

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR  
MAUDE-JOSÉE BLONDIN

ALGORITHME CONTRAINT D'OPTIMISATION POUR LA SYNTONISATION  
DE CORRECTEURS AVEC SATURATION

SEPTEMBRE 2014

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

À celui qui voit loin,  
il n'est rien d'impossible.  
*-Henry Ford*

## Résumé

Plusieurs méthodes et règles ont été proposées pour l'ajustement de contrôleurs PID *classiques*. Par contre, elles sont fixes concernant le nombre de paramètres à ajuster et leur critère d'optimisation. Ainsi, des variations au sein de la structure de commande tel l'ajout d'un retour d'anti-saturation ne permettent plus l'utilisation de ces méthodes. Des approches complémentaires ont été développées pour l'ajustement de ce type de contrôleur. Par contre, dans certains cas, elles n'offrent pas une syntonisation optimale. En effet, c'est, entre autres, dû au fait que l'interaction entre les paramètres du contrôleur PID et du retour d'anti-saturation n'est pas considérée. S'intéresser à cette dépendance entre les paramètres de la structure de contrôle amène à représenter leur ajustement comme un problème d'optimisation. Dans la classe des métaheuristiques, les algorithmes à colonies de fourmis (ACO) permettent et répondent au besoin d'une syntonisation globale des paramètres de contrôleurs tout en offrant plusieurs avantages. Cependant, des points faibles tel le temps de calculs les rendent moins intéressants. D'un autre côté, dans les méthodes d'optimisation aléatoire, la méthode Nelder-Mead (NM) permet un ajustement adéquat de diverses structures de contrôle pour des conditions de démarrage appropriées. Comme la grandeur des termes du contrôleur peut être inconnue, il est difficile d'initialiser correctement NM. En fait, selon les résultats de simulation, NM appliqué seul n'est pas efficace. De plus, il peut converger vers des valeurs de paramètres négatives que le système ne peut accepter pour un bon fonctionnement. Delà, l'intérêt de développer une nouvelle méthode

**contrainte** et **globale** de réglage de structure de commande avec retour d'anti-saturation à base des ACO et de NM pour ne garder que les forces de chacun. Les résultats de simulation montrent que l'algorithme contraint proposé permet une syntonisation efficace de différentes structures de contrôle avec retour d'anti-saturation pour un temps de convergence presque 1.67 à 5.7 fois plus rapide que ACO seul. De plus, le test de positivité pour contraindre l'algorithme NM qui considère l'interaction entre les paramètres est jusqu'à 2.5 fois plus rapide que de simplement fixer la valeur du paramètre à sa contrainte s'il la dépasse. Qui plus est, le critère de performance de l'algorithme contraint proposé est facilement modifiable et permet d'imposer des contraintes souples et rigides au système. Ces avantages combinés à son nombre de paramètres à syntoniser flexible viennent, dans certains cas, remplacer la lourdeur d'une syntonisation empirique des structures de commande avec anti-saturation tout en ouvrant la voie sur l'optimisation de divers systèmes.

## Remerciements

Tout d'abord, je remercie les membres évaluateurs de mon mémoire, professeur Loïc Boulon, professeur Hicham Chaoui et bien évidemment, mon directeur de recherche Pierre Sicard.

Spécialement, je tiens à exprimer toute ma reconnaissance à Pierre Sicard. Son sens de l'éthique combiné à sa rigueur scientifique me sert de modèle. Je le remercie pour son dévouement et sa générosité. Il m'a toujours encouragé dans mon travail en m'orientant dans l'écriture de mes articles scientifiques et en m'appuyant dans les nombreuses demandes de bourse d'études. Tout au long de mon travail, il m'a conseillé et aidé, ce qui m'a outillé pour grandir dans le merveilleux monde de la recherche universitaire. Je le remercie pour sa confiance et les opportunités qu'il m'a offertes. Merci Pierre!

Je veux également remercier le professeur Adel Omar Dahmane pour ses précieux conseils et son support moral. D'une disponibilité hors pair, il était présent pour m'écouter et m'encourager. Il a fortement contribué à rendre plus agréable et fluide mon parcours académique à la maîtrise.

Merci aux Fonds de recherche du Québec – Nature et technologies (FRQNT) pour la bourse de maîtrise, au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) pour la bourse Alexander-Graham Bell (maîtrise), à la fondation C.E.U pour la bourse d'excellence, à l'Association des diplômés de l'UQTR pour la bourse des diplômés

et aux décanats des études supérieures pour les bourses d'aide à la diffusion. Votre support financier m'a permis l'obtention de ce diplôme.

Plus personnellement, je tiens à remercier mon père Serge Blondin pour sa présence et son support financier. Il a été un pilier important dans l'atteinte de mes objectifs et de mes rêves. Merci pour ta patience et ton dévouement auprès de mes trois enfants. Je remercie également ma mère Marie-Claire Caya pour son aide et ses encouragements qu'elle peut m'offrir.

Un merci du fond du cœur à Lucille et à Laurianne alias belle Laurie qui ont su prendre soin de mes enfants lors de mes périodes d'études et lors de mes voyages pour assister à des conférences. J'ai pu me concentrer à 100% sur mon travail, car je savais que mes enfants étaient en sécurité.

Finalement, je remercie de tout mon être mes enfants Andrea, Rocco et Rosie. Merci d'apporter un équilibre à ma vie. ☺ Je vous aime!

# Table des matières

Résumé.....	iii
Remerciements.....	v
Table des matières.....	vii
Liste des tableaux.....	x
Liste des figures .....	xi
Chapitre 1 - Introduction.....	1
1.1 Problématique.....	2
1.2 Objectif.....	7
1.3 Méthodologie.....	7
1.4 Structure du mémoire .....	9
Chapitre 2 - Description des contrôleurs et du système.....	10
2.1 Contrôleur PID .....	10
2.1.1 Description d'un contrôleur PID.....	10
2.1.2 Mesures de performance de la réponse transitoire.....	12
2.1.3 Retour d'anti-saturation .....	15
2.2 Système à l'étude.....	18



2.3 Conclusion.....	22
Chapitre 3 - Algorithme d'optimisation contraint à base des ACO et de NM	
pour l'ajustement de PID avec retour d'anti-saturation.....	23
3.1 Les algorithmes d'optimisation à colonies de fourmis.....	23
3.1.1 Les métaheuristiques.....	23
3.1.2 Source d'inspiration.....	25
3.1.3 Description des ACO.....	25
3.1.4 Application spécifique à la syntonisation de contrôleurs .....	28
3.2 La méthode Nelder-Mead.....	30
3.2.1 Description de NM.....	31
3.2.2 Application spécifique à l'ajustement de contrôleurs.....	33
3.3 Fonctionnement de l'algorithme contraint proposé.....	35
3.4 Conclusion.....	38
Chapitre 4 - Résultats de simulation .....	
4.1 Ajustement des structures de commande par ACO.....	39
4.1.1 Structure de commande simple.....	40
4.1.2 Structures de commande complexes.....	45
4.1.3 Conclusion .....	51
4.2 Ajustement de structures de commande par NM .....	51
4.2.1 Ajustement de la structure simple - points de départ variés .....	52

4.2.2	Ajustement de la structure simple - points de départ ACO .....	54
4.2.3	Conclusion .....	57
4.3	Ajustement de structure de commande – algorithme contraint proposé .....	58
4.3.1	Ajustement de structure simple.....	58
4.3.2	Ajustement de structures complexes.....	62
Chapitre 5 - Conclusion générale.....		70
Références.....		73
Annexe A – Articles scientifiques .....		76

## Liste des tableaux

Tableau 2-1	Effet typique de l'augmentation des paramètres d'un contrôleur PID.....	13
Tableau 4-1	Mesures de performance pour la structure de contrôle simple.....	42
Tableau 4-2	Mesures de performance pour le deuxième scénario de simulation. ....	47
Tableau 4-3	Mesures de performance pour le troisième scénario de simulation. ....	50
Tableau 4-4	Résultats de simulation selon différents points de départ de recherche NM. ....	53
Tableau 4-5	Résultats de simulation selon différents points de départ de recherche NM (suite). ....	54
Tableau 4-6	Valeurs des paramètres selon trois algorithmes de syntonisation. ....	61
Tableau 4-7	Valeurs des paramètres des trois algorithmes de syntonisation au résultat intermédiaire $n = 25$ . ....	64

## Liste des figures

Figure 2-1	Structure mixte d'un contrôleur PID.....	11
Figure 2-2	Contrôleur PID <i>classique</i> avec saturateur entre le signal de commande et le procédé (système). .....	15
Figure 2-3	Contrôleur PID avec retour d'anti-saturation ( <i>back-caculation</i> ). .....	16
Figure 2-4	Système de positionnement d'une meule. ....	18
Figure 2-5	Modèle du système pour l'application de l'algorithme contraint proposé pour le réglage de paramètres de contrôleurs. ....	19
Figure 2-6	Contrôleur PID <sub>F</sub> de rétroaction de position. ....	20
Figure 2-7	Contrôleur pondéré PID <sub>F</sub> avec un retour d'anti-saturation sur le courant. ....	21
Figure 2-8	Contrôleur pondéré PID <sub>F</sub> avec un retour d'anti-saturation sur la vitesse du moteur.....	21
Figure 2-9	Contrôleur pondéré PID <sub>F</sub> avec retour d'anti-saturation sur la vitesse du moteur ainsi que sur le courant.....	22
Figure 3-1	Processus de diversification. ....	24
Figure 3-2	Processus d'intensification. ....	24
Figure 3-3	Structure des ACO.....	27
Figure 3-4	Graphe d'optimisation pour les différentes structures de commande. ....	28
Figure 3-5	Structure de la méthode NM. ....	33
Figure 3-6	Graphe décisionnel de la procédure NM avec le test de contrainte de positivité. ....	35
Figure 3-7	Structure de l'algorithme contraint proposé.....	37
Figure 4-1	Structure de l'environnement de simulation des itérations des ACO.....	39

Figure 4-2	Réponse à un échelon d'entrée pour le contrôleur $PID_F$ avec retour d'anti-saturation – réglage par ACO. ....	41
Figure 4-3	Réponse à un échelon d'entrée pour le contrôleur $PID_F$ avec retour d'anti-saturation – réglage par GA. ....	43
Figure 4-4	Agrandissement de la réponse à l'échelon d'entrée pour le contrôleur $PID_F$ avec retour d'anti-saturation selon différents $Kt$ ( $\sigma = 2$ et $\beta = 0$ ) – réglage par ACO. ....	44
Figure 4-5	Agrandissement de la réponse à l'échelon de perturbation pour le contrôleur $PID_F$ avec retour d'anti-saturation selon différents $Kt$ ( $\sigma = 2$ et $\beta = 0$ ) – réglage par ACO. ....	45
Figure 4-6	Réponse à l'échelon d'entrée et de perturbation et agrandissement de la réponse à l'échelon d'entrée pour le contrôleur pondéré $PID_F$ avec retour d'anti-saturation sur le courant de référence ( $\sigma = 2$ , $Kt = 0.75$ ) - ACO vs GA. ....	46
Figure 4-7	Agrandissement de la réponse à la perturbation de la Figure 4.6. ....	47
Figure 4-8	Agrandissement de la réponse entrée échelon et à la perturbation pour le contrôleur pondéré $PID_F$ avec limitation de vitesse du moteur ( $\sigma = 2$ , $Kt = 0.075$ ) - ACO vs GA. ....	49
Figure 4-9	Agrandissement de la réponse à la perturbation de la Figure 4.8. ....	50
Figure 4-10	Vitesse du moteur avec et sans compensation de saturation. ....	50
Figure 4-11	Réponse à l'échelon pour le contrôleur $PID_F$ avec retour d'anti-saturation ( $\sigma = 1$ ) en fonction de conditions initiales (C.I.) variées- méthode NM de base. ....	53
Figure 4-12	Temps moyen de calculs pour différentes stratégies d'optimisation. ....	54
Figure 4-13	Réponse à l'échelon pour la syntonisation de la structure simple – ACO réduit ( $\beta = 0$ , $\sigma = 1$ et $Kt = 0$ ). ....	55
Figure 4-14	Environnement de simulation de l'algorithme ACO réduit-NM. ....	56
Figure 4-15	Réponse à l'échelon pour ACO réduit-NM selon deux pas de discrétisation. ....	57
Figure 4-16	Temps moyen de calcul pour NM contraint et ACO. ....	60
Figure 4-17	Temps de simulation pour les trois algorithmes de syntonisation. ....	62

Figure 4-18	Réponse à l'échelon pour le contrôleur $PID_F$ avec retour d'anti-saturation pour le positionnement de la table.....	62
Figure 4-19	Agrandissement de la réponse à l'échelon pour le contrôleur pondéré $PID_F$ avec un retour d'anti-saturation sur le courant. ....	65
Figure 4-20	Agrandissement de la réponse à la perturbation contrôleur pondéré $PID_F$ avec un retour d'anti-saturation sur le courant. ....	65
Figure 4-21	Réponse à l'échelon et à la perturbation avec agrandissement de la réponse transitoire pour le contrôleur pondéré $PID_F$ avec retour d'anti-saturation sur la vitesse du moteur. ....	67
Figure 4-22	Agrandissement de la réponse à la perturbation pour le contrôleur pondéré $PID_F$ avec retour d'anti-saturation sur la vitesse du moteur.....	67
Figure 4-23	Réponse à l'échelon et agrandissement de la réponse à la perturbation pour le contrôleur pondéré $PID_F$ avec un retour d'anti-saturation sur la vitesse du moteur et sur le courant de référence. ....	68

## Chapitre 1 - Introduction

L'utilisation de contrôleurs touche divers domaines de sciences appliquées tels que la mécanique, la pneumatique et l'électronique. Leur gamme d'application est très étendue; du contrôle dans les lecteurs CD et DVD, elle va jusqu'au contrôle de température dans les réacteurs chimiques. Par exemple, en industrie, les applications de positionnement de pièces sont nombreuses telle la gestion de déplacement d'une meule pour ses opérations. Un opérateur peut la déplacer manuellement avec précision pour assurer un meulage adéquat ou il est possible d'automatiser son positionnement. Pour ce faire, un moteur est relié à la meule qui nécessite une excitation appropriée pour répondre correctement aux changements de position. Par conséquent, son alimentation est modulée par un algorithme provenant d'un contrôleur qui délivre un signal de commande en fonction de la différence (erreur) entre sa consigne (position désirée) et sa mesure (position actuelle). Dans le même ordre d'idées, la stabilisation automatique de la vitesse dans les véhicules automobiles se fait via un « régulateur de vitesse ». La vitesse fixée par le conducteur est maintenue constante par un système qui contrôle la valve de l'accélérateur. En fait, le signal reçu par la valve provient d'un contrôleur. En pratique, le contrôleur PID, « Proportionnel-Intégral-Dérivé », est l'un des plus populaires dans les systèmes de contrôle, ce qui en fait le contrôleur de base [AST06], [OGA10].

Plusieurs variantes au contrôleur PID existent. En effet, selon les exigences de l'application, il est possible de conserver seulement la partie proportionnelle (P), ou former

une combinaison proportionnelle-dérivée (PD), proportionnelle-intégrale (PI) et autres architectures. Quelle que soit la structure du régulateur, ses paramètres doivent être ajustés avec minutie pour délivrer un signal de commande approprié. Leur réglage constitue donc une étape importante dans la conception globale d'un système, car il a un impact direct et significatif sur la qualité du comportement du système.

### 1.1 Problématique

Plusieurs méthodes et règles efficaces ont été proposées spécifiquement pour le réglage du contrôleur PID *classique* tels que Ziegler-Nichols, ajustement par placement de pôles, *lambda tuning* etc. [PAD12] Leur critère de performance est précis et fixe, par exemple, l'intégrale de la valeur absolue de l'erreur (IAE). Il est donc impossible de cibler une caractéristique indésirable dans la réponse du système afin de la pénaliser davantage, ce qui pourtant, en pratique, serait un avantage. Par exemple, dans les opérations de meulage, l'erreur de dépassement de positionnement peut endommager les pièces. Il serait donc utile de pouvoir diminuer/éliminer cette caractéristique néfaste. D'autres parts, ces méthodes classiques ajustent un nombre fixe de paramètres. Ainsi, si la structure de commande varie et contient plus de paramètres à syntoniser, par exemple, dans le cas d'une structure PID avec un coefficient de retour d'anti-saturation, ces techniques ne sont plus performantes [BOH95].

En effet, dû au terme supplémentaire du retour d'anti-saturation, les méthodes classiques ne peuvent s'appliquer pour un ajustement **simultané** de l'ensemble des paramètres du système de contrôle. Des méthodes complémentaires [KUM12] basées sur un ajustement **séquentiel** (en deux phases) ont été développées, c'est-à-dire; (i) la syntonisation des paramètres PID et (ii) réglage du coefficient du retour d'anti-saturation.



Dans [BOH95] et [PAD12], l'application de cette approche a soulevé l'existence d'une relation entre les coefficients du PID et du retour d'anti-saturation. Cette interaction s'observe sur la qualité de la réponse du système qui peut être très lente. Ainsi, des ajustements empiriques peuvent être nécessaires pour tenter d'améliorer les performances du contrôleur.

De ce fait, le développement d'une approche **globale** pour régler le système PID et son retour d'anti-saturation est pertinent. De cette façon, l'effet d'interaction entre les termes du PID et son retour d'anti-saturation sera considéré en globalité dans le but d'obtenir des performances satisfaisantes du procédé.

L'ajustement simultané des paramètres des différentes fonctions du correcteur peut être vu comme un problème d'optimisation auquel est associé une fonction coût, souvent dite fonction objective. En fait, cette fonction qui tend à être minimiser par rapport à tous les paramètres est le critère de performance de la réponse du système. Puisqu'aucun algorithme ne permet de localiser l'optimum global (la meilleure solution possible) à coup sûr et en un nombre fini de calculs, la syntonisation de contrôleur s'insère dans la définition de problème d'optimisation difficile. Pour résoudre ce type de problème, les métaheuristiques, comprenant notamment les algorithmes d'optimisation par essais particuliers (PSO) [WAI11], les algorithmes génétiques (GA) [HAU98], les algorithmes à colonies de fourmis (ACO) [DOR96] et la programmation évolutive (EP) [LIE01], deviennent très intéressantes et appropriées. L'existence de nombreuses métaheuristiques amène la difficulté de choisir une méthode efficace pour s'approcher le plus près possible de la solution optimale en un temps de calcul raisonnable. Dans [SHA11] et [NAG10], ces méthodes ont été appliquées à pour l'ajustement de contrôleurs PID *classiques*. Les performances en termes de qualité de

la réponse du système et du temps de convergence sont comparées. ACO et GA représentent les deux méthodes avec le plus grand potentiel en termes de performance. Cependant, ACO contient moins de termes à paramétrer que GA simplifiant donc son utilisation.

D'ailleurs, d'après [CHI12], [VAR04] et [HSI04], l'ajustement de contrôleurs PID par ACO [DOR04] possède plusieurs avantages. Tout d'abord, il a été démontré que le critère de performance est variable et ne nécessite pas de solution analytique, ce qui permet de l'adapter en fonction de l'application. Par exemple, certaines caractéristiques de la réponse du système peuvent être spécifiquement pénalisées (comme le dépassement de la consigne). Également, l'approche basée sur les ACO est indépendante de la structure de commande. Cette méthode permet donc un nombre variable de coefficients à ajuster. Les ACO pourraient donc syntoniser des structures de commande avec retour d'anti-saturation, ce qui à ma connaissance n'a pas été effectué jusqu'à présent. En revanche, l'augmentation du nombre de paramètres peut conduire à un temps excessif de calculs afin que l'algorithme converge vers une solution souhaitée. Autre point faible, les métaheuristiques telles ACO sont des stratégies de haut niveau; leur degré de précision (défini comme la différence entre la solution trouvée par les ACO et la solution optimale du problème) peut varier et dépend fortement des paramètres de l'algorithme. De plus, l'utilisation de cette approche nécessite une détermination préalable de l'espace de recherche. Comme la grandeur de la valeur des termes du contrôleur peut être inconnue, il peut être délicat de le déterminer adéquatement. Cette détermination comprend l'intervalle de recherche et son pas de discrétisation. Tous les deux ont un effet significatif sur la qualité de la réponse du système et sur le temps de convergence de l'algorithme. En effet, l'intervalle de l'espace de recherche doit être

suffisamment grand pour augmenter la probabilité d'inclure la région de la solution optimale. Par contre, plus il est vaste, plus l'algorithme prendra du temps à converger. Quant au pas de discrétisation, il doit être suffisamment fin pour s'approcher de la réponse optimale, mais l'augmentation du nombre d'unités discrètes alourdit également le temps de calcul.

Par ailleurs, une autre famille de méthodes d'optimisation, soit l'optimisation aléatoire, peut s'appliquer à la syntonisation de structure de commande puisque leur recherche se base uniquement sur l'évaluation d'une fonction objective quelconque sans l'évaluation de dérivée. Par exemple, dans [TAJ11], l'algorithme Nelder-Mead (NM) [LAG98] syntonise un régulateur PID pour un système d'actionnement électrohydraulique. Dans ce cas-ci, le critère de performance est la réponse du système. L'ajustement est donc adaptable à une application précise rendant possible la diminution de l'effet de certaines caractéristiques indésirables dans la réponse du système. Puisque cette méthode utilise le concept de simplexe (forme géométrique spéciale de  $n + 1$  variables) qui se déplace dans un espace continu de recherche, son nombre de variables à optimiser est flexible. De plus, aucune discrétisation de l'espace de recherche n'est nécessaire. Ces caractéristiques (avantages) font qu'elle pourrait syntoniser globalement l'ensemble des paramètres du contrôleur PID avec son retour d'anti-saturation, ce qui n'a pas été considéré dans [LAG98]. Cependant, entre autres dans [LUE03], une relation importante de cette technique d'optimisation est soulevée. Le temps de convergence et la qualité de l'ensemble de la solution trouvée par l'algorithme NM dépendent fortement du point initial de recherche de l'algorithme. Pour un point de départ de l'algorithme convenablement déterminé, sa convergence vers la solution souhaitée est rapide. Par contre, comme expliqué, la grandeur des termes d'une structure de

commande peut être inconnue; il est donc difficile de fournir un ensemble pertinent de paramètres initiaux pour démarrer une recherche efficace. Une phase d'exploration de l'espace de solutions possibles est nécessaire pour augmenter les probabilités d'attraper la région de l'optimum global. Dans [LUE03], cette phase passe par le lancement simultané et successif de la procédure NM défini entre autres par un algorithme probabiliste. Par contre, il n'y a aucune garantie sur la rapidité de convergence de l'algorithme, ce qui rend son application moins avantageuse. Point important; la méthode NM est non contrainte. Pour la syntonisation d'un régulateur, une plage limitée de valeur de paramètres permet d'obtenir la stabilité du système et des considérations pratiques imposent ainsi la positivité de certains paramètres. Ces considérations pratiques permettent notamment d'obtenir une meilleure robustesse. Pour un système de positionnement typique, le gain proportionnel du correcteur PID admettra une valeur préférentielle, soit positive ou négative, selon la fonction qu'il réalise. Donc, certains ensembles de paramètres ne sont pas acceptables. Il serait possible de rejeter les ensembles solution non permis comme dans [TAJ11]. Toutefois, cette procédure n'est pas optimale. Limiter la valeur d'un coefficient proche de la contrainte peut être appliqué, inspiré de [LUE03]. Par contre, cette méthode ne tiendrait pas compte de l'interaction et l'interdépendance entre les divers paramètres.

À ma connaissance, aucune technique de syntonisation globale de plusieurs structures de commande avec retour d'anti-saturation n'a été développée possédant simultanément les avantages suivants :

1. Critère de performance flexible;
2. Imposition de contraintes rigides et souples au système via le critère de performance;

3. Nombre de termes à ajuster flexible;
4. Convergence rapide;
5. Méthode robuste et stable;
6. Applicable à différentes structures de commande avec retour d'anti-saturation.

## 1.2 Objectif

L'objectif de ce travail est de développer une nouvelle technique **globale et simultanée** à base des ACO et de la méthode NM pour l'ajustement de l'**ensemble** des paramètres de diverses structures de contrôle comprenant un retour d'anti-saturation. Ainsi, le nombre de paramètres à ajuster doit être facilement altérable. Son critère de performance doit être flexible/modulable pour imposer certaines caractéristiques à la réponse système et imposer le respect de certaines limites internes du système. Le temps de convergence doit être rapide. Cette méthode pourrait donc être considérée comme une boîte blanche où il faut préciser le nombre de paramètres à ajuster ainsi qu'un critère d'optimisation précis. Le but du développement d'une telle technique est de permettre une syntonisation plus simple et autonome de **différentes structures de contrôle** de plusieurs domaines et subséquentement, pour certains cas, de remplacer la lourdeur d'un ajustement empirique, comme ça peut être le cas pour l'ajustement de structure de commande avec retour d'anti-saturation.

## 1.3 Méthodologie

Le développement du nouvel algorithme se base sur l'étude des ACO et de la méthode NM. Les étapes suivantes permettront d'atteindre l'objectif de ce travail:

1. Programmer les ACO et les valider avec un problème d'optimisation connu, soit le problème du voyageur de commerce;
2. Analyser le système à l'étude et son environnement de modélisation et simulation (Matlab®/Simulink™);
3. Adapter les ACO pour l'ajustement de structures de commande;
4. Ajuster globalement par les ACO les paramètres de différentes structures de contrôle complexes avec retour de saturation et valider par comparaison avec GA;
5. Appliquer seule la méthode NM pour le réglage d'une structure de commande simple avec retour d'anti-saturation;
6. Développer le nouvel algorithme et proposer une nouvelle procédure pour le contraindre;
7. Valider l'algorithme contraint proposé;
  - i. Convergence vers des termes positifs seulement;
  - ii. Temps de convergence;
  - iii. Flexibilité du critère de performance (imposer des contraintes rigides et souples au système);
  - iv. Flexibilité du nombre de paramètres à ajuster;
  - v. Application à différentes structures de contrôle;
  - vi. Qualité de la réponse système.

## 1.4 Structure du mémoire

Le présent travail se divise en quatre sections. Le contrôleur PID *classique*, le principe du retour d'anti-saturation, la description du système et les structures de commandes utilisées seront détaillés au Chapitre 2. Le Chapitre 3 comprend la description des ACO et de la méthode NM; leur application spécifique au réglage de contrôleur sera exposée ainsi que le nouvel algorithme contraint proposé. Au Chapitre 4, les résultats de différents scénarios de simulation seront présentés. Finalement, un retour sur la problématique, sur sa résolution et sur la contribution scientifique de ce projet concluront le travail au Chapitre 5.

## Chapitre 2 - Description des contrôleurs et du système

Ce chapitre présente le fonctionnement du contrôleur PID *classique* avec ses diverses mesures de performance. Le fonctionnement et l'impact du retour d'anti-saturation dans une structure de commande sont détaillés. Le système utilisé pour développer et valider l'algorithme contraint proposé y est décrit ainsi que ses quatre structures de contrôle.

### 2.1 Contrôleur PID

Tel que mentionné, le contrôleur PID de son acronyme « Proportionnel-Intégral-Dérivé » est l'un des plus utilisés dans l'industrie dû à la simplicité relative de sa structure [AST06],[OGA10]. Généralement, il est utilisé pour réguler le comportement de différents systèmes dynamiques tels que le contrôle d'une valve électromagnétique ou d'un niveau de liquide dans un réservoir. Malgré sa structure simple, trouver un bon ensemble de paramètres pour atteindre les performances désirées peut devenir complexe puisque chacun d'eux influence différemment la réponse du système. Par conséquent, dans cette section, la structure du contrôleur PID sera décrite plus en détails et les critères de performance seront expliqués.

#### 2.1.1 Description d'un contrôleur PID

Un régulateur PID est une structure de calculs dont le rôle est d'envoyer un signal de commande au système tout en respectant certaines exigences tant au niveau des limitations



du système qu'à des caractéristiques spécifiques de sa réponse. Les calculs pour délivrer le signal commande,  $I^*$ , sont basés sur la différence entre sa consigne,  $P_T^*$ , (position/état souhaité) et sa mesure,  $P_T$ , (position/état actuel du système). Cette différence est appelée « erreur »,  $e(s)$ . La Figure 2-1 présente l'architecture mixte d'un contrôleur PID. Il existe plusieurs variantes d'architectures : (i) parallèle; (ii) en série; et (iii) mixte et plusieurs combinaisons possibles telles que PI, IP, PD, etc.

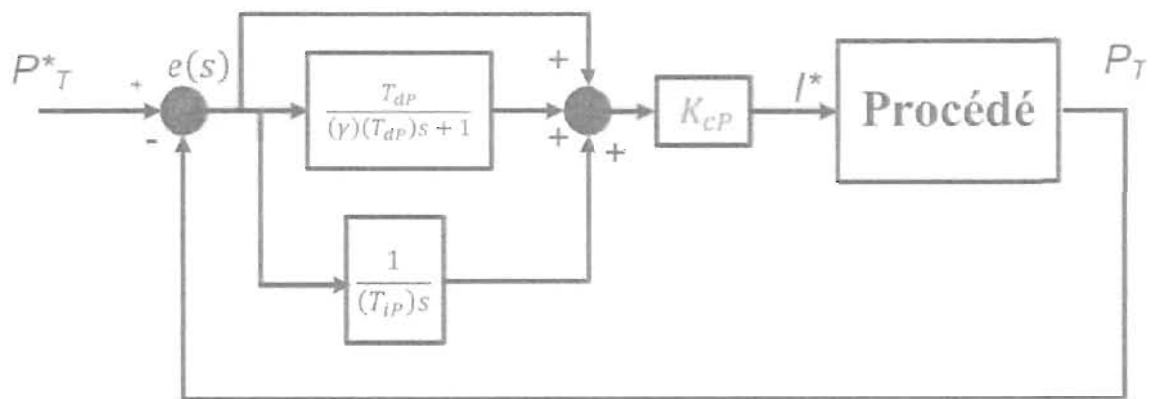


Figure 2-1 Structure mixte d'un contrôleur PID.

où  $K_{cP}$  est le gain proportionnel,  $T_{dP}$  est le temps de dérivation,  $\gamma T_{dP}$  est la constante de temps du filtre du terme de dérivatif,  $T_{iP}$  est le temps de rappel.

Selon la Figure 2-1, une combinaison de 3 sous-calculs (opérations) est faite :

1. Opération intégrale I:  $e(s)$  est intégrée et divisée par un temps de rappel  $T_{iP}$ ;
2. Opération dérivée D:  $e(s)$  est dérivée et multipliée par un temps de dérivation  $T_{dP}$ ;
3. Opération proportionnelle P :  $e(s)$ .

La somme des trois termes est multipliée par un gain  $K_{cP}$ .

Chaque paramètre du PID joue un rôle distinct sur la réponse du système. La fonction de transfert de la structure de la Figure 2-2 (sans le filtre sur l'action dérivée) est définie :

$$G_c(s) = K_{cP} \left( 1 + \frac{1}{T_{iP}s} + T_{dP}s \right) \quad (2-1)$$

Les valeurs de  $K_{cP}$ ,  $T_{iP}$  et  $T_{dP}$  doivent être déterminées afin de permettre au système d'atteindre les performances recherchées. En fait, la performance d'un contrôleur repose en grande partie sur son ajustement. Comme plusieurs techniques pour paramétrer les régulateurs existent, il s'agit de sélectionner celle qui est la plus appropriée pour le problème à résoudre.

### 2.1.2 Mesures de performance de la réponse transitoire

Pour analyser la performance de la réponse du système, les caractéristiques suivantes sont typiquement mesurées [OGA10];

1.  $t_r$  est le temps de montée; temps requis pour que la réponse du système passe de 0 à 100% de sa valeur finale;
2.  $t_p$  est le temps nécessaire pour que le système atteigne son premier plus haut point de dépassement;
3.  $M_p$  est le pourcentage de dépassement calculé selon;

$$M_p = \frac{P_T(t_p) - P_T(\infty)}{P_T(\infty)} \times 100\% \quad (2-2)$$

4.  $t_s$  est le temps de stabilisation; temps requis pour que le système atteigne et demeure dans un intervalle spécifique autour de sa valeur finale, généralement de 2 à 5 %;

5.  $e(\infty)$  est l'erreur statique; différence entre la consigne et la valeur finale en régime permanent.

Les réponses à la perturbation peuvent être caractérisées par;

1.  $t_{pert}$  est le temps de stabilisation à la perturbation; temps nécessaire pour que la réponse converge à l'intérieur d'un intervalle précis autour de sa valeur finale;
2.  $M_{pert}$  est la déviation maximale normalisée;

$$M_{pert} = \frac{P_T(\infty) - P_{Tmin}}{\Delta T_{pert}} \quad (2-3)$$

où  $\Delta T_{pert}$  est la variation de la perturbation.

La variation de valeur de  $K_{cP}$ ,  $T_{iP}$  et  $T_{dP}$  au sein de la structure de commande aura donc un impact direct sur ces caractéristiques. En fait, le Tableau 2-1 résume les effets typiques d'augmenter ces valeurs sur la réponse du système.

Tableau 2-1 Effet typique de l'augmentation des paramètres d'un contrôleur PID [OGAT10].

	Temps de montée ( $t_r$ )	Dépassement ( $M_p$ )	Temps de stabilisation ( $t_s$ )	Erreur statique ( $e(\infty)$ )
$K_{cP}$	Diminue	Augmente	Peu d'effet	Diminue
$1/T_{iP}$	Diminue	Augmente	Augmente	Éliminée
$T_{dP}$	Peu d'effet	Diminue	Diminue	Peu d'effet

Outre ces mesures de performance, des calculs de critères de qualité peuvent être utilisés pour améliorer le réglage d'un PID. Quatre critères de base considérant l'erreur  $e(t)$  entre la consigne et la mesure sur l'ensemble du temps sont souvent utilisés :

- Intégrale de l'erreur absolue (*Integral Absolute Error*):

$$IAE = \int_0^{\infty} |e(t)| dt \quad (2-4)$$

- Intégrale de l'erreur au carré (*Integral of Squared Error*):

$$ISE = \int_0^{\infty} (e(t))^2 dt \quad (2-5)$$

- Intégrale du temps multipliée par l'erreur absolue (*Integral of Time multiply Absolute Error*) :

$$ITAE = \int_0^{\infty} t|e(t)| dt \quad (2-6)$$

- Intégrale du temps multipliée par l'erreur au carré (*Integral of Time multiply Squared Error*):

$$ITSE = \int_0^{\infty} t(e(t))^2 dt \quad (2-7)$$

Il n'y a pas de « meilleurs » critères, mais seulement des critères plus appropriés selon l'application du système. En effet, selon les utilisations, certaines caractéristiques sont à considérer davantage pour améliorer le procédé. Cependant, tel que mentionné, les méthodes classiques ont été développées pour des structures de commande standards avec des critères de performances précis. Une augmentation du nombre de paramètres au sein de la structure influence les effets des paramètres PID.

Ainsi, pour d'autres types de structures de commande par exemple avec retour d'anti-saturation, des méthodes et/ou procédures distinctes doivent être mises en place pour assurer un réglage adéquat afin de répondre aux exigences de l'application.

### 2.1.3 Retour d'anti-saturation

Pour protéger et respecter les limites du système à contrôler, il est commun d'ajouter un saturateur entre le signal de commande et le système (voir Figure 2-3) au sein de la structure de contrôle.

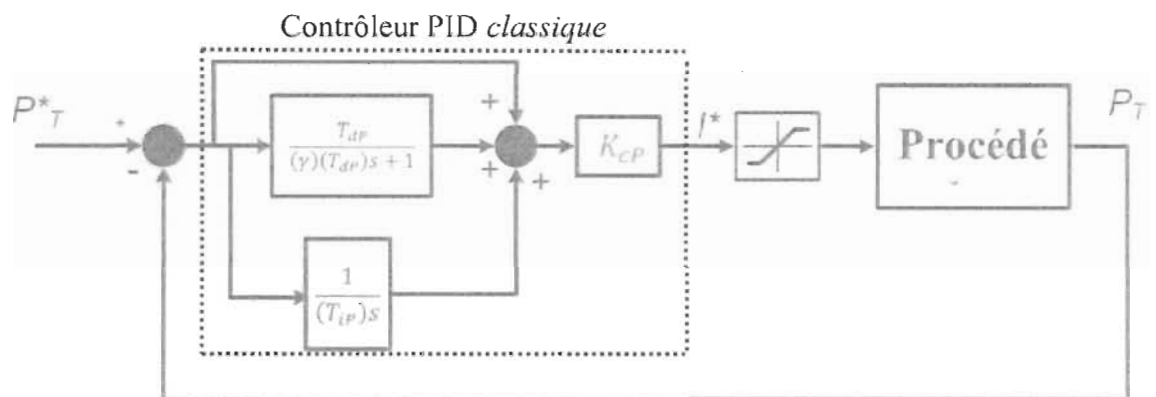


Figure 2-3 Contrôleur PID *classique* avec saturateur entre le signal de commande et le procédé (système).

Cependant, l'utilisation d'un saturateur peut amener le système dans une région non linéaire. Autrement dit, l'augmentation du signal de commande n'a plus d'effet sur la sortie du système. Ceci est causé soit momentanément par le terme proportionnel, ou de façon prolongée par la valeur de sortie de l'intégrateur. En effet, cette valeur continue d'augmenter lors de la saturation, ce qui provoque une diminution considérable des performances conduisant généralement à un dépassement important et à un temps de stabilisation plus long. Pour contrer cette problématique, communément appelé *integrator*

*windup* (enroulement de l'intégrateur), différentes options ont été suggérées tel qu'éviter la saturation en limitant le changement de consigne ou inhiber l'intégrateur quand certaines conditions sont vérifiées. Par contre, elles ne sont pas conseillées à cause d'un effort supplémentaire important dans la conception du contrôleur et à la difficulté de déterminer convenablement les paramètres de condition [VIS06] [AST89]. De là est née la solution d'insérer un retour d'anti-saturation ( $K_{satP}$ ) dans la structure de commande pour diminuer l'effet d'enroulement [BOH95] (Figure 2-4).

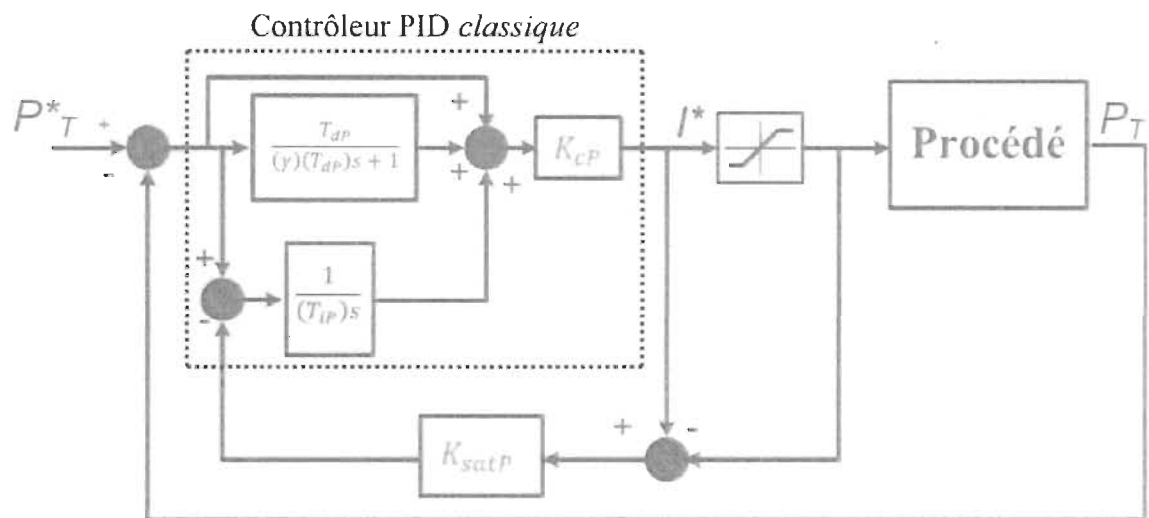


Figure 2-4 Contrôleur PID avec retour d'anti-saturation (*back-caculation*).

Lorsque le système ne sature pas, le signal du retour d'anti-saturation est nul. Par conséquent, il n'a aucun effet sur les opérations du contrôleur PID. En revanche, quand le système entre en saturation, le signal de rétroaction vient réduire l'entrée de l'intégrateur (prévient l'emballement) améliorant ainsi les performances du système. Dû au terme supplémentaire,  $K_{satP}$ , les méthodes classiques ne peuvent s'appliquer menant ainsi au développement de méthodes complémentaires pour la synthèse du correcteur.

Dans la littérature, la syntonisation des paramètres de structures de commande avec retour d'anti-saturation se fait généralement en deux phases : (i) ensemble des paramètres PID et (ii) le retour d'anti-saturation. Dans [BOH95], les coefficients du PID sont ajustés par une méthode implémentée dans MatLab (*relay autotuning*). Comme deuxième étape, le retour d'anti-saturation est fixé selon l'inverse du temps de rappel. Autrement dit, toute variation de valeur du temps de rappel se répercute sur la valeur du retour d'anti-saturation. Des variations empiriques sur la valeur du coefficient du retour d'anti-saturation peuvent être effectuées pour tenter d'améliorer la réponse du système. Qui plus est, de manière analogue, dans [PAD12], l'ajustement des paramètres de leur contrôleur PID est effectué par deux méthodes : (i) *Extremum Seeking* (ES) et (ii) *Internal Model Controller* (IMC). Le retour d'anti-saturation est syntonisé séquentiellement où cette fois-ci sa valeur est fonction du produit du temps de rappel et du temps de dérivation :

$$K_{satP} = 1 / \sqrt{T_{iP} T_{dP}}. \text{ En comparant la syntonisation de deux méthodes, dans cet article,}$$

il a été observé qu'il existe une dépendance entre l'efficacité de cette approche séquentielle et la méthode utilisée pour l'ajustement des paramètres PID [PAD12]. Ceci s'explique par l'étroite interaction entre le coefficient du retour d'anti-saturation et les valeurs du temps de rappel et de dérivation. La grandeur de ces coefficients déterminée par une méthode quelconque aura directement un impact sur la valeur du retour d'anti-saturation. Comme le coefficient du retour d'anti-saturation gouverne la rapidité à laquelle l'entrée de l'intégrateur diminue, la réponse du système peut être excessivement lente comme il a été observé avec l'ajustement des termes du PID avec IMC. Bref, dû au terme supplémentaire,  $K_{satP}$ , et à l'inconstance de performance d'une syntonisation séquentielle, un ajustement

**simultané** de l'ensemble des paramètres du système de contrôle devient intéressant et adéquat pour considérer l'interaction des paramètres et ainsi obtenir de performances satisfaisantes.

## 2.2 Système à l'étude

Le système à l'étude est la gestion du positionnement d'une table pour des opérations de meulage industriel. La Figure 2-5 présente le système physique qui a servi de référence pour modéliser le système de ce présent travail.

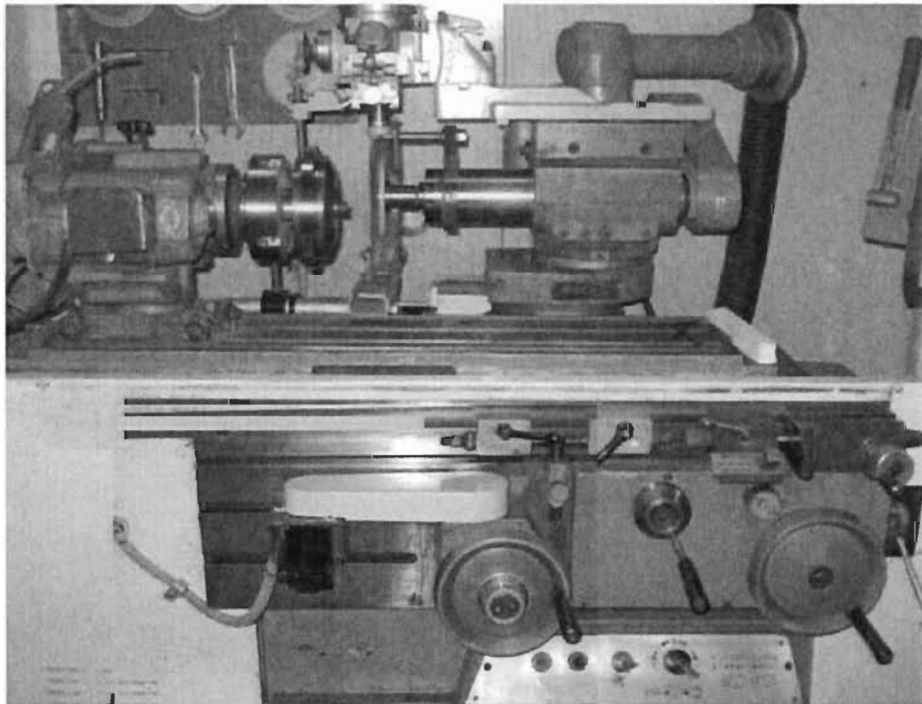


Figure 2-5 Système de positionnement d'une meule.

La Figure 2-6 représente la modélisation du système avec ses paramètres d'entrée et de sortie utilisés par la méthode d'optimisation pour ajuster la structure de contrôle avec le retour d'anti-saturation. Plus précisément, un moteur à courant continu de 3/4 HP, 5200 rpm, 3.5 N•m est couplé à un axe d'une table de meulage via une vis sans fin où un radian



de rotation du moteur provoque un déplacement de 0.1 mm de la table. En raison des éléments de transmission, le couplage flexible est modélisé comme un ressort équivalent. Les critères de performance incluent une grande précision, un dépassement minimal et une bonne performance dynamique pour améliorer la qualité du produit et réduire le cycle de fonctionnement. Le couplage flexible complexifie le processus de réglage afin d'atteindre une haute qualité de performance de ce système de commande en boucles en cascade.

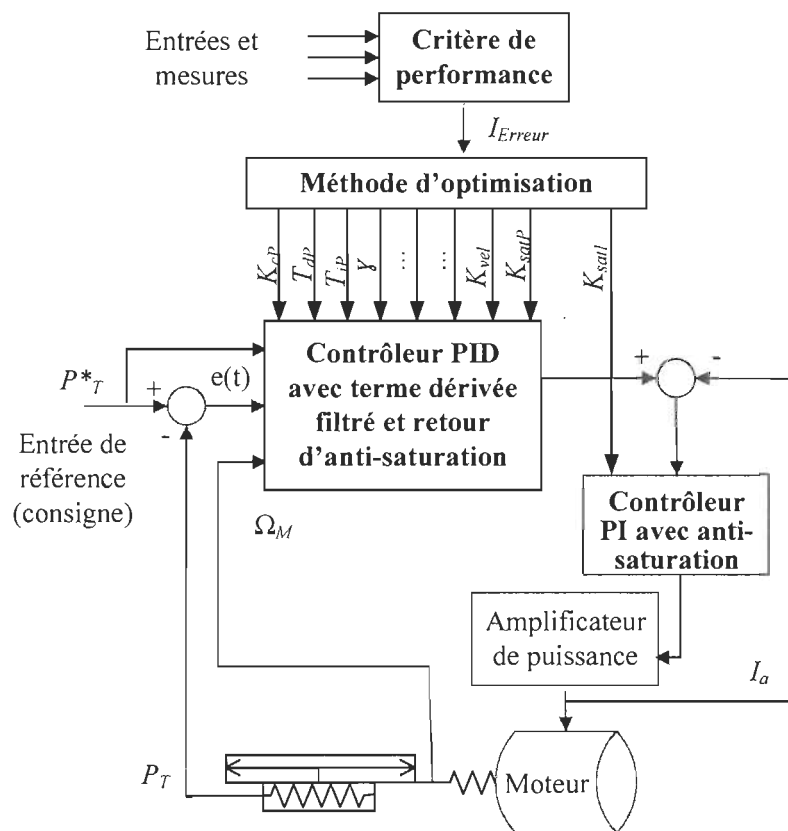


Figure 2-6 Modèle du système pour l'application de l'algorithme contraint proposé pour le réglage de paramètres de contrôleurs.

Quatre structures de régulation différentes sont à ajuster. Dans un premier temps, le dispositif de commande de rétroaction de position est illustré sur la Figure 2-7. Il se compose d'un régulateur  $PID_F$  à terme dérivée filtrée, une rétroaction de vitesse du moteur

( $K_{vel}$ ) et un retour d'anti-saturation. Le régulateur de courant interne PI a été ajusté séparément, mais son gain d'anti-saturation  $K_{satI}$  (Figure 2-6) est syntonisé par la méthode d'optimisation développée dans le cadre de ce travail.

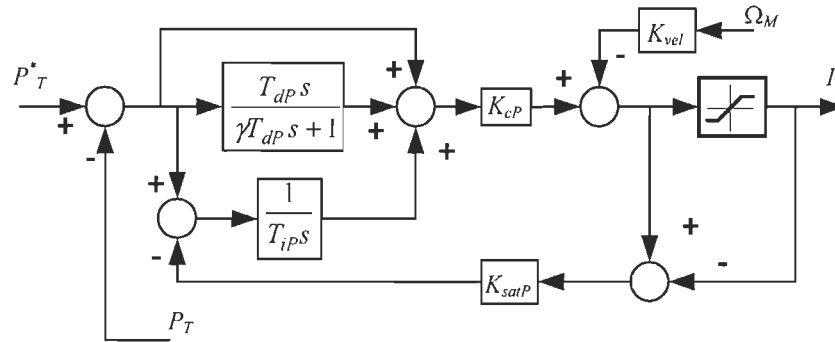


Figure 2-7 Contrôleur  $PID_F$  de rétroaction de position.

Dans le domaine industriel, il est fréquent que le procédé doive gérer des perturbations tout en répondant correctement à sa consigne, c'est notamment le cas pour les opérations de meulage avec les irrégularités de la surface de l'objet traité. Obtenir une réponse satisfaisante à ces deux entrées avec un régulateur PID *classique* avec ou sans retour d'anti-saturation représente un défi. L'ajout de degrés de liberté à la structure de commande peut permettre d'atteindre des exigences de performance plus rigoureuses. Avec ses degrés de liberté supplémentaires, le régulateur pondéré  $PID_F$  fournit globalement de meilleurs résultats à l'échelon de consigne et à l'échelon de perturbation que le régulateur PID *classique*. La Figure 2-8 présente la structure du dispositif de commande  $PID_F$  pondéré où les paramètres  $Wp$  et  $Wv$  représentent des facteurs de pondération sur les termes proportionnel et dérivé, modifiant efficacement les zéros de la fonction de transfert de la position de référence à la position réelle de la table de meulage.

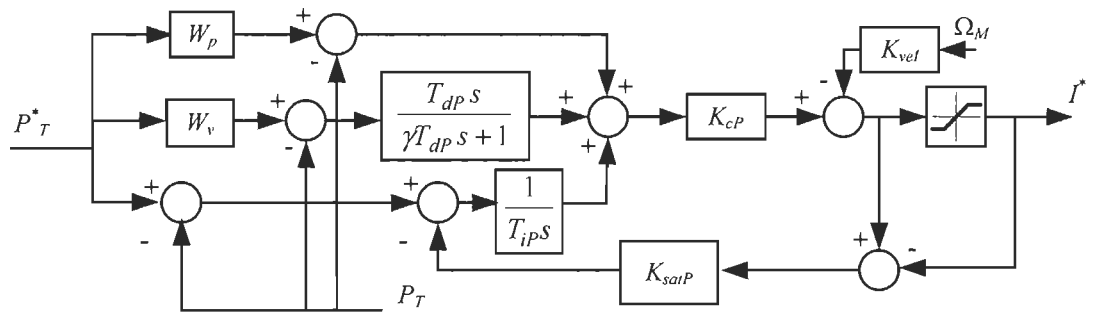


Figure 2-8 Contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur le courant.

La Figure 2-9 présente la limitation de la vitesse du moteur qui est considérée au lieu de la saturation du courant.

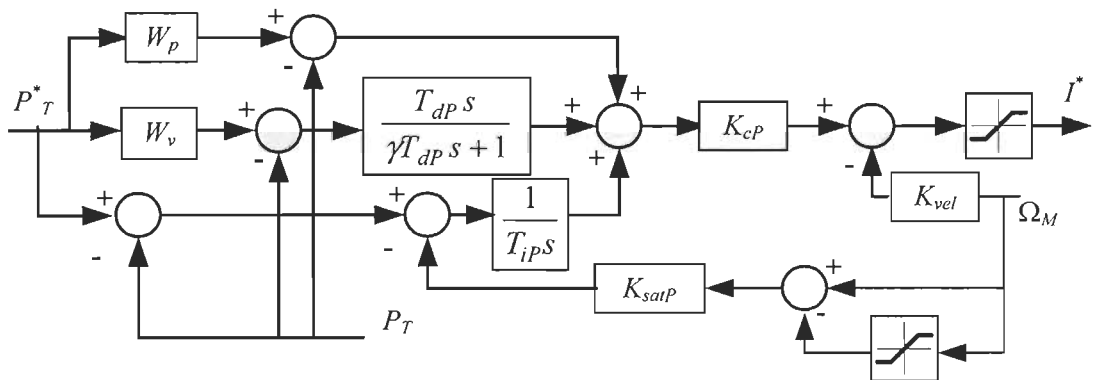


Figure 2-9 Contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur la vitesse du moteur.

La Figure 2-10 considère simultanément la saturation du courant (coefficient  $K_{satP}$ ) et la limitation de la vitesse du moteur (coefficient  $K_{satV}$ ).

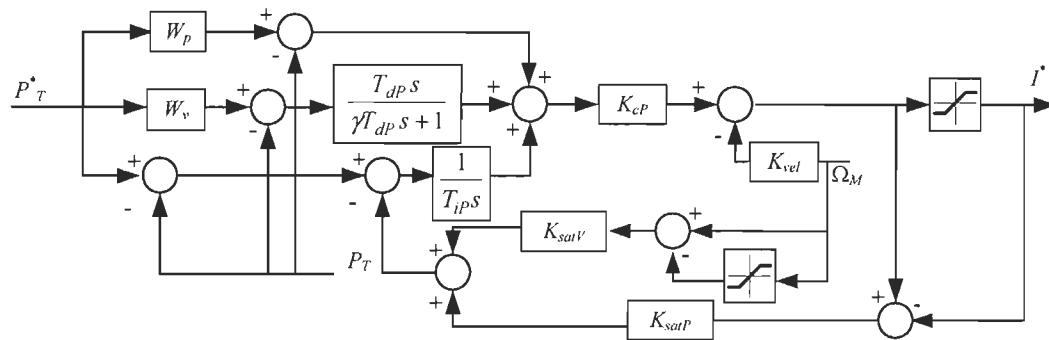


Figure 2-10 Contrôleur pondéré PID<sub>F</sub> avec retour d'anti-saturation sur la vitesse du moteur ainsi que sur le courant.

### 2.3 Conclusion

Ces quatre structures de commande ne peuvent être syntonisées efficacement avec une même méthode classique. En fait, la solution typique pour ces types de système consiste à utiliser en cascade des boucles de régulation de courant et de position de la table avec une boucle de vitesse du moteur afin d'améliorer l'amortissement interne et de réduire les oscillations. Dans ce cas, l'ajustement représente un plus grand défi qu'une simple boucle de contrôle, en particulier lorsque les contraintes du système sont considérées tels que les niveaux de saturation ainsi que les limites de vitesse.

Développer un algorithme contraint pouvant ajuster les quatre structures devient donc intéressant. En effet, une méthode commune de réglage permettrait de comparer honnêtement l'efficacité des structures de commande et ainsi faire ressortir les avantages de chacune.

## **Chapitre 3 - Algorithme d'optimisation contraint à base des ACO et de NM pour l'ajustement de PID avec retour d'anti-saturation**

Ce chapitre détaille le fonctionnement des ACO et de la méthode NM lesquels sont utilisés pour le développement de l'algorithme contraint proposé. Leur application spécifique à l'ajustement de structure de commande est expliquée. Finalement, l'algorithme proposé avec sa nouvelle méthode de contrainte est présenté.

### **3.1 Les algorithmes d'optimisation à colonies de fourmis**

Cette section présente les algorithmes à colonies de fourmis dont son abréviation est ACO pour ses initiales en anglais, *Ant Colony Optimization*. Ces derniers furent proposés dans les années 1990 par M. Dorigo et ses collègues. Cette approche est destinée à la résolution de problèmes d'optimisation combinatoire et fait partie de la classe des métaheuristiques [DOR96],[DOR04],[LEE08].

#### *3.1.1 Les métaheuristiques*

Les métaheuristiques, telles que les algorithmes d'optimisation par essaims particulaires (PSO) [WAI11], les algorithmes génétiques (GA) [HAU98], les algorithmes à colonies de fourmis (ACO) [DOR96] et la programmation évolutive (EP) [LIE01], reposent sur trois concepts communs; (i) effet *mémoire* qui sauvegarde les données recueillies par l'algorithme; (ii) *diversification* qui se réfère à l'exploration de l'espace de recherche

continu ou discret (promouvoir le processus de recherche en explorant de nouvelles régions pour générer des solutions qui se différencient de manière significative des solutions déjà visitées dans le but d'atteindre la région de l'optimum global (Figure 3-1<sup>1</sup>); et (iii) *intensification* qui fait référence à l'utilisation des informations déjà récoltées pour intensifier les recherches dans des zones intéressantes (concentre la recherche dans le voisinage des meilleures solutions rencontrées (Figure 3-2<sup>2</sup>) [ARI10].

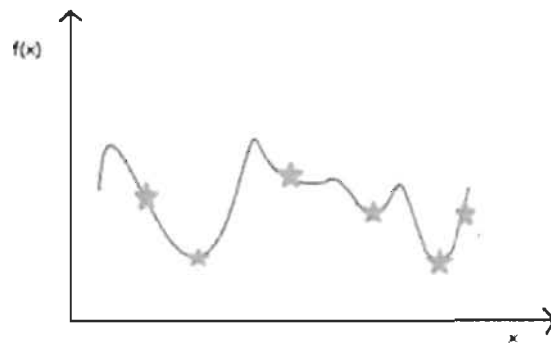


Figure 3-1 Processus de diversification.

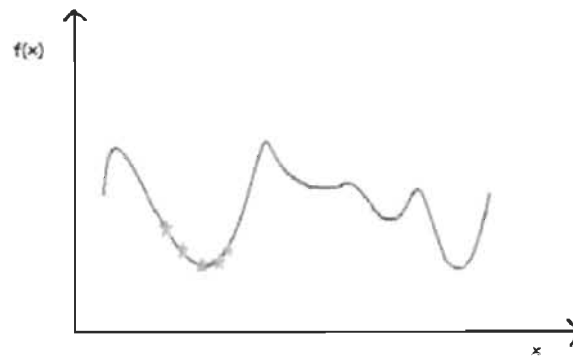


Figure 3-2 Processus d'intensification.

---

<sup>1</sup> Les images ont été inspirées de Wikipédia : <http://fr.wikipedia.org/wiki/M%C3%A9taheuristique>

(juillet 2014)

La performance de l'algorithme repose principalement sur l'équilibre entre les processus de diversification et d'intensification. L'atteinte essentielle de cet équilibre passe par le réglage des paramètres de la méthode. Un réglage adéquat représente donc un défi, ce qui peut être considéré comme un point faible général associé aux métaheuristiques.

### *3.1.2 Source d'inspiration*

Les ACO sont une technique d'optimisation inspirée par le comportement de vraies fourmis au sein de la colonie. Il a été observé qu'une colonie de fourmis face à plusieurs chemins menant à une source de nourriture avait tendance à prendre le chemin le plus court. Ceci s'explique par le fait que les fourmis communiquent entre elles via l'environnement en laissant des traces de phéromones. Les fourmis détectent localement la présence de phéromones. Plus un chemin contient de phéromones, plus il est attirant. Donc, la première fourmi qui emprunte, par hasard, le chemin le plus court fera plus d'aller-retour que les autres pour le même laps de temps. Conséquemment, la quantité de phéromones augmentera plus rapidement sur ce trajet entraînant de plus en plus de fourmis. De cette façon, toutes les fourmis finiront par emprunter le même chemin, soit le plus court. La solution émerge donc d'une interaction collective et d'une coopération naturelle entre les fourmis. La mémorisation des trajets repose donc sur les quantités de phéromones présentes sur les différents chemins. De cette caractéristique sont nés les ACO.

### *3.1.3 Description des ACO*

Les ACO peuvent être utilisés pour la majorité des problèmes d'optimisation combinatoire pour laquelle une heuristique constructive peut être employée [DOR04],[ARI10]. L'un des premiers problèmes pour lequel les ACO ont été appliqués

afin d'être validés est le problème du voyageur de commerce, soit en anglais *Traveling Saleman Problem* (TSP). Ce problème d'optimisation combinatoire est l'un des plus connus et probablement le plus utilisé pour tester des algorithmes puisque sa représentation est simple. En fait, l'objectif est de trouver la route la plus courte reliant un ensemble  $N$  de villes  $n$ . Le trajet doit commencer et se terminer sur une même ville  $n$  et ne passer qu'une seule fois par ville  $n$ . Les fourmis se déplacent donc sur le graphe du TSP générant différentes solutions. La probabilité avec laquelle la fourmi  $k$  sur une ville  $i$  choisisse une ville  $j$  s'énonce;

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \text{ où } j \in N_i^k \quad (3-1)$$

où  $\tau_{ij}(t)$  est la quantité de phéromones entre la ville  $i$  et la ville  $j$ ,  $\eta_{ij}(t) = 1/d_{ij}$  est l'inverse de la distance séparant une ville  $i$  de la ville  $j$ ,  $\alpha$  et  $\beta$  sont deux paramètres qui déterminent respectivement l'importance relative des traces de phéromones et de l'information heuristique. Pour une valeur  $\alpha = 0$ , ce sont les villes les plus proches qui sont priorisées alors que pour une valeur  $\beta = 0$ , ce sont les phéromones qui sont au travail. Pour une recherche optimale, il est nécessaire d'établir un juste équilibre entre l'importance des phéromones et l'information heuristique. Un déséquilibre peut causer une stagnation rapide, ce qui augmente la probabilité d'atteindre un minimum local seulement.

Une fois que toutes les fourmis ont parcouru leur tour, chacune d'elles dépose des phéromones sur les arcs parcourus;

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad \forall (i,j) \in L \quad (3-2)$$



où  $\Delta \tau_{ij}^k(t)$  est la quantité de phéromones déposée sur les arcs visités par une fourmi  $k$  durant son tour.

La quantité de phéromones déposée sur chaque arc du graphe est fonction de la qualité de la solution qui est définie par

$$\Delta \tau_{ij}^k(t) = \begin{cases} 1/L^k \\ 0 \text{ sinon} \end{cases} \quad (3-3)$$

où  $L$  est la longueur totale du tour. Ainsi, plus le trajet est court, plus grande sera la quantité de phéromones déposée.

De plus, les traces de phéromones associées à chaque arc sont réduites par un facteur constant  $\rho$ ;

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad \forall (i, j) \in L \quad (3-4)$$

où  $0 < \rho < 1$  est le taux d'évaporation. Le phénomène d'évaporation est en quelque sorte un facteur d'oubli utilisé pour éviter une accumulation illimitée de phéromones et également de permettre d'« oublier » les mauvaises décisions prises auparavant. La Figure 3-3 présente la structure de l'algorithme.

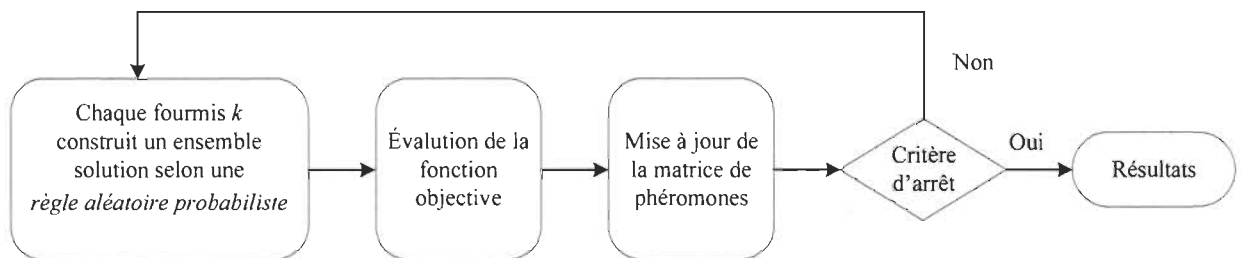


Figure 3-3 Structure des ACO.

### 3.1.4 Application spécifique à la syntonisation de contrôleurs

Pour utiliser ACO, l'ajustement de paramètres doit être transformé en un problème d'optimisation combinatoire [CHI12], [HSI04], [VAR04] (Figure 3-4). Chaque colonne du graphique correspond à un paramètre à ajuster de la structure de commande alors que chaque nœud (point) présent dans une colonne possède une valeur unique. Cette représentation permet donc facilement l'ajout ou la suppression de coefficients de contrôleur. L'intervalle de chacun des paramètres et le pas de quantification entre chaque nœud sont à déterminer pour compléter le graphe. Une syntonisation approximative des paramètres de contrôle peut être utilisée pour définir la plage de paramètres. Les fourmis se déplacent, de manière probabiliste, sur le graphe de gauche à droite. Elles choisissent un nœud par colonne pour former une solution complète, soit une combinaison de paramètres pour la structure de commande à ajuster. Selon la qualité de la solution (réponse du système), des traces de phéromones sont déposées sur les arcs de chemin parcouru par la fourmi lors de la création de l'ensemble solution. Donc, une matrice de phéromone est mise à jour en fonction de la qualité de la solution de manière à influencer l'exploration des prochaines fourmis.

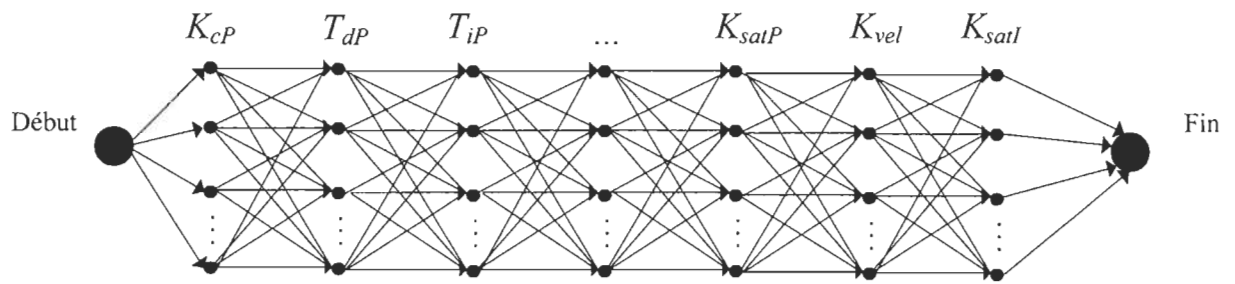


Figure 3-4 Graphe d'optimisation pour les différentes structures de commande.

La performance du trajet est évaluée par l'indice  $I_{Erreur}$  (3-5) où les fourmis coopèrent pour trouver l'ensemble de paramètres minimisant ce critère (défini spécifiquement pour le système à l'étude). Cette spécification du critère de performance est possible due à la flexibilité d'optimisation des ACO.

$$I_{Erreur} = \sum \left( \int_{t_s}^{t_s+T_s} \frac{|e(t)|}{\Delta P_T^*} dt \right) + K_T \sum \left( \int_{t_{pp}}^{t_{pp}+T_p} \frac{|e(t)|}{\Delta T_{pert}} dt \right) + \sigma \int_0^{T_f} \frac{e_d(t)}{\Delta P_T^*} dt + \beta \int_0^{T_f} \frac{e_{vd}(t)}{\Omega_{Mm}} dt \quad (3-5)$$

où

- Les deux premiers termes sont les sommations respectives de la réponse transitoire normalisée par rapport à l'amplitude de la variation de l'entrée échelon  $\Delta P_T^*$  et de la réponse transitoire due à l'application d'une perturbation d'amplitude  $\Delta T_{pert}$ ;  $t_s$  est le temps du dernier point de variation de consigne;  $T_s$  est le temps approximatif de stabilisation pour l'entrée de référence échelon;  $t_{pp}$  est le temps de la dernière variation d'entrée de perturbation;  $T_p$  est le temps approximatif de stabilisation de l'entrée de perturbation;  $e(t) = P_T^*(t) - P_T(t)$  est l'erreur de sortie;  $P_T$  et  $P_T^*$  sont respectivement la position actuelle et la position de référence de la table;  $\Delta P_T^*$  est l'amplitude de variation du dernier échelon utilisé pour normaliser l'erreur de position;  $\Delta T_{pert}$  est l'amplitude de la dernière entrée de perturbation;  $K_T$  est une constante positive utilisée comme rigidité équivalente pour normaliser l'erreur de position due à la perturbation;

- Le troisième terme pénalise les erreurs de dépassement;  $T_f$  est le temps de simulation où le terme est pénalisé;  $\sigma$  est une constante de poids positive;  $e_d(t)$  est défini tel que

$$e_d(t) = \begin{cases} |e(t)| & \text{durant le dépassement} \\ 0 & \text{sinon.} \end{cases} \quad (3-6)$$

- Le quatrième terme pénalise le dépassement des limites de vitesse du moteur ;

$$e_{v_d}(t) = \begin{cases} ||\Omega_M(t)| - \Omega_{Mm}| & \text{si excède les limites du moteur} \\ 0 & \text{sinon} \end{cases} \quad (3-7)$$

où  $\Omega_{Mm}$  est la vitesse limite maximale du moteur.

Des valeurs plus élevées de  $\sigma$  dans (3-5) pénalisent davantage le dépassement, lequel est dommageable durant les opérations de meulage. L'augmentation de  $K_T$  est utilisée pour réduire l'impact de la perturbation sur la réponse du système. La constante  $\beta$  est utilisée pour contraindre la vitesse du moteur dans le but de respecter les limites du système. Le choix des grandeurs de ces paramètres est justifié dans le prochain chapitre.

### 3.2 La méthode Nelder-Mead

Cette section présente la méthode du simplexe Nelder-Mead (NM). Cet algorithme fût publié en 1965 par Nelder et Mead et depuis c'est un des plus populaires pour les problèmes d'optimisation non linéaire sans contrainte [NEL65]. L'algorithme NM tend à minimiser une fonction scalaire non linéaire de  $n$  paramètres sans l'utilisation de dérivée. En fait, à partir d'un point de départ, NM évalue la fonction objective et oriente ses recherches. Cette méthode fait partie de la classe générale des méthodes de recherche locale directe.

### 3.2.1 Description de NM

Cet algorithme utilise un simplexe de  $n + 1$  sommets évaluant la fonction pour un vecteur  $x$  de dimension  $n$ . Initialement, l'algorithme construit un simplexe à partir d'un point de recherche initial  $x_o$ . Pour le former, un certain pourcentage est ajouté (5% dans ce travail) à chaque  $x_o(i)$  à  $x_o$ , où  $i = 1, \dots, n$ . Par la suite, le simplexe se déplace dans l'espace de recherche non contraint dans le but de minimiser une fonction objective. La procédure suivante est suivie pour les futures itérations conformément à [LAG98] et la Figure 3-5 présente la structure de l'algorithme:

1. **Ordonner.** À l'itération  $k$ , les sommets identifiés tels que  $x_1^{(k)}, \dots, x_{n+1}^{(k)}$  sont ordonnés selon la valeur de la fonction  $f_1^{(k)} \leq f_2^{(k)} \leq \dots \leq f_{n+1}^{(k)}$  où  $f_i^{(k)}$  fait référence à  $f(x_i^{(k)})$ . Le point  $x_1^{(k)}$  est donc associé au meilleur point ou sommet alors que  $x_{n+1}^{(k)}$  au pire. Dans le même ordre d'idées,  $f_{n+1}^{(k)}$  fait référence à la pire valeur de la fonction à minimiser. À chaque itération, soit (i) le pire point actuel  $x_{n+1}^{(k)}$  est remplacé par un nouveau sommet dans le simplexe ou (ii) l'algorithme atteint l'étape 5 (rétrécissement), le simplexe sera  $x_1, v_2, \dots, v_{n+1}$ .

2. **Réflexion.** Calculer le point de réflexion  $x_r$  selon;

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) \quad (3-8)$$

où  $\bar{x} = \sum_{i=1}^n x_i/n$  est le centroïde des  $n$  meilleurs points et  $\rho = 1$  (valeur usuelle) est le coefficient de réflexion. Évaluer la fonction  $f_r = f(x_r)$ . Si la valeur de  $f_r$  est tel que  $f_1 \leq f_r < f_n$ , le point de réflexion  $x_r$  remplace le point  $x_{n+1}$  dans le simplexe. L'itération est terminée.

3. **Expansion.** Si  $f_r < f_1$ , calculer le point d'expansion  $x_e$  selon;

$$x_e = \bar{x} + \chi(x_r - \bar{x}) \quad (3-9)$$

où  $\chi = 2$  (valeur usuelle) est le coefficient d'expansion. Évaluer la fonction;  $f_e = f(x_e)$ . Si  $f_e < f_r$ , le point  $x_e$  est accepté dans le simplexe et l'itération est terminée. Sinon, si  $f_r \leq f_e$ ,  $x_r$  est accepté et l'itération est terminée.

4. **Contraction.** Si  $f_r \geq f_n$ , une contraction est effectuée entre  $\bar{x}$  et le meilleur de  $x_{n+1}$  et  $x_r$ .

a. **Extérieure.** Si  $f_n \leq f_r < f_{n+1}$ , signifiant que  $x_r$  est strictement meilleur que  $x_{n+1}$ , une contraction à l'intérieur du simplexe est effectuée. Le point  $x_c$  est calculé selon;

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) \quad (3-10)$$

où  $\gamma = 1/2$  (valeur usuelle) est le coefficient de contraction. Évaluer  $f_c = f(x_c)$ . Si  $f_c < f_r$ , le point  $x_c$  est introduit dans le simplexe et l'itération est terminée. Sinon, passer à l'étape 5 (rétrécissement).

b. **Intérieure.** Si  $f_r \geq f_{n+1}$ , le point de la contraction intérieure est calculé selon;

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) \quad (3-11)$$

où  $\gamma = 1/2$  (valeur usuelle) est le coefficient de contraction. Évaluer  $f_{cc} = f(x_{cc})$ . Si  $f_{cc} < f_{n+1}$ , le point  $x_{cc}$  est introduit dans le simplexe et l'itération est terminée. Sinon, passer à l'étape 5 (rétrécissement).

5. **Rétrécissement.** Évaluer  $f$  au  $n$  points selon;

$$v_i = x_1 + \sigma(x_i - x_1) \quad (3-12)$$

où  $i = 2, \dots, n+1$  et  $\sigma = 1/2$  (valeur usuelle) est le coefficient de rétrécissement.

Ainsi, le simplexe à la prochaine itération est  $x_1, v_2, \dots, v_{n+1}$ .

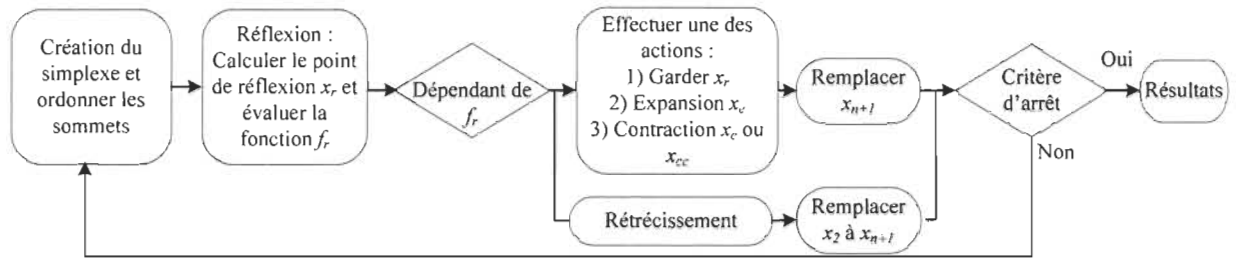


Figure 3-5 Structure de la méthode NM.

### 3.2.2 Application spécifique à l'ajustement de contrôleurs

Chaque sommet du simplexe représente donc un ensemble de paramètres de la structure de commande. Une modification importante à l'algorithme NM de base doit être apportée pour permettre une syntonisation efficace et avantageuse. En effet, tel que mentionné, pour assurer le bon fonctionnement, seuls des termes positifs sont acceptables dans les systèmes de commande à l'étude<sup>2</sup>. Ainsi, contraindre l'algorithme devient incontournable. Une contrainte de positivité est introduite dans les itérations de NM en ajoutant une nouvelle opération. En fait, avant d'insérer un nouveau terme  $x$  dans le simplexe, une vérification de signe est effectuée sur chaque valeur de paramètre pour s'assurer qu'il soit positif. Puisque les paramètres du contrôleur PID et les paramètres d'anti-saturation sont interdépendants, il

<sup>2</sup> Il est à noter que les paramètres de la structure de commande doivent se trouver dans une région avec paramètres positifs, ou avec un signe bien défini dans le cas du gain  $K_{cP}$  en particulier. Le travail est limité au cas positif.

est donc important de considérer équitablement l'ensemble des paramètres lorsqu'un paramètre doit être limité, différemment de [LUE03]. La procédure de la nouvelle étape est la suivante et elle est insérée tel que présenté dans la Figure 3-6 du graphe décisionnel de la méthode NM originale:

#### Test de positivité -Réduction globale

1. Calculer la grandeur de dépassement;

$$\delta(i) = \begin{cases} |x(i) - \varphi| & \text{si } x(i) < 0 \\ 0 & \text{sinon} \end{cases} \quad i = 1, \dots, n \quad (3-13)$$

où  $x(i)$  est le point préalablement calculé et prêt à être inséré dans le simplexe,  $n$  est le nombre de variables et  $\varphi$  est la valeur de la contrainte (valeur très proche de zéro).

2. Calculer la différence entre chaque point;

$$d(i) = |x(i) - x_{n+1}^{(k)}(i)|, \quad i = 1, \dots, n \quad (3-14)$$

où  $x_{n+1}^{(k)}(i)$  est le point à remplacer dans le simplexe.

3. Calculer le rapport du dépassement par rapport à la distance;

$$\rho(i) = \frac{\delta(i)}{d(i)} \quad (3-15)$$

4. Procéder à la réduction globale;

$$x(i) = \begin{cases} x(i) + (\max_i \rho(i)) d(i) & \text{si } x(i) < x_{n+1}^{(k)}(i) \\ x(i) - (\max_i \rho(i)) d(i) & \text{sinon} \end{cases}, \quad i = 1, \dots, n \quad (3-16)$$

Le point  $x(i)$  est introduit dans le simplexe et l'itération est terminée.



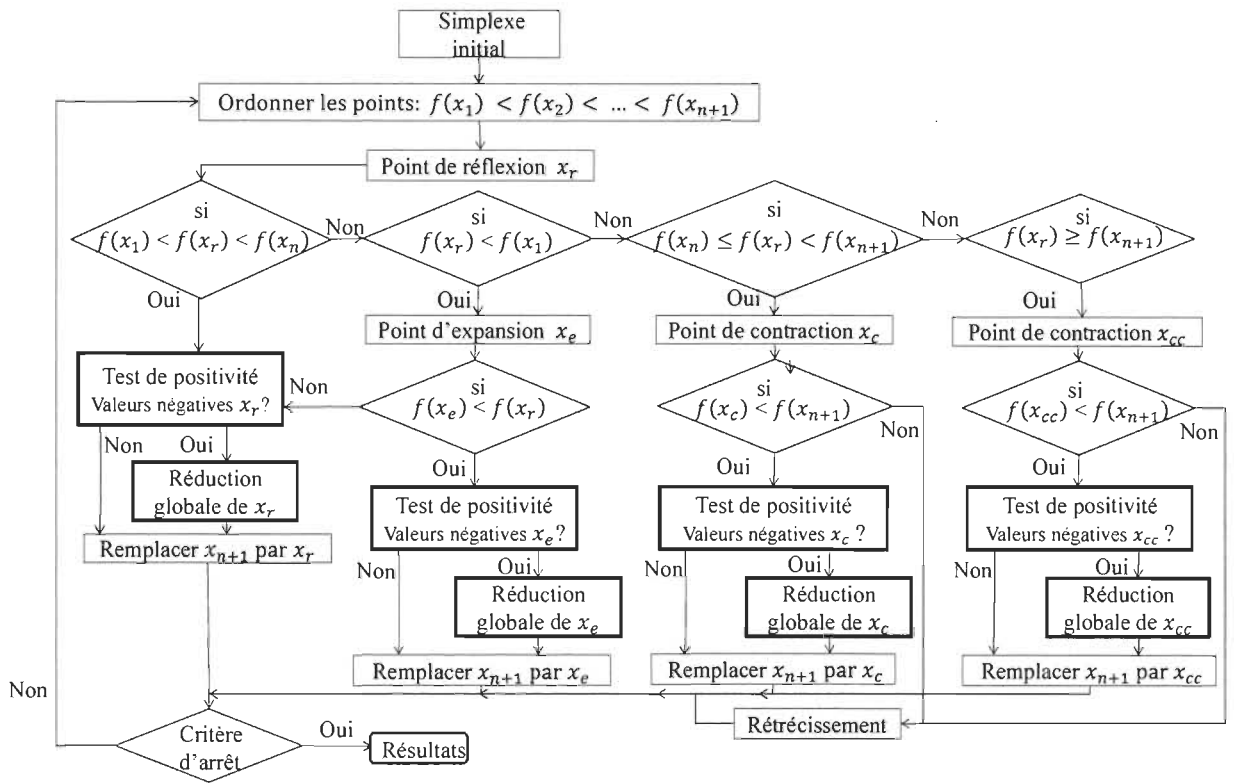


Figure 3-6 Graphe décisionnel de la procédure NM avec le test de contrainte de positivité.

### 3.3 Fonctionnement de l'algorithme contraint proposé

L'algorithme contraint proposé est composé de deux étapes principales: (i) l'optimisation globale utilisant une quantification approximative de paramètres à l'aide des ACO pour obtenir des conditions initiales de haute qualité pour NM; (ii) l'optimisation locale utilisant NM contraint pour le réglage fin des paramètres. Le mariage de deux algorithmes dépasse la solution élémentaire de combiner séquentiellement les deux algorithmes de base, c'est-à-dire mettre une à la suite de l'autre les procédures schématisées à la Figure 3-3 et à la Figure 3-5. Cette façon ne serait pas optimale en temps de calcul et en qualité de la réponse du système. En fait, la Figure 3-7 présente le graphe décisionnel de l'algorithme contraint proposé. Selon une règle de démarrage, la méthode NM est lancée à

plusieurs reprises à partir de solutions intermédiaires obtenues par ACO, ce qui permettra de mieux explorer l'espace de recherche pour s'approcher davantage de l'optimum de la fonction tout en réduisant le temps de calcul. De plus, contraindre l'algorithme NM selon le nouveau test de positivité assure la praticabilité de l'ensemble des coefficients trouvés tout en conservant l'interaction des paramètres.

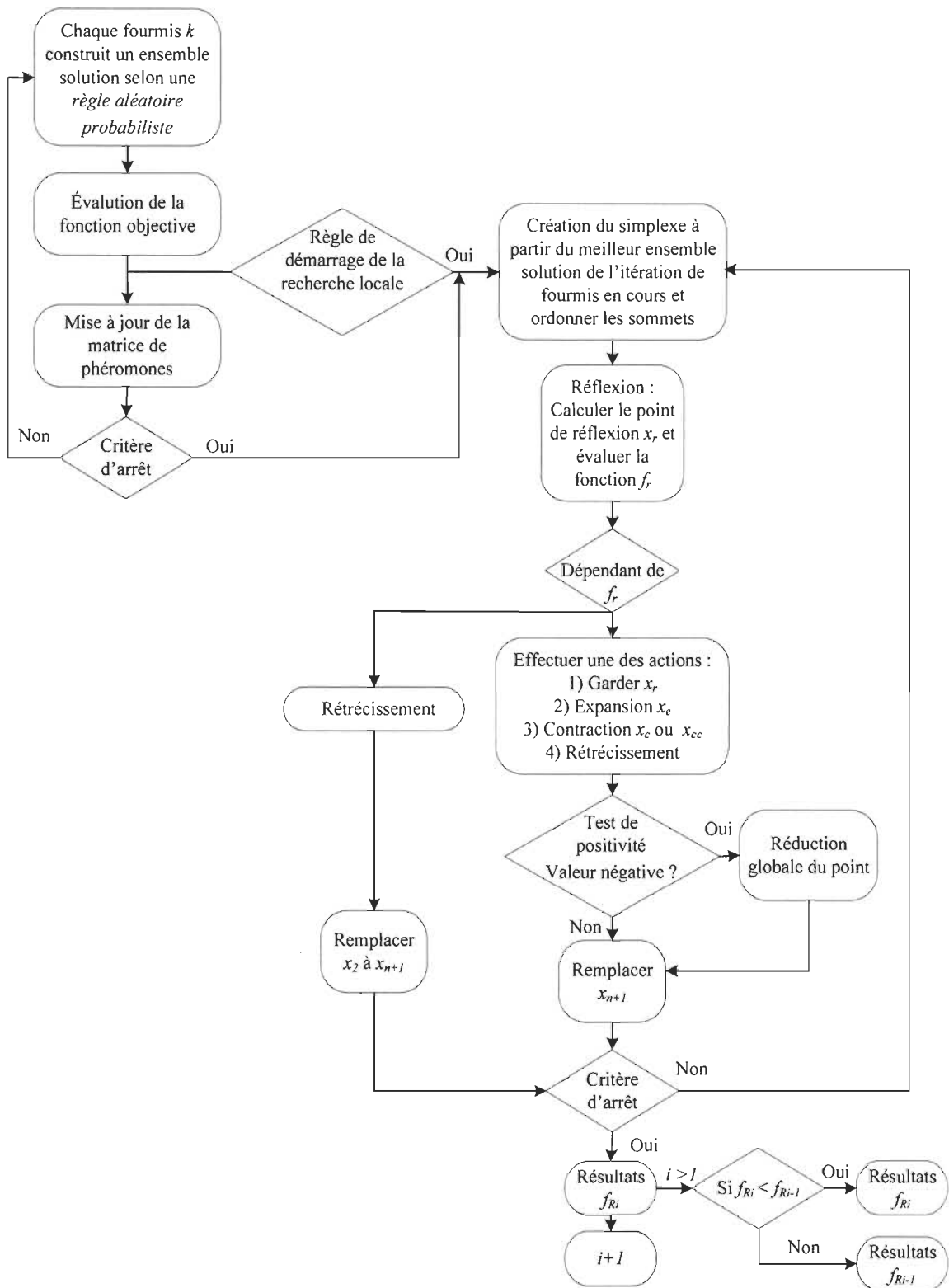


Figure 3-7 Structure de l'algorithme contraint proposé.

Il est important de mentionner que le mariage des deux algorithmes est possible puisqu'ils peuvent optimiser un nombre variable de termes et parce que leur critère de performance est adaptable. Cette fusion contrecarre les inconvénients de chacune des techniques d'optimisation pour conserver uniquement leurs forces respectives. Par conséquent, les fourmis explorent un grand espace de recherche afin de trouver de bons ensembles de paramètres initiaux pour des démarrages successifs de NM contraint. Concernant ACO, les exigences au niveau de la précision des paramètres d'optimisation, de la détermination de l'intervalle ainsi que du pas de discrétisation de l'espace de recherche sont moins élevées. En effet, le but de la recherche n'est plus de trouver un ensemble solution le plus optimal possible, mais un ensemble de départ suffisamment bon pour démarrer efficacement la méthode NM contraint. De ce fait, le temps de convergence des ACO est réduit tout comme celui de la méthode NM puisque son point de départ est convenablement bon.

### **3.4 Conclusion**

L'algorithme contraint proposé à base d'ACO et de NM concilie les points forts de chacun tout en écartant leurs inconvénients respectifs. En effet, l'algorithme contraint proposé conserve l'avantage de l'approche globale des ACO pour s'assurer d'englober l'optimum global de la fonction et la force de précision offerte par la méthode locale de NM. Cette combinaison réduit considérablement le temps de convergence de l'algorithme (Chapitre 4). De plus, leur critère de performance adaptable ainsi que leur nombre de paramètres à ajuster flexible permettent une application étendue de l'algorithme contraint proposé à plusieurs contrôleurs. Il est donc plus facile de comparer les performances entre les différentes structures de commande en ayant une méthode commune d'ajustement.

## Chapitre 4 - Résultats de simulation

Ce chapitre présente les résultats des scénarios de simulations d'ajustement de différentes structures de contrôle effectué sur le système de positionnement flexible décrit au Chapitre 2. En premier lieu, les résultats de syntonisation effectuée par ACO [BLOa13],[BLO13b] sur trois structures de commande seront présentés, expliqués et comparés [BLO14a]. Par la suite, la méthode NM de base (non contrainte) sera appliquée seule pour l'ajustement d'une structure de commande simple [BLO13c]. Finalement, les résultats du nouvel algorithme contraint proposé appliqué à quatre structures de commande seront présentés et détaillés [BLO14b].

### 4.1 Ajustement des structures de commande par ACO

La Figure 4-1 présente l'environnement de simulation développé pour appliquer les ACO à l'ajustement de paramètres des différentes structures de contrôle préalablement décrites. ACO est implémenté dans Matlab® et le modèle du système dans Simulink™.

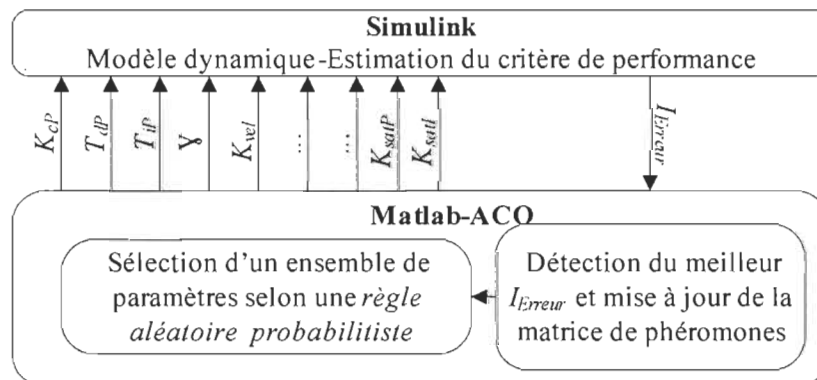


Figure 4-1 Structure de l'environnement de simulation des itérations des ACO.

#### 4.1.1 Structure de commande simple

Pour déterminer la grandeur de l'espace de recherche utilisé par les ACO, les paramètres du contrôleur  $PID_F$  (Figure 2-7) ont d'abord été déterminés par placement de pôles sur une version approximée du modèle rigide du système. L'intervalle choisi est large pour s'assurer d'englober la région de l'optimum global. L'espacement entre chaque paramètre est logarithmique.

$$\begin{array}{llll} K_{SatP}: [1,1000] & K_{cP}: [0.001, 10] & T_{dP}: [0.001,10] & K_{SatI}: [1,1000] \\ T_{iP}: [0.003,30] & \gamma: [10,50] & K_{vel}: [0.01,0.1] & \end{array}$$

Comme premier scénario de simulation, la méthode ACO a été appliquée, hors ligne, au contrôleur  $PID_F$  de rétroaction de position illustrée à la Figure 2-7, où 7 paramètres sont à ajuster. Le nombre de fourmis  $m = 100$ , le nombre d'itérations  $n = 100$  et le nombre de pas de discrétisation  $r = 75$ . Les paramètres  $K_t$  et  $\beta$  de l'indice de performance (3-5) sont fixés à 0. Étant donné que le dépassement est très dommageable pour les opérations de meulage, réduire ce dernier tout en maintenant de grandes performances dynamiques est hautement souhaitable. Pour ce faire, différentes valeurs de  $\sigma$ , dans (3-5), ont été utilisées. Cette variation de  $\sigma$  permet de vérifier la corrélation entre la recherche par ACO et son critère de performance. Inévitablement, l'ajustement par les ACO de cette structure de commande est comparé avec l'ajustement séquentiel; syntonisation du bloc de paramètres  $PID_F$  (par la méthode *root locus*) suivi des deux retours d'anti-saturation. La Figure 4-2 montre que l'ajustement séquentiel des paramètres  $PID_F$  suivit du réglage du retour d'anti-saturation produit une réponse suramortie de faible performance pour le système à l'étude. Comme expliqué, une dépendance réside entre les paramètres du contrôleur et son retour de saturation. Ainsi, dépendamment de la valeur des termes du contrôleur  $PID_F$ , il est possible

qu'aucune valeur de retour d'anti-saturation ne puisse délivrer une performance de qualité. Dans notre cas, en analysant les valeurs des paramètres, ce résultat est dû au fait que, sans saturation, la solution du régulateur  $PID_F$  tend à s'appuyer sur son terme dérivé ( $T_{dP}$ ) pour accélérer sa réponse initiale conduisant ainsi à temps de réponse très lent avec saturation. Delà, l'intérêt d'une approche **globale** de syntonisation. D'un autre côté, l'efficacité des ACO est démontrée par la Figure 4-2. Les résultats de la recherche sont consistants avec le critère de performance à minimiser. En effet, plus grande est la valeur  $\sigma$ , moins il y a présence de dépassement dans la réponse transitoire et la valeur  $I_{Erreur}$  décroît conformément (Tableau 4-1).

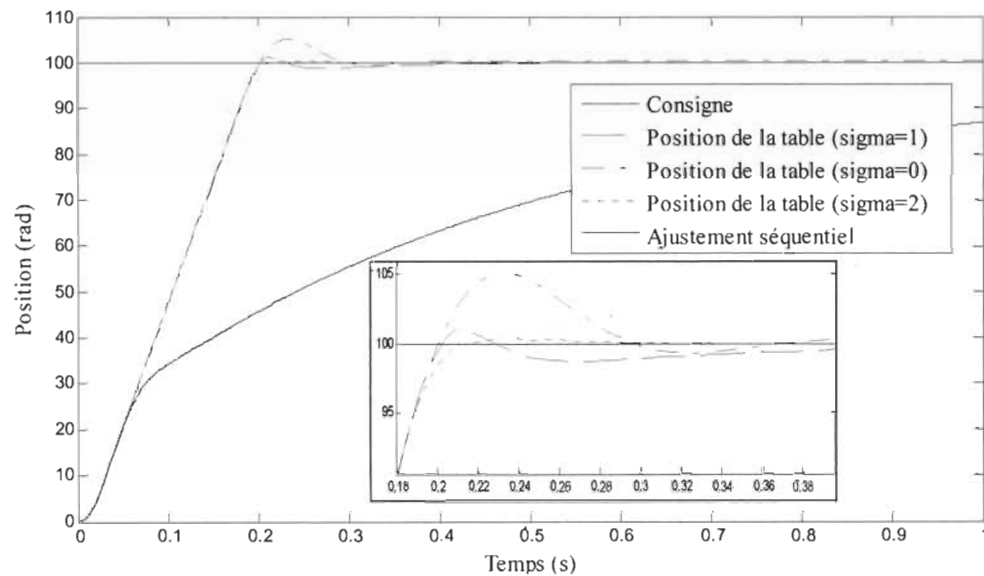


Figure 4-2 Réponse à un échelon d'entrée pour le contrôleur  $PID_F$  avec retour d'anti-saturation – réglage par ACO.

Tableau 4-1 Mesures de performance pour la structure de contrôle simple.

Méthode	$\sigma=0$		$\sigma=1$		$\sigma=2$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.201	0.200	0.203	0.221	0.213	0.753
$t_p$ (s)	0.230	0.223	0.211	0.221	0.230	0.753
$M_p$ (%)	5.100	0.700	1.100	0	0.400	0
$t_s(2\%)$ (s)	0.271	0.192	0.196	0.193	0.198	0.243
$I_{Erreur}$	0.119	0.105	0.107	0.104	0.105	0.107

Une façon d'évaluer convenablement et efficacement les performances des ACO est de comparer ses résultats avec une autre méthode métaheuristique prometteuse. Puisque GA présente également un grand potentiel pour l'ajustement de contrôleur PID *classique* [SHA11],[NAG10], cette méthode a été appliquée à la même structure de contrôle (Figure 2-7). Sa population est de 100 chromosomes pour 100 itérations. Ses limites de l'espace de recherche et les paramètres de (3-5) sont identiques à ceux des ACO. La Figure 4-3 montre que les réponses transitoires du système oscillent plus qu'avec ACO. Lorsque  $\sigma = 2$ , le système est suramorti et les mesures de performance (expliquées à la section 2.3.2) sont médiocres (Tableau 4-1). De plus, la valeur  $I_{Erreur}$  ne décroît pas conformément selon l'augmentation de la valeur de  $\sigma$ .



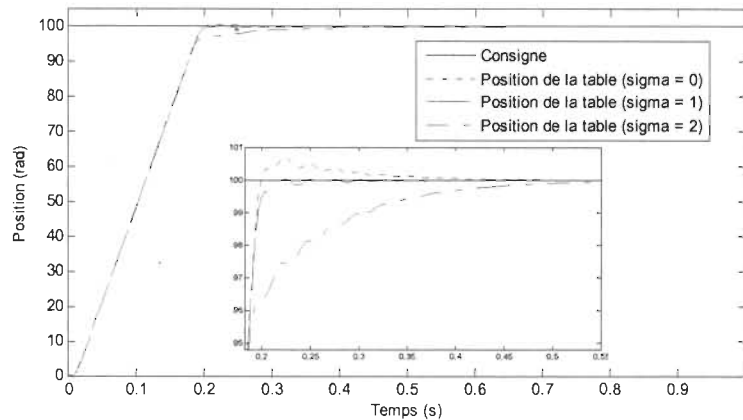


Figure 4-3 Réponse à un échelon d'entrée pour le contrôleur  $PID_F$  avec retour d'anti-saturation – réglage par GA.

Selon les résultats obtenus, l'optimisation par ACO produit des performances supérieures. En effet, pour ce scénario, sa forte tendance de corrélation entre son critère d'optimisation et sa recherche permet d'offrir un plus grand contrôle sur le système, ce qui est un avantage important.

Un échelon de perturbation correspondant à 20% du couple nominal (0.75 Nm) (représentant une valeur de perturbation réaliste) est appliqué au système à deux secondes. Pour améliorer le rejet de perturbation, le coefficient  $K_t$  est utilisé dans le critère de performance (3-5) pour pénaliser l'erreur de perturbation de sortie. La sélection des valeurs  $K_t$  est basée sur l'interprétation de (3-5) par un modèle de ressort équivalent et en posant un déplacement maximal autorisé de la table de meulage pour une perturbation donnée. À partir de (3-5), il est possible d'écrire (ce n'est qu'une interprétation et le facteur d'unité a été utilisé pour normaliser ce terme à 1):

$$1 = K_t \frac{|e_{max}|}{\Delta T_{pert}} \quad (4-1)$$

et de supposer qu'une perturbation de 20% du couple nominal du moteur amène un déplacement de 1 mm à 0.1 mm de la table de meulage satisfaisant ainsi (4-1). Les valeurs de  $K_t$  sont donc respectivement 0.075 à 0.75. La Figure 4-4 présente les réponses optimales obtenues à l'échelon et les réponses à la perturbation de charge pour différentes valeurs de  $K_t$ . La Figure 4-5 présente l'effet positif du coefficient  $K_t$  sur l'amélioration de la réponse à la perturbation. Ces deux figures montrent que ceci a été réalisé au prix d'un compromis sur la qualité de la réponse transitoire de l'entrée. Delà, la pertinence d'ajouter des degrés de liberté en utilisant un contrôleur pondéré  $PID_F$ .

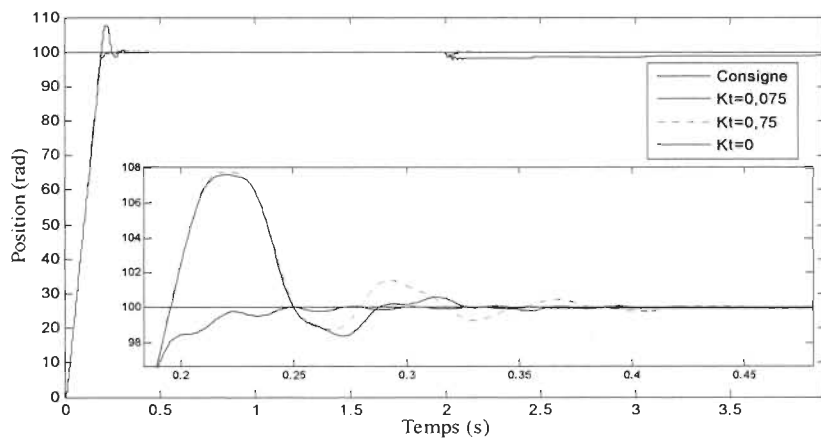


Figure 4-4 Agrandissement de la réponse à l'échelon d'entrée pour le contrôleur  $PID_F$  avec retour d'anti-saturation selon différents  $K_t$  ( $\sigma = 2$  et  $\beta = 0$ ) – réglage par ACO.

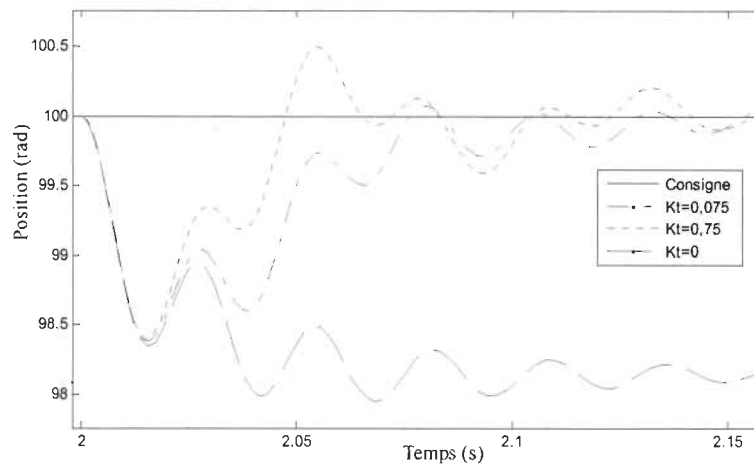


Figure 4-5 Agrandissement de la réponse à l'échelon de perturbation pour le contrôleur  $PID_F$  avec retour d'anti-saturation selon différents  $K_t$  ( $\sigma = 2$  et  $\beta = 0$ ) – réglage par ACO.

#### 4.1.2 Structures de commande complexes

Comme deuxième scénario de simulation, le contrôleur pondéré  $PID_F$  avec anti-saturation sur le courant de référence (Figure 2-7) est utilisé afin de mieux gérer les perturbations. Deux termes sont donc ajoutés à l'espace de recherche des ACO, soit  $W_v$  avec un intervalle de  $[0 \text{ à } 1]$  tout comme  $W_p$  de  $[0 \text{ à } 1]$ . Ces paramètres additionnels offrent des degrés de liberté supplémentaires permettant au contrôleur d'atteindre de meilleures performances à la réponse à l'échelon et au rejet de perturbation. Due à cette augmentation du nombre de paramètres à ajuster, le nombre de fourmis  $m$  et d'itérations  $n$  sont augmentés à 200. Dans la même visée qu'à la sous-section précédente, une comparaison des performances avec GA est effectuée. Ainsi, la population et le nombre d'itérations de GA sont également augmentés à 200. Dans l'indice de performance (3-5),  $\sigma = 2$ ,  $\beta = 0$  et le  $K_t$  prend l'une des trois valeurs: 0, 0.075 et 0.75. La Figure 4-6 et la Figure 4-7 montrent les résultats obtenus avec l'ajustement par ACO et par GA pour  $K_t = 0.75$ . Les réponses système pour les deux

autres valeurs de  $K_t$  possèdent une allure/comportement semblable à la Figure 4-6. Le Tableau 4-2 résume les caractéristiques des réponses transitoires à l'échelon et à la perturbation pour les différentes valeurs de  $K_t$ . Tel que montré sur la Figure 4-6 et la Figure 4-7, le réglage par ACO est plus efficace que par GA. En effet, la réponse transitoire possède moins de dépassement et le temps de stabilisation est plus court. Concernant la réponse à la perturbation, même si la déviation maximale normalisée est légèrement plus grande, le temps de stabilisation,  $t_{pert}$ , est beaucoup plus court et le dépassement est moins important. De plus, globalement, l'amplitude et la fréquence des oscillations présentes dans les réponses transitoires à l'échelon et à la perturbation sont plus faibles avec l'ajustement par ACO.

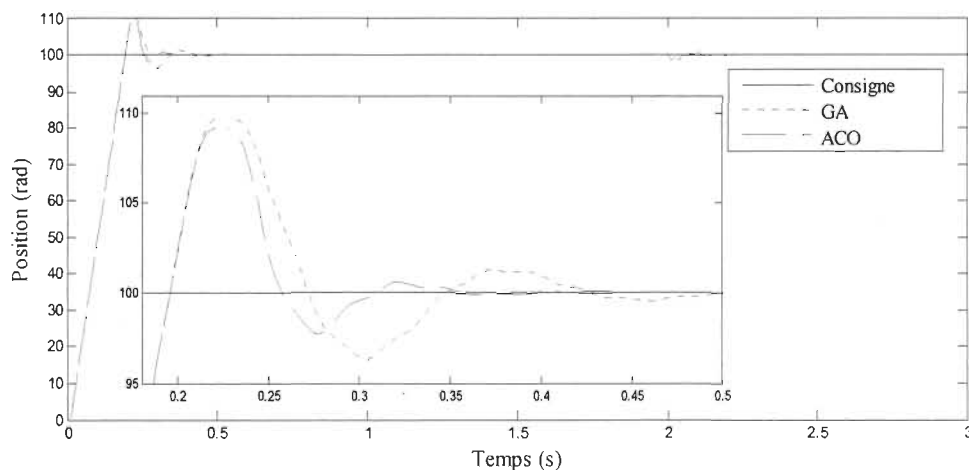


Figure 4-6 Réponse à l'échelon d'entrée et de perturbation et agrandissement de la réponse à l'échelon d'entrée pour le contrôleur pondéré  $PID_F$  avec retour d'anti-saturation sur le courant de référence ( $\sigma = 2$ ,  $K_t = 0.75$ ) - ACO vs GA.

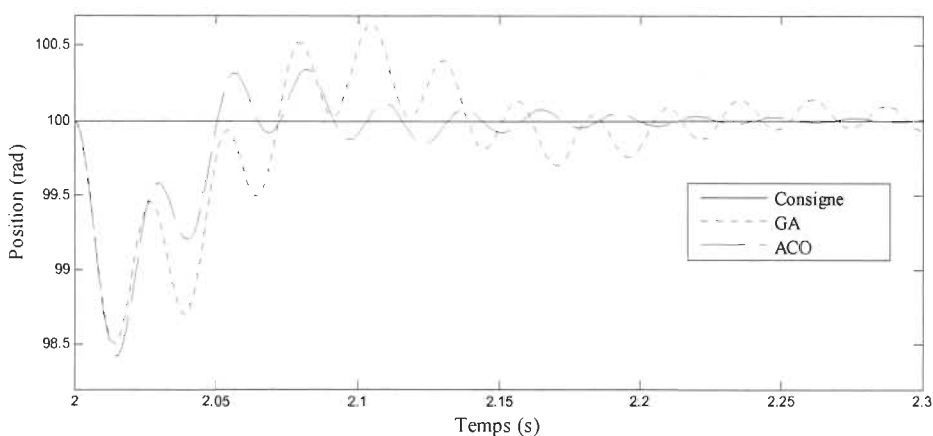


Figure 4-7 Agrandissement de la réponse à la perturbation de la Figure 4.6.

Tableau 4-2 Mesures de performance pour le deuxième scénario de simulation.

Méthode	$K_t=0$		$K_t=0.075$		$K_t=0.75$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.222	0.214	0.213	0.217	0.196	0.196
$t_p$ (s)	0.234	0.218	0.220	0.219	0.221	0.221
$M_p$ (%)	0.500	0.400	0.600	0.100	9.020	9.700
$t_s(2\%)$ (s)	0.202	0.193	0.194	0.194	0.283	0.329
$M_{pert}$ (rad/Nm)	9.253	2.080	2.453	2.520	2.107	2.000
$t_{pert}$ (s)(1%)	0.266	0.047	0.070	0.070	0.022	0.044
$I_{Erreur}$	0.107	0.104	0.116	0.117	0.167	0.213

Pour avoir un contrôle précis sur le système, l'amélioration de la réponse à la perturbation doit être directement liée à l'augmentation de la valeur  $K_t$ . Selon le Tableau 4-2, cette tendance semble se dessiner avec ACO, mais pas avec GA. Avec l'ajustement par ACO, la déviation maximale normalisée,  $M_{pert}$ , et le temps de stabilisation,  $t_{pert}$ , décroissent graduellement selon l'augmentation de la valeur de  $K_t$  alors que le pourcentage de dépassement,  $M_p$ , augmente. Alors, selon le critère de performance choisi, la syntonisation par ACO semble être plus en mesure d'imposer un certain comportement à la réponse du système tout en conservant des performances dynamiques satisfaisantes.

Comme troisième scénario de simulation, le contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur la vitesse du moteur, illustré à la Figure 2-9, est utilisé pour améliorer l'ensemble de la réponse du système. En effet, le dépassement excessif présent dans la réponse transitoire (Figure 4-6) est causé par l'emballement de l'intégrateur dans le régulateur de position dû à la saturation naturelle de la vitesse du moteur. Alors, des limitations internes sur la vitesse du moteur doivent être considérées. Elles seront pénalisées dans l'indice de performance (3-5) via  $\beta$ . De plus, lors de l'optimisation, pour éviter que la réponse soit dominée par la grande entrée échelon, 100 rad, une petite variation échelon, 5 rad, est appliquée à deux secondes. La perturbation de 20% du couple nominal est maintenant appliquée au système à quatre secondes. Dans (3-5),  $\beta = 100$ ; cette valeur plus élevée permet d'implémenter une contrainte rigide sur la vitesse du moteur, alors qu'une petite valeur implémenterait une contrainte plus souple. Conséquemment, dans (3-5), la valeur de  $\sigma = 100$  est employé pour également imposer une contrainte rigide sur le dépassement de position. La variable  $K_t$  prend les valeurs suivantes: 0, 0.075, 0.750. Les paramètres d'optimisation des deux techniques sont identiques au scénario de simulation précédent. La Figure 4-8 et la Figure 4-9 démontrent que l'ajustement par ACO est plus efficace que GA pour cette structure de commande pour un  $K_t = 0.075$ . En effet, la réponse à l'échelon est plus rapide et l'impact de la perturbation sur la réponse système est plus faible. Le Tableau 4-3 résume les mesures de performance pour la réponse transitoire à l'échelon et à la perturbation.

Les courbes des réponses système sont en concordance avec l'augmentation de  $K_t$ ; la réponse à la perturbation est améliorée au coût de la réponse transitoire de l'échelon d'entrée. L'ajustement par ACO génère de meilleurs résultats pour 2 des 3 valeurs de  $K_t$ .

( $K_t = 0$ ,  $K_t = 0.075$ ). En fait,  $t_r$ ,  $t_p$  et  $t_s$  sont deux fois plus rapide,  $M_{pert}$  est réduit de moitié et  $t_{pert}$  est approximativement six et trois fois plus court respectivement. Quand  $K_t = 0.75$ , la réponse à l'échelon est meilleure avec ACO alors que la réponse à la perturbation est meilleure avec GA.

Dans l'ensemble la méthode ACO est plus efficace que GA pour l'ajustement de cette structure de commande. La Figure 4-10 montre l'efficacité du critère de performance à limiter la vitesse du moteur. ( $\Omega_{Mm} = 386 \text{ rad/s}$ ).

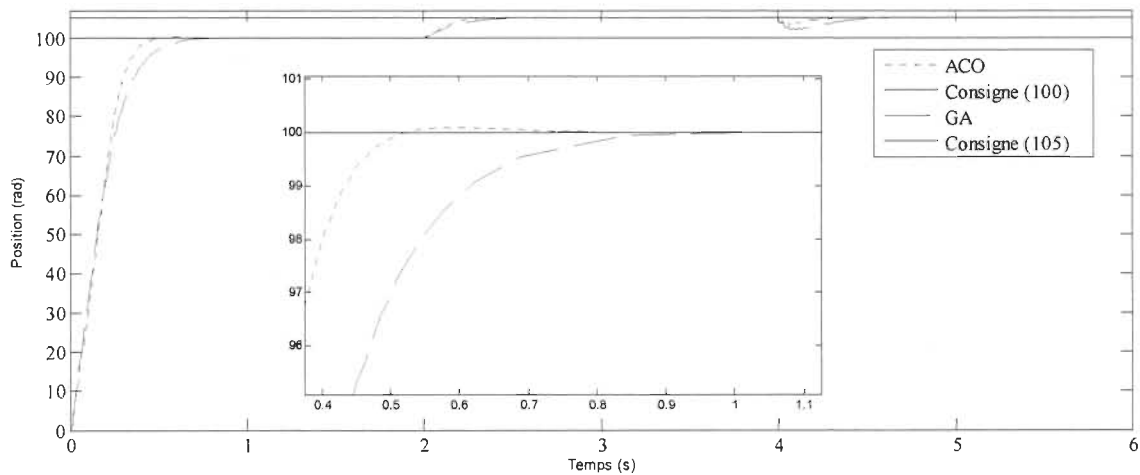


Figure 4-8 Agrandissement de la réponse entrée échelon et à la perturbation pour le contrôleur pondéré  $PID_F$  avec limitation de vitesse du moteur ( $\sigma = 2$ ,  $K_t = 0.075$ ) - ACO vs GA.

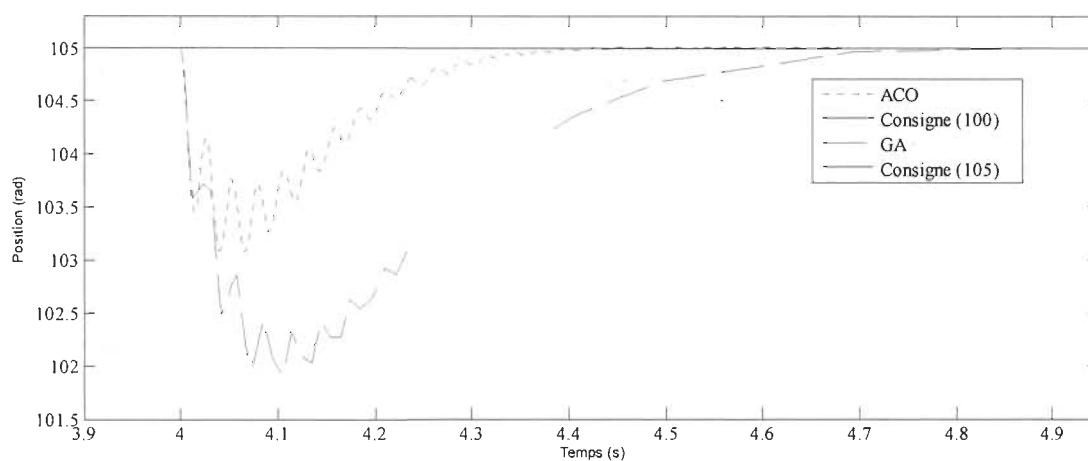


Figure 4-9 Agrandissement de la réponse à la perturbation de la Figure 4.8.

Tableau 4-3 Mesures de performance pour le troisième scénario de simulation.

Méthode	$K_t=0$		$K_t=0.075$		$K_t=0,75$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.564	1.251	0.524	1.042	0.943	1.126
$t_p$ (s)	0.564	1.251	0.537	1.042	0.943	1.255
$M_p$ (%)	0	0	0.100	0	0	0
$t_s(2\%)$ (s)	0.400	0.750	0.400	0.547	0.475	0.514
$M_{pert}$ (rad/Nm)	2.533	4.800	2.533	4.133	2.000	1.467
$t_{pert}$ (s)(1%)	0.132	0.587	0.151	0.353	0.098	0.051
$I_{Erreur}$	0.321	0.445	0.354	0.450	0.483	0.457

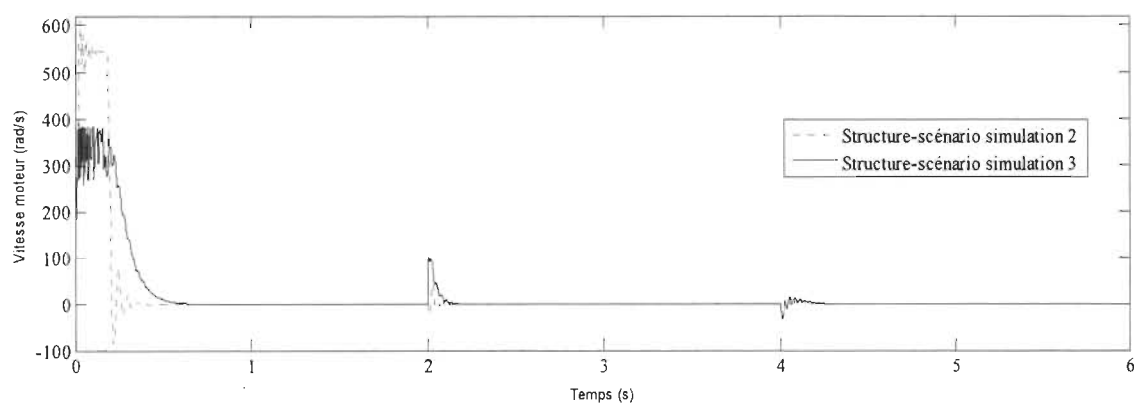


Figure 4-10 Vitesse du moteur avec et sans compensation de saturation.



### 4.1.3 Conclusion

Ces trois scénarios de simulation montrent l'efficacité des ACO à ajuster différentes structures de contrôle. Ceci démontre leur souplesse concernant le nombre de paramètres à optimiser ainsi que l'étendue de leur application. Également, par le biais de variation du critère de performance, il est possible d'imposer un certain comportement à la réponse du système. Leur performance a été également validée par comparaison avec la syntonisation par GA. Dans l'ensemble, la méthode ACO a montré une plus grande efficacité et a fourni de meilleurs résultats. Pour les mêmes conditions, les réponses du système avec les ACO semblent être plus corrélées avec le critère de performance, ce qui est un avantage substantiel. Cependant, le temps de calculs des ACO dépend entre autres du nombre de fourmis  $m$  au sein de la colonie et du nombre d'itérations  $n$  choisi. Ces deux variables dépendent du nombre de paramètres à optimiser. Ainsi, plus il y a de termes, plus le nombre de fourmis  $m$  et d'itérations  $n$  tendent à être élevé. Dans le cadre de ce travail, les paramètres de la méthode ACO ont été raisonnablement choisis selon des résultats de simulation préliminaires avec différentes valeurs de  $m$  et  $n$  et selon les résultats obtenus avec le *TSP*. Selon les paramètres qui ont été choisis, le temps de calcul tend tout de même à être long, soit environ (i) 1 500 s pour la structure de la Figure 2-7; (ii) 10 900 s pour celle de la Figure 2-8; et (iii) 16 351 s pour la Figure 2-9. De ce fait, son temps de calculs excessif rend la technique moins intéressante à utiliser seule.

## 4.2 Ajustement de structures de commande par NM

Cette section présente les résultats de simulation de l'ajustement de la structure de commande simple (Figure 2-7) effectué avec la méthode NM de base. Un des intérêts d'utiliser NM est que l'espace de recherche est continu, alors que ACO nécessite un

espace de recherche discrétisé, limitant ainsi la qualité de la solution optimale. La qualité de la convergence de NM dépend des conditions de recherche initiales [LUE03][BLO13c]. Puisque la grandeur de chaque paramètre est inconnue, il peut être difficile d'établir un bon point de départ. Ainsi, plusieurs simulations ont été réalisées avec différents points de recherche initiaux afin de mesurer leur impact.

#### 4.2.1 Ajustement de la structure simple - points de départ variés

Au sein d'un contrôleur, chaque paramètre joue un rôle distinct sur la réponse du système. Les coefficients  $K_{CP}$ ,  $T_{dP}$  et  $T_{iP}$  sont ceux qui ont le plus d'impact sur la qualité de la réponse du système. Ainsi, comme premier scénario de simulation, ces trois valeurs sont variées afin de former différents points de départ pour démarrer la recherche NM de base. En fait, huit combinaisons différentes de  $K_{CP}$ ,  $T_{dP}$  et  $T_{iP}$  ont été testées afin de mesurer les répercussions sur la qualité de la réponse du système. Ces combinaisons sont basées sur les limites des valeurs minimales et maximales de l'espace de recherche des ACO alors que les autres paramètres ont été initialisés aux valeurs médianes de l'intervalle de recherche des ACO. Les paramètres de (3-5) sont :  $\beta = 0$ ,  $\sigma = 1$  et  $K_t = 0$  (aucune perturbation n'est introduite). La Figure 4-11 présente la piètre qualité des solutions trouvées par NM pour le système à l'étude; seulement quelques résultats sont présentés. Pour la majorité des paramètres initiaux, NM converge vers des ensembles solutions comprenant des valeurs négatives (Tableau 4-4 et Tableau 4-5). Pour la présente application, des valeurs négatives pourraient amener le système en instabilité ou en opération inappropriée. Un mécanisme devrait être mis en place pour générer un bon ensemble de départ pour la méthode NM de base. D'ailleurs, le temps moyen de calcul dépend fortement du point de recherche initial. En moyenne,

pour de mauvaises conditions initiales, il est tout de même significatif (Figure 4-12).

D'après les résultats obtenus pour le système à l'étude, appliquer NM de base seul pour syntoniser un contrôleur n'est pas efficace.

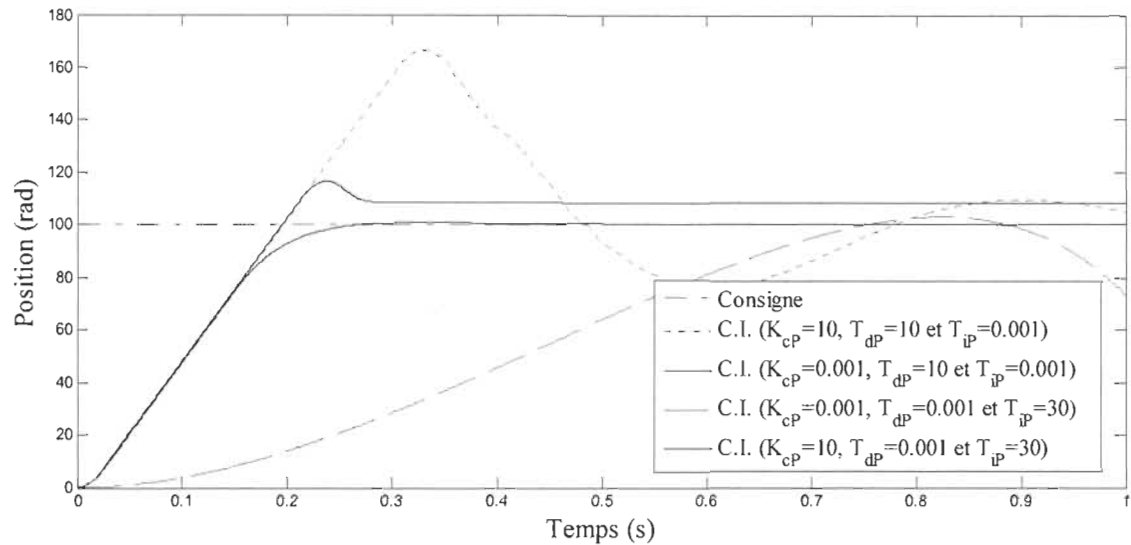


Figure 4-11 Réponse à l'échelon pour le contrôleur  $PID_F$  avec retour d'anti-saturation ( $\sigma = 1$ ) en fonction de conditions initiales (C.I.) variées-méthode NM de base.

Tableau 4-4 Résultats de simulation selon différents points de départ de recherche NM.

Scénarios Paramètres	1		2		3		4	
	Point de départ	Résultat	Point de départ	Résultat	Point de départ	Résultat	Point de départ	Résultat
$K_{cP}$	0.001	0.003	0.001	0.003	0.001	0.034	0.001	0.003
$T_{dP}$	0.001	1.54e-4	0.001	9.77e-4	10.000	1883.900	10.000	25.772
$T_{iP}$	0.001	0.001	30.000	0.835	30.000	3.542	0.001	0.251
$\gamma$	29.460	-0.009	29.460	-0.007	29.460	18.833	29.460	683.240
$K_{vel}$	0.009	0.014	0.010	5.59e-4	0.009	0.069	0.009	0.0467
$K_{SatI}$	2.710	2.239	2.710	1.447	2.710	-75.368	2.710	-284.955
$K_{SatP}$	2.710	2.228	2.710	5.720	2.710	35.350	2.710	-10.187
Temps de convergence (s)	-	406.510	-	203.370	-	478.000	-	670.050
$I_{Erreur}$	-	13.129	-	43.350	-	10.372	-	11.040

Tableau 4-5 Résultats de simulation selon différents points de départ de recherche NM (suite).

Scénarios Paramètres	5		6		7		8	
	Point de départ	Résultat	Point de départ	Résultat	Point de départ	Résultat	Point de départ	Résultat
$K_{CP}$	10.000	30.050	10.000	5.591	10.000	0.104	10.000	0.203
$T_{dP}$	0.001	0.001	10.000	5.676	0.001	0.0013	10.000	10.400
$T_{iP}$	0.001	4.44e-05	0.001	0.001	30.000	35.356	30.000	45.162
$\gamma$	29.460	-0.054	29.460	103.718	29.460	32.055	29.460	0.005
$K_{vel}$	0.009	-0.0145	0.009	-0.015	0.009	0.010	0.009	0.0190
$K_{SatI}$	2.710	4.608	2.710	-2.353	2.710	3.301	2.710	3.795
$K_{SatP}$	2.710	8.159	2.710	3.036	2.710	2.309	2.710	1.398
Temps de convergence (s)	-	165.840	-	789.890	-	62.250	-	270.630
$I_{Erreur}$	-	25.270	-	23.567	-	11.330	-	10.480

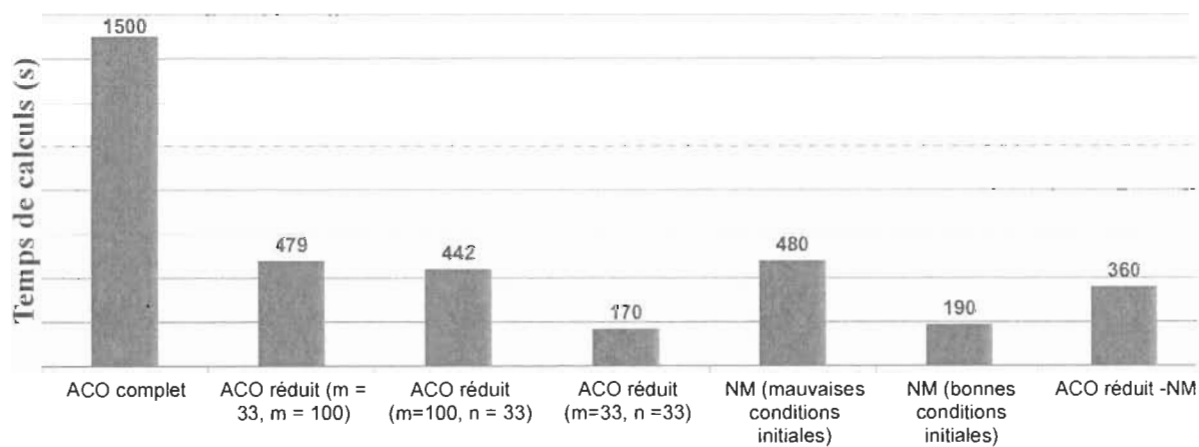


Figure 4-12 Temps moyen de calculs pour différentes stratégies d'optimisation.

#### 4.2.2 Ajustement de la structure simple - points de départ ACO

Tel que démontré à la section précédente, ACO permet une excellente syntonisation de plusieurs structures de commande au prix d'un temps de calculs considérable pour

converger vers une solution désirée. Une façon de diminuer ce temps de calculs est de réduire le nombre de fourmis,  $m$ , et d'itérations,  $n$ . Ainsi, comme deuxième scénario de simulation, trois combinaisons ont été testées : 1)  $m = 33$  et  $n = 33$ , 2)  $m = 33$  et  $n = 100$  et 3)  $m = 100$  et  $n = 33$ . Le temps de calculs est effectivement réduit (Figure 4-12). Cependant, la Figure 4-13 montre la qualité médiocre des solutions obtenues. Ceci est dû au fait que les fourmis ne peuvent converger vers une solution adéquate en un faible nombre d'itérations ou si leur colonie est trop petite.

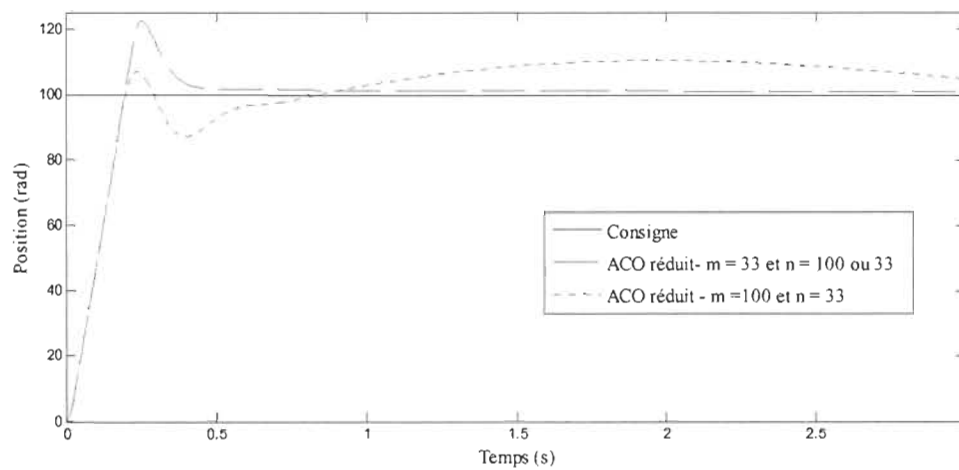


Figure 4-13 Réponse à l'échelon pour la syntonisation de la structure simple – ACO réduit ( $\beta = 0$ ,  $\sigma = 1$  et  $K_t = 0$ ).

Ainsi, dans le but d'atteindre une solution optimale avec un temps de calcul réduit, un nouveau scénario de simulation a été entrepris avec une combinaison élémentaire des ACO et NM où l'environnement de simulation est représenté à la Figure 4-14.

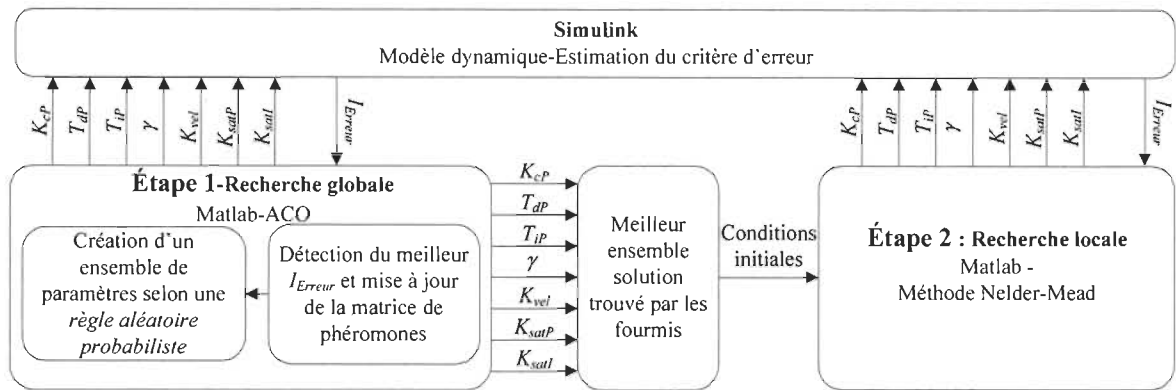


Figure 4-14 Environnement de simulation de l'algorithme ACO réduit-NM.

Inspiré par [HAG12], les ACO sont utilisés pour explorer un grand espace de recherche pour trouver un ensemble de départ suffisamment bon pour démarrer efficacement la recherche locale de NM de base. Pour obtenir un temps de calcul le plus faible possible, la recherche par les ACO se fait avec  $m = 33$  et  $n = 33$  et la solution trouvée initialise la recherche NM. Il est important de mentionner que le nombre de pas de discrétisation  $r$  doit être réduit également; il est donc fixé à 25 au lieu de 75. En effet, la Figure 4-15 exhibe les résultats obtenus avec ACO réduit - NM de base selon deux nombres de pas de discrétisation alors que la Figure 4-12 présente la diminution significative du temps de calculs. Un ratio est observé entre le nombre de fourmis  $m$ , le nombre d'itérations  $n$  et le nombre de pas de discrétisation  $r$  afin de générer un bon point de départ pour NM. Par contre, ce ratio peut varier selon le type de structure de commande diminuant ainsi l'intérêt de cette combinaison.

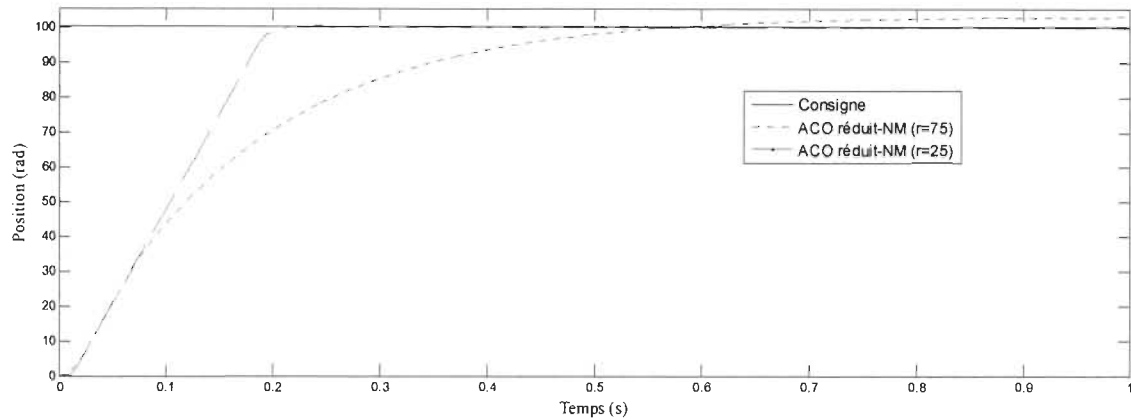


Figure 4-15 Réponse à l'échelon pour ACO réduit-NM selon deux pas de discrétisation.

#### 4.2.3 Conclusion

En résumé (à partir de la Figure 4-12), l'efficacité des ACO avec  $m = 100$  et  $n = 100$  a été obtenue au prix d'un temps de calculs excessif parmi tous les scénarios de simulation. Lorsque  $m$  et/ou  $n$  sont réduits, le temps de calculs diminue drastiquement. Par contre, les solutions obtenues sont de faible performance dynamique. Quant à NM seul, le temps moyen de calculs est environ le tiers de celui des ACO et ce dernier est étroitement lié aux conditions initiales de NM. De plus, étant non-contraint, NM ne garantit pas une convergence vers un ensemble solution acceptable (paramètres positifs seulement) pour le système à l'étude. La combinaison simple de ACO réduit – NM de base permet d'obtenir une meilleure solution que ACO seul en un temps de calculs 75% plus rapide. Toutefois, fournir un bon point de départ à NM pour converger rapidement vers une solution désirée nécessite une paramétrisation précise de ACO, laquelle est sujette à changement d'une structure de contrôle à l'autre. Ainsi, dans le but d'étendre facilement l'application d'un algorithme de syntonisation de structure de commande tout en s'assurant d'obtenir un

ensemble solution positif, il est essentiel de réduire la précision de paramétrisation nécessaire des ACO et de contraindre la méthode NM de base.

### 4.3 Ajustement de structure de commande – algorithme contraint proposé

Cette section présente les résultats de l'algorithme contraint proposé à base des ACO et NM contraint (test de positivité présenté à la sous-section 3.2.2) appliqué à la syntonisation de quatre structures de commande différentes. Tout comme à la section 4.1, l'espace de recherche pour les ACO est défini à partir d'un ajustement par placement de pôles de la structure de commande simple où le système est modélisé comme étant rigide. L'espacement entre chaque valeur de paramètres est logarithmique et est défini (certains paramètres sont inutilisés dépendamment de la structure de contrôle à syntoniser) :

$$K_{SatP}: [0.01, 1000]; \quad T_{iP}: [0.001, 30]; \quad W_p: [0, 1]; \quad K_{cP}: [0.001, 10]; \quad T_{dP}: [0.001, 10]; \\ K_{SatI}: [0.01, 1000]; \quad \gamma: [10, 50]; \quad W_v: [0, 1]; \quad K_{vel}: [0.001, 0.1]; \quad K_{SatV}: [0.01, 1000]$$

L'intervalle de recherche des ACO doit être suffisamment grand pour inclure la région de l'optimum global. Puisque NM est contraint pour éviter les valeurs négatives, la limite minimale est fixée à  $\varphi = 0.0001$  pour tous les paramètres et aucune limite maximale n'est imposée.

#### 4.3.1 Ajustement de structure simple

Le premier scénario de simulation est appliqué au contrôleur  $PID_F$  de rétroaction de position illustré à la Figure 2-7. Pour le critère de performance (3-5), la valeur des variables sont :  $\beta = 0$ ,  $\sigma = 1$  et  $K_f = 0$  (aucune perturbation n'est appliquée). Maintenant, ACO explore le grand intervalle de recherche dans le but de fournir plusieurs points de départ



pour démarrer NM contraint. Pour de nombreux cas de combinaisons d'algorithmes [DOR04], effectuer simultanément une recherche locale et globale est efficace. Par contre, pour ce contrôleur appliqué au système à l'étude, ce n'est pas le cas. Appliquer ACO et NM contraint en parallèle serait moins efficace qu'une approche séquentielle. En effet, la Figure 4-16 montre le temps de calculs des deux méthodes. Tel que mentionné, pour NM contraint, le temps de convergence est directement lié à son point de recherche initial. Pour un bon point de départ, l'algorithme converge rapidement vers la solution souhaitée alors que la convergence est lente avec un point de départ médiocre. Le temps moyen de convergence de NM contraint démarré avec des solutions intermédiaires des ACO est de 300 s. Ceci représente un temps de calculs significatif sachant que les ACO ( $m = 100$  et  $n = 100$ ) prennent environ 1 500s à converger vers une solution de bonne qualité. Alors, environ quatre recherches de NM contraint peuvent être lancées durant l'exploration des fourmis afin de conserver le même temps de calculs que ACO seul. Cette procédure ne représente donc pas une amélioration significative comparativement à ACO seule et le temps de calculs serait toujours significatif. Tel que montré, à la Figure 4-16, pour les ACO, le temps de calculs est lié au nombre d'itérations  $n$  et de fourmis  $m$ . Une réduction de ces deux variables fait descendre drastiquement le temps de convergence de l'algorithme<sup>3</sup>. Dans ce cas-ci, il devient donc intéressant de procéder NM contraint après une exploration d'un petit nombre de fourmis et d'itérations.

---

<sup>3</sup> Il est important de diminuer également le nombre de pas de discrétisation  $r$  au sein de l'espace de recherche des ACO.

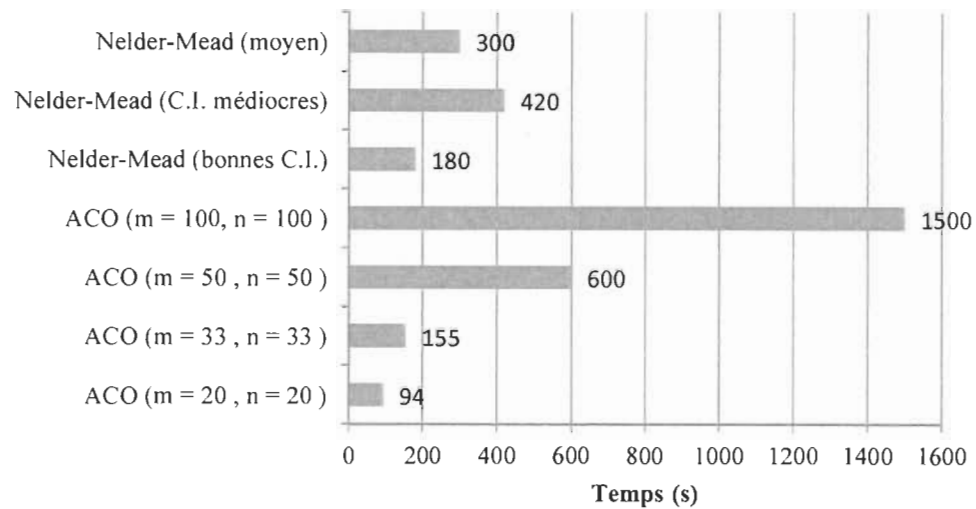


Figure 4-16 Temps moyen de calcul pour NM contraint et ACO.

Ces résultats mènent au deuxième scénario de simulation où trois combinaisons d'algorithmes à base des ACO et NM sont comparées. Les trois algorithmes sont :

1. Algorithme contraint proposé;
2. Combinaison ACO réduit- NM contraint par une valeur fixe (le paramètre prend la valeur de sa contrainte lorsqu'il la dépasse);
3. Combinaison élémentaire ACO réduit-NM de base (section 4.2.2) [BLO13c].

Dans le but d'obtenir un très faible temps de calcul,  $m = 20$  et  $n = 20$ . À partir de la solution des fourmis, l'algorithme NM construit son simplexe initial. Ainsi, aucune règle de démarrage de NM contraint n'est utilisée (Figure 3-7). Le Tableau 4-6 compare les résultats obtenus selon les trois algorithmes de syntonisation. L'ensemble des paramètres trouvé par la combinaison élémentaire ACO réduit-NM de base possède une valeur négative pour  $K_{sat}$ . Pour le système à l'étude, cette composition ne peut être acceptée pour assurer une opération adéquate du retour d'anti-saturation. Ceci démontre donc la nécessité de

contraindre NM. Les deux autres algorithmes contraints convergent vers des groupes de paramètres de valeurs similaires pour une valeur,  $I_{Erreur}$ , identique.

Tableau 4-6 Valeurs des paramètres selon trois algorithmes de syntonisation.

Paramètres	Algorithme contraint proposé	ACO réduit-NM contraint (valeur fixe)	ACO réduit -NM de base
$K_{cP}$	0.491	0.499	0.489
$T_{dP}$	0.016	0.016	0.017
$T_{iP}$	12.249	11.736	12.363
$\gamma$	28.617	27.909	21.043
$K_{vel}$	0.035	0.036	0.039
$K_{SatI}$	0.034	0.944	<b>-29.861</b>
$K_{SatP}$	0.584	0.593	0.679
$I_{Erreur}$	10.387	10.387	10.384

La Figure 4-17 présente le temps de simulation requis par les trois algorithmes pour qu'ils convergent vers un ensemble solution de bonne qualité. Il est important de mentionner que ACO réduit-NM de base nécessite plus de fourmis,  $m$ , et d'itérations,  $n$ , pour obtenir un ensemble positif de paramètres (sous-section 4.2.2) comparativement à  $m = 20$  et  $n = 20$  pour les deux algorithmes contraints. Une réduction substantielle est atteinte avec l'algorithme contraint proposé; une amélioration de 27 % du temps de calculs est obtenue par rapport à ACO réduit-NM de base; 14 % plus rapide que ACO réduit-NM contraint par une valeur fixe et; 5.7 fois plus rapide qu'avec ACO seul. L'algorithme contraint proposé est efficace répondant ainsi aux besoins d'avoir une méthode contrainte rapide et fiable.

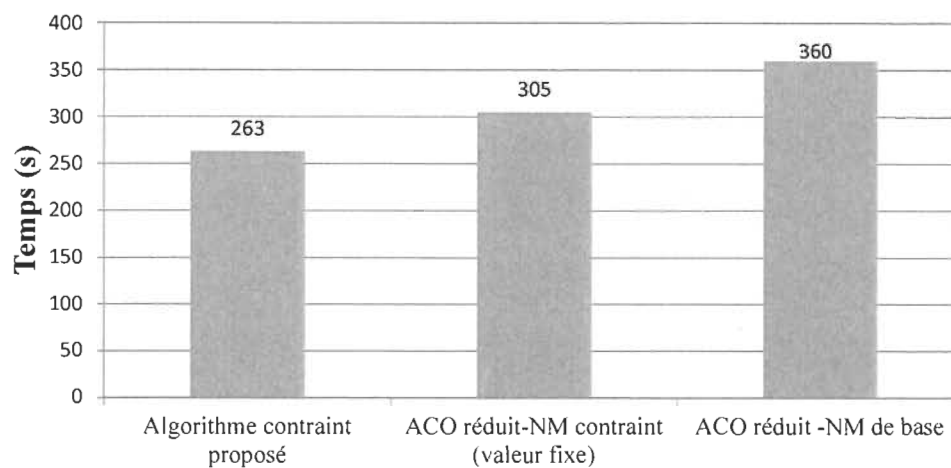


Figure 4-17 Temps de simulation pour les trois algorithmes de syntonisation.

La Figure 4-18 exhibe la réponse du système obtenue par l'algorithme contraint proposé. La réponse est rapide et pratiquement sans dépassement satisfaisant ainsi les exigences des opérations de meulage.

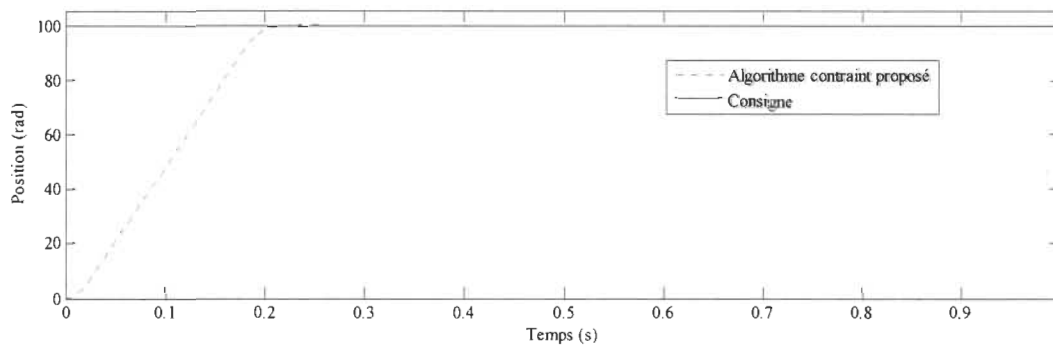


Figure 4-18 Réponse à l'échelon pour le contrôleur  $PID_F$  avec retour d'anti-saturation pour le positionnement de la table.

#### 4.3.2 Ajustement de structures complexes

Dans cette sous-section, les trois algorithmes ajustent trois différentes structures de commande complexes. La même perturbation qu'à la section 4.1.2 est appliquée au système, soit 20 % du couple nominal (0.75 Nm).

Comme premier scénario de simulation de structure complexe, la structure de commande est celle présentée à la Figure 2-8 (contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur le courant). Dans l'indice de performance (3-5),  $\beta = 0$ ,  $\sigma = 2$  et  $K_T = 0.075$ . Pour les ACO, le nombre de combinaisons possibles est relié au nombre de paramètres à ajuster et au nombre de niveaux de quantification  $r$ . Alors, puisqu'il y a deux paramètres supplémentaires à syntoniser, les valeurs de  $m$ ,  $n$  et  $r$  doivent être augmentées comparativement à l'ajustement de la structure simple. Les valeurs sont  $m = 200$ ,  $n = 75$  et  $r = 50$ , ce qui est significativement plus faible que les ACO seuls où  $m = 200$ ,  $n = 200$  et  $r = 75$ . Les valeurs plus élevées de  $m$  et  $n$  (comparativement au scénario de la structure simple) permettent donc d'initialiser efficacement des recherches locales à partir de points intermédiaires trouvés par ACO. Les trois algorithmes préalablement présentés sont comparés. NM contraint ou non est lancé à partir de la meilleure solution trouvée par les ACO aux itérations  $n = 1, 25, 50$  et  $75$  (règle de démarrage). Les meilleurs résultats intermédiaires ont été obtenus à  $n = 25$  et sont présentés dans le Tableau 4-7. Une fois de plus, la nécessité d'avoir une méthode contrainte est démontrée par l'ensemble trouvé par la combinaison ACO réduit-NM de base : deux termes négatifs doivent être rejetés pour éviter d'avoir des zéros dans le demi-plan droit et afin d'assurer l'activation convenable du retour d'anti-saturation.

Tableau 4-7 Valeurs des paramètres des trois algorithmes de syntonisation au résultat intermédiaire  $n = 25$ .

Paramètres	Algorithme contraint proposé	ACO réduit-NM contraint (valeur fixe)	ACO réduit -NM de base
$K_{cP}$	3.841	3.859	4.707
$T_{dP}$	0.001	0.001	0.0001
$T_{iP}$	0.116	0.116	0.123
$\gamma$	22.842	22.050	57.405
$K_{vel}$	0.018	0.016	0.015
$K_{SatI}$	0.092	0.186	<b>-33.537</b>
$K_{SatP}$	89.421	122.222	252.884
$W_p$	0.198	0.190	0.053
$W_v$	0.531	0.480	<b>-0.326</b>
$I_{Erreur}$	0.117	0.116	0.115
Temps (s) de convergence NM	84	222	208

La valeur de  $I_{Erreur}$  obtenus par les algorithmes contraints est similaire. Cependant, la façon de contraindre NM impacte significativement sur le temps de convergence. Conséquemment, comme la structure simple, le temps de convergence est plus rapide (plus de 2.5 fois en rapidité) avec le test de positivité proposé. La Figure 4-19 et la Figure 4-20 illustrent le résultat intermédiaire de l'algorithme contraint proposé avec sa valeur finale comparée à la solution trouvée par les ACO seuls (section 4.1) [BLO13b]; les temps de convergence sont également présentés. Le résultat intermédiaire est obtenu presque 4 fois plus rapidement pour une qualité de réponse semblable. Ceci signifie que cet ensemble de paramètres peut être utilisé par le système alors la réponse du système continue d'être améliorée par l'algorithme contraint proposé. La réponse finale obtenue par l'algorithme contraint proposé présente moins de dépassement que les autres, rencontrant les besoins spécifiques pour les opérations de meulage. Qui plus est, cette réponse est produite presque

2 fois plus rapidement que par la méthode ACO pour une meilleure réponse à l'échelon et à la perturbation.

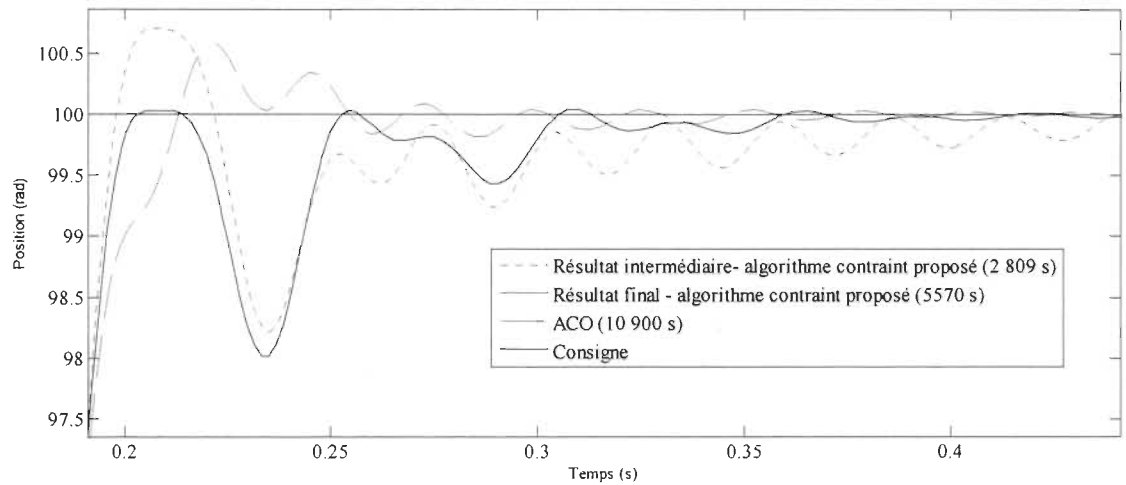


Figure 4-19 Agrandissement de la réponse à l'échelon pour le contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur le courant.

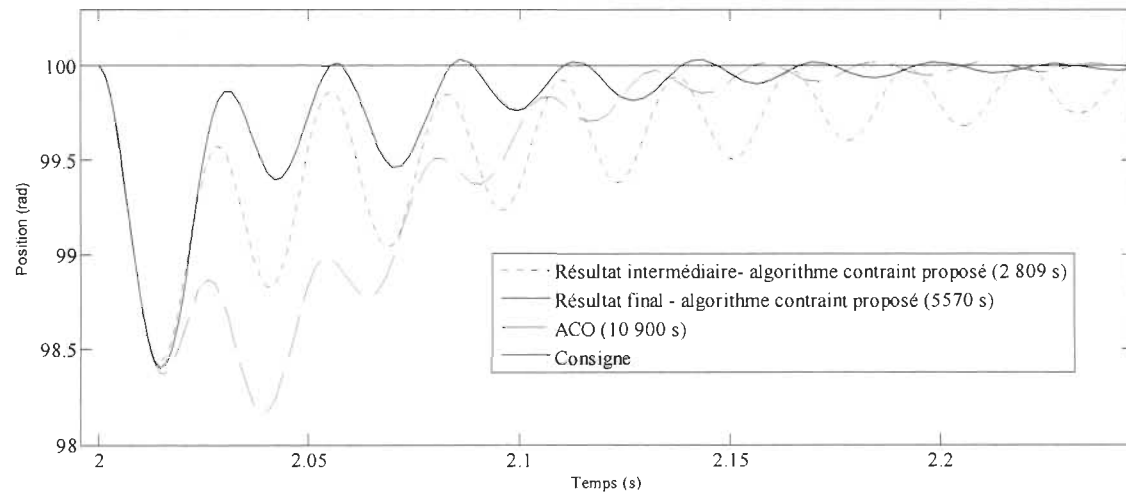


Figure 4-20 Agrandissement de la réponse à la perturbation contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur le courant.

Pour le deuxième scénario de simulation, les mêmes trois algorithmes sont appliqués pour la syntonisation du contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur la

vitesse du moteur (Figure 2-9), qui est utilisé pour régler le phénomène expliqué à la sous-section 4.1.2. En effet, il est nécessaire de pénaliser explicitement les violations de limites de vitesse. La même perturbation et la même variation d'échelon sont introduites dans le système qu'à la sous-section 4.1.2, soit une variation d'échelon de 5 rad à deux secondes et une perturbation de 20% du couple nominal à quatre secondes. Les paramètres du critère de performance (3-5) sont :  $\beta = 100$ ,  $\sigma = 100$  et  $K_t = 0.075$ . Des démarrages intermédiaires de NM contraint ou non sont effectués à la meilleure solution trouvée aux itérations  $n = 1, 25, 50$  et  $75$  (règle de démarrage) Pour ce scénario, aucune valeur négative n'a été obtenue. Ainsi, les trois algorithmes convergent, en un temps de calculs similaire, vers les mêmes solutions pour tous les résultats intermédiaires et finaux. La Figure 4-21 et la Figure 4-22 comparent donc le meilleur résultat intermédiaire et le résultat final obtenus par l'algorithme contraint proposé avec ACO appliqué seul. Le meilleur résultat trouvé par l'algorithme contraint proposé est obtenu à partir d'une solution sous optimale des ACO, soit à l'itération  $n = 50$ , pour démarrer la recherche NM contraint. Cela peut s'expliquer par la dépendance entre la convergence de NM versus son point d'initialisation du simplexe. Ce résultat supporte donc la pertinence de mener des recherches locales par NM contraint durant le processus de recherche global des ACO afin d'obtenir une réponse système de qualité supérieure. Une amélioration significative en temps de calculs est remarquée; la réponse est obtenue 1.67 plus rapidement avec l'algorithme contraint proposé qu'avec la méthode ACO seule pour obtenir une meilleure réponse.



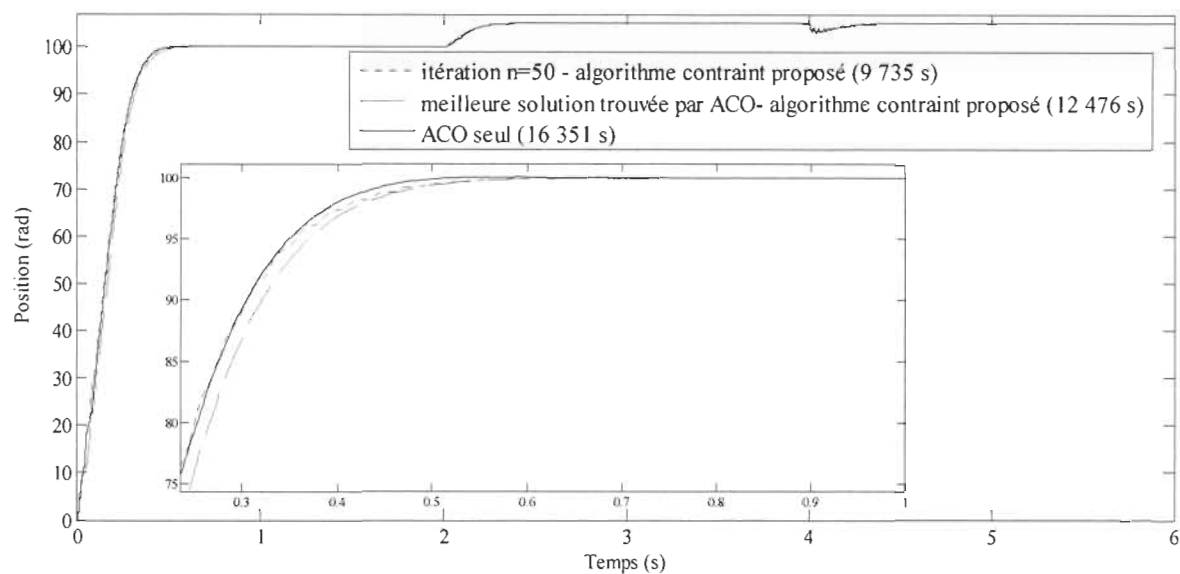


Figure 4-21 Réponse à l'échelon et à la perturbation avec agrandissement de la réponse transitoire pour le contrôleur pondéré  $PID_F$  avec retour d'anti-saturation sur la vitesse du moteur.

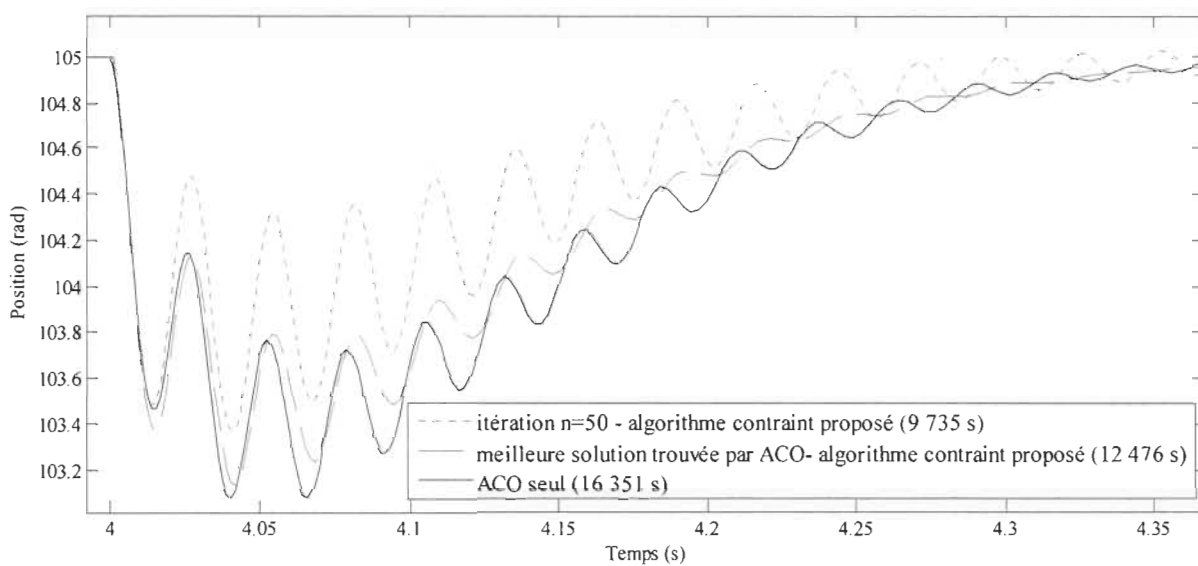


Figure 4-22 Agrandissement de la réponse à la perturbation pour le contrôleur pondéré  $PID_F$  avec retour d'anti-saturation sur la vitesse du moteur.

Comme dernier scénario de simulation, le contrôleur pondéré  $PID_F$  avec retour d'anti-saturation sur la vitesse du moteur ainsi que sur le courant est utilisé (Figure 2-10). Le réglage du scénario de simulation est identique au scénario précédent à la différence que ce contrôleur contient un paramètre supplémentaire à syntoniser pour un total de 10 paramètres. La procédure de contrainte est utile puisque des valeurs négatives sont présentes pour certains résultats intermédiaires. Cependant, les trois algorithmes comparés obtiennent la même meilleure réponse finale pour un temps de calculs comparable (aucune contrainte n'était donc nécessaire dans ce cas). La Figure 4-23 présente cette réponse comparée à celle obtenue avec la structure précédente. La réponse à l'entrée de variation à l'échelon est plus rapide avec cette présente structure et la réponse à la perturbation est mieux gérée. Avantageusement, le fait d'avoir une méthode d'optimisation facilement adaptable à différentes structures permet donc de comparer efficacement et justement les forces et faiblesses de chacune de ces structures afin de sélectionner la meilleure pour l'application visée.

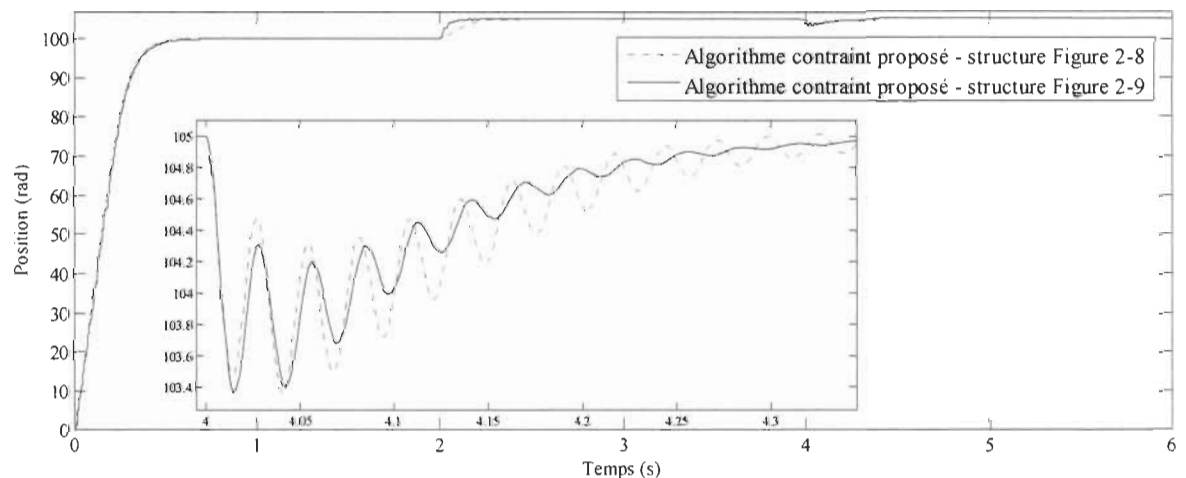


Figure 4-23 Réponse à l'échelon et agrandissement de la réponse à la perturbation pour le contrôleur pondéré  $PID_F$  avec un retour d'anti-saturation sur la vitesse du moteur et sur le courant de référence.

L'ensemble des résultats de simulation montre qu'il est nécessaire d'avoir une méthode d'optimisation contrainte. Le simple fait de restreindre la valeur d'un paramètre à sa limite ne permet pas d'avoir une solution de bonne qualité en un faible temps de calculs. L'algorithme contraint proposé vient donc répondre aux besoins d'avoir une méthode fiable et adaptable à différentes structures de commande avec un critère de performance flexible.

## Chapitre 5 - Conclusion générale

Un nouvel algorithme contraint à base des ACO et de NM a été proposé pour la synthèse de structures de commande simple et complexes pour un système de positionnement flexible. Les résultats de simulation démontrent l'efficacité et les avantages de cette méthode d'ajustement pour quatre structures différentes de contrôleurs PID avec retour d'anti-saturation. Il a été démontré qu'il était essentiel d'avoir un algorithme d'optimisation contraint et que la nouvelle procédure proposée, test de positivité, insérée dans la méthode NM de base permet une syntonisation jusqu'à 2.5 fois plus rapide que de contraindre l'algorithme par valeur fixe. De plus, l'algorithme contraint proposé converge vers une meilleure solution environ 1.67 à 5.7 fois plus rapidement que ACO appliqué seul, dépendamment de la structure. Également, dans certains cas, la procédure de démarrage intermédiaire de NM contraint permet d'obtenir une qualité de réponse supérieure en un temps de convergence plus court. En plus d'un temps de calculs rapide, cet algorithme contraint proposé est combiné à un critère de performance flexible qui permet d'imposer des contraintes souples et rigides au système. L'algorithme contraint proposé possède les avantages escomptés; (i) gestion du retour d'anti-saturation; (ii) adaptable à plusieurs structures; (iii) nombre variable de termes à syntoniser; (ix) critère de performance flexible; (x) méthode contrainte; et (xi) méthode rapide.

Étant donné que l'optimisation de l'ajustement des paramètres est indépendante du contrôleur, cette méthode d'optimisation serait facilement adaptable à d'autres systèmes

ouvrant la voie à plusieurs nouveaux domaines d'application. À court terme, cet algorithme sera appliqué pour ajuster la structure de régulation basée sur *backstepping*.

Le présent travail a permis la publication et les présentations de cinq articles scientifiques à des conférences internationales [BLO13a] [BLO13b] [BLO13c] et [BLO14a] [BLO14b]. Suite à la conférence, l'article [BLO14a] a été sélectionné pour une potentielle publication dans ELECTRIMACS 2014 Special Issue of Elsevier Trans. of IMACS on Mathematics and Computers in Simulation (MATCOM.) pour octobre 2014. De plus, une partie de ce travail de maîtrise a été présenté en tant que conférencière invitée dans le cadre de l'école d'été REM, de l'Université de Lille, en France. Également, des sections ont été présentées au concours d'affiches scientifiques de l'Université du Québec à Trois-Rivières (hiver 2013- 2<sup>ième</sup> place /hiver 2014- 1<sup>ère</sup> place – département de génie électrique).

Pour travaux futurs, cet algorithme et les notions acquises serviront de base pour entamer mon projet doctoral portant sur le développement d'algorithmes pour les appliquer à des problèmes d'optimisation de complexité structurelle croissante : (i) synthèse de correcteurs adaptatifs à base d'intelligence artificielle pour la MSAP visant à minimiser les essais sur les machines; (ii) synthèse de lois de répartition d'effort/puissance pour des systèmes multimachines ou multisources dans un but d'optimisation énergétique; et (iii) synthèse de lois de gestion énergétique dans les véhicules hybrides tenant compte des limites internes, notamment de l'état de charge des batteries. Ces objectifs sont dans le but de rendre l'électrification des transports de plus en plus rentable ce qui est en parfaite concordance avec les récents investissements du gouvernement fédéral dans la recherche et

développement du secteur de l'automobile ainsi qu'avec la politique du gouvernement québécois pour l'électrification des transports.

## Références

- [ARI10] F. L. A. Arito, « Advanced Control of SPWM dc/cc Inverters. Mémoire. San Luis, Argentina », Universidad Nacional de San Luis, 2010.
- [AST06] K. J. Aström et T Häggglund, « Advanced PID Control, » ISA, 2006.
- [AST89] K. J. Aström et L. Rundqwist, « Integrator Windup and How to Avoid It », IEEE, American Control Conference, Pittsburgh, Pennsylvanie, juin 1989.
- [BLO13a] M. J. Blondin et P. Sicard, « ACO Based Controller and Anti-Windup Tuning for Motion System with Flexible Transmission », IEEE Canadian Conference on Electrical and Computer Engineering, Regina, Canada, mai 2013.
- [BLO13b] M. J. Blondin et P. Sicard, « PID controllers and anti-windup systems tuning using Ant Colony Optimization », European Conference on Power Electronics and applications, Lille, France, septembre 2013.
- [BLO13c] M. J. Blondin et P. Sicard, « Combined ACO Algorithm – Nelder-Mead Simplex Search For Controller and Anti-Windup Tuning for Motion System With Flexible Transmission », Conference of the IEEE Industrial Electronics Society , Vienne, Autriche, novembre 2013.
- [BLO14a] M. J. Blondin et P. Sicard, « A Metaheuristic approach for Controller and Anti-Windup Tuning of a Motion System with Flexible Transmission: ACO Algorithm vs Genetic Algorithm », International Conference on Modeling and Simulation of Electric Machines, Converters and Systems, Valence, Espagne, mai 2014.
- [BLO14b] M. J. Blondin et P. Sicard « ACO Algorithm — New Constrained Nelder-Mead Simplex Search Combination for Controller and Anti-Windup Tuning for a Motion System with Flexible Transmission », European Conference on Power Electronics and applications, Lappeenranta, Finlande, août 2014.
- [BOH95] C. Bohn et D. P. Atherton, « An Analysis Package Comparing PID Anti-Windup Strategies », IEEE Control Systems, vol.15, no.2, pp.34-40, avril 1995.
- [CHI12] I. Chiha, N. Liouane, et P. Borne, « Tuning PID Controller Using Multiobjective Ant Colony Optimization », Applied Computational Intelligence and Soft Computing, Hindawi Publishing Corporation, 2012.

- [DOR04] M. Dorigo et T. Stützle, « Ant Colony Optimization », The MIT Press, Cambridge, Massachusetts, 2004.
- [DOR96] M. Dorigo, V. Maniezzo, et A. Colorni, « Ant system: optimization by a colony of cooperating agents », IEEE Transactions on Systems, Man, and Cybernetics-part B Cybernetics, vol. 26, no. 1, pp. 29-41, février 1996.
- [HAU98] L. Haupt, et S. E. Haupt, Practical Genetic Algorithms, Wiley-Interscience, New York, N.Y., 1998.
- [HAG12] M. Haghifam, M.-R. Haghifamand and B.S. Chabok, « State estimation electric distribution networks in presence of distributed generation using the PMUs », CIRED Workshop 2011, Lisbon, mai 2012.
- [HSI04] Y.-T. Hsiao, C.-L. Chuang, et C.-C. Chien, « Ant Colony Optimization for Designing of PID Controllers », IEEE, International Symposium on Computer Aided Control System Design, Taipei, Taiwan, septembre 2004.
- [KUM12] S. Kumar et R. Negi, « A Comparative Study of PID Tuning Methods Using Anti-Windup Controller », IEEE, International Conference on Power, Control and Embedded Systems, Uttar Pradesh, Inde, décembre 2012.
- [LEE08] K. Y. Lee et M. A. El-Sharkawi, « Modern heuristic optimization techniques : Theory and Applications to Power Systems », Wiley-IEEE Press, Hoboken, New Jersey, 2008.
- [LAG98] J. C. Lagarias, J. A. Reeds, M. H. Wright et P. E. Wright, « Convergence properties of the Nelder-Mead simplex method in low dimension », SIAM J. Optimization, Society for Industrial and Applied Mathematics, vol. 9 no.1, pp 112-147, 1998.
- [LIE01] J. Lieslehto "PID controller tuning using Evolutionary programming" IEEE, American Control Conference, Arlington, Virginia, juin 2001.
- [LUE03] M. A. Luersen, R. Le Riche et F. Guyon, « A constrained, globalized, and bounded Nelder-Mead method for engineering optimization », Struc. Multidisc Optim, Springle-Verlag, 2003.
- [NAG10] B. Nagaraj et N. Muruganath, « A comparative study of PID controller tuning using GA, EP, PSO and ACO », IEEE International Conference on Communication Control and Computing Technologies, oct. 2010.
- [NEL65] J. A. Nelder et R. Mead, « A Simplex Method for Function Minimization », Computer Journal, vol. 7, pp. 308-313, 1965.
- [OGA10] K. Ogata, « Modern Control Engineering », Pearson, Prentice Hall, 2010.



- [PAD12] F. Padula et A. Visioli, « Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes », *Control Theory & Applications, IET*, vol. 6, no. 6, pp. 776-786, 2012.
- [SHA11] M. F. N. Shah, M. A. Zainal, A. Faruq et S.S. Abdullah, « Metamodeling approach for PID controller optimization in an evaporator process, » *IEEE, International Conference on Modeling, Simulation and Applied Optimization*, avril 2011.
- [TAJ11] M. Tajjudin, N. Ishak, H. Ismail, M. H. Fazalul et R. Adnan, « Optimized PID control using Nelder-Mead Method for electro-hydraulic actuator system », *IEEE Control and System Graduate Research Colloquium, Shah Alam, Malaisie*, juin 2011.
- [VAR04] H. A. Varol, et Z. Bingul, “A New PID Tuning Technique Using Ant Algorithm,” *IEEE, American Control Conference, Boston, Massachusetts*, juin 2004.
- [VIS06] A. Visioli, « *Practical PID control* », Springer-Verlag London Limited, 2006.
- [WAI11] R.-J. Wai; J.-D. Lee et K.-L. Chuang, « Real-Time PID Control Strategy for Maglev Transportation System via Particle Swarm Optimization, », *IEEE Transactions on Industrial Electronics*, vol.58, no.2, pp.629-646, fév. 2011.

## Annexe A – Articles scientifiques

Les articles scientifiques publiés sont présentés dans cette section selon leur ordre de publication :

1. [BLO13a] M. J. Blondin et P. Sicard, «ACO Based Controller and Anti-Windup Tuning for Motion System with Flexible Transmission, » IEEE Canadian Conference on Electrical and Computer Engineering, Regina, Canada, mai 2013.
2. [BLO13b] M. J. Blondin et P. Sicard, «PID controllers and anti-windup systems tuning using Ant Colony Optimization, » European Conference on Power Electronics and applications, Lille, France, septembre 2013.
3. [BLO13c] M. J. Blondin et P. Sicard, «Combined ACO Algorithm – Nelder-Mead Simplex Search For Controller and Anti-Windup Tuning for Motion System With Flexible Transmission», Conference of the IEEE Industrial Electronics Society , Vienne, Autriche, novembre 2013.
4. [BLO14a] M. J. Blondin et P. Sicard, «A Metaheuristic approach for Controller and Anti-Windup Tuning of a Motion System with Flexible Transmission: ACO Algorithm vs Genetic Algorithm », International Conference on Modeling and Simulation of Electric Machines, Converters and Systems, Valence, Espagne, mai 2014.
5. [BLO14b] M. J. Blondin et P. Sicard « ACO Algorithm — New Constrained Nelder-Mead Simplex Search Combination for Controller and Anti-Windup Tuning for a Motion System with Flexible Transmission », European Conference on Power Electronics and applications, Lappeenranta, Finlande, août 2014.

# ACO BASED CONTROLLER AND ANTI-WINDUP TUNING FOR MOTION SYSTEMS WITH FLEXIBLE TRANSMISSION

*Maude-Josée Blondin and Pierre Sicard*

GREI, Université du Québec à Trois-Rivières, Trois-Rivières (Québec), Canada

## ABSTRACT

Optimal controller and anti-windup tuning is a difficult problem for practical motion systems with flexible coupling and cascaded loops control structure especially when time scale separation is insufficient. An Ant Colony Optimization (ACO) based methodology for global off-line tuning of Proportional-Integral-Derivative (PID) controllers with anti-windup is proposed for such systems. Given the core ACO algorithm, the designer can modify or create new performance indices to fully grasp the desired behavior with multi-objective criteria.

*Index Terms*— ACO, PID Controller, Integrator anti-windup, Optimization, Motion systems

## 1. INTRODUCTION

Many methods and rules (lambda tuning, Ziegler-Nichols, frequency response methods, heuristic methods, etc.) have been proposed for tuning PID controllers, either to obtain initial setting of their parameters or for fine tuning [1]. The tuning rules are typically defined for the basic (ideal) PID controller or for controllers with specific factorization or form. Moreover, the rules typically respond to particular performance criteria (Integral of Absolute value of Error (IAE), Integral of Squared Error (ISE), Integral of Time multiplied by the Absolute value of Error (ITAE), etc.) and do not take into account controller saturation. Rules also do exist to tune anti-windup controllers and it is customary to sequentially tune the PID and the anti-windup controllers. However, there exists close interaction between controller and anti-windup tuning. It was noted that the sequential approach is not efficient [2]. In fact, it may lead to sluggish performance, henceforth the interest of developing a more global approach to tune the PID system.

To cope with this problem, we propose an offline tuning of PID controllers using ACO which offers many advantages. Firstly, tuning may be controller-structure independent. It also allows considering variations of the structure of the controller, such as tuning of input filters, weighted PID controllers, etc. Performance criterion is not fixed and does not require an analytic solution so that specific behaviors such as overshoot can be specifically penalized. Cascaded control loops may also be tuned

simultaneously in particular when time scale separation is not sufficient, which results into a large number of interdependent parameters to be tuned.

In this paper, the ACO based methodology is applied to a motion system with flexible coupling and cascaded loops control structure. We demonstrate the efficiency of ACO to tune simultaneously the position loop PID controller with filtered derivative term, its anti-windup feedback gain and the anti-windup gain of the current control loop. Moreover, we validate the improvement of the response by using a modified IAE performance index to reduce overshoot.

## 2. ANT COLONY OPTIMIZATION

ACO is an optimization technique inspired by the behavior of real ants within a colony [3] [4]. It was observed that a colony facing multiple paths to a food source has the tendency of taking the shortest road. This is explained by the fact that ants communicate with each other via their environment by depositing traces of pheromones. The paths with the most pheromones are more attractive to other ants. Therefore, the amount of pheromones increases more rapidly on the shortest path, attracting more and more ants. Eventually, all ants will take the same path, i.e. the shortest path. Henceforth, the solution emerges from the collective interaction of the ants.

ACO can be used to solve most of the combinatorial optimization problems for which a constructive heuristic can be employed. One of the first problems ACO was applied to is the Traveling Salesman Problem (TSP), which allows illustrating the method. The ants move on the TSP graph and generate different solutions. The probability for an ant  $k$  located in city  $i$  to choose city  $j$  is stated as

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \quad (1)$$

where  $N_i^k$  represents the number of cities to be visited by the ant  $k$ ,  $j \in N_i^k$ ;  $\tau_{ij}(t)$  is the quantity of pheromones deposited between cities  $i$  and  $j$ ;  $\eta_{ij}(t) = 1/d_{ij}$  is the inverse of the distance separating cities  $i$  and  $j$ ;  $\alpha$  and  $\beta$  are parameters that determine respectively the relative weight of pheromone traces and the heuristic information.

At each step, each ant deposits pheromones on its travelled path:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad \forall (i, j) \in L \quad (2)$$

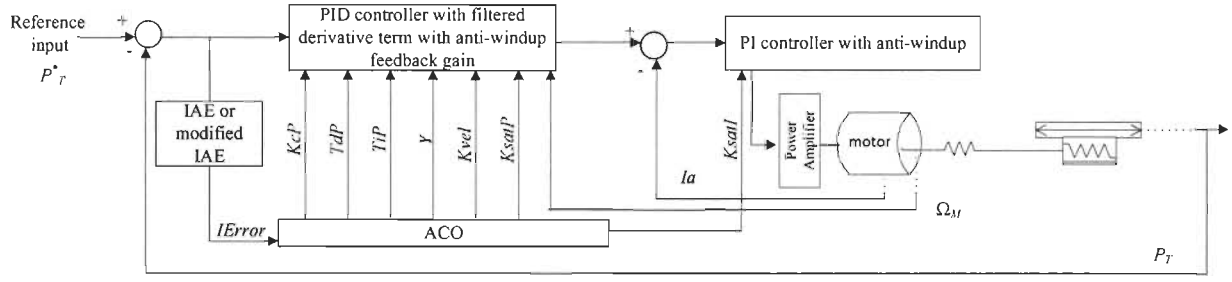


Figure 1- System overview with ACO application for PID parameters tuning

where  $\Delta \tau_{ij}^k(t)$  is the quantity of pheromones deposited on segments that ant  $k$  has visited during this step. The quantity of deposited pheromones on each segment of the graph is function of the quality of the solution:

$$\Delta \tau_{ij}^k(t) = \begin{cases} 1/L^k & \forall (i, j) \in L \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $L$  is the total length of travelled distance for the iteration. The pheromone traces associated to each segment are reduced by a constant evaporation rate  $\rho$ :

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad \forall (i, j) \in L \quad (4)$$

where  $0 < \rho < 1$ . The evaporation process constitutes a forgetting factor used to avoid unlimited accumulation of pheromones and to allow forgetting previous bad decisions.

### 3. ACO FOR PID AND ANTI-WINDUP TUNING

The positioning table of an industrial grinder system is considered in this study. A 3/4 HP dc motor is coupled to the grinder table by a transmission box including a worm-gear. The significant flexible coupling is modeled as an equivalent lumped spring. Performance criteria include high accuracy and good dynamic performance in order to decrease the operation cycle. The flexible coupling complexifies the tuning process to achieve high performance for this cascaded loop control system.

Figure 1 summarizes the system model and ACO's main input and output parameters for PID and anti-windup parameter tuning. The position controller is illustrated in Figure 2, along with its parameters. It is composed of a filtered PID controller with motor velocity feedback and anti-windup controller. The PI current controller's anti-windup feedback gain is identified as  $K_{satI}$  on Figure 1.

To use ACO, controller tuning must be transformed into a combinatorial optimization problem [5]-[7] (Figure 3). Each slice of the graph corresponds to nodes of quantified controller parameters. This representation allows easy addition or removal of free controller coefficients. Range of parameters and quantization steps must be chosen to complete the graph of Figure 3. Rough tuning of the control parameters can be used to define the range of parameters.

The ants move on the graph from left to right by choosing the nodes they travel from probabilistic rules. They leave pheromone traces as a function of the quality of the solution.

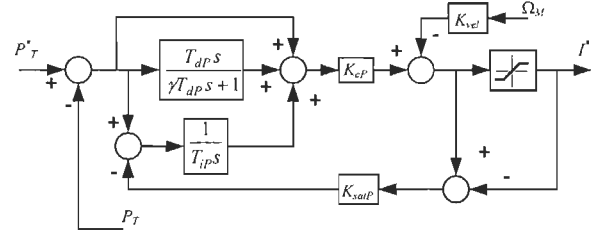


Figure 2- Position feedback controller

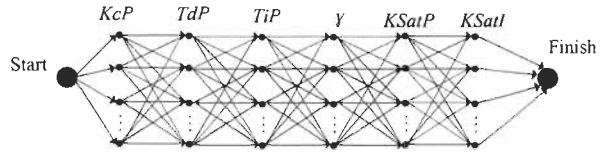


Figure 3- Optimization graph for filtered position PID controller and anti-windup coefficients

The performance is evaluated by  $IError$  index resulting from the set of parameters on the travelled path. The ants work together to find the best set of parameters to minimize the performance criterion  $IError$  defined as the modified Integral of Absolute value of Error (IAE) index:

$$IError = \int_0^{T_s} |e(t)| dt + \sigma \int_0^{T_s} e_d(t) dt \quad (5)$$

where  $e(t) = P_r^* - P_T(t)$  is the output error;  $P_T$  and  $P_r^*$  are respectively the measured and reference positions of the grinder table;  $T_s$  is the settling time;  $\sigma$  is a positive constant weight and

$$e_d(t) = \begin{cases} |e(t)| & \text{during response overshoots} \\ 0 & \text{otherwise} \end{cases}$$

The performance index is chosen to emphasize the desired response or to penalize undesired behavior. For  $\sigma=0$ , equation (5) corresponds to the IAE index; larger values of  $\sigma$  are used to penalize the overshoot, which is very detrimental to the quality of the grinding operations.

### 4. SIMULATION RESULTS

Figure 4 shows the simulation environment used to apply ACO to the system described in the previous section.

The position PID controller's parameters were initially determined by pole placement from the approximate rigid model of the system and the following ranges were chosen for the parameter coefficients for the search:

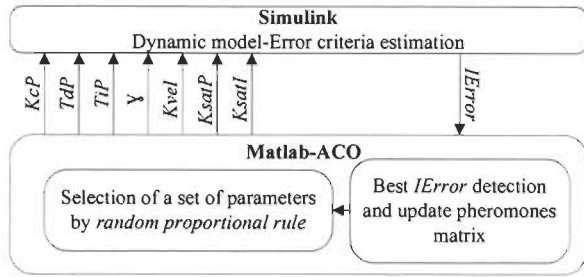


Figure 4- Simulation environment structure of ACO iteration

$$\begin{array}{ll}
 K_{SatP}: [0,01 \text{ to } 1000] & K_{SatI}: [0,01 \text{ to } 1000] \\
 K_{cP}: [0,001 \text{ to } 10] & T_{iP}: [0,003 \text{ to } 30] \\
 T_{dP}: [0,001 \text{ to } 10] & \gamma: [10 \text{ to } 50]
 \end{array}$$

In a first simulation scenario, the algorithm is evaluated for motor positioning with a rigid transmission. The inner loop motor velocity feedback is hence not used ( $K_{vel}=0$ ). The number of quantization steps was fixed to 75. The quantization step size was imposed by using even logarithmic spacing between the coefficient values (constant ratio of consecutive values). The iteration number is fixed at 100 and the number of ants to 100.

Figure 5 shows the simulation results for both initial controller setting and ACO tuning results without saturation (in the figures, all the positions are scaled to the motor side of the gearbox). The IAE performance index ( $\sigma=0$ ) results into an overshoot of less than 10%. By changing the index criterion to reduce the undesirable characteristics of the time response, ACO has the ability to generate new parameters without additional reconfiguration of the computer program. As expected for the modified IAE index (with  $\sigma=1$ ), the overshoot has decreased (Figure 5).

If the ACO tunes the controller and the anti-windup blocks sequentially (optimizing the PID controller without saturation and afterward the anti-windup feedback gains) then the time response is mediocre (Figure 6). This is due to the fact that without saturation, the optimal solution for the PID controller tends to rely on the derivative term of the controller to speed-up the initial response. This leads to very slow time response with saturation. Sequential optimization is not efficient hence the importance of simultaneous tuning of the parameters by ACO.

Figure 6 also show the results of global optimization with the two performance indices. The obtained step responses demonstrate a huge improvement compare to sequential tuning. Furthermore, as expected, the overshoot is lower when  $\sigma = 1$ , which shows again the effectiveness of the modified performance criterion.

The current response is shown on Figure 7 (with saturation level set to 7.8A) demonstrating proper tuning of the anti-windup controller. We note that the current saturates only for a short time period, but that this situation was detrimental to proper tuning of the system with the sequential approach (Figure 6), which favored a larger derivative term in the controller.

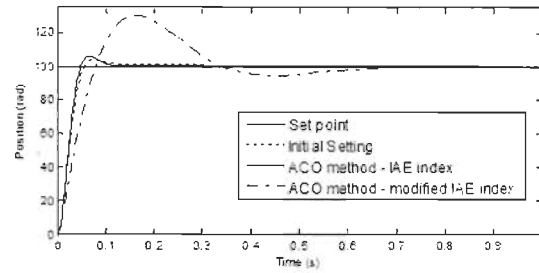


Figure 5- Step response for initial setting of PID position controller and optimized coefficients for IAE index and modified IAE index ( $\sigma = 1$ ), without saturation

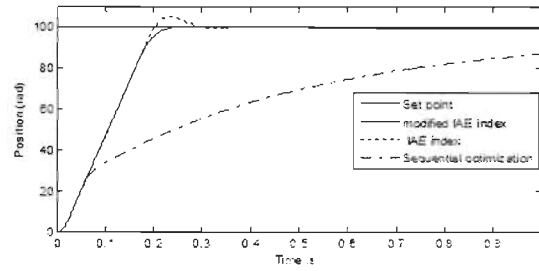


Figure 6- Step response for sequential optimization of PID controller and anti-windup feedback coefficients, and for global optimization with the IAE and modified IAE indices ( $\sigma = 1$ ), with saturation

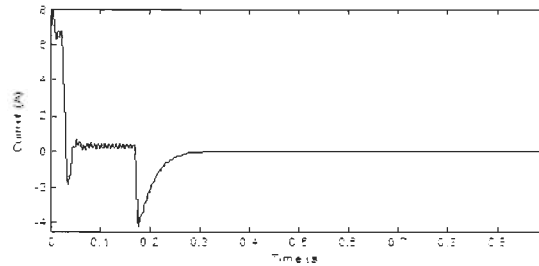


Figure 7- Motor current for a position step response with modified IAE index ( $\sigma = 1$ ), with saturation

We now consider the more realistic model where the rigid transmission model is replaced by a flexible transmission for the positioning of the grinder table as shown by Figure 1. Therefore, the inner loop motor velocity feedback coefficient  $K_{vel}$  must now be adjusted. This term is required to damp oscillations due to the drive flexibility. The increased number of parameters to tune is easily handled by the representation of the optimization problem. Indeed, only one slice needs to be added to the graph of Figure 3. The range of  $K_{vel}$  is  $[0,001 \text{ to } 0,1]$ . The small values for  $K_{vel}$  are due to the gearbox ratio.

It is important to note that the model becomes more complex due to the flexible coupling between the motor and the grinder table. The system is non-linear due to the saturation terms in the controllers and an analytic method cannot be used to tune the PID controller. Therefore, the proposed method based on ACO responds to a lack of efficient tuning method for this kind of application when high performance is required.

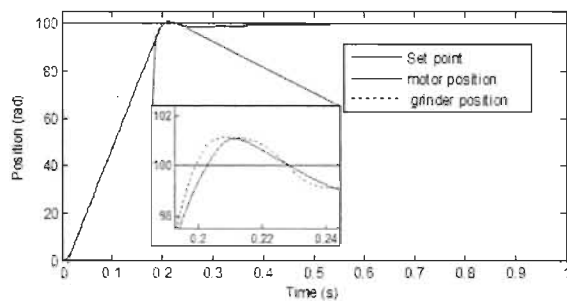


Figure 8- Step response for PID controller for motor position and grinder position using modified IAE index ( $\sigma = 1$ ), with saturation

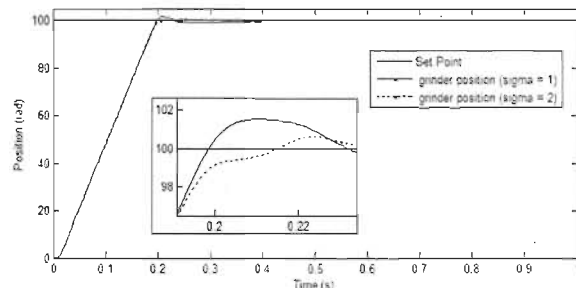


Figure 9- Step response for the PID controller for the grinder position using modified IAE index with saturation

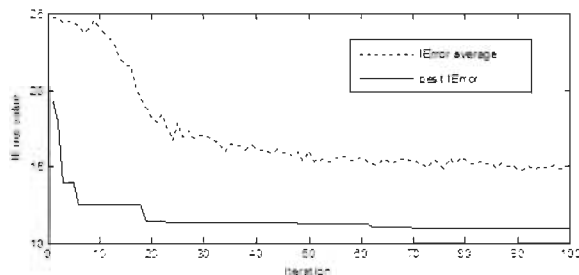


Figure 10- Convergence rate of the algorithm for step response for the modified IAE ( $\sigma = 2$ ) with saturation

Figure 8 demonstrates the efficiency of ACO for tuning of PID controllers with anti-windup for this motion system with flexible coupling and cascaded control loops structure using the modified IAE index. The drive flexibility is observed in the difference between the motor position and grinder table scaled position. As the overshoot is very detrimental to the grinding operations, decreasing overshoot without reducing dynamic performances is highly desirable. Figure 9 compares the grinder position step responses for  $\sigma = 1$  and  $\sigma = 2$ . Once again, it substantiates the relevance of using the modified IAE performance index to reduce overshoot.

The results confirm that ants adapt their search based on specific problem constraints while minimizing the performance index *IError* value. Figure 10 shows the convergence rate of the algorithm and of the best achieved *IError*. The average index values decrease with the iterations, as a result of the ants converging towards the best

solutions. We notice very little improvement of the best index value after 20 iterations. This means that the search can be stopped after an iteration number lower than 100 to approach optimal tuning of the controller with a good degree of confidence.

## 5. CONCLUSION

An ACO based methodology is proposed for controller tuning for a motion system with flexible coupling and cascaded loops control structure. This approach fills a gap for adjusting PID controller with saturation. Simulation results demonstrate the efficiency of the method for tuning PID controllers with anti-windup. The flexibility given by the choice of performance criteria and the flexibility on the number of parameters to be adjusted make this method very attractive. Moreover, it can be easily adapted to a specific application. Thus, as future work, we will apply this method to other types of controllers, including weighted controllers and controllers with many degrees of freedom.

Only a sub-optimal solution can be guaranteed since the ACO is a randomized optimization technique and that the controller coefficients are discretized to define the search space. Strategies to reduce the execution time of the ACO and to combine ACO with local search algorithms to yield better solutions are being explored.

## 6. ACKNOWLEDGEMENT

This work was supported by an Undergraduate Student Research Awards from the Natural Sciences and Engineering Research Council of Canada.

## 7. REFERENCES

- [1] F. Padula, and A. Visioli, "Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes," *Control Theory & Applications*, IET, 6(6), pp. 776-786, 2012.
- [2] N.J. Killingsworth and M. Krstic, "PID Tuning Using Extremum Seeking," *IEEE Control Systems Magazine*, pp. 70-79, February 2006.
- [3] M. Dorigo, V. Maniezzo, and A. Coloni. "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE TSMCB*, 26(1), pp. 29-41, 1996.
- [4] Dorigo, M., and T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts, 2004.
- [5] I. Chiha, N. Liouane, and P. Borne, "Tuning PID Controller Using Multiobjective Ant Colony Optimization," *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, Article ID 536326, 2012.
- [6] H.A. Varol, and Z. Bingul, "A New PID Tuning Technique Using Ant Algorithm," *Proceeding of the 2004 ACC*, Boston, Massachusetts, pp. 2154-2159, June 30 - July 2, 2004.
- [7] Y.-T. Hsiao, C.-L. Chuang, and C.-C. Chien, "Ant Colony Optimization for Designing of PID Controllers," *2004 IEEE ISCAS*, Taipei, Taiwan, pp. 321-326, September, 2004.

# PID controllers and anti-windup systems tuning using ant colony optimization

Maude-Josée Blondin and Pierre Sicard

Groupe de Recherche en Électronique Industrielle, Université du Québec à Trois-Rivières

Trois-Rivières (Québec), Canada

Tel.: +1 (819) – 376-5011.

Fax: +1 (819) – 376-5219.

E-Mail: MaudeJosee.Blondin@uqtr.ca; Pierre.Sicard@uqtr.ca

URL: <http://www.grei.ca>

## Acknowledgements

This work was supported by a fellowship and a research grant from the Natural Sciences and Engineering Research Council of Canada.

## Keywords

Optimisation, Motion control, Active damping, Autotuning, DC machine, Non-linear control

## Abstract

Ant Colony Optimization allows simultaneous tuning of a large number of control parameters and provides flexibility in the choice of the performance criterion. An off-line tuning methodology is proposed for PID-type controllers and their associated anti-windup subsystem for cascaded loop control structures. Insight is given on the design process, on parameter setting and on handling of soft and hard constraints. The method is illustrated by tuning a weighted PID controller and saturation management for a positioning axis in a grinder application.

## 1. Introduction

Many methods and rules (lambda tuning, Ziegler-Nichols, frequency response methods, heuristic methods, etc.) have been proposed for tuning Proportional-Integral-Derivative (PID) controllers, either to obtain initial setting of their parameters or for fine tuning [1]. The tuning rules are typically defined for the basic (ideal) PID controller or for controllers with a specific factorization or form. Moreover, the rules typically respond to particular performance criteria (Integral of Absolute value of Error (IAE), Integral of Squared Error (ISE), Integral of Time multiplied by the Absolute value of Error (ITAE), etc.) and do not take into account controller saturation. Rules also do exist to tune anti-windup controllers and it is customary to sequentially tune the PID and the anti-windup controllers. However, there exists close interaction between controller and anti-windup tuning. It was noted that the sequential approach is not efficient [2], [3]. In fact, it may lead to sluggish performance, henceforth the interest of using a more global approach, which can handle a large set of parameters, to tune adequately and simultaneously the PID system with anti-windup. Moreover, in order to manage more realistically the system, it would be relevant to have a tuning method with a flexible performance criterion that takes into account system limitations and constraints.

In [4], Ant Colony Optimization (ACO) has been proposed to tune PID controllers. We have used ACO as an offline tuning method [3] and we observed that this method offers several advantages. Firstly, tuning may be controller-structure independent. It also allows considering variations of the structure of the controller, such as tuning of input filters, fractional-order controllers, etc. Moreover, performance criterion is not fixed and does not require an analytic solution so that specific behaviours such as overshoot can be specifically penalized. Cascaded control loops may also be tuned simultaneously; in particular when time scale separation is not sufficient, which results into a large number of interdependent parameters to be tuned.

In industrial applications, it is common that the system has to cope with disturbances while responding correctly to a desired set point. It is a challenge to provide satisfactory response to both these inputs with a classical PID controller and additional degrees of freedom are needed for stringent performance requirements. With its added degrees of freedom, the weighted PID controller provides globally better results for step response and step disturbance than the PID controller.

In this paper, we propose an ACO based methodology applied to a motion system with flexible coupling for tuning cascaded loops control structures with saturation. We show the efficiency of ACO to tune simultaneously: the weighted position loop PID controller with filtered derivative term (PID<sub>f</sub>); its anti-windup feedback gain and the anti-windup gain of the current control loop. Furthermore, through an appropriate choice of the performance criterion, we demonstrate the flexibility provided by our method, e.g. to impose hard or soft current and motor velocity limitation, or to penalize specific behaviours such as output overshoot.

## 2. Ant Colony Optimization

ACO is an optimization technique inspired by the behaviour of real ants within a colony [5] [6]. It was observed that a colony facing multiple paths to a food source has the tendency of taking the shortest road. This is explained by the fact that ants communicate with each other via their environment by depositing traces of pheromones. The paths with the most pheromones are more attractive to other ants. Therefore, the amount of pheromones increases more rapidly on the shortest path, attracting more and more ants. Eventually, all ants will take the same path, i.e. the shortest path. Henceforth, the solution emerges from the collective interaction of the ants.

ACO can be used to solve combinatorial optimization problems for which a constructive heuristic can be employed. One of the first problems ACO was applied to is the Traveling Salesman Problem (TSP), which allows illustrating the method. The ants move on the TSP graph and generate different solutions. The probability for an ant  $k$  located in city  $i$  to choose city  $j$  is stated as

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \quad (1)$$

where  $N_i^k$  represents the number of cities to be visited by the ant  $k$ ,  $j \in N_i^k$ ;  $\tau_{ij}(t)$  is the quantity of pheromones deposited between cities  $i$  and  $j$ ;  $\eta_{ij}(t) = 1/d_{ij}$  is the inverse of the distance separating cities  $i$  and  $j$ ;  $\alpha$  and  $\beta$  are parameters that determine respectively the relative weight of pheromone traces and the heuristic information.

At each step, each ant deposits pheromones on its travelled path:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad \forall (i,j) \in L \quad (2)$$

where  $\Delta \tau_{ij}^k(t)$  is the quantity of pheromones deposited on segments that ant  $k$  has visited during this step. The quantity of deposited pheromones on each segment of the graph is function of the quality of the solution:

$$\Delta \tau_{ij}^k(t) = \begin{cases} 1/L^k & \forall (i,j) \in L \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $L$  is the total length of travelled distance for the iteration. The pheromone traces associated to each segment are reduced by a constant evaporation rate  $\rho$ :

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad \forall (i,j) \in L \quad (4)$$

where  $0 < \rho < 1$ . The evaporation process constitutes a forgetting factor used to avoid unlimited accumulation of pheromones and to allow forgetting previous bad decisions.

## 3. ACO for PID<sub>f</sub> and anti-windup tuning

To demonstrate the tuning methodology, one axis of the positioning table of an industrial grinder



system is considered in this study. A 3/4 hp, 5200 rpm, 3.5 N·m, dc motor is coupled to the grinder table by a gear box and a worm-gear such that 1 rad of motor rotation causes a displacement of 0.1 mm of the grinder table. The significant flexible coupling due to the transmission elements is modelled as an equivalent lumped spring. Performance criteria include high accuracy, minimal overshoot and good dynamic performance to improve product quality, avoid stress on the grinder and decrease the operation cycle. The flexible coupling complexifies the tuning process to achieve high performance for this cascaded loop control system. The typical solution for this type of system is to use cascaded current and table position control loops with an inner motor velocity loop to improve damping and reduce oscillations. Tuning for performance is more challenging than simple cascaded control loops, in particular when considering system constraints such as saturation levels and velocity limitations.

Figure 1 summarizes the system model and ACO's main input and output parameters for controller and anti-windup parameter tuning, which is applied to various cases of system constraints (Figures 2, 3 and 4). The first position controller is illustrated in Figure 2, along with its parameters. It is composed of a PID controller with filtered derivative term, a motor velocity feedback ( $K_{vel}$ ) and an anti-windup controller. The inner PI current controller was tuned separately but its anti-windup feedback gain  $K_{sat}$  (Figure 1) will be tuned by the ACO. Figure 3 presents the structure of the weighted  $PID_f$  controller where parameters  $W_p$  and  $W_v$  represent the weighting factors on the proportional and derivative terms of the  $PID_f$  controller, effectively modifying the zeroes of the transfer function from the reference position to the actual grinder table position. Figure 4 shows the last controller structure used where motor velocity limitation is considered instead of current saturation for anti-windup. Both could have been considered simultaneously, but considering that the motor is lightly loaded, its current does not saturate for significant periods and a simple saturator was deemed sufficient for this scheme.

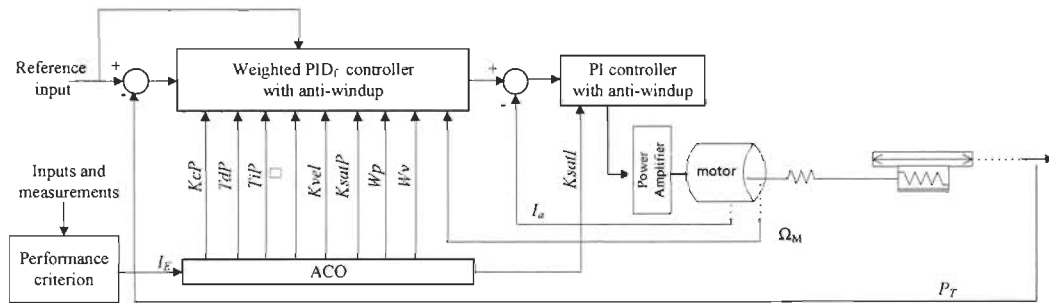


Fig. 1: System overview with ACO application for tuning  $PID_f$  and anti-windup system parameters.

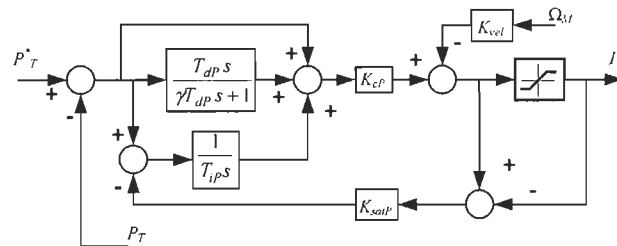


Fig. 2: Position feedback controller and anti-windup scheme.

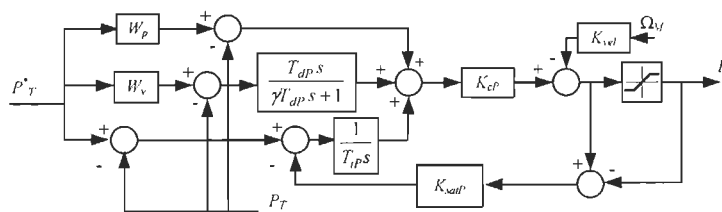


Fig. 3: Weighted  $PID_f$  controller structure with anti-windup for current reference limitation.

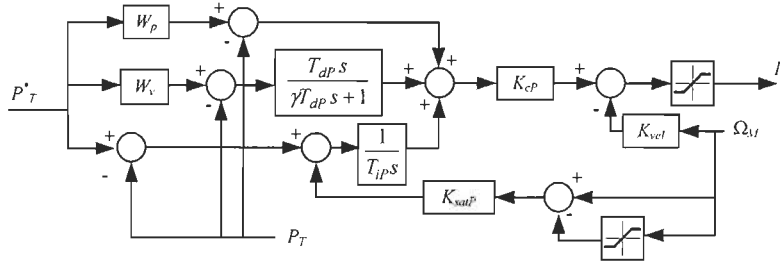


Fig. 4: Weighted PID<sub>f</sub> controller structure with anti-windup for motor velocity limitation.

To use ACO, controller tuning must be transformed into a combinatorial optimization problem [4], [7], [8] (Figure 5). Each slice of the graph corresponds to nodes of quantized controller parameters. This representation allows easy addition or removal of free controller coefficients. Range of parameters and quantization steps must be chosen to complete the graph of Figure 5. Rough tuning of the control parameters can be used to define the range of parameters. The ants move on the graph from left to right by choosing the nodes they travel from probabilistic rules. They leave pheromone traces as a function of the quality of the solution (c.f. Section 2).

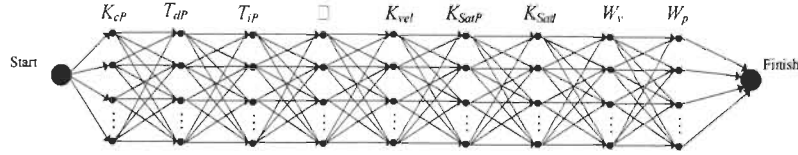


Fig. 5: Optimization graph for weighted PID<sub>f</sub> controller and anti-windup coefficients.

The performance is evaluated by  $I_E$  index resulting from the set of parameters on the travelled path. The objective of the ACO is to find the best set of parameters to minimize  $I_E$ . The choice of the performance criterion is critical to obtain the desired behaviour with regards to transients, steady state and inner variables limitation. To simultaneously consider set point performance and disturbance rejection criteria, a performance index with normalized quantities was defined as follows (some coefficients are set to zero for some of the tests):

$$I_E = \sum \left( \int_{t_s}^{t_s+T_s} \frac{|e(t)|}{\Delta P_T^*} dt \right) + K_t \sum \left( \int_{t_p}^{t_p+T_p} \frac{|e(t)|}{\Delta T_{pert}} dt \right) + \sigma \int_0^{T_f} \frac{e_d(t)}{\Delta P_T^*} dt + \beta \int_0^{T_f} \frac{e_{v_d}(t)}{\Omega_{Mm}} dt \quad (5)$$

where:

- the first two terms are summations over each of the set point step input and disturbance input transients respectively:  $t_s$  is the time of last set point variation;  $T_s$  is the approximate settling time for the set point step response;  $t_p$  is the time of last disturbance input variation;  $T_p$  is the approximate settling time for the step disturbance;  $e(t) = P_T^* - P_T(t)$  is the output error;  $P_T$  and  $P_T^*$  are respectively the measured and reference positions of the grinder table;  $\Delta P_T^*$  is the amplitude of the last step input variation used to normalize the position error;  $\Delta T_{pert}$  is the amplitude of the last force disturbance variation;  $K_t$  is a positive constant of equivalent stiffness used to normalize the position error due to force disturbance;
- the third term penalizes position overshoots:  $T_f$  is the simulation time as this term is penalized continually;  $\sigma$  is a positive constant weight;  $e_d(t)$  is defined as

$$e_d(t) = \begin{cases} |e(t)| & \text{during response overshoots} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

- the fourth term penalizes motor velocity limit violations:

$$e_{v_d}(t) = \begin{cases} ||\Omega_M(t)| - \Omega_{Mm}| & \text{if exceeding the motor velocity limit} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where  $\Omega_{Mm}$  is the upper motor velocity limit.

Larger values of  $\sigma$  in (5) penalize more the overshoot, which is very detrimental to the quality of the grinding operations. Larger values of  $K_t$  are used to reduce the impact of the disturbance on the system.  $\beta$  is used to constrain the motor speed to respect the system limits. Choice of the weights is analysed in the next section.

#### 4. Simulation Results

The ACO algorithm was implemented in Matlab® and the system’s model on Simulink™ to implement off-line parameter tuning (Figure 6). The position  $PID_f$  controller’s parameters were initially determined by pole placement from the approximate rigid model of the system and the following ranges were chosen for the parameter coefficients for the search (wide ranges were chosen to extend the space of possible solutions):

$$\begin{array}{ll}
 K_{SatP}: [1 \text{ to } 1000] & K_{SatI}: [1 \text{ to } 1000] \\
 K_{cP}: [0.001 \text{ to } 10] & T_{iP}: [0.003 \text{ to } 30] \\
 T_{dP}: [0.001 \text{ to } 10] & \gamma: [10 \text{ to } 50] \\
 W_v: [0 \text{ to } 1] & W_p: [0 \text{ to } 1] \\
 K_{vel}: [0.01 \text{ to } 0.1] & 
 \end{array}$$

The number of quantization steps was fixed to 75. The quantization step size was imposed by using even logarithmic spacing between the coefficient values (constant ratio of consecutive values).

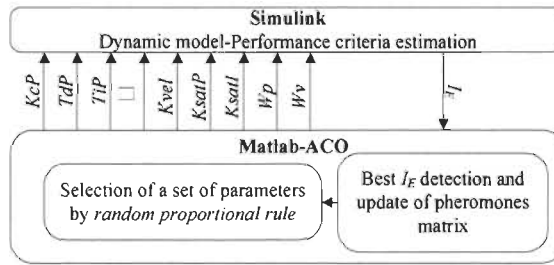


Fig. 6: Simulation environment structure of ACO iteration.

##### 4.1. PID<sub>f</sub> controller with current based anti-windup

In a first simulation scenario, the control structure shown in Figure 2 is used. The iteration number for ACO methodology is fixed at 100 and the number of ants to 100. The parameters  $K_t$  and  $\beta$  of the performance criteria are set to 0 and no load disturbance is applied. Therefore, the performance criterion is the IAE performance index augmented with an additional term to penalize overshoot. As this characteristic is very detrimental to the grinding operations, decreasing overshoot without reducing dynamic performances is highly desirable. Figure 7 shows the best response obtained by the ACO for  $\sigma = 1$ . The drive flexibility is observed in the difference between the motor position and scaled grinder table position.

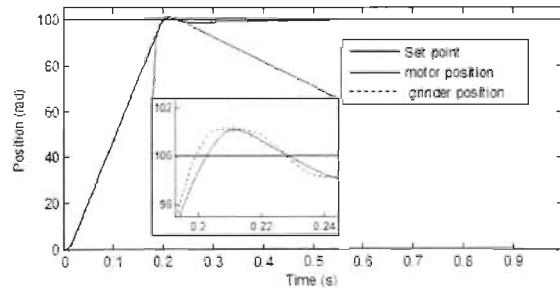


Fig. 7: Step response for  $PID_f$  controller for motor position and grinder position using modified IAE index ( $\sigma = 1$ ), with saturation.

Figure 8 compares the grinder position step responses for  $\sigma = 1$  and  $\sigma = 2$ . It substantiates the relevance of using the modified IAE performance index to reduce overshoot.

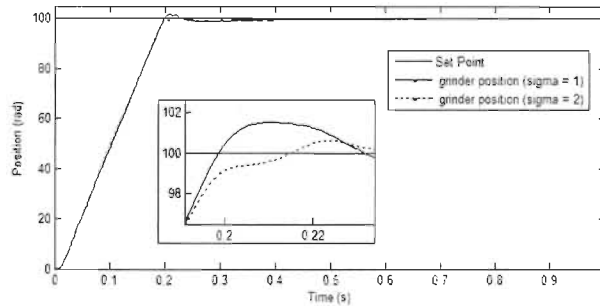


Fig. 8: Step response for the  $PID_f$  controller for the grinder position using modified IAE index with saturation.

Figure 9 shows the convergence rate of the algorithm and the best achieved  $I_E$ . The average index values decreases with the iterations, as a result of the ants converging towards the best solutions. We notice very little improvement of the best index value after 60 iterations. This means that less than 100 iterations are required to approach optimal tuning of the controller with a good degree of confidence.

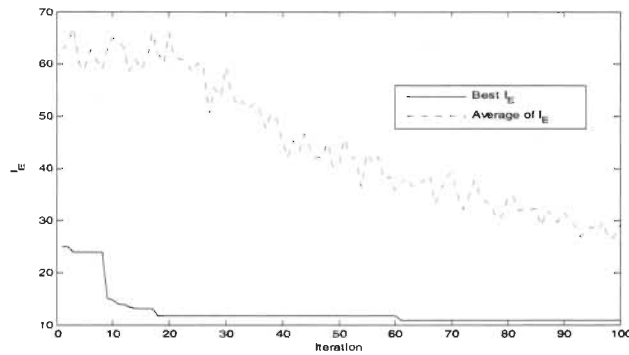


Fig. 9: Convergence rate of the algorithm for set point step response for the modified IAE index ( $\sigma = 2$ ) with saturation.

A step load disturbance corresponding to 20% of the nominal torque (0.75 Nm) and representing a realistic disturbance value is applied to the system at 2 s. Figure 10 shows poor system performance when responding to the disturbance with the parameter values obtained using only the reference input portion of the response.

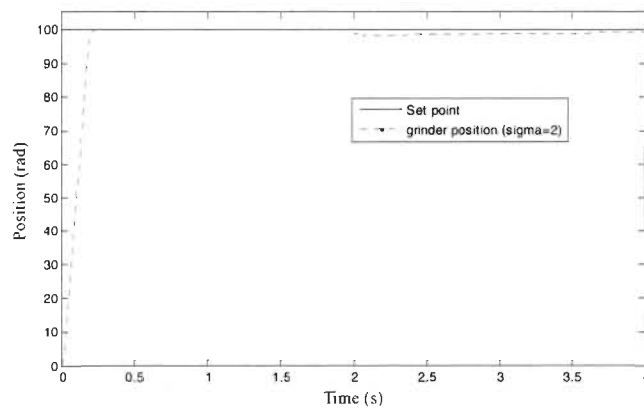


Fig. 10: Step response and step disturbance response for  $PID_f$  controller for grinder position using modified IAE index ( $\sigma = 2$ ) evaluated over the first 2 s.

To improve disturbance rejection, coefficient  $K_i$  is used in the performance criterion (5) to penalize the output disturbance error. The selection of  $K_i$  values is based on the interpretation of the corresponding term in (5) by a spring model and posing a maximum permitted displacement of the grinder table for a given disturbance. From (5), if we write out (this is only an interpretation and the unity factor was used to normalize this term to 1):

$$1 = K_t \frac{|e_{max}|}{\Delta T_{pert}} \quad (8)$$

and assume that for a disturbance of 20% the motor nominal torque, a 1 mm to 0.1 mm of grinder table displacement satisfies (8), the  $K_i$  values are respectively 0.075 to 0.75. These values are used as a starting point to eventually “tune” (5). Figure 11 presents the obtained optimal set point step responses and the step load disturbance responses for different  $K_i$  values. Figure 11b demonstrates the positive effect of coefficient  $K_i$  on improving the disturbance response. On the other side, Figure 11c shows that this was achieved at the cost of a compromise on the transient response to the reference input. Added degrees of freedoms using the weighted  $PID_f$  controller and motor velocity limitation management are now introduced to deal with this issue.

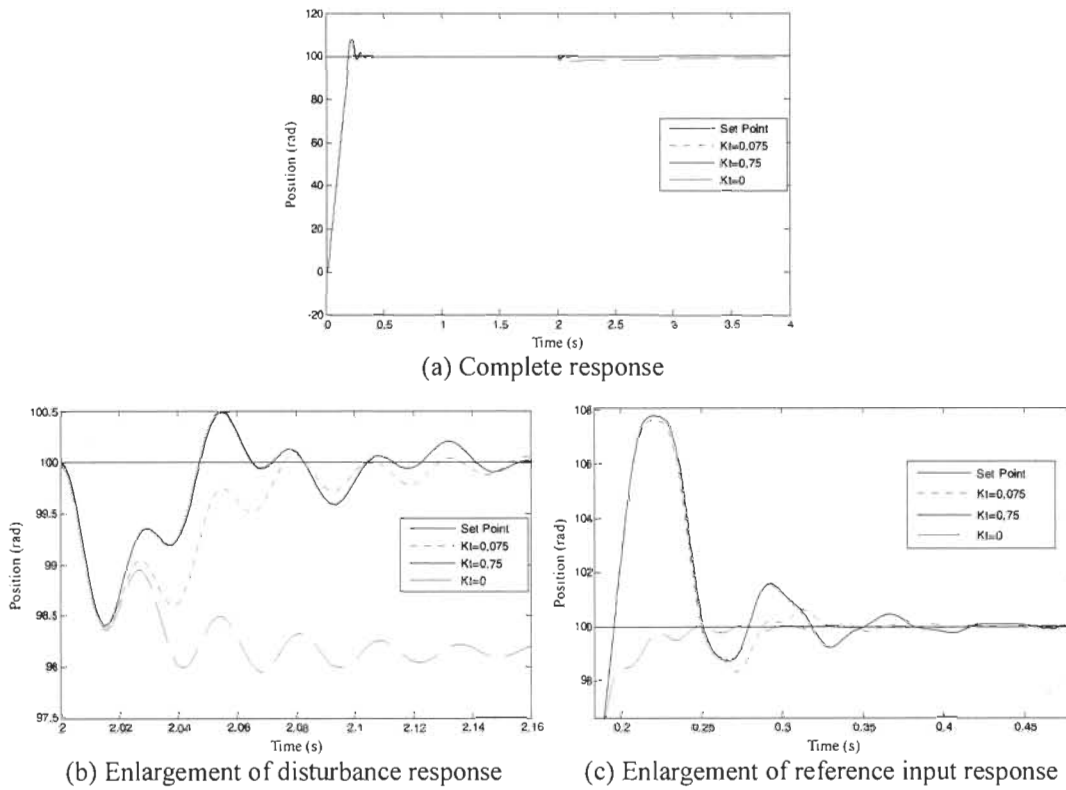


Fig. 11: Step response and step disturbance response for PID controller for grinder position ( $\sigma = 2$ ) versus different  $K_i$  values.

#### 4.2. Weighted $PID_f$ controller with current based anti-windup

As a second simulation scenario, the control structure shown on Figure 3 is used. The iteration number for ACO methodology is augmented at 200 and the number of ants to 200 due to a larger number of parameters to be adjusted.  $\sigma$  is set to 2 and  $\beta$  is set to 0 in the performance index (5). Coefficients  $W_p$  and  $W_v$  offer additional degrees of freedom for control so that the controller can simultaneously yield better performance for set point response and disturbance rejection.

Globally, Figure 12 shows better results than those obtained in the first scenario (Figure 11). The weighted controller with  $K_i = 0$  improves the input reference step response at the cost of poorer

disturbance rejection, which was expected since the disturbance response was not considered for optimization in this case. For  $K_i = 0.075$ , input response has improved and disturbance response is comparable to the previous settings. Improvements are also observed for  $K_i = 0.75$  but with excessive position overshoot (Figures 11c and 12c). This overshoot is due to integrator windup in the position controller due to natural motor velocity saturation as shown on Figure 13.

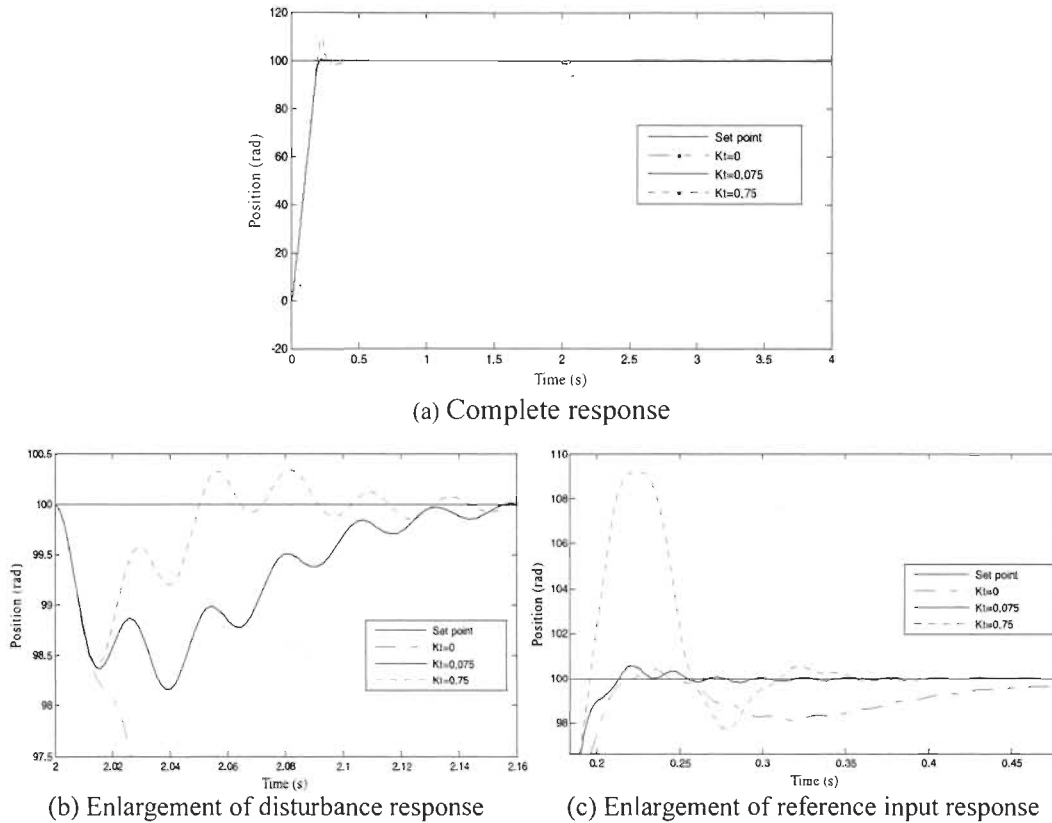


Fig. 12: Step response and step disturbance response for weighted PID controller for grinder position ( $\sigma = 2$ ) versus different  $K_i$  values.

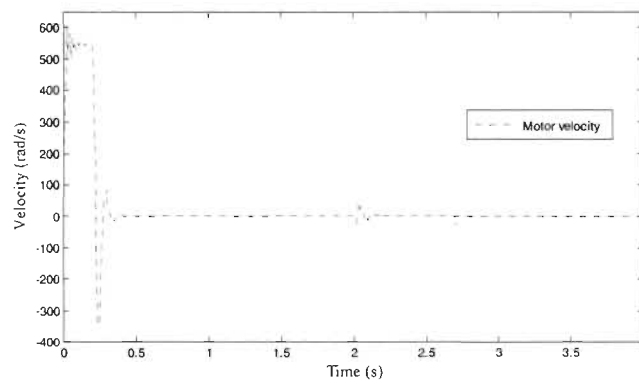


Fig. 13: Motor velocity response;  $K_i = 0.75$ .

### 4.3. Weighted PID controller with motor velocity based anti-windup

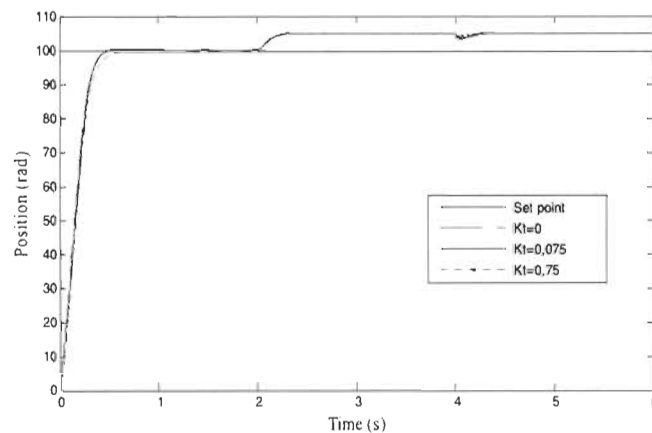
In order to improve the overall response, inner limitations must be considered and penalized in the performance index. Henceforth, for the third simulation scenario, the control structure described in Figure 4 is used. We have observed that only including anti-windup feedback based on velocity limitation as illustrated on Figure 4 is insufficient and that it is necessary to explicitly penalize velocity

limit violations (last term in (5)) to obtain the desired effect. Moreover, in order for the response in saturation mode during large input step variations not to dominate in the performance criterion over the small signal operation, a large (100 rad at 0 s) and a small (5 rad at 2 s) input step variations are applied to the system for optimization. A step disturbance corresponding to 20% the motor's nominal torque is applied at 4 s. Parameter  $\beta$  is set to 100 in (5). A larger value of  $\beta$  allows to implement hard constraints on motor velocity, whilst a small value would implement a softer constraint.

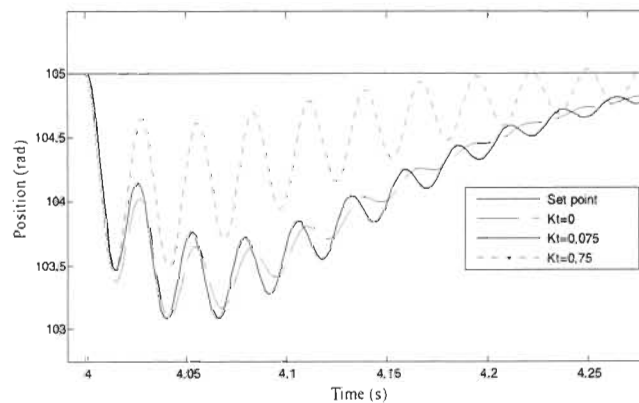
Also, in the previous simulation scenarios, overshoot was present. As small as it may be, no overshoot should be present when precision work is required such as in grinding operation. Henceforth, the value of  $\sigma$  is increased to 100 to implement a hard constraint on position overshoot. The iteration number for ACO methodology and the number of ants are still fixed at 200.

Figure 14 show the results obtained when motor velocity limit is set to 386 rad/s and Figure 15 the effectiveness of the performance index to limit the motor velocity. As expected, no position overshoot is present and Figure 14b demonstrates the efficiency of higher  $K_f$  values to improve disturbance rejection, eventually at the cost of decreased damping of the transmission's flexibility effects.

These results show that the weighted controller provides a better control over the system response to load disturbances than a PID controller. Also, the ACO method offers flexibility in its performance criterion to adjust the parameters of the controller. This advantage allows to specifically penalising certain aspects of the system behaviour to respect system limitations and to adapt the system to the application.



(a) Complete response



(b) Enlargement of response to disturbance

Fig. 14: Step response with motor velocity saturation for weighted  $PID_f$  controller for grinder position versus different  $K_f$  values.

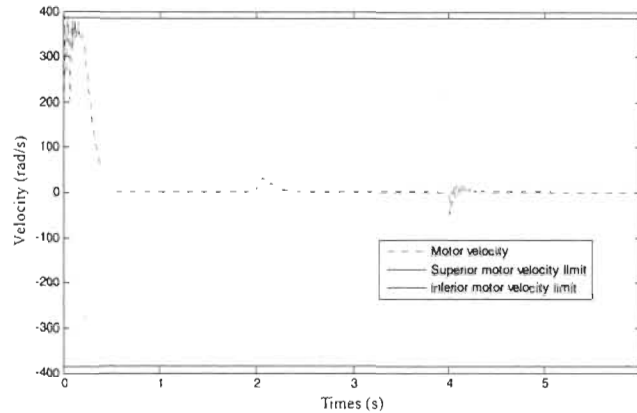


Fig. 15: Motor velocity with saturation compensation;  $K_t = 0.075$  .

## 5. Conclusion

Controller tuning in nonlinear systems is a complex task to obtain a satisfactory response to set point and disturbances inputs while respecting the limits of the system and the constraints imposed by the application. The ACO methodology has the advantage of tuning a large number of parameters simultaneously, but also to use a performance criterion tailored to the application. Hard or soft constraints can be imposed by properly choosing the relative weights in the performance criterion. It is also important to choose properly the number of ants and of iterations for the ACO so that the solution is statistically valuable with regards to the number of output parameters and of their quantization. It was also shown that the weighted controller alleviates the need for compromises that must be made between set point response and disturbance rejection. The methodology can be applied to tune industrial systems for which a reasonable model is known. Simulation time may become a problem so that we have initiated work to develop a hybrid algorithm that combines ACO for global search and Nelder-Mead algorithm for controller fine tuning [9].

## References

- [1] F. Padula, and A. Visioli, "Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes," *Control Theory & Applications*, IET, 6(6), pp. 776-786, 2012.
- [2] N.J. Killingsworth and M. Krstic, "PID Tuning Using Extremum Seeking," *IEEE Control Systems Magazine*, pp. 70-79, Feb. 2006.
- [3] M.-J. Blondin, and P. Sicard, "ACO Based Controller and Anti-Windup Tuning for Motion Systems with Flexible Transmission," *2013 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'13)*, Regina, Saskatchewan, Canada, 5-8 May 2013.
- [4] I. Chiha, N. Liouane, and P. Borne, "Tuning PID Controller Using Multiobjective Ant Colony Optimization," *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, Article ID 536326, 2012.
- [5] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics-part B Cybernetics*, 26(1), pp. 29-41, February 1996.
- [6] Dorigo, M., and T. Stützle, *Ant Colony Optimization*. The MIT Press, Cambridge, Massachusetts, 2004.
- [7] H.A. Varol, and Z. Bingul, "A New PID Tuning Technique Using Ant Algorithm," *Proceeding of the 2004 ACC*, Boston, Massachusetts, pp. 2154-2159, June 30 - July 2, 2004.
- [8] Y.-T. Hsiao, C.-L. Chuang, and C.-C. Chien, "Ant Colony Optimization for Designing of PID Controllers," *2004 IEEE ISCASD*, Taipei, Taiwan, pp. 321-326, September, 2004.
- [9] M.-J. Blondin, and P. Sicard, "Combined ACO Algorithm — Nelder-Mead Simplex Search for Controller and Anti-Windup Tuning for a Motion System with Flexible Transmission," submitted to *2013 IEEE 39th Annual Conference of the IEEE Industrial Electronics Society (IECON'13)*, Vienna, Austria, 10-13 November 2013.



# Combined ACO Algorithm — Nelder-Mead Simplex Search for Controller and Anti-Windup Tuning for a Motion System with Flexible Transmission

Maude-Josée Blondin and Pierre Sicard

Research Group on Industrial Electronics, ECE department  
Université du Québec à Trois-Rivières  
Trois-Rivières (Québec), Canada  
MaudeJosee.Blondin@uqtr.ca; Pierre.Sicard@uqtr.ca

**Abstract**— Optimal controller and anti-windup tuning for motion systems with flexible coupling and cascaded loops control structure is a practical and difficult problem, in particular when time scale separation is insufficient. An Ant Colony Optimization (ACO) based methodology for global off-line tuning of Proportional-Integral-Derivative (PID) controllers with anti-windup can be used at the expense of excessive computation time. A new hybrid algorithm based on ACO and Nelder-Mead (NM) improves computation time to reach accurate optimal solutions. This hybrid algorithm handles large sets of parameters to tune and the designer can modify or create new performance indices to fully grasp the desired behavior with multi-objective criteria.

**Keywords**— *Ant colony optimization, Nelder-Mead algorithm, Heuristic algorithms, PID Controller, Integrator anti-windup, Optimization, Motion systems.*

## I. INTRODUCTION

Many methods and rules (lambda tuning, Ziegler-Nichols, frequency response methods, heuristic methods, etc.) have been proposed for tuning PID controllers, either to obtain initial setting of their parameters or for fine tuning [1]. The tuning rules are typically defined for the basic (ideal) PID controller or for controllers with specific factorization or form. Moreover, the rules typically respond to particular performance criteria (Integral of Absolute value of Error (IAE), Integral of Squared Error (ISE), Integral of Time multiplied by the Absolute value of Error (ITAE), etc.) and do not take into account controller saturation. Rules also do exist to tune anti-windup controllers and it is customary to sequentially tune the PID and the anti-windup controllers. However, there exists close interaction between controller and anti-windup tuning. It was noted that the sequential approach is not efficient [2]. In fact, it may lead to sluggish performance, henceforth the interest of using a more global approach to tune adequately the PID system with anti-windup.

In [3], Ant Colony Optimization (ACO) has been proposed to tune PID controllers. We have used ACO algorithm as an offline tuning method [4], [5] and we observed that this method offers several advantages. Firstly, tuning may be controller-structure independent. It also allows considering variations of the structure of the controller, such as tuning of input filters, weighted PID controllers, etc. Performance

criterion is not fixed and does not require an analytic solution so that specific behaviors such as overshoot can be specifically penalized. Cascaded control loops may also be tuned simultaneously in particular when time scale separation is not sufficient, which results into a large number of interdependent parameters to be tuned. However, the increase in number of parameters to tune leads to an excessive computation time for the algorithm to converge to a desired solution.

In this paper, and inspired by [6], we propose a new way to combine ACO algorithm with Nelder-Mead search (NM) for tuning a motion system with flexible coupling and cascaded loops control structure. We show that the proposed hybrid algorithm can tune simultaneously and effectively the position loop PID controller with filtered derivative term, its anti-windup feedback gain and the anti-windup gain of the current control loop, whilst responding to the need of having a reliable and fast global tuning method.

The choice of NM algorithm is motivated by the fact that it presents the same benefits than ACO methodology. Contrary to [7], we show in this paper that NM algorithm is not efficient when used alone for tuning PID controller with anti-windup for the system under study. Indeed, the performance of the NM algorithm in terms of computation time and efficiency depends on the choice of initial conditions. When initial conditions are suitably chosen, the algorithm converges rapidly to reach the desired solution. Therefore, the proposed hybrid algorithm allows the ants to explore a large search space in order to find a good set of parameters for starting the initial search of NM algorithm. Simulation results show that sequentially using global search with ACO algorithm and local search with NM method provides faster and more precise results. We validate the new algorithm by comparing three simulation scenarios; ACO algorithm applied alone, NM method used alone and the application of our new hybrid algorithm. The simulations confirm that the ACO-NM combination provides accurate results faster.

## II. ANT COLONY OPTIMIZATION

ACO is an optimization technique inspired by the behavior of real ants within a colony [8], [9]. It was observed that a colony facing multiple paths to a food source has the tendency of taking the shortest road. This is explained by the fact that

ants communicate with each other via their environment by depositing traces of pheromones. The paths with the most pheromones are more attractive to other ants. Therefore, the amount of pheromones increases more rapidly on the shortest path attracting more and more ants. Eventually, all ants will take the same path, i.e. the shortest path. Henceforth, the solution emerges from the collective interaction of the ants.

ACO can be used to solve most of the combinatorial optimization problems for which a constructive heuristic can be employed. One of the first problems ACO was applied is the Traveling Salesman Problem (TSP), which allows to illustrate the method. The ants move on the TSP graph and generate different solutions. The probability for an ant  $k$  located in city  $i$  to choose city  $j$  is stated as

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \quad (1)$$

where  $N_i^k$  represents the number of cities to be visited by the ant  $k$ ,  $j \in N_i^k$ ;  $\tau_{ij}(t)$  is the quantity of pheromones deposited between cities  $i$  and  $j$ ;  $\eta_{ij}(t) = 1/d_{ij}$  is the inverse of the distance separating cities  $i$  and  $j$ ;  $\alpha$  and  $\beta$  are parameters that determine respectively the relative weight of pheromone traces and the heuristic information.

At each step, each ant deposits pheromones on its travelled path:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad \forall (i, j) \in L \quad (2)$$

where  $\Delta \tau_{ij}^k(t)$  is the quantity of pheromones deposited on segments that ant  $k$  has visited during this step. The quantity of deposited pheromones on each segment of the graph is function of the quality of the solution:

$$\Delta \tau_{ij}^k(t) = \begin{cases} 1/L^k & \forall (i, j) \in L \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $L$  is the total length of travelled distance of the iteration. The pheromone traces associated to each segment are reduced by a constant evaporation rate  $\rho$ :

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad \forall (i, j) \in L \quad (4)$$

where  $0 < \rho < 1$ . The evaporation phenomenon constitutes a forgetting factor used to avoid unlimited accumulation of pheromones and to allow forgetting previous bad decisions.

### III. NELDER-MEAD SIMPLEX SEARCH

Nelder-Mead simplex search is one of the most popular algorithms for solving unconstrained nonlinear optimization problems. By its name, it should not be confused with the Dantzig's simplex algorithm applicable to linear optimization problem with constraints. Both of them rely on sequences of simplexes, but they are completely different and separate. NM algorithm tends to minimize a nonlinear scalar function of  $n$  variables from a starting point estimated by using only the evaluation of the objective function without using derivative.

Therefore, this method is part of general class of direct search methods and it is generally used for local optimization.

Initially, NM algorithm performs a simplex of  $n + 1$  vertices from a starting point  $x_0$ . A certain percentage (usually 5%) is added on each  $x_0(i)$  to  $x_0$  where  $i = 1, \dots, n$ . If initially a component of  $x_0$  is zero, a very small value is used. Then, the first simplex is created. Thereafter, the simplex moves in the unconstrained search space in order to minimize an objective function following the next procedure [10].

**1. Ordering.** At iteration  $k$ , the vertices identified such as  $x_1^{(k)}, \dots, x_{n+1}^{(k)}$  are ordered depending on the value of the function  $f_1^{(k)} \leq f_2^{(k)} \leq \dots \leq f_{n+1}^{(k)}$  where  $f_i^{(k)}$  denotes  $f(x_i^{(k)})$ . Then, because we seek to minimize the objective function,  $x_1^{(k)}$  point refers to the best point or vertex while  $x_{n+1}^{(k)}$  to the worst point. Similarly,  $f_{n+1}^{(k)}$  refers to the worst function value, and so on. For each iteration, the worst actual point  $x_{n+1}^{(k)}$  is replaced by a new vertex on the simplex.

**2. Reflection.** In this step, reflection point  $x_r$  is calculated using

$$x_r = \bar{x} + \rho(\bar{x} - x_{n-1}) \quad (5)$$

where  $\bar{x} = \sum_{i=1}^n x_i/n$  is the centroid of the  $n$  best points and  $\rho = 1$  (usual value) is the reflection coefficient. Then,  $f_r = f(x_r)$  is evaluated. If  $f_r$  is such that  $f_1 \leq f_r < f_n$ , the reflection point  $x_r$  replaces the point  $x_{n+1}$  in the simplex and the iteration is terminated.

**3. Expansion.** If  $f_r < f_1$ , the expansion point  $x_e$  is obtained using

$$x_e = \bar{x} + \chi(x_r - \bar{x}) \quad (6)$$

where  $\chi = 2$  (usual value) is the expansion coefficient. Then,  $f_e = f(x_e)$  is evaluated. If  $f_e < f_r$ ,  $x_e$  point is accepted in the simplex and the iteration stops. Otherwise, if  $f_r \leq f_e$ ,  $x_r$  is accepted and the iteration is complete.

**4. Contraction.** If  $f_r \geq f_n$ , a contraction is done between  $\bar{x}$  and the best between  $x_{n+1}$  and  $x_r$ .

**a. Outside.** If  $f_n \leq f_r < f_{n+1}$ , meaning that  $x_r$  is strictly better than  $x_{n+1}$ , a contraction is performed outside the simplex. The point  $x_c$  is computed as follows:

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) \quad (7)$$

where  $\gamma = \frac{1}{2}$  (usual value) is the contraction coefficient. Then,  $f_c = f(x_c)$  is evaluated. If  $f_c \leq f_r$ , the point  $x_c$  is introduced in the simplex and the iteration is terminated. Otherwise, a shrink operation is performed (go to step 5).

**b. Inside.** If  $f_r \geq f_{n+1}$ , the inside contraction point is computed according to

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) \quad (8)$$

where  $\gamma = 0.5$  (usual value) is the contraction coefficient.

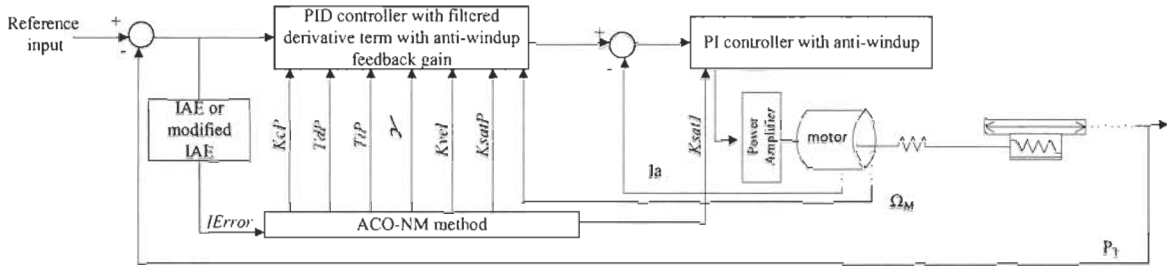


Fig. 1. System overview with ACO-NM application for PID parameters' tuning.

Evaluate  $f_{cc} = f(x_{cc})$ . If  $f_{cc} \leq f_{n+1}$ , then  $x_{cc}$  is becoming a part of the simplex and the iteration is done. Otherwise, a shrink operation is performed (go to step 5).

5. **Shrinkage.** Evaluate  $f$  at  $n$  points according to,

$$v_i = x_1 + \sigma(x_i - x_1) \quad (9)$$

where  $i = 2, \dots, n+1$  and  $\sigma = 0.5$  (usual value) is the shrinkage coefficient. Thereby, the simplex for the next iteration is  $x_1, v_2, \dots, v_{n+1}$ .

#### IV. HYBRID ACO—NELDER-MEAD ALGORITHM FOR PID AND ANTI-WINDUP TUNING

The proposed hybrid algorithm implements ACO to quickly explore a large search space to find a good set of initial parameters for NM algorithm. Both techniques can handle a large set of parameters to minimize an objective function easily adaptable depending of the application field. These two advantages fulfill the requirements to respond to a lack of tuning method for PID controllers with anti-windup system.

To demonstrate the methodology, one axis of the positioning table of an industrial grinder system is considered in this study. A 3/4 hp dc motor is coupled to the grinder table by a gear box and a worm-gear. The significant flexible coupling is modeled as an equivalent lumped spring. Performance criteria include high accuracy and good dynamic performance in order to decrease the operation cycle. The flexible coupling complicates the tuning process to achieve high performance for this cascaded loop control system.

Fig. 1 summarizes the system model and ACO-NM hybrid combination's main input and output parameters for PID and anti-windup parameter tuning. The position controller is illustrated in Fig. 2, along with its parameters. It is composed of a filtered PID controller with motor velocity feedback ( $K_{vel}$ ) and anti-windup controller.  $K_{vel}$  term is required to damp oscillations due to the drive flexibility. The PI current controller's anti-windup feedback gain is identified as  $K_{satI}$  on Fig. 1.

To use ACO, controller tuning must be transformed into a combinatorial optimization problem [3], [11] (Fig. 3). Each slice of the graph corresponds to nodes of quantified controller parameters, allowing for easy addition or removal of free controller coefficients. Range of parameters and quantization steps must be chosen to complete the graph of Fig. 3.

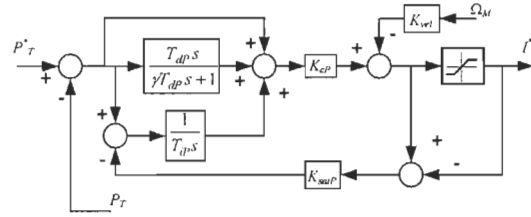


Fig. 2. Position feedback controller.

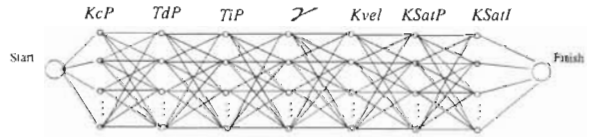


Fig. 3. Optimization graph for filtered position PID controller and anti-windup coefficients.

The ants move on the graph from left to right by choosing the nodes they travel from probabilistic rules. They leave pheromone traces as a function of the quality of the solution. The performance is evaluated by  $IError$  index resulting from the set of parameters on the travelled path. The ants work together to find the best set of parameters to minimize the performance criterion  $IError$  defined as a modified Integral of Absolute value of Error (IAE) index:

$$IError = \int_0^{T_s} |e(t)| dt + \sigma \int_0^{T_s} e_d(t) dt \quad (10)$$

where  $e(t) = P_r^*(t) - P_r(t)$  is the output error,  $T_s$  is the settling time,  $\sigma$  is a positive constant weight and

$$e_d(t) = \begin{cases} |e(t)| & \text{during response overshoots} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

The performance index is chosen to emphasize the desired response or to penalize undesired behavior. For  $\sigma=0$ , (10) corresponds to the IAE index; larger values of  $\sigma$  are used to penalize the overshoot, which is very detrimental to the quality of the grinding operations.

The best set of parameters found by the ants at a given iteration number is used as initial conditions for NM method research. This search follows the rules described in Section III. The objective function minimized by NM method is the same as for the ACO.

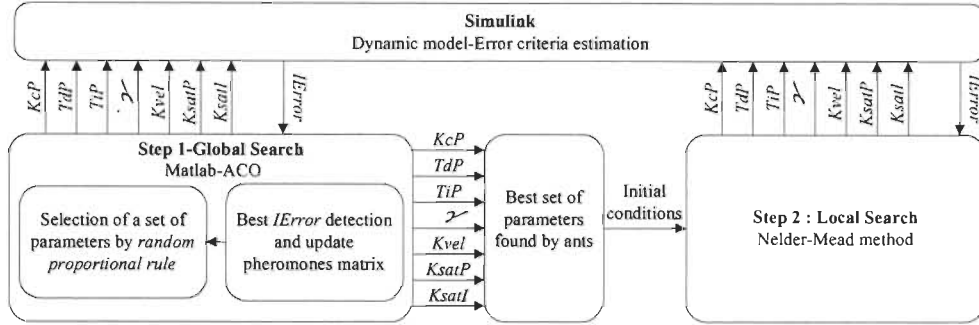


Fig. 4. Simulation environment structure of hybrid algorithm.

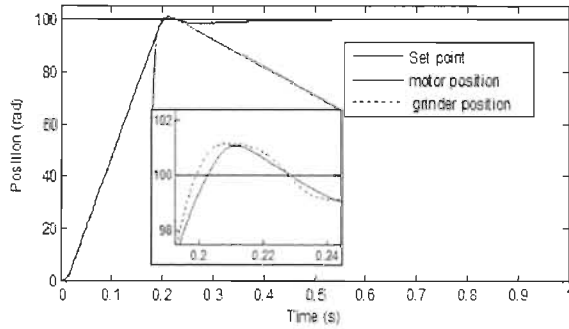


Fig. 5. Step response for PID controller for motor position and grinder position using modified IAE index ( $\sigma = 1$ ) with ACO.

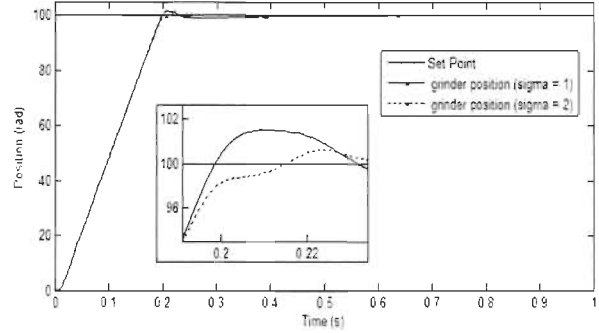


Fig. 6. Step response for the PID controller for the grinder position using modified IAE index with ACO.

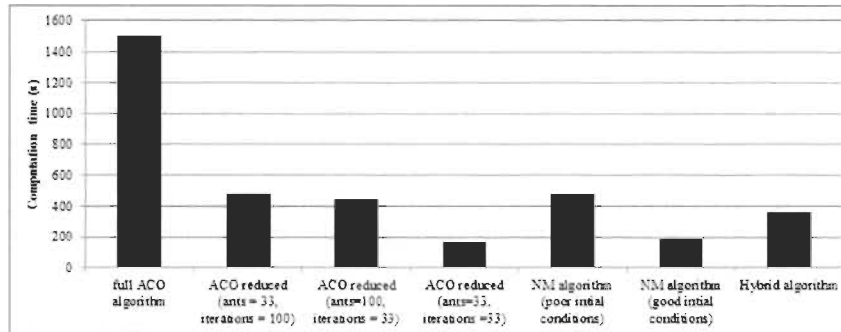


Fig. 7. Computation time for various optimization strategies.

## V. SIMULATION RESULTS

Fig. 4 shows the simulation environment used to apply the hybrid algorithm to the system described in the section IV.

The position PID controller's parameters were initially determined by pole placement from the approximate rigid model of the system to determine the range of parameter coefficients for the search. A large search region is chosen to avoid ignoring good candidate solutions:

$$\begin{aligned}
 K_{SatP}: [0,01 \text{ to } 1000] & \quad K_{SatI}: [0,01 \text{ to } 1000] \\
 K_{cP}: [0,001 \text{ to } 10] & \quad T_{iP}: [0,001 \text{ to } 30] \\
 T_{dP}: [0,001 \text{ to } 10] & \quad \gamma: [10 \text{ to } 50] \\
 K_{vel}: [0,001 \text{ to } 0,1] &
 \end{aligned}$$

As a first simulation scenario, ACO algorithm is used alone. In this case, the number of quantization steps must be

large enough in order to obtain a precise solution. All coefficient intervals have been divided into 75 logarithmically spaced values (i.e. with constant ratio of consecutive values). The iteration number and the number of ants are both fixed to 100.

Fig. 5 demonstrates the efficiency of ACO for tuning of PID controllers with anti-windup for this motion system with flexible coupling and cascaded control loops structure using the modified IAE index. The drive flexibility is observed in the difference between the motor position and grinder table scaled position. As the overshoot is very detrimental to the grinding operations, decreasing overshoot without reducing dynamic performances is highly desirable. Fig. 6 compares the grinder position step responses for  $\sigma = 1$  and  $\sigma = 2$  in performance index (10). These results are obtained at the

expense of excessive computation time (Fig. 7, full ACO algorithm).

To decrease computation time, it is possible to reduce the number of ants or/and the iteration number (Fig. 7, ACO reduced). However, Fig. 8 shows that poor quality solutions are obtained. This is due to the fact that the ants could not converge to an adequate solution in a low number of iterations or if their number is too small.

As a second simulation scenario, NM algorithm alone is applied to the PID controller tuning problem. The interest of using NM is that its search space is real numbers whilst the ACO uses a discretized search space, limiting the quality of its optimal solution. The quality of convergence of NM algorithm depends on the initial conditions. Since the magnitude of each parameter is unknown, it's difficult to establish good initial conditions. It is important to note that it is not sufficient to use the solution found by the initial pole placement since it may not lead to the global or best solution. Several simulations have been carried out using different initial conditions. The coefficients  $KcP$ ,  $TdP$  and  $TiP$  have the most significant impact on the system response. For that reason, these values are varied. Eight possible combinations of initial values for these parameters have been tested; from minimum and maximum values of ACO search space. The remaining parameters have been initialized using the center values of the search space used for ACO.

Fig. 9 shows the poor quality of the solutions found by the NM algorithm for this problem, noting that only a few of the best results are presented. For some initial search parameters, NM method converged to a set of parameters with negative value since parameters search is unconstrained with NM. For the current application, negative values cannot be allowed to assure the system stability. A mechanism must be put into place to generate good quality initial parameter values to use NM. We also observe that computation time is still significant, in particular for certain initial conditions (Fig. 7).

To reach optimal solutions with a reduced computational time, a third simulation scenario has been undertaken using the proposed hybrid ACO-NM algorithm. The proposed algorithm uses the ACO algorithm with to explore rapidly the discretized search space to find a set of initial parameters for a local search using NM algorithm. To reduce computation time while obtaining the desired solution, the previous results obtained with ACO with the iteration number and the number of ants both fixed to 33 (the lowest computation time), have been tested as initial conditions for NM method. Fig. 10 shows the final response of the hybrid algorithm, which are not as good as the first scenario (ACO only). This is explained by the fact that the ACO did not provide initial conditions with sufficient quality to the NM method. This is due to the fact that, if quantization step is too small, the number of ants and iterations must be set to a high number to improve probability of converging close to optimal solutions.

For the ACO to provide better initial conditions to NM algorithm, without increasing execution time, the number of quantization steps must be reduced. In fact, we observed that the ratio between iteration number, number of ants and the number of quantization steps must be properly set, in our

experiment to the same level as the standard ACO algorithm in order to provide good initial conditions for NM method.

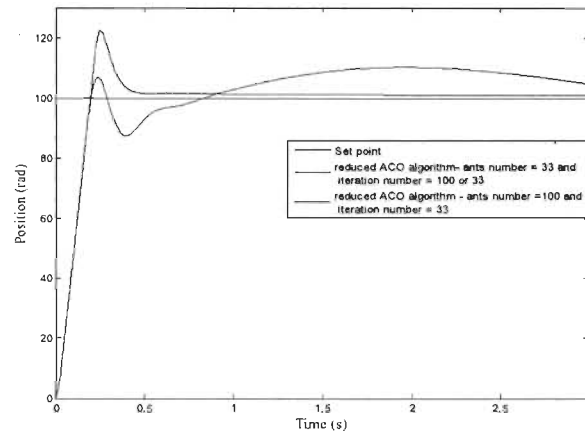


Fig. 8. Step response for PID controller for grinder position using modified LAE index ( $\sigma=1$ ) with ACO reduced.

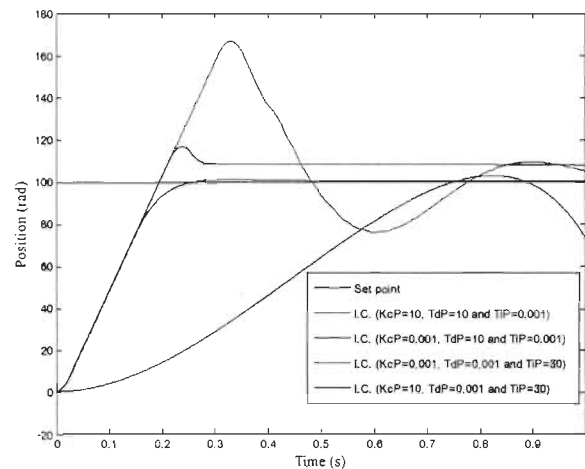


Fig. 9. Step response for the PID controller for the grinder position ( $\sigma=1$ ) using different initial conditions for NM algorithm.

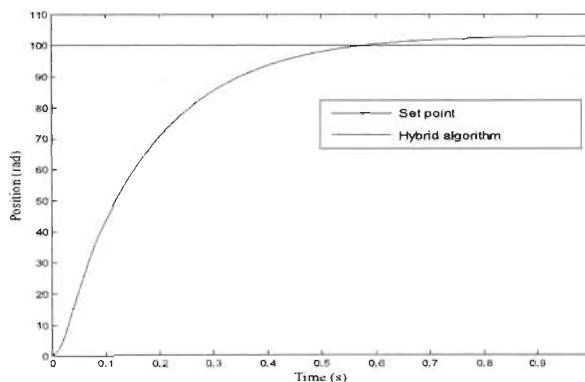


Fig. 10. Step response for PID controller for motor position and grinder position using modified LAE index ( $\sigma=1$ ) with proposed hybrid algorithm (large search space).

Fig. 11 shows the efficiency of the proposed hybrid algorithm when the number of quantization steps is reduced to 25 (with 33 ants and 33 iterations), resulting into improved results as compared to Fig. 5 (ACO with 100 iterations and ants), in approximately 25% of computation time. We also note improvement of the quality of the response with respect to only reducing the number of quantization steps (ACO reduced, with 100 iterations and ants and 25 quantization steps), emphasizing the effect of controller parameter quantization on the solution search.

Fig. 7 shows the mean computation time the algorithms implemented on Matlab took to converge. The efficiency of the ACO algorithm with 100 ants and 100 iterations (full ACO condition) has been reached at the expense of the highest computation time among all the simulated techniques. When reducing the number of ants or/and number of iterations (reduced ACO condition), the computation time reduces drastically; but the solutions obtained with the reduced ACO condition result into poor dynamical performance. When using NM algorithm alone, the mean simulation time is about one third the time required by the full ACO condition. We have noticed that simulation time is tightly related to the initial conditions provided to the NM algorithm. Moreover, NM algorithm alone did not guarantee good performance. Finally, the proposed hybrid algorithm was able to reach better performance than the full ACO condition with approximately 75% reduction in computation time. Therefore, the proposed hybrid algorithm combines the efficiency of full ACO condition in terms of performance and of reduced ACO condition in terms of computation time.

## VI. CONCLUSION

A new hybrid algorithm based on ACO and NM method was proposed for controller tuning for a motion system with flexible coupling and cascaded loops control structure. This approach is better in computing time than ACO algorithm and provides better results than ACO algorithm and NM method applied alone. Simulation results demonstrate the efficiency and the improvement of the method for tuning PID controllers with anti-windup. The flexibility given by the choice of performance criteria, the flexibility on the number of parameters to be adjusted and the reduced computation time make this method very attractive for off-line tuning for complex problems. Moreover, it can be easily adapted to a specific application.

## REFERENCES

[1] F. Padula, and A. Visioli, "Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes," *Control Theory & Applications, IET*, 6(6), pp. 776-786, 2012.

[2] N.J. Killingsworth and M. Krstic, "PID tuning using extremum seeking," *IEEE Control Systems Magazine*, pp. 70-79, February 2006.

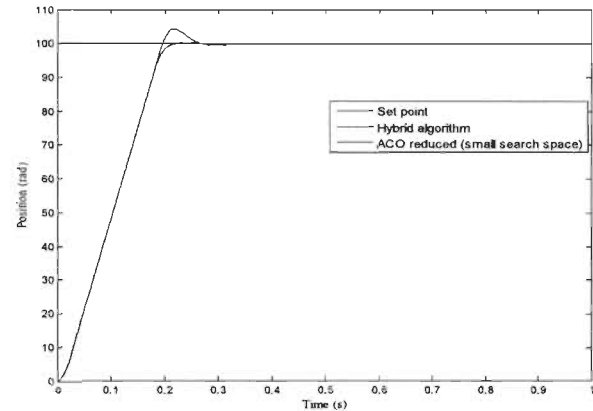


Fig. 11. Step response for PID controller for grinder position using modified IAE index ( $\sigma=1$ ) with proposed hybrid algorithm (properly sized search space with 25 quantization steps).

[3] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective Ant Colony Optimization," *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, Article ID 536326, 2012.

[4] M.-J. Blondin, and P. Sicard, "ACO based controller and anti-windup tuning for motion systems with flexible transmission," *2013 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'13)*, Regina, Saskatchewan, Canada, 5-8 May 2013.

[5] M.-J. Blondin, and P. Sicard, "PID controllers and anti-windup systems tuning using Ant Colony Optimization," *EPE '13-ECCE Europe, the 15th European Conference on Power Electronics and Applications*, Lille, France, 3-5 September 2013.

[6] M. Haghifam, M.-R. Haghifamand and B.S. Chabok, "State estimation electric distribution networks in presence of distributed generation using the PMUs", *CIREN Workshop 2011*, Lisbon, May 29-30, 2012.

[7] M. Tajjudin, N. Ishak, H. Ismail, M. H. Fazalul and R. Adnan, "Optimized PID control using Nelder-Mead Method for electro-hydraulic actuator systems", *2011 IEEE Control and System Graduate Research Colloquium*, Malaysia, pp. 90-93, 2011.

[8] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics-part B Cybernetics*, 26(1), pp. 29-41, February 1996.

[9] Dorigo, M., and T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts, 2004.

[10] J.C. Lagarias, J.A. Reeds, M.H. Wright and P.E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimension," *SIAM J. Optimization*, Society for Industrial and Applied Mathematics, 9(1), pp 112-147, 1998.

[11] Y.-T. Hsiao, C.-L. Chuang, and C.-C. Chien, "Ant Colony Optimization for designing of PID controllers," *2004 IEEE International Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, pp. 321-326, September 24, 2004.

# A Metaheuristic approach for Controller and Anti-Windup Tuning of a Motion System with Flexible Transmission: ACO Algorithm vs Genetic Algorithm

Maude-Josée Blondin and Pierre Sicard

GRÉI, Université du Québec à Trois-Rivières, 3351, boul. des Forges, Trois-Rivières (Qc), Canada  
MaudeJosee.Blondin@uqtr.ca, Pierre.Sicard@uqtr.ca

**Abstract** — Optimal controller and anti-windup tuning for motion systems with flexible coupling and cascaded loops control structure is a practical and difficult problem; in particular when time scale separation is insufficient. Performances of an Ant Colony Optimization (ACO) based methodology and genetic algorithm (GA) are compared for global off-line tuning of Proportional-Integral-Derivative (PID) controllers with anti-windup to counter nonlinearities and saturation problems. Contrary to classical tuning methods, these methods allow simultaneous tuning of a large number of control parameters and provide flexibility in the choice of the performance criterion. Results show that ACO based tuning method provides better system responses than GA approach. Moreover, an advantage with ACO approach is that the search is more correlated with the performance criterion reflecting method flexibility and adaptability.

**Keywords** — Ant colony optimization, Genetic algorithm, PID Controller, Integrator anti-windup, Optimization, Motion systems.

## 1. INTRODUCTION

Tuning PID controllers is important in industrial control systems since it has a direct and significant impact on the system response quality. Methods and rules ( $\lambda$  tuning, Ziegler-Nichols, frequency response methods, heuristic methods, etc.) have been proposed for tuning PID controllers, either to obtain initial setting of their parameters or for fine tuning [1]. Typically, the tuning rules are defined for the basic (ideal) PID controller or for controllers with specific factorization or form. The rules also characteristically respond to particular performance criteria (IAE, ISE, ITAE, etc.). For a nonlinear system with saturation and feedback loop, it is challenging to adjust controller parameters with conventional methods. Moreover, it is customary to sequentially tune the PID and the anti-windup controllers. Since close interaction exists between controller and anti-windup tuning, this may lead to sluggish performance [2]. Therefore, a global approach has to be adopted to tune adequately the PID system with anti-windup.

In [3], Ant Colony Optimization (ACO) was proposed to tune PID controllers with anti-windup. Use of an ACO methodology for offline tuning [4],[5] offers several advantages. Tuning is controller structure independent and it allows considering variants of the structure of the controller, such as tuning of input filters, weighted PID controllers, etc. Cascaded control loops may also be tuned simultaneously in particular when time scale separation is not sufficient. Moreover the choice of performance criterion is flexible and does not require an analytic solution.

To fairly evaluate the performance of the ACO algorithm [4],[5], another metaheuristic algorithm is

used, namely a genetic algorithm (GA). In [6], the effectiveness of the GA-tuned simple PID controller was established. As demonstrated in [4]-[6], a common advantage of this technique and of ACO is the objective function flexibility. Consequently, it is relevant and interesting to analyze the performance of GA in the same environment as [4],[5] to adjust different and more complex controllers, including an increasing number of free parameters.

In this paper, we propose a comprehensive comparison between the ACO tuning method and GA tuning method in order to identify the strengths and weaknesses of both methods. The comparison is applied to a motion system with flexible coupling for tuning cascaded loops control structures with saturation.

In a first scenario, we compare the efficiency of ACO versus GA to tune simultaneously the position loop PID controller with filtered derivative term, its anti-windup feedback gain and the anti-windup gain of the current control loop. In a second and a third scenario, we study the weighted PID controller tuning in the system to cope with disturbances while responding correctly to a desired set point [5]. We compare the efficiency of ACO and GA to tune simultaneously: the weighted position loop PID controller with filtered derivative term (PIDf), its anti-windup feedback gain and the anti-windup gain of the current control loop or with anti-windup for motor velocity limitation. We compare the objective function adaptability and flexibility in order to impose hard or soft current and motor velocity limitation and to penalize specific behaviors such as output overshoot. Performance characteristics in terms of transient response to a step input and step disturbance are also considered.

## 2. ANT COLONY OPTIMIZATION

ACO is an optimization technique inspired by the behavior of ants within a colony [7]. It was observed that a colony facing multiple paths to a food source has the tendency of taking the shortest road. This is explained by the fact that ants communicate by depositing traces of pheromones. The paths with the most pheromones are more attractive to other ants. Therefore, the amount of pheromones increases more rapidly on the shortest path attracting more and more ants. Eventually, all ants will take the same path, i.e. the shortest path. Henceforth, the solution emerges from the collective interaction of the ants.

ACO can be used to solve most of the combinatorial optimization problems for which a constructive heuristic can be employed [4],[5]. The optimization problem is considered as a combinatorial optimization problem and represented by a graph optimization. The ants move probabilistically on the optimization graph trying to minimize an objective function. Therefore, they deposit pheromones on the arcs of the travelled graph depending on the quality of the response. Fig. 1 illustrates the structure of the algorithm.

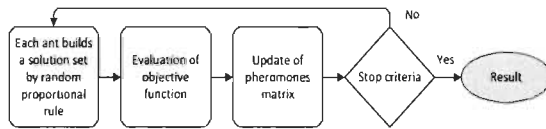


Fig. 1- ACO algorithm structure

## 3. GENETIC ALGORITHM

GA is an optimization technique belonging to the family of evolutionary algorithms (EA) inspired by the concept of natural selection developed by Charles Darwin. From an initial population of various candidate solutions called chromosomes, formed by a set of independent variables (gene), the search space is explored by generating new populations based on natural biological evolution [8]. Objective function of each individual in the population is evaluated. Then, the individuals are scaled under different manners and selected in various ways depending on the application. Fig. 2 shows the scheme of a GA.

The population evolution is gradual and occurs via two main operations which are crossover and mutation in order to minimize or maximize a specific objective function. The crossover mixes some genes of two selected parents to produce a child to create potentially better solutions for the next generation. The mutation operator makes small random alterations on the chromosome to diversify the population in order to allow exploring new search areas and to prevent a population accumulation on a local optimum. Selection among the different ways to perform the crossover and the mutation is based on the application problem.

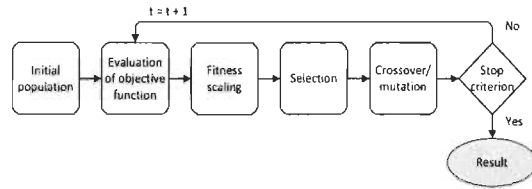


Fig. 2 Evolution scheme of the genetic algorithm

## 4. ACO ALGORITHM AND GA FOR PID AND ANTI-WINDUP TUNING

To evaluate the algorithms, one axis of the positioning table of an industrial grinder system is considered. A 3/4 hp, 5200 rpm, 3.5 N·m dc motor is coupled to a grinder table by a gear box and a worm-gear. The significant flexible coupling due to the transmission elements is modelled as an equivalent lumped spring. Performance criteria include high accuracy, minimal overshoot and good dynamic performance to improve product quality, avoid stress on the grinder and decrease the operation cycle. The flexible coupling complexifies the tuning process to achieve high performance for this cascaded loop control system. The typical solution for this type of system is to use cascaded current and table position control loops with an inner motor velocity loop to improve damping and reduce oscillations.

Tuning for performance is more challenging than for a simple control loop, in particular when considering system constraints such as saturation levels and velocity limitations. Fig. 3 summarizes the system model and optimization method's main input and output parameters for controller and anti-windup parameter tuning, which is subjected to various cases of system constraints.

The first position controller is illustrated in Fig. 4. It is composed of a PID controller with filtered derivative term, motor velocity feedback ( $K_{vel}$ ) and an anti-windup controller. The inner PI current controller was tuned separately but its anti-windup feedback gain  $K_{sat}$  (Fig. 3) is to be tuned by ACO or GA.

Fig. 5 presents the structure of the weighted  $PID_f$  controller where parameters  $W_p$  and  $W_v$  represent the weighting factors on the proportional and derivative terms of the  $PID_f$  controller. They affect the zeroes of the transfer function from the reference position to the actual grinder table position.  $W_p$  and  $W_v$  offer additional degrees of freedom for control so that the controller can yield better performance for set point response and disturbance rejection simultaneously.

Fig. 6 shows the last controller structure used where motor velocity limitation is considered instead of current saturation for anti-windup. Both could have been considered simultaneously, but considering that the motor is lightly loaded, its current does not saturate for significant periods and a simple saturator was deemed sufficient for this scheme.



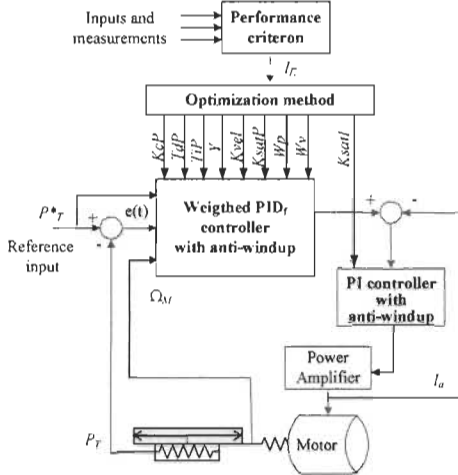


Fig. 3: System overview for tuning  $PID_f$  and anti-windup system parameters

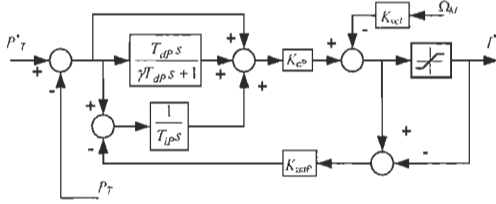


Fig. 4: Position feedback controller

The combinatorial optimization graph for the application of ACO is shown on Fig. 7 [3]-[5]. Each slice of the graph corresponds to nodes of quantized controller parameters. This representation allows easy addition or removal of free controller coefficients. Range of parameters and quantization steps must be chosen to complete the graph of Fig. 7. Rough tuning of the control parameters can be used to define the range of parameters. The ants move on the graph from left to right by choosing the nodes they travel from probabilistic rules. The performance is evaluated by  $I_E$  index resulting from the set of parameters on the travelled path. A phomone matrix is updated as a function of the quality of the solutions.

To use GA for controller tuning, an individual (chromosome) in the population is represented by  $Ch = [K_{cP} \ T_{dP} \ T_{iP} \ K_{vel} \ \dots \ K_{satf} \ W_v \ W_p]$  where each gene represents a parameter to tune. This representation can handle different number of parameters to tune by adding or removing a gene in the chromosome. The operations are performed in order to minimize the performance index  $I_E$ . The search space is constrained by a lower and upper bounds. An important difference of both methods is that ACO has a beforehand discretized search space comparatively to a continuous search space for GA.

The objective of the ACO and GA is to find the best set of parameters to minimize  $I_E$ . The choice of the performance criterion is critical to obtain the desired behaviour with regards to transients, steady state and inner variables limitation. To simultaneously consider

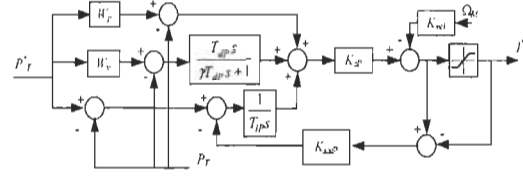


Fig. 5: Weighted  $PID_f$  controller structure with anti-windup for current reference limitation.

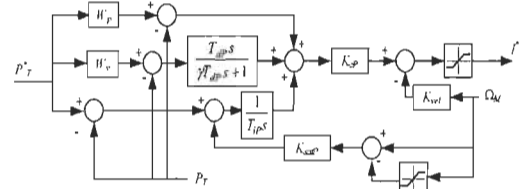


Fig. 6: Weighted  $PID_f$  controller structure with anti-windup for motor velocity limitation.

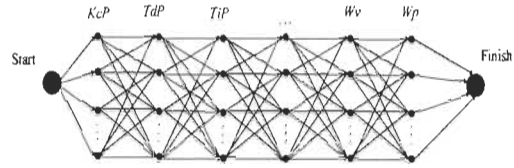


Fig. 7: Optimization graph for controller structure coefficients used by ACO tuning method

set point performance and disturbance rejection criteria, a performance index with normalized quantities was defined as follows [5]:

$$I_E = \sum \left( \int_{t_s}^{t_s+T_s} \frac{|e(t)|}{\Delta P_T^*} dt \right) + K_t \sum \left( \int_{t_p}^{t_p+T_p} \frac{|e(t)|}{\Delta T_{pert}} dt \right) + \sigma \int_0^{T_f} \frac{e_d(t)}{\Delta P_T^*} dt + \beta \int_0^{T_f} \frac{e_{v_d}(t)}{\Omega_{Mm}} dt \quad (1)$$

The first two terms in (1) are summations over each of the set point step input and disturbance input transients respectively:  $t_s$  is the time of last set point variation;  $T_s$  is the approximate settling time for the set point step response;  $t_p$  is the time of last disturbance input variation;  $T_p$  is the approximate settling time for the step disturbance;  $e(t) = P^*(t) - P(t)$  is the output error;  $P^*$  and  $P$  are respectively the measured and reference positions of the grinder table;  $\Delta P_T^*$  is the amplitude of the last step input variation used to normalize the position error;  $\Delta T_{pert}$  is the amplitude of the last force disturbance variation;  $K_t$  is a positive constant of equivalent stiffness used to normalize the position error due to force disturbance;

The third term penalizes position overshoots:  $T_f$  is the simulation time as this term is penalized continually;  $\sigma$  is a positive constant weight;  $e_d(t)$  is defined as

$$e_d(t) = \begin{cases} |e(t)| & \text{during response overshoots} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The fourth term penalizes motor velocity limit violations:

$$e_{vd}(t) = \begin{cases} | |\Omega_M(t)| - \Omega_{Mm} | & \text{if limit exceeded} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $\Omega_{Mm}$  is the upper motor velocity limit.

Larger values of  $\sigma$  in (1) penalize more the overshoot, which is very detrimental to the quality of the grinding operations. Larger values of  $K_r$  are used to reduce the impact of the disturbance on the system.  $\beta$  is used to constrain the motor speed to respect the system limits.

For analysis, tuning methods are compared with some characteristic measures on the system responses [9]:

- $t_r$  is the rise time; time required for the response to rise from 0% to 100 % of its final value;
- $t_p$  is the peak time; time required for the system to reach its first peak overshoot;
- $M_p$  is maximum (percent) overshoot

$$M_p = \frac{P_T(t_p) - P_T(\infty)}{P_T(\infty)} \times 100 \quad (4)$$

- $t_s$  is the settling time; time required to reach and stay within a specific range around its final value.

Responses to disturbances are characterized by:

- $t_{pert}$  is the disturbance settling time; time required to converge within a specific range around its final value.
- $M_{pert}$  is maximum normalized deviation

$$M_{pert} = \frac{c(\infty) - c_{min}}{\Delta T_{pert}} \quad (5)$$

## 5. SIMULATION RESULTS

The ACO and GA algorithms were implemented in Matlab® and the system's model on Simulink™. To determine a range of parameter coefficients for the search, the position PID<sub>r</sub> controller's parameters were initially determined by pole placement from the approximate rigid model of the system. The subsequent ranges were chosen for the search parameter coefficients. The search space of possible solutions is enlarged to avoid ignoring good candidate solutions. The same range is used as lower and upper bounds for GA.

$$\begin{array}{lll} K_{SatP}: [1,1000] & W_p: [0,1] & T_{iP}: [0.003,30] \\ K_{cP}: [0.001,10] & K_{vel}: [0.01,0.1] & \gamma: [10,50] \\ T_{dP}: [0.001,10] & K_{SatI}: [1,1000] & W_p: [0,1] \end{array}$$

As a first scenario, the optimization methods are applied on the control structure shown in Fig. 4 where 7 parameters must be tuned. For ACO methodology, the iteration number is fixed at 100 and the number of ants to 100. For GA method, the population is composed of 100 chromosomes and the number of generations is fixed to 100. Parameters  $K_r$  and  $\beta$  of the performance criterion (1) are set to 0 and no step load disturbance is applied. Therefore, the IAE performance index augmented with an additional term to penalize overshoot is being used. As overshoot is

very detrimental to the grinding operations, decreasing overshoot without reducing dynamic performances is highly desirable. Different value of  $\sigma$  are used to highlight the optimization method adaptability relative to performance criterion.

Fig. 8 demonstrates the efficiency of ACO to tune PID controllers with anti-windup for this motion system with flexible coupling and cascaded control loops structure. The results show that the search is consistent with the performance criterion to be minimized. Higher is  $\sigma$  value, lower is the overshoot and  $I_E$  value also decreases in accordance (Table I).

Fig. 9 shows the step responses obtained with GA parameters tuning. The performances of the GA algorithm are not acceptable since only one of three results is satisfactory. To obtain adequate results (Fig. 10), simulation time of the model had to be increased to 4s instead of 1s as required for ACO tuning. In this case, computation time increases considerably (Fig. 11). Response curves obtained by GA tuning present more oscillations before reaching stabilization and when  $\sigma=2$  with GA tuning, the system is overdamped and performance specifications are middling.  $I_E$  value did not decrease with increasing  $\sigma$  (Table I).

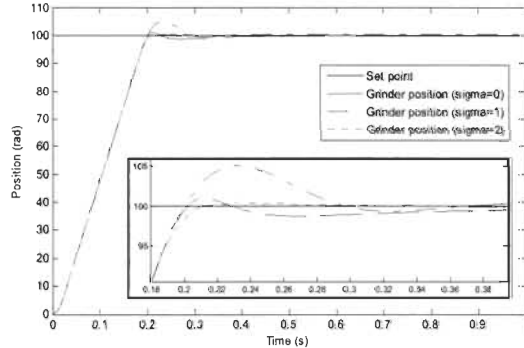


Fig. 8- Input step response for PID controller with current anti-windup — ACO tuning

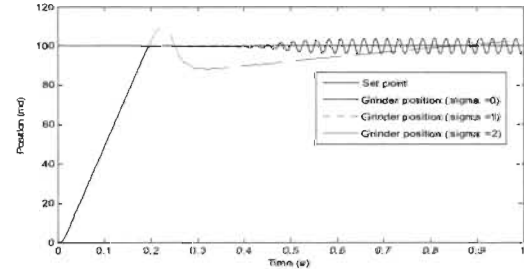


Fig. 9- Input step response for PID controller with current anti-windup — GA tuning

Table I Performance measures for 1<sup>st</sup> scenario

Method	$\sigma=0$		$\sigma=1$		$\sigma=2$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.201	0.200	0.203	0.221	0.213	0.753
$t_p$ (s)	0.230	0.223	0.2107	0.221	0.230	0.753
$M_p$ (%)	5.100	0.700	1.100	0	0.400	0
$t_s(2\%)$ (s)	0.271	0.192	0.1957	0.193	0.198	0.243
$I_E$	0.119	0.105	0.107	0.104	0.105	0.107

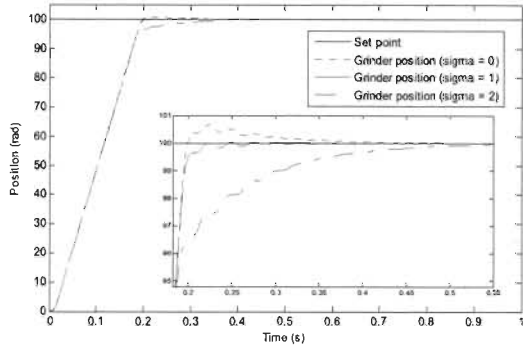


Fig. 10- Input step response for the PID controller with current anti-windup — GA tuning with simulated time = 4s

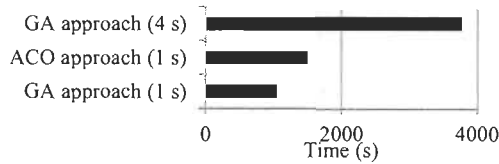


Fig. 11. Computation time of ACO and GA tuning for simulated time of 1s and 4s

According to the overall results, optimization using ACO provides better performance. By its stronger correlation tendency with the performance criteria in this experiment, it offers greater control over the system response; this may be explained by a greater sensitivity to the performance criterion. Also, the computation time is significantly lower.

As a second simulation scenario, the control structure shown on Fig. 5 is used. The iteration number and the number of ants for ACO methodology are both increased to 200 due to a larger number of parameters to be adjusted (9 parameters). For GA tuning, the population is also increased to 200 chromosomes and the number of generations to 200. In the performance index (1),  $\sigma$  is set to 2,  $\beta$  is set to 0 and the  $K_t$  takes one of the values: 0, 0.075 and 0.75. Increasing of  $K_t$  value improves disturbance response at the cost of the transient step response quality. A step load disturbance corresponding to 20% of the nominal torque (0.75 Nm) is applied to the system at 2 s.

Fig. 12 and Fig. 13 show the results obtained with ACO tuning versus GA tuning for  $K_t = 0.75$ . Other results to different  $K_t$  values exhibit the same behaviour than Fig. 12. Table II resumes transient-response and disturbance specifications of both methods with different  $K_t$  values. As shown in Fig. 12 and 13, ACO tuning response is better than GA tuning response. In the transient response, the overshoot is lower and the settling time  $t_s$  is shorter. For the disturbance response, even though the maximum normalized deviation is slightly larger, the settling time  $t_{peri}$  is much shorter and the overshoot is less important. Moreover, globally, the amplitude and the oscillation frequency on transient and disturbance responses are smaller with ACO tuning.

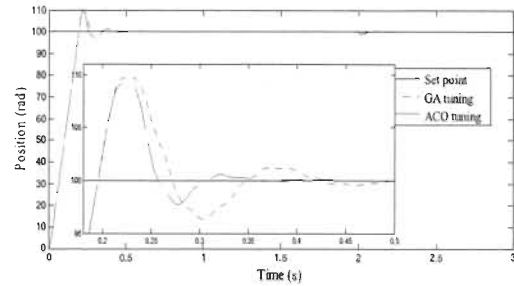


Fig. 12- Input and disturbance steps response and enlargement of input step response for weighted PID controller ( $\sigma = 2, K_t = 0.75$ )- ACO vs GA tuning

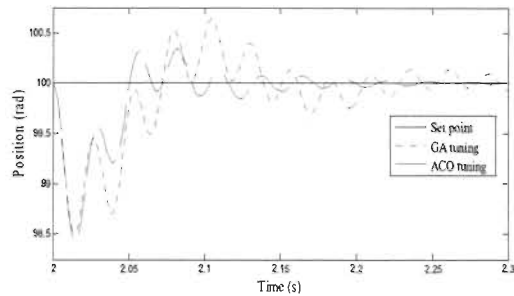


Fig. 13 Enlargement of disturbance response of Fig 12

Table II. Performance measures for 2<sup>nd</sup> scenario

Method	$K_t=0$		$K_t=0.075$		$K_t=0.75$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.222	0.214	0.213	0.217	0.196	0.196
$t_n$ (s)	0.234	0.218	0.220	0.219	0.221	0.221
$M_p$ (%)	0.500	0.400	0.6	0.100	9.020	9.7
$t_s(2\%)$ (s)	0.202	0.193	0.194	0.194	0.283	0.329
$M_{peri}$ (rad/Nm)	9.253	2.080	2.453	2.520	2.107	2.000
$t_{peri}$ (s)(1%)	0.266	0.047	0.070	0.070	0.022	0.044
$I_E$	0.107	0.104	0.116	0.117	0.167	0.213

To have precise control over the system, disturbance response improvement should be directly linked to the increasing  $K_t$  value. From Table II, this is the case for ACO tuning, but not for GA tuning. With ACO tuning, the maximum normalized deviation value ( $M_{peri}$ ) and disturbance settling time ( $t_{peri}$ ) decrease gradually when  $K_t$  value increases. According to the chosen performance criterion, ACO tuning is able to better impose the system response behavior while keeping good dynamic performance.

As a third simulation scenario, the control structure shown on Fig. 6 is used to improve the overall response. The excessive overshoot shown on Fig. 12 is due to the integrator windup in the position controller due to natural motor velocity saturation. Then, inner limitations must be considered and penalized in the performance index. Moreover, in order for the response in saturation mode during large input step variations not to dominate the performance criterion during tuning over the small signal operation, large (100 rad at 0 s) and small (5 rad at 2 s) input step variations are applied to the system for optimization. A step disturbance corresponding to 20% the motor's nominal torque is applied at 4 s.

Parameter  $\beta$  is set to 100 in (1). A larger value of  $\beta$  allows implementing hard constraints on motor velocity, whilst a small value would implement a softer constraint. Henceforth, the value of  $\sigma$  is increased to 100 to implement a hard constraint on position overshoot. The optimization parameters for both tuning methods and the variation of  $K_t$  value are the same as for the second simulation scenario.

Fig. 14 and Fig. 15 demonstrate the better efficiency of ACO method than GA method for tuning the weighted PIDf controller response with motor velocity saturation when  $K_t = 0.075$ . The step response is faster and disturbance impact is lower for ACO tuning. Table III resumes transient-response and disturbance specifications for both methods.

The response curves are coherent with increasing values of  $K_t$ , and disturbance response improves at the cost of transient step response. ACO tuning provides the best results for two of three  $K_t$  values ( $K_t = 0$ ,  $K_t = 0.075$ ). Indeed,  $t_r$ ,  $t_p$ ,  $t_s$  are halved,  $M_{pert}$  is reduced in half and  $t_{pert}$  is approximately six and three times shorter. When  $K_t = 0.75$ , step response is better with ACO tuning while disturbance response is better with GA tuning. Overall, ACO method is more efficient than GA method for tuning the PID controller structures.

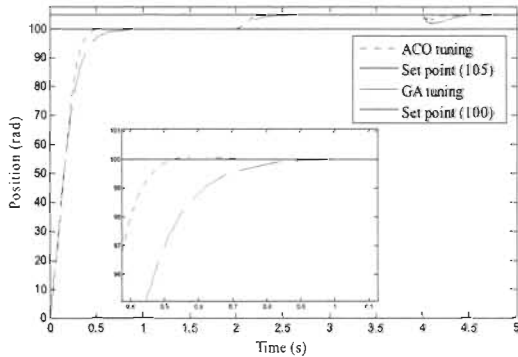


Fig 14. Enlargement of input and disturbance steps response for weighted PIDf controller and motor velocity saturation ( $K_t = 0.075$ ) ACO vs GA tuning

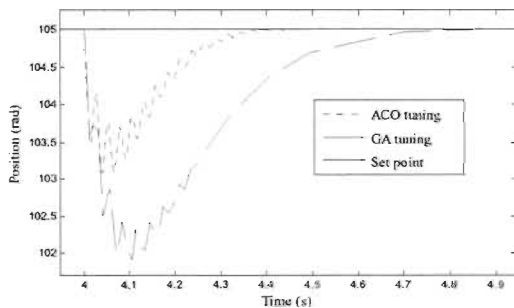


Fig 15. Enlargement of disturbance response of Fig.

Table III. Performance measures for 3<sup>rd</sup> scenario

Method	$K_t=0$		$K_t=0.075$		$K_t=0.75$	
	ACO	GA	ACO	GA	ACO	GA
$t_r$ (s)	0.564	1.251	0.524	1.042	0.943	1.126
$t_p$ (s)	0.564	1.251	0.537	1.042	0.943	1.255
$M_p$ (%)	0	0	0.100	0	0	0
$t_s$ (2%) (s)	0.400	0.750	0.400	0.547	0.475	0.514
$M_{pert}$ (rad/Nm)	2.533	4.800	2.533	4.133	2.00	1.467
$t_{pert}$ (s) (1%)	0.132	0.587	0.151	0.353	0.098	0.051
$I_E$	0.321	0.445	0.354	0.450	0.483	0.457

## 6. CONCLUSION

A comparison between ACO method and GA method was proposed for parameters tuning for different controller structures. The effectiveness of both methods was tested on three simulation scenarios with different performance criteria related to the transient-response specifications and disturbance response specifications. Nonetheless, another desirable feature is to obtain a global tuning method by which performance criterion can provide precise control for various controller structures. Overall, ACO method has shown higher efficiency and provided better results. The comparison leads us to highlight the strength and advantage of ACO tuning. For the same conditions, this method tends to better correlate the results and the performance criterion, for shorter simulated periods. Since ACO and GA are probabilistic, tests with a large number of simulations are being conducted to draw more conclusions.

## REFERENCES

- [1] F. Padula, and A. Visioli, "Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes," *Control Theory & Applications*, IET, 6(6), pp. 776-786, 2012.
- [2] N.J. Killingsworth and M. Krstic, "PID tuning using extremum seeking," *IEEE Control Syst Mag.*, pp. 70-79, February 2006.
- [3] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective Ant Colony Optimization," *Appl Comp Intell and Soft Computing*, Hindawi Publishing Corporation, Article ID 536326, 2012.
- [4] M.-J. Blondin, and P. Sicard, "ACO based controller and anti-windup tuning for motion systems with flexible transmission," *2013 IEEE CCECE Conference*, Regina, Sask., Canada, 5-8 May 2013.
- [5] M.-J. Blondin, and P. Sicard, "PID controllers and anti-windup systems tuning using Ant Colony Optimization," *EPE '13-ECCE Europe Conference*, Lille, France, 3-5 Sept. 2013.
- [6] M.J. Neath, A.K. Swain, U.K. Madawala, and D.J. Thrimawithana, "An Optimal PID Controller for a Bidirectional Inductive Power Transfer System Using Multiobjective Genetic Algorithm," *IEEE Trans. Power Electron.*, vol.29, no.3, pp.1523-1531, March 2014
- [7] M. Dorigo, and T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts, 2004.
- [8] L. Haupt, R., and S. E. Haupt, *Practical Genetic Algorithms*, Wiley-Interscience, New York, N.Y., 1998.
- [9] K.Ogata, *Modern Control Engineering*, Prentice Hall, 5<sup>th</sup> ed. Pearson, 2010.

# A Hybrid ACO and Nelder-Mead constrained Algorithm for Controller and Anti-Windup Tuning

Maude-Josée Blondin and Pierre Sicard

Groupe de Recherche en Électronique Industrielle, Université du Québec à Trois-Rivières  
Trois-Rivières (Québec), Canada

Tel.: +1 (819) 376-5011.

E-Mail: MaudeJosee.Blondin@uqtr.ca; Pierre.Sicard@uqtr.ca

## Acknowledgements

This work was supported by a fellowship and a research grant from the Natural Sciences and Engineering Research Council of Canada.

## Keywords

Optimal control, Motion control, Active damping, Autotuning, DC machine, Non-linear control

## Abstract

Nonlinear controller tuning with anti-windup is not trivial since the parameters within the structure are interdependent. A global optimization-based tuning approach can handle this interdependence. A hybrid Ant Colony Optimization (ACO) and Nelder-Mead (NM) algorithm for controller and anti-windup tuning efficiently solves this problem. This paper proposes a new way to constrain ACO-NM algorithm to ensure positivity of key parameters in order to avoid searches in instability or low robustness regions and to improve computation time to reach an optimal solution.

## 1. Introduction

Tuning PID controllers is important in industrial control systems since it has a direct and significant impact on the system response quality. For a nonlinear system with saturation and feedback loops, it is challenging to adjust controller parameters with conventional methods [1]. It is customary to sequentially tune the PID and the anti-windup controllers. In [2], PID coefficients are tuned using the relay autotuning method and the anti-windup gain is fixed to the inverse of the integrator gain. Empirical adjustments are performed with the intention to improve the system response. However, since close interaction exists between controller and anti-windup tuning, the sequential approach is not efficient. In fact, it may lead to sluggish performance [3]. Therefore, a global approach is favored to tune adequately in one step the PID system with anti-windup. In [4] and [5], global optimization such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolution Programming (EP) and Particle Swarm Optimization (PSO) is applied to tune PID controllers for different systems. Their performances with regards to the quality of the system response and the convergence time were compared. ACO and GA are the methods with the highest potential in terms of those performances. However, ACO has fewer terms to set within the method than GA, which simplifies its application.

In [6] and [7], a global tuning method based on Ant Colony Optimization (ACO) has been proposed. This method is easier to apply and provides better results than the sequential approach [3][6]. However, the computation time increases with the number of parameters to adjust, which leads to an excessive computation time for the algorithm to converge to a desired solution. In [8], to reduce the excessive computation time, a hybrid algorithm based on ACO [9] and Nelder-Mead simplex search (NM) [10] has been proposed to tune a motion system with flexible coupling and cascaded loops control structure. The hybrid algorithm allows the ants to explore a large search space in order to find a good set of parameters as initial search point for the NM algorithm.

Since NM algorithm performances in terms of computation time and efficiency depend on the initial search point, the algorithm converges rapidly with the suitable initial conditions provided by ACO. However, being unconstrained and for some initial search parameters, the NM method converges to a

set of parameters with negative values. For the current application, negative values cannot be allowed to assure system stability and robustness. Therefore, in order to determine a good NM starting point to reach satisfactory performances, the parameterization of the algorithm must be precise, which is subject to change from one controller structure to another. Then, in order to easily extend the applicability of the hybrid method to other control structures, it is mandatory to get rid of the dependency between ACO parameters and the convergence of NM method parameters to positive values.

In part of [11], limiting parameter values to their constraint values can be applied to constrain the values for NM algorithm. For PID controller tuning, this procedure wouldn't take into account the interaction between different parameters to reach a global optimum. In this paper, a new way to constraint the NM method within the hybrid algorithm is proposed. The solutions found by ants are used to execute the NM local search. The latter algorithm searches the neighborhood of those initial points. If the solution of ants is slightly away from an optimal region with positive parameters, and since NM algorithm is constrained, it is now able to compensate this deviation bringing the solution to a positive region. The information provided by the parameter value when exceeding the limit is used to rectify the negative value while maintaining consistency in the ratio between all parameters. The proposed approach, allows reducing the number of ants and the number of iterations for minimizing the computation time. In addition, as a major advantage, local searches can be performed earlier in the optimization process for some controller structures.

This new constrained hybrid method can tune simultaneously and more effectively the parameters of cascaded loops control structures with saturation for a motion system, whilst responding to the need of having a reliable, fast and constrained global tuning method. The method flexibility is demonstrated by modifying the performance criteria. Hard or soft current and motor velocity limitation can be imposed. Also, specific behaviors such as output overshoot can be penalized. These advantageous properties are observed on four different controller structures thereby verifying the diversity of the application of the proposed optimization method.

The remaining paper is organized as follows. Section 2 introduces ACO and NM algorithms. Section 3 explains the new constrained hybrid algorithm for PID and anti-windup tuning for the controller structure. Simulation results are presented in Section 4. Finally, main conclusions are drawn and future work is introduced in section 5.

## 2. ACO and NM algorithms

### 2.1 ACO algorithm

ACO is an optimization technique inspired by the behavior of real ants within a colony [9]. It was observed that a colony facing multiple paths to a food source has the tendency of taking the shortest road. This is explained by the fact that ants communicate with each other via their environment by depositing traces of pheromones. The paths with the most pheromones are more attractive to other ants. Therefore, the amount of pheromones increases more rapidly on the shortest path attracting more and more ants. Eventually, all ants will take the same path, i.e. the shortest path. Henceforth, the solution emerges from the collective interaction of the ants.

The optimization problem must be considered as a combinatorial optimization problem and represented by an optimization graph. The ants move probabilistically on the optimization graph trying to minimize an objective function. Therefore, they deposit pheromones on the arcs of the travelled graph depending on the quality of the response. Fig. 1 illustrates the structure of the algorithm.

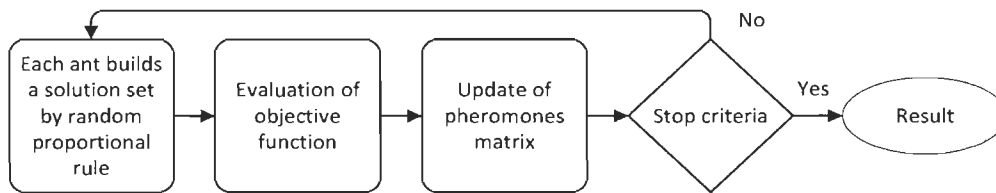


Fig. 1- ACO algorithm structure.

## 2.2 Nelder-Mead algorithm

Nelder-Mead simplex search is part of general class of direct search methods. It is one of the most popular algorithms for solving unconstrained nonlinear optimization problems without using derivatives [10] and it is generally used for local optimization. NM algorithm tends to minimize a nonlinear scalar function of  $n$  variables from a starting point estimated by using only the evaluation of the objective function.

Initially, NM algorithm performs a simplex of  $n+1$  vertices from a starting point  $x_0$ . A certain percentage (usually 5%) is added on each  $x_0(i)$  to  $x_0$  where  $i=1, \dots, n$ . If initially a component of  $x_0$  is zero, a very small value is used. Then, the first simplex is created. Thereafter, the simplex moves in the unconstrained search space in order to minimize an objective function. The procedure for the iterations is done according to 5 steps: 1) Ordering 2) Reflection, 3) Expansion, 4) Contraction and 5) Shrinkage [10]. Fig. 2 presents the NM algorithm scheme.

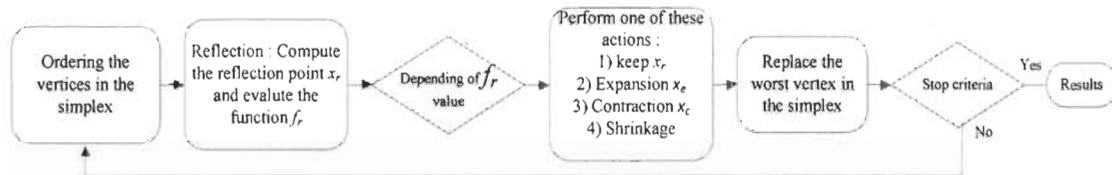


Fig. 2- NM algorithm Scheme.

## 3 Hybrid ACO-NM constrained algorithm for PID anti-windup tuning

The hybrid ACO-NM algorithm consists of two main steps: (i) global optimization using rough quantization of parameters using ACO to obtain high quality initial conditions for NM; (ii) local optimization using NM for fine tuning of parameters.

To evaluate the hybrid algorithm, one axis of the positioning table of an industrial grinder system is considered. A 3/4 hp, 5200 rpm, 3.5 N·m dc motor is coupled to a grinder table by a gear box and a worm-gear. The significant flexible coupling due to the transmission elements is modelled as an equivalent lumped spring. Performance criteria include high accuracy, minimal overshoot and good dynamic performance to improve product quality, avoid stress on the grinder and decrease the operation cycle. The flexible coupling complicates the tuning process to achieve high performance for this cascaded loop control system.

The typical solution for this type of system is to use cascaded current and table position control loops with an inner motor velocity loop to improve damping and reduce oscillations. Tuning in this case is more challenging than for a simple control loop, in particular when considering system constraints such as saturation levels and velocity limitations. Fig. 3 summarizes the system model and optimization method's main input and output parameters for controller and anti-windup parameter tuning, which is subjected to various cases of system constraints.

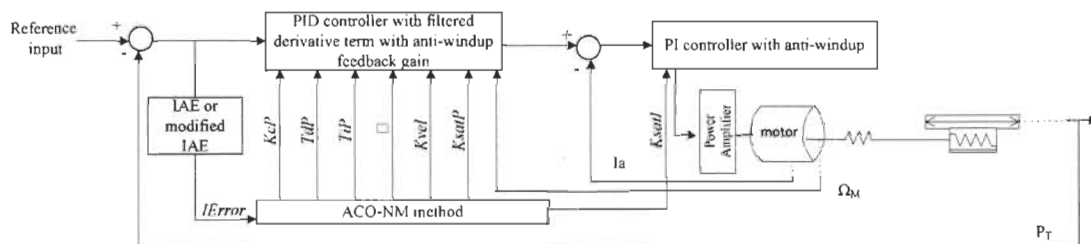


Fig. 3- System overview with ACO-NM application for PID parameters' tuning.

The proposed hybrid algorithm is applied to four different controller structures. Firstly, the position feedback controller is illustrated in Fig. 4. It is composed of a PID controller with filtered derivative term, motor velocity feedback ( $K_{vel}$ ) and an anti-windup controller. The inner PI current controller was tuned separately but its anti-windup feedback gain  $K_{sat}$  (Fig. 3) is to be tuned by the hybrid ACO-

NM constrained algorithm. Fig. 5 presents the structure of the weighted  $PID_f$  controller where parameters  $W_p$  and  $W_v$  represent the weighting factors on the proportional and derivative terms of the  $PID_f$  controller, effectively modifying the zeroes of the transfer function from the reference position to the actual grinder table position. Motor velocity limitation is managed by an anti-windup scheme. Fig. 6 shows the controller structure with current saturation managed with anti-windup. Fig.7 considers simultaneously both current saturation and velocity limitation.

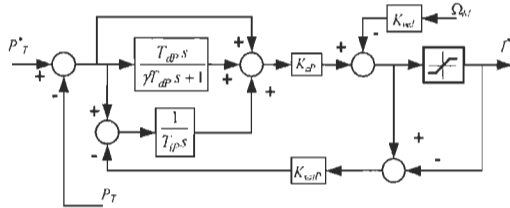


Fig. 4- Position feedback controller

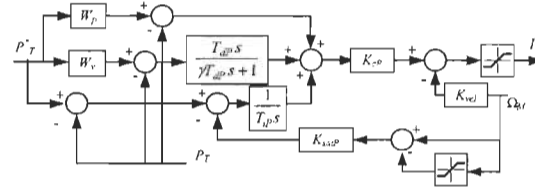


Fig. 5- Weighted  $PID_f$  controller structure with anti-windup for motor velocity limitation.

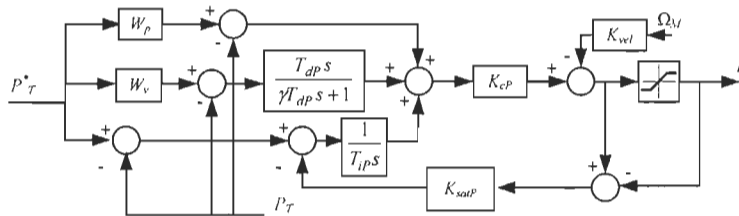


Fig. 6- Weighted  $PID_f$  controller structure with anti-windup for current reference limitation.

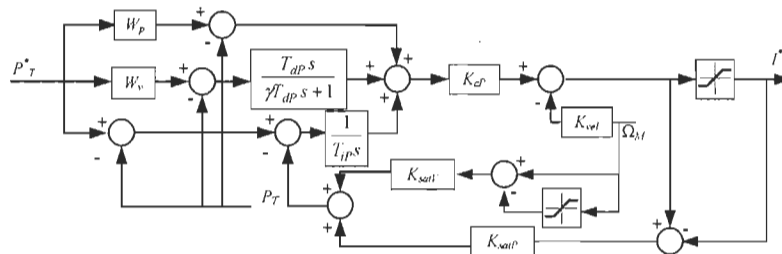


Fig. 7- Weighted  $PID_f$  controller structure with anti-windup for motor velocity limitation and current reference saturation.

In order to apply ACO, PID anti-windup tuning should be considered as a combinatorial optimization problem which is shown on Fig. 8 [12].

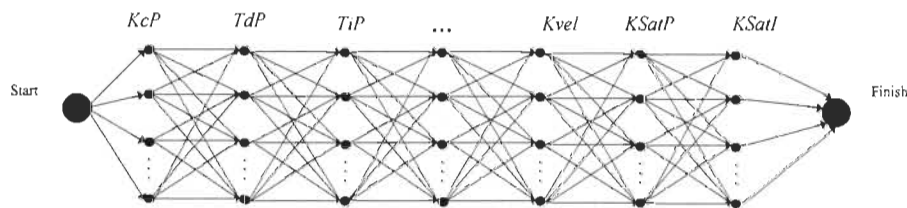


Fig. 8- Optimization graph for filtered position PID controller and anti-windup coefficients.

Each slice (column) of the graph corresponds to one controller parameter and each node is a quantized value. This representation allows easy addition or removal of controller coefficients. The ants move on the graph from left to right by choosing the nodes they travel from probabilistic rules. The performance is evaluated by  $I_E$  index (5) resulting from the set of parameters on the travelled path. A



pheromone matrix is updated as a function of the quality of the solution so as to influence the next ants' exploration.

The best solutions found by ants are used as initial conditions to perform NM constrained algorithm. The positivity constraint is introduced in the NM iterations by adding a new operation: before inserting a new term  $x$  in the simplex, sign verification is performed on each parameter value to ensure that it is positive. Since the PID and anti-windup parameters are interdependent, all parameter are considered when a negative parameter is constrained. The procedure of the new step is as follow:

**Step positivity constraint: Overall reduction.** Compute the value of the exceeding amount on the negative parameter relative to a value near the constraint value.

$$\delta(i) = \begin{cases} |x(i) - \varphi| & \text{if } x(i) < 0 \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, n \quad (1)$$

where  $x(i)$  is the point computed previously for the simplex,  $n$  is the number of variables and  $\varphi$  is the constraint value on this parameter;  $\varphi = 10^{-4}$  is used in the remainder of the paper.

Compute the difference between each point.

$$d(i) = |x(i) - x_{n+1}^{(k)}(i)|, \quad i = 1, \dots, n \quad (2)$$

where  $x_{n+1}^{(k)}(i)$  is the point to be replaced in the simplex.

Calculate the ratio of the exceeding measure relative to the distance.

$$\rho(i) = \frac{\delta(i)}{d(i)} \quad (3)$$

Proceed to the overall reduction

$$x(i) = \begin{cases} x(i) + \left( \max_i(\rho(i)) \right) d(i) & \text{if } x(i) < x_{n+1}^{(k)}(i) \\ x(i) - \left( \max_i(\rho(i)) \right) d(i) & \text{otherwise} \end{cases}, \quad i = 1, \dots, n \quad (4)$$

The point  $x(i)$  is introduced in the simplex and the iteration is terminated.

The hybrid constrained algorithm minimizes the performance criterion  $I_E$  defined in (5) [7]: To simultaneously consider set point performance and disturbance rejection criteria, a performance index with normalized quantities was defined as follows (some coefficients are set to zero for some of the tests):

$$I_E = \sum \left( \int_{t_s}^{t_s+T_s} \frac{|e(t)|}{\Delta P_T^2} dt \right) + K_t \sum \left( \int_{t_p}^{t_p+T_p} \frac{|e(t)|}{\Delta T_{pert}} dt \right) + \sigma \int_0^{T_f} \frac{e_d(t)}{\Delta P_T^2} dt + \beta \int_0^{T_f} \frac{e_{v_d}(t)}{\Omega_{Mm}} dt \quad (5)$$

Larger values of  $\sigma$  in (5) penalize more the overshoot, which is very detrimental to the quality of the grinding operations. Larger values of  $K_t$  are used to reduce the impact of the disturbance on the system.  $\beta$  is used to constrain the motor speed to respect the system limits.

## 4 Simulation results

The ACO and NM constrained algorithms were implemented in Matlab® and the system's model on Simulink™. The search space for ACO is defined to include the results of an initial rough controller tuning (pole placement for rigid model of the system) and is defined such as logarithmically spaced intervals (some parameters are unused depending on the controller structure):

$$K_{Satp}: [0.01, 1000]; \quad T_{ip}: [0.001, 30]; \quad W_p: [0, 1]; \quad K_{cP}: [0.001, 10]; \quad T_{dP}: [0.001, 10]; \\ K_{Satl}: [0.01, 1000]; \quad \gamma: [10, 50]; \quad W_v: [0, 1]; \quad K_{vel}: [0.001, 0.1]; \quad K_{Satv}: [0.01, 1000]$$

As the magnitude of the terms of the best set of parameters is unknown, a large search region is chosen to insure inclusion of the global optimum region. Since NM is constrained to avoid negative values, for the NM search, the lower bound is 0.0001 for all parameters and no upper bounds are imposed.

#### 4.1 PID controller with current based anti-windup

The first simulation scenario is applied to the controller structure shown on Fig. 4. In the performance index (5),  $\sigma$ ,  $K$ , and  $\beta$  are set to 1, 0 and 0 respectively (no perturbation is applied). The local and global searches have been executed simultaneously. ACO explores a large search space to find several sets of initial parameters for NM constrained algorithm. In many cases of hybrid algorithms [9] performing simultaneously local search and global search is effective, but for the controller under study, it is not the case. Applying ACO and constrained NM in parallel can be less efficient than a sequential approach. Fig. 9 shows the computation time of both methods. For NM constrained algorithm, convergence time is directly related to the initial starting point. With good initial conditions, the algorithm converges rapidly to a desired solution while convergence with poor initial conditions is slower. The average convergence time of NM algorithm used in parallel with ACO is 300 s. It represents a significant computation time knowing that ACO (100 iterations and 100 ants) takes 1500 s to converge through a desired solution. Then, a maximum of 4 NM searches can be launched during the exploration of the ants in order to keep the same computation time than ACO alone. This procedure doesn't provide significant improvement on the system response relative to ACO alone and the computation time is still expensive. As shown on Fig. 9, for ACO, computation time is correlated with the number of ants ( $m$ ) and iterations ( $n$ ). A reduction of both of them decreases substantially the computation time. Therefore, it becomes interesting to perform NM constrained algorithm after the search space exploration by a small number of ants and iterations.

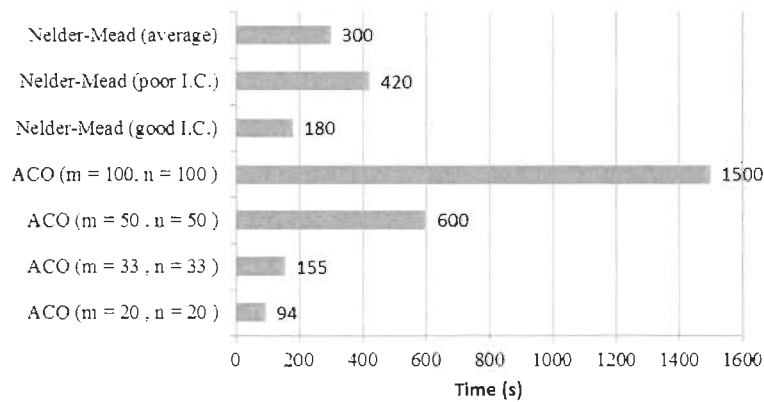


Fig. 9- Computation time of NM constrained algorithm and ACO.

As a second simulation scenario, three different ACO-NM algorithms applied sequentially is compared. In order to get a very low computation time,  $m = 20$  and  $n = 20$ . From the solution found by ants, NM algorithm builds a simplex around this point. Table I compares the results obtained by the new hybrid ACO-NM constrained algorithm to the previous unconstrained hybrid ACO-NM algorithm [8] and to a hybrid ACO-NM algorithm constrained with fixed value. The set of parameters found by unconstrained ACO-NM has a negative value for  $K_{Sat}$ . Therefore, for the system under study; this solution with negative value cannot be accepted to ensure proper anti-windup operation. This demonstrates the necessity to constrain the NM algorithm. Both ACO-NM constrained algorithms provide almost the same  $I_E$  value with comparable parameters value.

Fig. 10 shows computation time required for the three different ACO-NM algorithms to converge to a good solution. It is important to mention that ACO-NM unconstrained algorithm requires  $m = 33$  and  $n = 33$  iterations to obtain positive sets of parameters comparatively to  $m = 20$  and  $n = 20$  for both ACO-NM constrained algorithms. A substantial time reduction is achieved with the new hybrid ACO-NM constrained algorithm: an improvement of 29% on computation time is obtained compare to the previous hybrid algorithm in [8]; and it is 14 % faster than ACO-NM constrained with fixed value. The proposed NM constrained algorithm is effective and responds to the need for a fast and reliable constrained method.

Fig. 11 presents the system response obtained by the new hybrid ACO-NM constrained algorithm. The response is fast without overshoot which satisfies the requirements for the grinding operations.

Table I. Resulting parameters value for three ACO-NM algorithms.

Parameters	ACO-NM constrained (proposed)	ACO-NM constrained (fixed value)	ACO-NM unconstrained
$K_{cP}$	0.491	0.499	0.489
$T_{dP}$	0.0159	0.0161	0.0169
$T_{iP}$	12.2487	11.736	12.363
$\dot{\gamma}$	28.617	27.909	21.0434
$K_{vel}$	0.0354	0.0359	0.0385
$K_{SatI}$	0.0339	0.944	-29.861
$K_{SatP}$	0.584	0.593	0.679
$I_E$	10.387	10.387	10.384

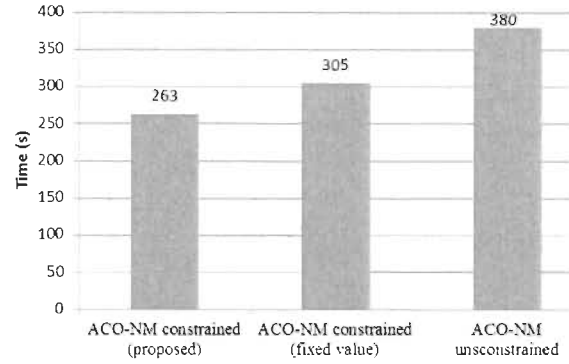


Fig. 10- Computation time for three ACO-NM algorithms.

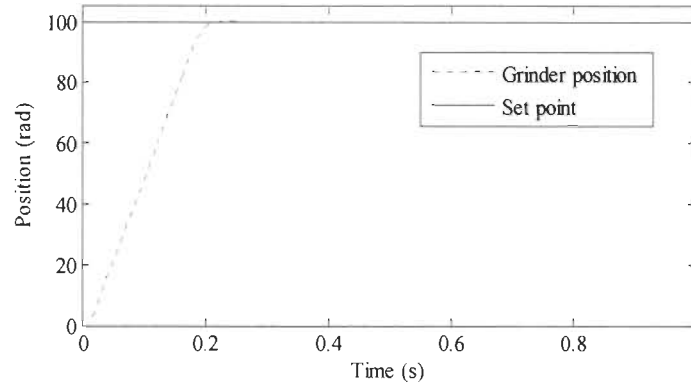


Fig. 11- Step response for the PID controller with anti-windup for the grinder position.

## 4.2 Weighted PIDf controller

A step load disturbance corresponding to 20% of the nominal torque (0.75 Nm) and representing a realistic disturbance value is applied to the weighted controller structures. Through to performance criteria, coefficient  $K_t$  is used (5) to penalize the output disturbance error in order to improve disturbance rejection. The selection of  $K_t$  values is based on the interpretation of the corresponding term in (5) by a spring model and posing a maximum permitted displacement of the grinder table for a given disturbance. From (5), if we write out (for the interpretation normalization to 1 is used):

$$1 = K_t \frac{|e_{max}|}{\Delta T_{pert}} \quad (8)$$

and assume that for a disturbance of 20% the motor nominal torque, a 1 mm to 0.1 mm of grinder table displacement satisfies (8), the required  $K_t$  value is contained within 0.075 to 0.75.

### 4.2.1 Weighted PIDf controller with current based anti-windup

As a third simulation scenario, the control structure shown on Fig. 6 is used. In the performance index (5),  $\sigma$ ,  $K_t$  and  $\beta$  are set to 2, 0.075 and 0 respectively. For ACO tuning, the number of possible combinations is related to the number of parameters to be adjusted and to the number of quantization levels. Then, since there are two more parameters to be tuned, the number of ants and iterations had to be augmented ( $m = 200$  and  $n = 75$ ) comparatively to the first controller structure. These values are still small relative to the optimization performed only by ACO where the values are  $m = 200$  and  $n = 200$ . This is important so as to have a faster optimization method. The higher values of  $m$  and  $n$  allow initiating effectively local searches from some intermediate sets of parameters found by ACO.

Table II compares the results obtained by the new hybrid ACO-NM constrained algorithm to the previous unconstrained hybrid ACO-NM algorithm [8] and to a hybrid ACO-NM algorithm constrained with fixed value. The NM methods are executed using the best set of parameters of the 25<sup>th</sup> iteration of ACO tuning. The necessity of having a constrained method is demonstrated by the final solution set of ACO-NM method: two negative values have to be rejected to avoid right half

plane zeroes and to ensure activation of the anti-windup loop. The parameter values and  $I_E$  value obtained by the two different ACO-NM constrained are very similar. However, the constraint procedure impacts significantly on convergence time. Consequently, as the first controller structure, the convergence time is faster (more than 2.5 faster) with the proposed constraint method. Fig. 12 and Fig. 13 illustrate an intermediate result of the proposed hybrid method with its final result compared to the solution found by the full ACO method alone [7]; the convergence times are also presented. The intermediate result is obtained almost 4 times faster than ACO applied alone for a similar system response. It means that this set of parameters can be used while the system response continues to be enhanced by the proposed method. The final system response obtained by ACO-NM constrained algorithm presents less overshoot than others, meeting the requirements for grinding operations as mentioned before. Also, the response is obtained 25 % faster than ACO alone for a better reference input and disturbance responses.

Table II. Parameters value of three different ACO-NM algorithms.

Parameters	ACO-NM constrained (step 6)	ACO-NM constrained (fixed value)	ACO-NM unconstrained
$K_{cP}$	3.8414	3.8593	4.7066
$T_{dP}$	0.0007	0.0007	0.0001
$T_{iP}$	0.1158	0.1155	0.1232
$\gamma$	22.8421	22.0504	57.4052
$K_{vel}$	0.0178	0.0162	0.0153
$K_{SatP}$	0.0916	0.1858	<b>-33.5372</b>
$K_{SatI}$	89.4210	122.2216	252.8843
$W_p$	0.1980	0.1895	0.0527
$W_v$	0.5310	0.4804	<b>-0.3259</b>
$I_E$	0.1166	0.1163	0.1149
<b>Time (s)</b>	<b>84</b>	<b>222</b>	<b>208</b>

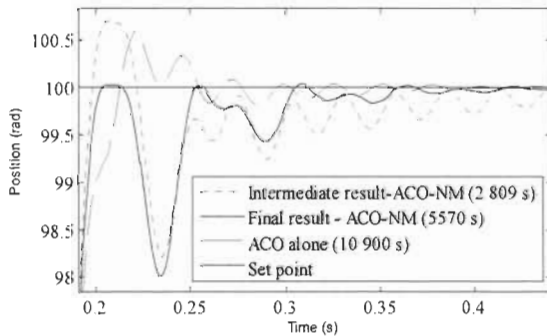


Fig. 12- Enlargement of reference input response.

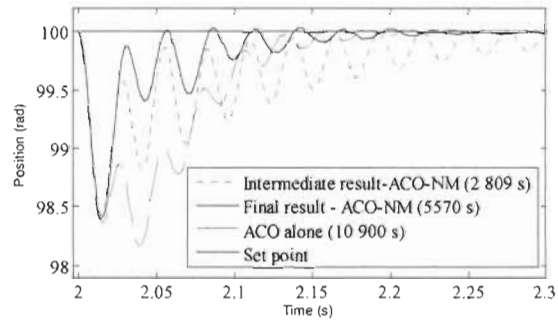


Fig. 13- Enlargement of disturbance response.

#### 4.2.2 Weighted PIDf controller with motor velocity based anti-windup

For the fourth simulation scenario, the control structure described in Fig. 5 is used. It has been observed that only including anti-windup feedback based on velocity limitation is insufficient and that it is necessary to explicitly penalize velocity limit violations (last term in (5)) to obtain the desired effect. Moreover, in order for the response in saturation mode during large input step variations not to dominate in the performance criterion over the small signal operation, large (100 rad at 0 s) and small (5 rad at 2 s) input step variations are applied to the system for optimization. A step disturbance corresponding to 20% the motor's nominal torque is applied at 4 s. In the performance index (5),  $\sigma$ ,  $K$ , and  $\beta$  are set to 100, 0.075 and 100 respectively. For this scenario, no negative values on the set of parameters were obtained. Henceforth, whether constrained or not, the hybrid methods provide the same results shown on Fig. 14. The ACO-NM constrained results are compared with the ACO applied alone. An evident improvement on time computing is achieved: the response obtained with the hybrid

method is 1.67 faster than ACO alone for reaching a better system response. Besides, the intermediate result of ACO-NM is better than when NM is applied from the best point of all ACO iteration. It can be explained by the dependence of the NM result versus its starting search point. Then, the best set of parameter of ACO might not be the better starting point for NM algorithm. It supports the relevance of conducting local NM searches during the search by ACO for reaching the best possible set of parameters.

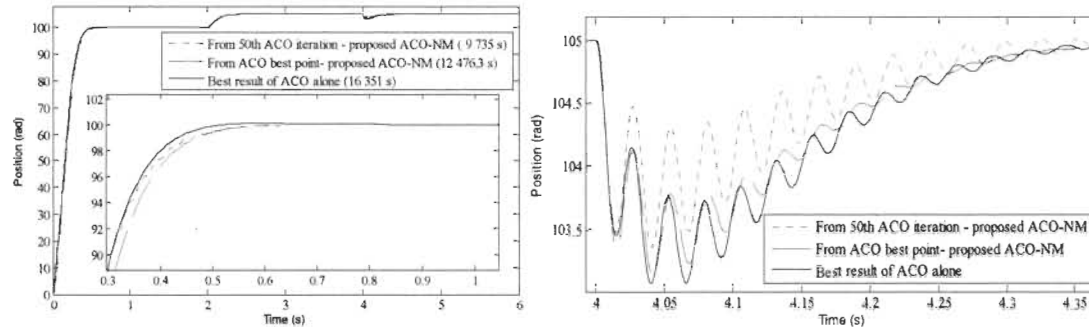


Fig. 14- Step response and enlargement of reference input response and disturbance response for weighted PID with motor velocity based anti-windup.

#### 4.2.4 Weighted PIDf controller structure with anti-windup for motor velocity limitation and current reference limitation.

As a fifth simulation scenario, the controller structure shown on Fig. 7 is used. The scenario setting is exactly the same than the previous scenario (section 4.2.2). However, this controller structure contains one more parameter to tune for a total of 10 parameters. The constrained method is effective because for some intermediate results, the unconstrained method converged to negative values. This controller structure provides a better system response than the previous controller structure (Fig. 15). The input step variation response is faster with the current controller structure. As well, the disturbance is better coped. Fig. 16 presents the effectiveness of the performance index to limit the motor velocity; motor velocity limit is set to 386 rad/s.

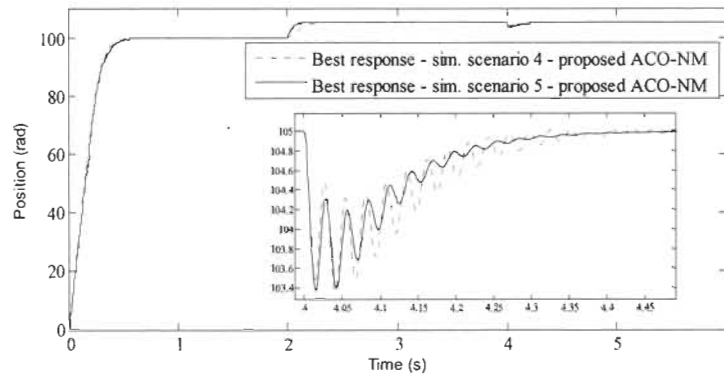


Fig. 15- Step response and enlargement of disturbance response for weighted PIDf controller structure with anti-windup for motor velocity limitation and current reference limitation.

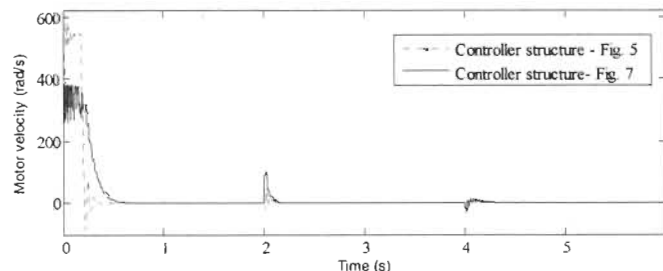


Fig. 16- Motor velocity without and with saturation compensation.

## 5. Conclusion

A new constrained hybrid algorithm based on ACO and NM method is proposed for controller tuning of a motion systems with flexible coupling and cascaded loops control structure. Simulation results demonstrate the efficiency and the improvement of the proposed method for tuning PID controllers with anti-windup and attest that it is necessary to constrain NM algorithm to ensure proper system operation. The additional step inserted in the NM algorithm in order to constraint it, allows faster convergence of the hybrid algorithm compared to the algorithm constrained to a fixed value. Also, local search by NM constrained can be launched while ACO algorithm is still exploring. A first satisfactory parameter set is acquired faster where the latter is improved by future iterations. In addition to a fast computation time, this new constrained hybrid method can be combined to a flexible performance criterion that has permitted to impose hard constraints to the system and to apply the algorithm to four different controller structures. It was demonstrated that this method is easily adaptable to other applications. In fact, tuning may be controller-structure independent. For future work, this algorithm will be applied to adjust controller structure based on backstepping.

## References

- [1] F. Padula, and A. Visioli, "Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes," *IET Control Theory & Applications*, Vol. 6, no. 6, pp. 776-786, 2012.
- [2] C. Bohn, and D. P. Atherton, "An Analysis Package Comparing PID Anti-Windup Strategies," *IEEE Control Systems*, vol.15, no.2, pp.34-40, April 1995.
- [3] N.J. Killingsworth, and M. Krstic, "PID tuning using extremum seeking," *IEEE Control Systems Magazine*, pp. 70-79, February 2006.
- [4] M. F. N. Shah, M. A. Zainal, A. Faruq, and S.S. Abdullah, "Metamodeling approach for PID controller optimization in an evaporator process," *IEEE International Conference on Modeling, Simulation and Applied Optimization*, April 2011.
- [5] B. Nagaraj, and N. Muruganath, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," *IEEE International Conference on Communication Control and Computing Technologies*, Oct. 2010.
- [6] M.-J. Blondin, and P. Sicard, "ACO based controller and anti-windup tuning for motion systems with flexible transmission," 2013 *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'13)*, Regina, Saskatchewan, Canada, 5-8 May 2013.
- [7] M.-J. Blondin, and P. Sicard, "PID controllers and anti-windup systems tuning using Ant Colony Optimization," *EPE '13-ECCE Europe, the 15th European Conference on Power Electronics and Applications*, Lille, France, 3-5 September 2013.
- [8] M.-J. Blondin, and P. Sicard, "Combined ACO Algorithm – Nelder-Mead Simplex Search For Controller and Anti-Windup Tuning for Motion System With Flexible Transmission," *39th IEEE Industrial Electronics Conference (IECON 2013)*, Vienna (Austria), 10-13 November 2013.
- [9] M. Dorigo, and T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts, 2004.
- [10] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimension," *SIAM J. Optimization*, Society for Industrial and Applied Mathematics, 9(1), pp 112-147, 1998.
- [11] M.A. Luersen, R. Le Riche, and F. Guyon, "A constrained, globalized, and bounded Nelder-Mead method for engineering optimization," *Structural and Multidisciplinary Optimization*, vol. 27, no. 1-2, pp. 43-54, 2004.
- [12] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective Ant Colony Optimization," *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, Article ID 536326, 2012.