].

2.7.1 Taille de l'essaim :

Plus il y a de particules dans l'essaim, plus la diversité initiale de l'essaim est grande à condition qu'un bon schéma d'initialisation uniforme soit utilisé pour initialiser les particules.Un grand essaim permet de couvrir plus de parties de l'espace de recherche par itération Cependant, elle augmente la complexité de calcul et dégrade le PSO en une recherche aléatoire. Néanmoins, il a été démontré dans un certain nombre d'études empiriques que le PSO est capable de trouver des solutions optimales avec de petits essaims de 10 à 30 particules[8]

2.7.2 La taille du voisinage

Pour tirer parti des avantages des tailles de voisinage petites et grandes, la meilleure solution est de commencer la recherche avec des petits voisinages et augmentez la taille proportionnellement à l'augmentation du nombre d'itérations.

Cette approche garantit une diversité initiale élevée avec une convergence plus rapide au fur et à mesure que les particules se déplacent vers une zone de recherche prometteuse[8]

2.7.3 Nombre d'itérations

Le nombre d'itérations pour atteindre une bonne solution dépend également du problème, car un nombre trop faible d'itérations peut mettre fin prématurément à la recherche alors qu'un trop grand nombre d'itérations a pour conséquence d'augmenter inutilement la complexité de calcul.

2.7.4 Coefficients d'accélération

Les coefficients d'accélération, **C1** et **C2**, ainsi que les valeurs aléatoires **R1** et **R2**, contrôlent l'influence stochastique des composantes cognitive et sociale sur la vitesse globale d'une particule.

C1 et **C2** sont également appelées paramètres de confiance, où **C1** exprime la confiance qu'une particule a en elle-même, tandis que **C2** exprime la confiance qu'une particule a en ses voisins.

Avec $\mathbf{C1} = \mathbf{C2} = 0$, les particules continuent de voler à leur vitesse actuelle jusqu'à ce qu'elles atteignent une limite de l'espace de recherche (en supposant l'absence d'inertie).

$\mathbf{C1} > 0$ et $\mathbf{C2} = 0$, toutes les particules sont des grimpeurs indépendants. Chaque particule trouve la meilleure position dans son voisinage en remplaçant la meilleure position actuelle si la nouvelle position est meilleure.

si $\mathbf{C2} > 0$ et $\mathbf{C1} = 0$, l'essaim entier est attiré vers un seul point alors l'essaim se transforme en un limiteur de pente stochastique.

Les particules tirent leur force de leur nature coopérative, et sont plus efficaces lorsque **C1** et **C2** coexistent dans un bon équilibre. Alors que la plupart des applications utilisent c1 = c2, le rapport entre ces constantes dépend du problème[9]

2.8 Implémentation de l'algorithme PSO

L'implémentation du PSO avec logiciel de programmation tel que MATLAB, passe par plusieurs étapes que nous allons définir ci-dessous :

2.8.1 Définition du problème

d'abord il faut définir la fonction objectif dont on cheche à optimiser. Ensuite nous devons choisir la taille de notre espace de recherche ainsi que le nombre de variables de décision.

2.8.2 Configuration des paramètres

Dans cette étape, le but est de bien choisir les paramètres PSO que nous avons évoqué plus haut tels que le nombre d'itérations , les coefficients d'accélération et la taille de l'essaim.

2.8.3 Initialisation

Une fois que la fonction objectif est définie et que les différents paramètres du PSO sont fixés, on passe à l'étape d'initialisation pour créer les particules de départ de l'algorithme. Pour une meilleure exploration de l'espace de recherche, il est préconisé de répartir uniformément les particules dans l'espace de recherche.

2.9 Application du PSO :

Pour tester les performances et le bon déroulement de l'algorithme PSO implémenté sous MATALB, on se propose dans cette section, d'étudier un problème de minimisation d'une fonction benchmark.

2.9.1 Fonction Ackley :

La fonction Ackley est largement utilisée pour tester les algorithmes d'optimisation. Dans sa forme bidimensionnelle, comme le montre la Figure 2.3 ci-dessus, elle se caractérise par une région extérieure presque plate et un grand trou au centre. La fonction présente un risque pour les algorithmes d'optimisation, en particulier les algorithmes de descente du gradient, d'être piégés dans l'un de ses nombreux minima locaux. Elle définie par :

$$F_{x,y} = -20exp[-0.2\sqrt{0.5(x^2+y^2)}] - exp[0.5cos(2\pi(x)) + cos(2\pi(y))] + e + 20 \qquad (2.3)$$

La représentation de la fonction Ackley est donné par la figure 2.3 comme suit :

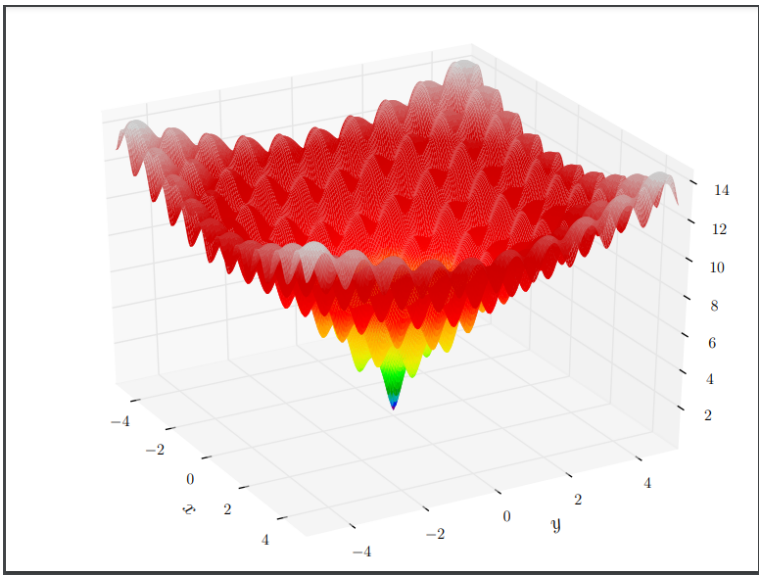

FIGURE 2.3 – Fonction Ackley

La Figure 2.4 donne l'évolution de la meilleure solution obtenue à chaque itération. Les valeurs des paramètres utilisés sont résumés dans le tableau 2.1.

Paramètres	C1	C2	w
Valeurs	1.5	2	1

Tableau 2.1 : Valeurs numériques des paramètres du PSO

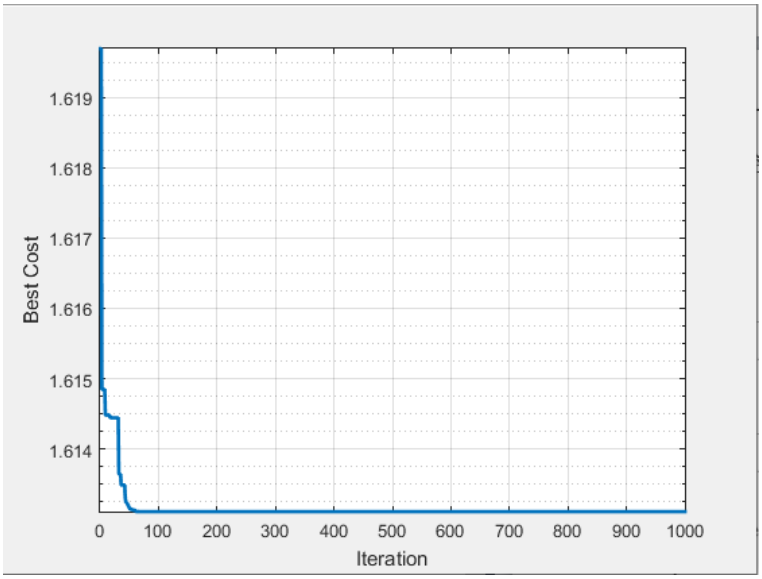

FIGURE 2.4 – résultat de la simulation

Comme on peut le voir sur la Figure, la phase d'optimisation est très rapide car en moins de 100 itérations l'algorithme arrive à localiser l'optimum global (F(0,0)=0) avec une très bonne précision.

2.10 Conclusion :

Dans ce chapitre nous avons d'abord abordé quelques définitions relatives au métaheuristiques avant de détailler l'algorithme d'optimisation par essaim de particules (PSO). Ce derniers est l'un des algorithmes les plus utilisés en raison de sa simplicité de mise en œuvre et d'être efficace dans la recherche globale. Nous avons donc décrit son principe de fonctionnement et les différents paramètres qui le constituent.

Un exemple d'optimisation de fonction benchmark a été réalisé pour évaluer les performances du PSO d'une part et d'autre part pour s'assurer du bon fonctionnement de l'algorithme avant d'être appliqué, dans le prochain chapitre, sur un problème plus complexe.

Chapitre 3

Commande du TRMS

3.1 Introduction :

Dans ce chapitre nous allons présenter la commande PID et les paramètres qui la compose puis nous montrerons comment nous avons appliquer l'algorithme PSO au PID afin d'optimiser notre commande. Enfin on présentera une simulation du système commander et les résultats obtenu ainsi qu'une validation des dits résultats sur notre modèle initial.

3.2 Définition de la commande PID :

Le régulateur PID, appelé aussi correcteur PID (proportionnel, intégrateur, dérivateur) est un système de contrôle, il est constitué d'un comparateur pour observer l'écart (erreur) entre la mesure et la consigne, et d'un correcteur dont l'algorithme permet d'obtenir une loi d'évolution de la mesure du procédé conforme au cahier des charges. En d'autres termes, Ils permettent d'effectuer un asservissement en boucle fermée du système étudié.[18]

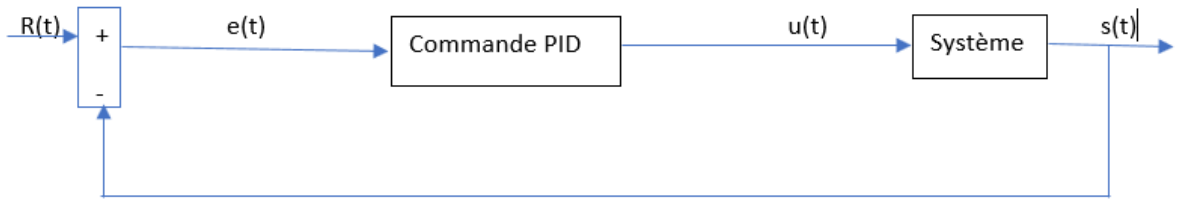

FIGURE 3.1 – Commande PID du TRMS

3.2.1 Types de contrôleurs :

Les régulateurs standards les plus utilisés dans l'industrie sont les régulateurs de la famille PID (proportionnel, intégral, dérivé), car ils permettent de régler à l'aide de ces trois paramètre les performances des régulations de procédés tel que : l'amortissement, le temps de réponse, le dépassement....[18]

L'adaptation de ce régulateur aux différents systèmes s'effectue par le réglage de ses coefficients propre à savoir les gains des trois actions : proportionnelle, intégrale et dérivée.

3.3 Principe de fonctionnement d'un régulateur PID :

Un contrôleur est un algorithme de calcul qui délivre un signal de commande à partir de la différence entre la consigne et la mesure. Il agit de trois manières.[18]

3.3.1 Action proportionnelle :

L'action est dite proportionnelle lorsque le signal de commande est proportionnel au signal d'erreur. Elle corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, elle permet de vaincre les grandes inerties du système. Afin de rendre le système plus rapide, on augmente le gain. Cependant, cet ajustement peut provoquer l'instabilité du système.

Le régulateur P est utilisé lorsqu'on désire régler un paramètre dont la précision n'est pas importante figure 3.2-a.

La fonction de transfert de l'action proportionnelle P est :

$$C(P) = K_p \tag{3.1}$$

3.3.2 Action intégral :

L'action est dite intégrale lorsque le signal de commande est proportionnel à l'intégrale du signal d'erreur,ceci complète l'action proportionnelle et permet d'éliminer l'erreur résiduelle en régime permanent. Afin de rendre le système plus dynamique (diminuer le temps de réponse), on diminue l'action intégrale mais, ceci provoque l'augmentation du déphasage ce qui engendre

l'instabilité en boucle fermée.

L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées figure 3.2-b.

La fonction de transfert de l'action intégrale I est :

$$C(P) = \frac{K_p}{T_i p} = \frac{K_i}{p} \tag{3.2}$$

3.3.3 Action dérivé :

L'action est dite dérivée lorsque le signal de commande est proportionnel à la dérivée du signal d'erreur. L'action dérivée, en compensant les inerties dues au temps mort, accélère la réponse du système et améliore la stabilité de la boucle de régulation, en permettant notamment un amortissement rapide des oscillations dues à l'apparition d'une perturbation ou à une variation subite de la consigne comme le montre la figure 3.2-c.

La fonction de transfert de l'action dérivée D est :

$$C(P) = K_d.T_d.P \tag{3.3}$$

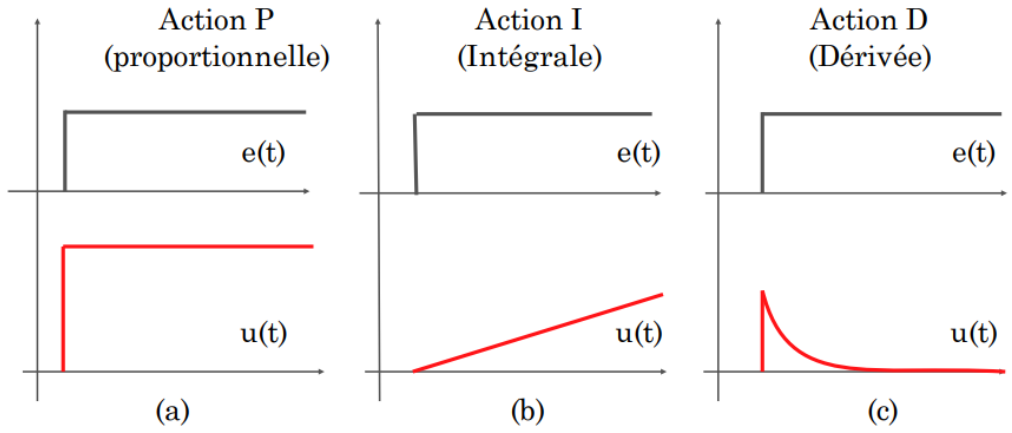

FIGURE 3.2 – Réponse des différentes actions

3.4 Différentes architectures des contrôleurs PID :

Dans un contrôleur PID, il existe plusieurs façons d'associer les paramètres P, I et D, en effet, le contrôleur PID peut avoir une architecture série, parallèle ou mixte.

3.4.1 Architecture parallèle :

L'architecture parallèle transforme le même signal de façon indépendante, puis somme les signaux.

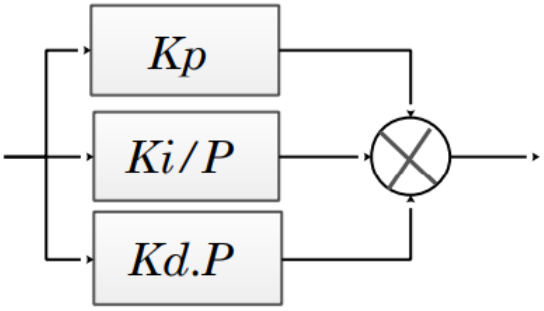

FIGURE 3.3 – Architecture parallèle[18]

La loi de commande de l'architecture parallèle est de la forme :

$$C_{parallle}(s) = K_p + \frac{K_i}{s} + K_d s \tag{3.4}$$

3.4.2 Architecture série :

L'architecture série applique les fonctions P, I et D en série, c'est-à-dire que le signal subit 3 transformation successives.

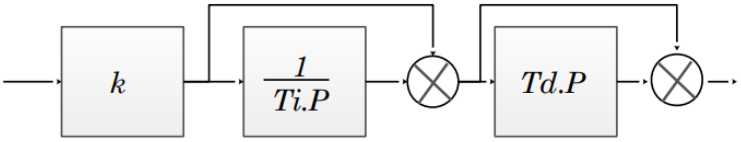

FIGURE 3.4 – Architecture série[18]

La loi de commande de l'architecture série est de la forme :

$$C_{srie}(s) = K(1 + \frac{1}{s\tau_i})(1 + s\tau_d) \tag{3.5}$$

3.4.3 Architecture mixte(série-parallèle) :

L'architecture standard effectue les actions I et D en parallèle et multiplie leur somme par p.

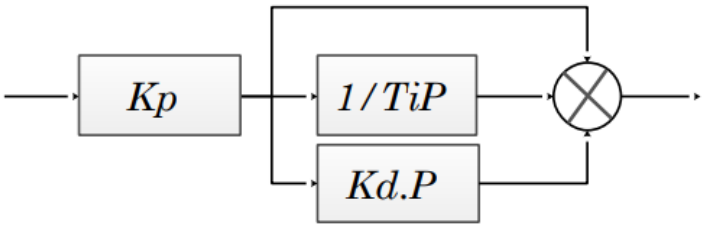

FIGURE 3.5 – Architecture série-parallèle[18]

La loi de commande est de la forme :

$$C(p) = K_p.e(p).[1 + K_i.\frac{1}{p} + K_d.p] \tag{3.6}$$

Il existe donc trois architecture différentes des mises en œuvre des régulateurs PID et elles sont presque mathématiquement équivalentes. Dans la plupart des cas (à l'exception par exemple de ki=0), il existe des formules pour transformer un régulateur d'une forme vers une autre forme.

3.4.4 La principale différence :

La principale différence entre ces différentes architectures concerne l'effet des coefficients de réglage sur le comportement du contrôleur. Ainsi l'architecture parallèle permet de découpler complètement les actions proportionnelle, intégrale et dérivée entre elles,alors que dans la forme série, une modification sur la valeur du coefficient Kp va modifier simultanément les actions proportionnelle, intégrale et dérivée.[18]

3.5 Réglage d'un PID :

Le réglage d'un PID consiste à déterminer les coefficients Kp ,Ki , Kd , afin d'obtenir une réponse adéquate du procédé et de la régulation. Il faut pour cela :

— dans le cas d'un fonctionnement en mode de régulation (consigne fixe) choisir des réglages permettant à la grandeur réglée de retourner dans un temps raisonnable à sa valeur de consigne

— dans le cas de fonctionnement de la boucle en mode d'asservissement (consigne variable), choisir des réglages permettant de limiter le ou les éventuels dépassements de la grandeur réglée

— la robustesse est sans doute le paramètre le plus important et délicat. On dit qu'un système est robuste si la régulation fonctionne toujours même si le modèle change un peu.

— la rapidité du régulateur dépend du temps de montée et du temps d'établissement du régime stationnaire

— le critère de précision est basé sur l'erreur statique

il existe différentes approches pour déterminer la combinaison optimale des réglages dont la méthode manuel, les méthodes classiques et les méthodes heuristiques .

3.5.1 La méthode manuelle

Si l'on possède suffisamment d'informations sur le processus à contrôler, il est possible de calculer les valeurs optimales de gain proportionnel, de gain d'intégrale et de gain de dérivée. Souvent, le processus est trop complexe, mais, en disposant de quelques données, notamment la vitesse à laquelle il répond aux corrections des erreurs, il est possible d'effectuer des réglages de niveau élémentaire.

Le réglage manuel advient en réglant le gain d'intégrale sur sa valeur maximale et le gain de dérivée sur zéro, et en augmentant le gain proportionnel jusqu'à ce que la boucle oscille à une amplitude constante (lorsque la réponse à une correction d'erreur est rapide, il est possible de recourir à un gain proportionnel plus élevé. Si la réponse est lente, il convient d'opter pour un gain proportionnel relativement faible). Ensuite, il faut diminuer le gain proportionnel de moitié et régler le gain d'intégrale de sorte à corriger toute déviation dans un délai acceptable. Enfin, il faut augmenter le gain de dérivée jusqu'à ce que le dépassement soit minimisé.

3.5.1.1 La méthode classique

Il y a plusieurs de méthodes qui permettent de calculer les paramètres du contrôleur PID, on peut citer par exemple la méthode de Cohen-Coon , la méthode de Halman , la méthode de Ziegler-Nichols ..., toutes ces méthode on l'avantage de facilité la démarche de synthèse du régulateur

PID, cependant elles restent assez complexe par fois si le système a une dynamique complexe.

3.5.2 Les méthodes heuristiques

D'autre part il existe aussi d'autres méthodes de synthèse basées principalement sur la recherche automatique des paramètres de ce régulateur. Ces méthodes utilisent principalement des algorithmes de recherche dits aussi algorithme d'optimisation. C'est cette approche que nous allons utiliser dans notre travail afin de synthétisé le régulateur PID que nous allons appliquer pour le système du TRMS.

3.6 Application du PID sur notre TRMS :

Dans notre cas nous avons opté pour une commande découplé avec quatre PID , deux lié à l'angle vertical (d'élévation) et les deux autres en charge de l'angle horizontal (azimuth) comme suit dans la figure (3.6) :

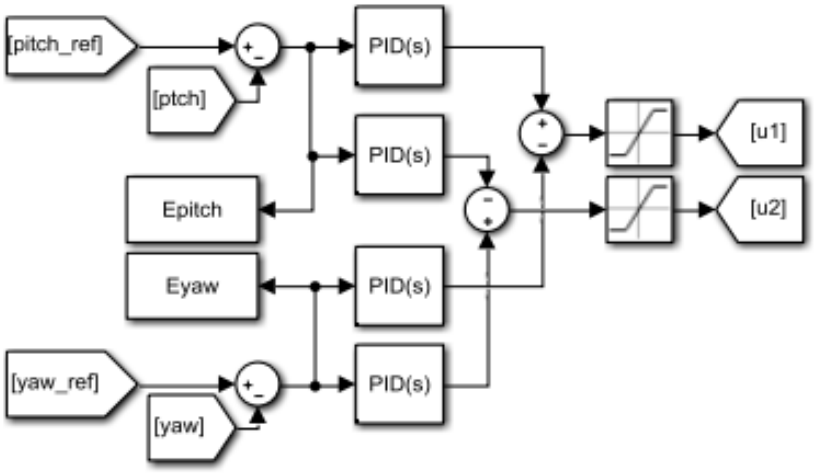

FIGURE 3.6 – Commande PID du TRMS

Comme on peut le voir sur la figure les PID sont découplé et en parallèles , deux PID lié à l'angle d'élévation et les deux autres à l'angle d'azimuth .
Afin de l'optimiser avec notre PSO on récupère les valeurs des erreurs qui sont envoyé vers le workspace pour qu'il calcul les meilleurs paramètres PID qui nous donne une erreur minimal.

On se qui concerne la configuration du PSO nous avons garder les mêmes paramètres que dans le chapitre 2 sauf pour la fonction objectif qui est devenu comme suit :

$$ISE = \sum_{i=1}^{N}(e1)^2 + \sum_{i=1}^{N}(e2)^2 \tag{3.7}$$

Avec N : le nombre d'échantillons récoltés durant la simulation.

Quant à la particule, elle est définie comme suit : [P1 I1 D1 P2 I2 D2 ... P4 I4 D4], cela permet de calculer les valeurs des PID en simultané pour avoir la meilleur optimisation possible.

3.6.1 Résultats de simulation pour la phase d'apprentissage :

Dans la figure ci-dessus , les PIDs sont numérotés du haut vers le bas (celui du haut est PID1 ... celui tous en bas et PID 4) , quant aux paramétres de simulation nous avons utilisé le od1(Euler) et un pas fixe et une consigne à suivre qui est carrée . Les resultats obtenus sont les suivants :

	P	I	D
PID1	49.9998	24.3058	35.1307
PID2	5.88147	-3.66964	4.50178
PID3	-49.9999	-16.3461	-14.7977
PID4	50	1.84389	19.6798

Nous allons dans la suite de ce chapitre utiliser ces valeurs optimales obtenu par l'algorithme PSO à la fin de la phase d'apprentissage pour le teste de robustesse avec des consignes différentes de la consigne choisie pour la phase d'apprentissage qui est carrée.(Voir la figure 3.8)

La figure suivante représente l'évolution du PSO durant la phase recherches(fitness) :

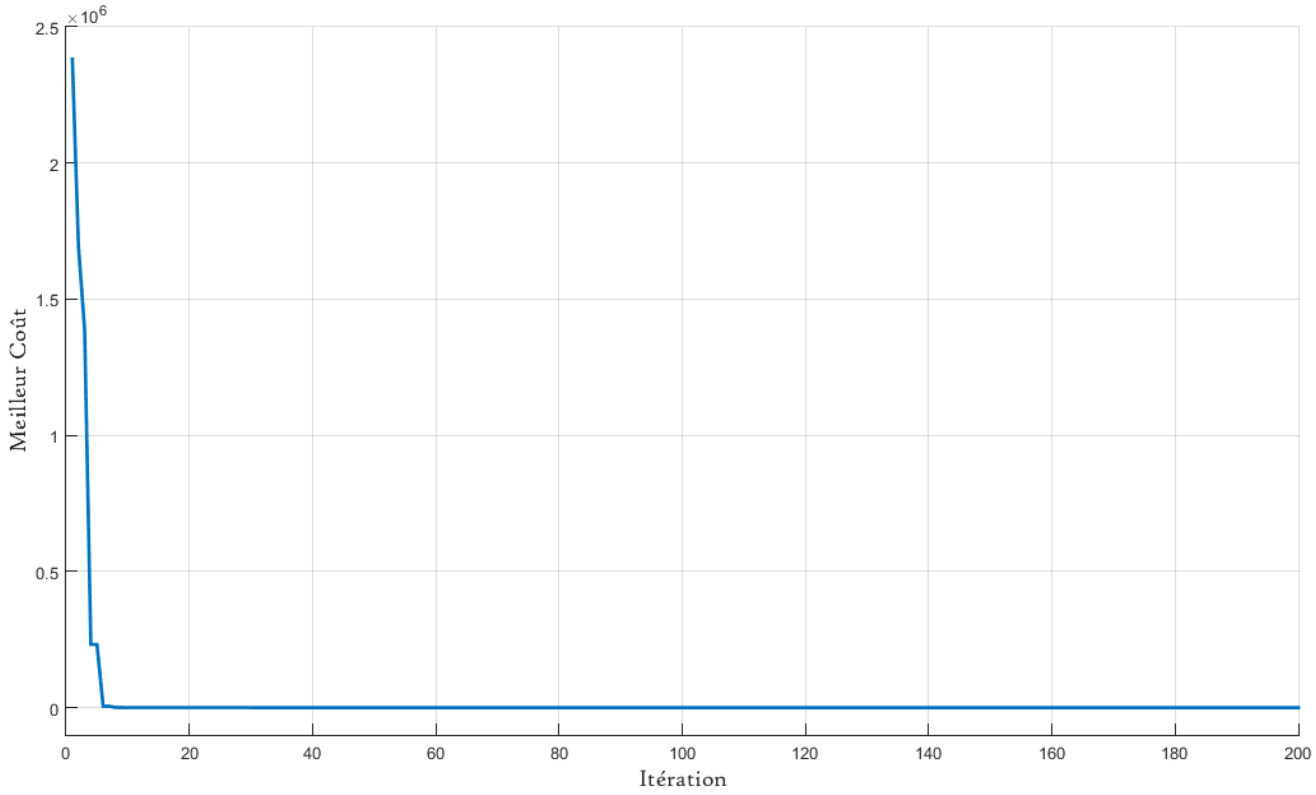

FIGURE 3.7 – L'evolution du PSO (fitness)

Comme on peut le voir sur la figure(3.7) la recherche est très rapide durant les 10 premières itérations, puis on remarque que la recherche se précise et se stabilise , donc nous pouvons on conclure que notre algorithme a pu atteindre la meilleure fitness qui par la suite va nous donner les paramètres des PID's optimisés pour notre système avec un ISE d'apprentissage de : 452.6921.

La figure suivante représente l'application des valeurs des PIDs optimisés.

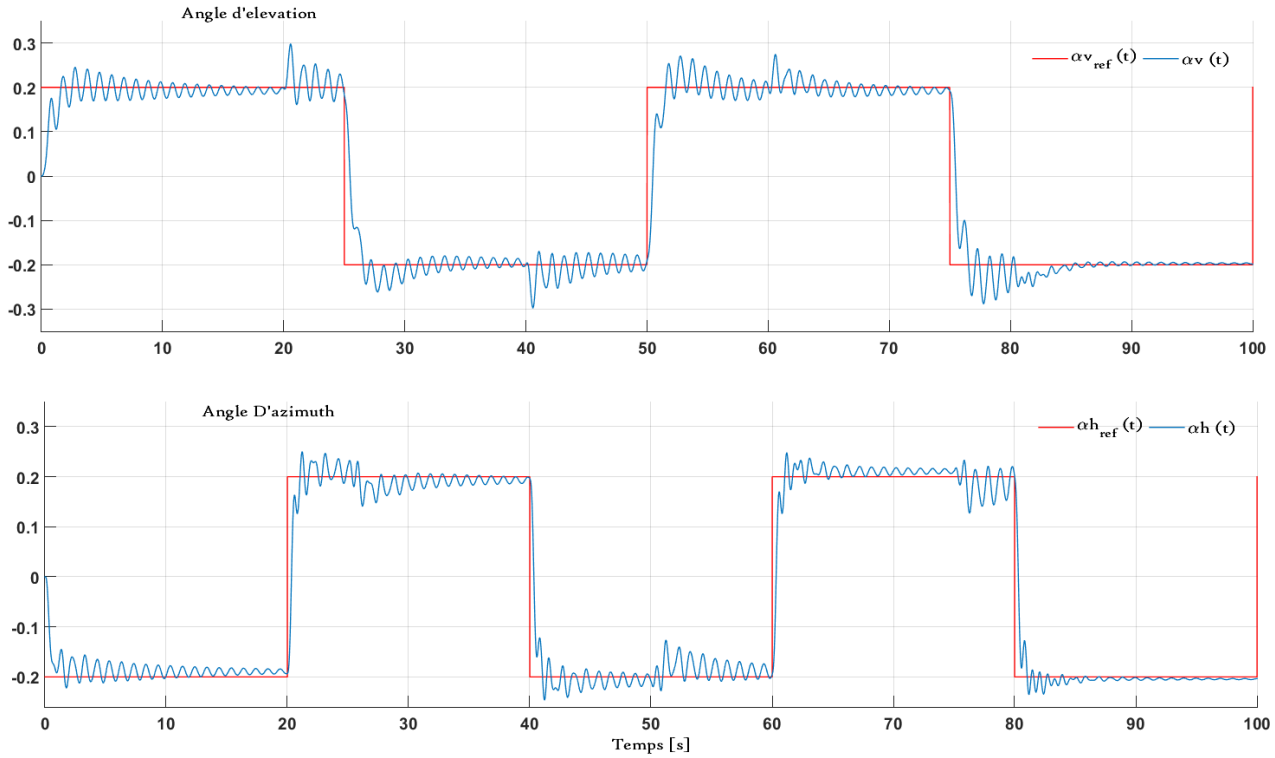

FIGURE 3.8 – Performances des meilleurs PIDs obtenus à la fin de la phase d'apprentissage

Comme nous pouvons le voir sur la figure 3.8 les deux angles (élévation et azimuth) suivent presque parfaitement la consigne carrée appliquée (signal de référence), mais nous pouvons aussi remarqué dés que l'un des deux angles essaie de se stabiliser il est aussitôt légèrement perturbé. se phénomène s'explique par la forte corrélation entre les deux angles de notre système. Tout du moins les deux angles restes très stable donc notre PID fonctionne sur se système.

3.6.2 Validation des résultats :

Afin de valider les résultats obtenus ci-dessus nous allons appliqués le PID sur notre modèle du TRMS en changeant la consigne carrée pour une sinusoïde qui est représenté par la figures suivante :

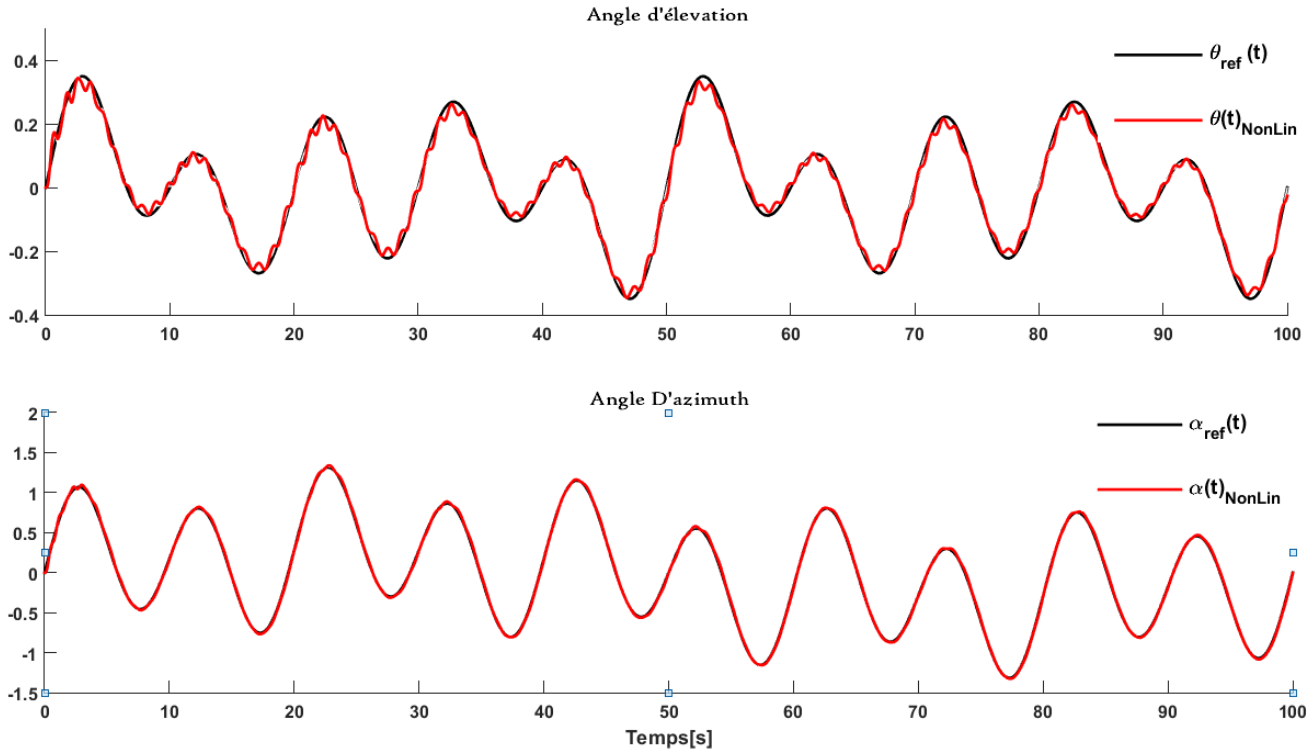

FIGURE 3.9 – Teste de robustesse

La figure ci-dessus montre la performance de régulation des quatre PIDs optimaux. On peut facilement constater qu'effectivement l'algorithme PSO a pu trouver les bonnes valeurs des différents paramètres.Avec ISE = 47,3199.

Les quatres PIDs optimaux travaillent en parfaite synchronisation pour commander le système du TRMS. Ce succès se traduit par la stabilisation des deux variables du système à savoir l'angle d'élévation et l'angle d'azimuth.

Ainsi les PIDs arrivent à stabilisés le système en agissant d'une manière optimale sur la force à appliquer. On peut en conclure que ce test confirme le bon fonctionnement du système de régulation et sa robustesse vis-à-vis des perturbation externes.

3.7 Conclusion :

Dans ce chapitre une étude sur les contrôleurs PIDs a été introduite pour explorer leur principe de fonctionnement. Dans un contrôleur PID il existe plusieurs façons d'associer les paramètres

P, I et D. En effet, le régulateur PID peut avoir une structure série, parallèle ou mixte. Il existe de nombreuses méthodes utilisées pour déterminer les paramètres du régulateur PID. Ensuite nous avons appliqués la commande PID sur notre système avec une recherche des paramètres optimisés grâce à l'algorithme PSO. Puis on à réaliser un dernier teste afin de valider les résultats obtenus.

Conclusion générale et perspectives

Pour pouvoir contrôler et commander des processus industriels, le développement d'outils de simulation est devenu une étape essentielle dans toute procédure de conception. Ces outils de simulations doivent permettre de couvrir la vaste problématique de commande allant des systèmes simples au plus compliqués.

Les industriels ont besoin de commander et contrôler le fonctionnement des composants, des appareils, des machines, et des processus. C'est dans ce contexte que s'inscrit ce travail de Master. Il a permis de lever certains verrous sur les techniques d'identifications des paramètres des régulateurs PID.

Ce mémoire a été consacré à la commande d'un simulateur TRMS optimiser par un algorithme évolutionnaire. Dans un premier temps, nous sommes amenés d'abord à comprendre les équations du système et à comprendre sa dynamique. La modélisation du système a été largement détaillée dans le chapitre 1.

Ensuite nous avons consacrés le chapitre 2 aux metaheuristiques et nous avons donnés un aperçu de l'algorithme génétique à titre d'exemple, avant d'étudier l'optimisation par essaim de particules ainsi que son principe de fonctionnement et les paramètres qui le définissent.

Finalement, dans le chapitre 3 nous avons introduit les contrôleurs PID et leurs principes de fonctionnements ainsi que les nombreuses méthodes utilisées pour déterminer les paramètres du régulateur PID. Et enfin on a identifié les paramètres du régulateur PID par la méthode PSO avant de finir ce chapitre sur une interprétation des résultats de la simulation.

Bibliographie

[1] A. CHARPENTIER, *Les hélicoptères d'hier à aujourd'hui,Patagon Diaporamas, 5KNA Productions, 2015.*

[2] J. TARIEL, *Histoire de l'hélicoptère, Revue XYZ.N°111-2^e metrimestre*, 2007.

[3] *https ://www.enpacapmatifou.com/Enpa2/Aero/MILITAIRE/LES HELICOPTERES.pdf*

[4] Feedback, *«Twin Rotor MIMO System Control Experiments», Manual 33-949S, edition 01/12/2006*

[5] A. MARTINI,*« Modélisation et commande de vol d'un hélicoptère drone soumis à unerafale de vent », Thèse de doctorat, université Paul Verlaine-Metz, France, 2008.*

[6] M.ILYAS, *al. « Control Law Design for Twin Rotor MIMO System with Nonlinear Control Strategy» Discrete Dynamics in Nature and Society, article en libre accès distribué sous licence Créative Commun, Pakistan,2016.*

[7]] http ://www.helicoptere-rc-24.com/blog/principe-de-vol-de-lhelicoptere/

[8]] Kennedy, J. ; Eberhart, R. (1995), *"Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks.IV. pp. 1942–1948. doi :10.1109/ICNN.1995.488968.*

[9]] Y. Shi and R. Eberhart, *"Parameter selection in particle swarm optimization," in Evolutionary Programming VII, pp. 591–600,Springer, 1998.*

[10]] P. Chalupa and J. Přikryl and , J. Novák, *"Modelling of Twin Rotor MIMO System" in25th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM 2014.*

[11]] A. Tastemirov, A. Lecchini-Visintini and R. M. Morales-Viviescas *Complete dynamic model of the Twin Rotor MIMO System (TRMS) withexperimental validation*

[12] http ://www.metaheuristics.org

[13]] I.H. Osman et G.Laporte *Metaheuristics : a bibliography. Annals of Operations Research 63, 513-623, 1996.*

[14]] T.stutzle *Local Search Algorithms for Combinatorial Problems –Analysis, Algorithms and New Applications. DISKI – Dissertationen zur Kunstliken Intelligenz. Infix, Sankt Augustin, Germany (1999).*

[15]] J. Holland. . *Outline for a logical theory of adaptive systems. Journal of the Association of Computing Machinery, 3, 1962.*

[16]] D.E Goldberg. *Genetic algorithms and walsh functions. part 1 and 2. Complex Systems, 3 :129–171, 1989.*

[17]] R. Dawkins. . *L'horloger aveugle, trad. B. Sigaud. Paris, Robert Laffont (coll.«La fontaine des sciences»), 15(5).1989.*

[18] S.Aberkan et T.moussaoui. *Commande d'un pendule inversé par un contrôleur PID optimisé par un algorithme évolutionnaire, Mémoire de master,université de Bejaia , 2020*

Résumé

La nature non linéaire des hélicoptères, et leurs fonctionnements en présence de plusieurs défauts et imperfection, provoque une baisse de performance. Toutes ses conséquences défavorables vont influencer la qualité et la stabilité de cette dernière.

L'objectif principal de ce travail est de créer un simulateur de commande de vol d'un hélicoptère de type TRMS et lui utiliser une commande PID afin d'assurer une bonne poursuite de la consigne.

Les résultats de la simulation obtenus par MATLAB/SIMULINK montrent l'efficacité de cette technique de commande.

Abstract

The nonlinear nature of helicopters, and their operation in the presence of several defects and imperfection, causes a drop in performance. All this negative consequences will influence the quality and stability of this latter.

The main objective of this work is to use and to create a simulator of a TRMS helicopter and use an PID control to assure a tracking of a setpoint, in spite of the presence of different disturbances. The simulation results obtained using MATLAB/Simulink show the effectiveness of this control strategy.