

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Supérieure Polytechnique
Département de Génie Electrique
Laboratoire de Commande des Processus



Thèse de Doctorat en Sciences en Génie Électrique Option : Automatique

Présenté par :

KHODJA Mohammed Abdallah

Magister en Informatique Industrielle et Robotique



Contribution et la Commande et la Supervision des Systèmes robotisés

Présente le 12 / 03 /2018 devant le jury compose de :

Mr. HEMICI Boualem	Professeur	ENP	Président
Mr. TADJINE Mohamed	Professeur	ENP	Directeur
Mr. BOUCHERIT S.Mohamed	Professeur	ENP	Co-directeur
Mr. BENMANSOUR Khelifa	Professeur	ESDAT/U. Medéa	Examineur
Mr. BOUCHOUCHA Mouloud	Professeur	EMP	Examineur
Mr. BOUCHIDA Wahid	Professeur	U. Medéa	Examineur
Mr. ILLOUL Rachid	Professeur	ENP	Examineur

ANNEE: 2018

Laboratoire de Commande des Processus (LCP), Ecole Nationale Polytechnique (ENP)
10, Avenue Pasteur, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie
www.enp.edu.dz

ملخص:

في هذي الأطروحة تهتم بإحدى الخوارزميات الحديثة للبحث عن القيم المثالية للأنظمة التحكمية. في بادئ البحث اهتمنا بإنشاء وتعريف نموذج خاصة بالحالة المدروسة وهو مروحية ذات اربع محركات صغيرة الحجم. تم تعريف النموذج نظريا وتطبيقيا ثم تطبيق عليه نظام تحكم كلاسيكي بقيمه المثالية في الحالة المدروسة باستعمال خوارزمية عناصر السرب. هذا العمل منجز من قبل فقط في حالة النظرية وبرنامج المحاكات فأضفنا في هذه الأطروحة الاثبات والدراسة العملية لهذا الموضوع. في المرحلة ثانية نستعمل نظرية المنطق المبهم من أجل التحكم في الذراع آلي، نهتم بدراسة التحكم بواسطة المنطق المبهم. فاعتمادا على البحوث السابقة لهذي النظرية بعض الباحثين يختار استعمال خوارزميات المثالية في البحث على العوامل المثالية لهذا نوع من التحكم مع الحفاظ على شكله المعروف به الذي هو متناظر وبعض الاخرين يزرع العوامل ويعمل في تغيير الشكل المعروف عليه فيصبح مجموعاته الضبابية غير متناظرة والغير المعروفة عادة. تم تطبيق هذي المقارنة ببرنامج المحاكات ماطلاب وتم استنتاج ان طريقة البحث عن المثالية المطبقة مع نظرية الضباب في حالة تغيير شكل المجموعات الضبابية افضل من البحث عن العوامل مع المحافظة على الشكل المعروف والنظامي المتناظر.

كلمات مفتاحية: خورزمية البحث عن القيم المثالية، مروحية ذات اربع محركات صغيرة الحجم، ذراع آلي، نظام تحكم خطي، نظام تحكم ضبابي.

Résumé

Dans cette thèse on a concentré sur la méthode d'optimisation par essaim de particules appelée pour l'optimisation des gains de commandes synthétisées. Puisque la méthode conventionnelle de Ziegler-Nichols (Z-N) pour le réglage des paramètres du régulateur PID est une méthode classique qui permet une identification facile des paramètres dans un court laps de temps, mais souvent la sortie du système sous réglage présente une réponse avec un dépassement, une erreur d'état stationnaire élevée et d'un grand temps de montée. Afin d'améliorer les performances de réglage, dans cette étude l'approche methaheuristique moderne de l'optimisation par essaim de particules a été utilisée avec validation expérimental afin d'optimiser les paramètres du PID pour la stabilisation de l'attitude d'un Quadrotor. Le deuxième cas traite de l'optimisation d'un contrôleur Proportionnel Dérivé (PD) Flou en utilisant aussi la technique d'optimisation par essais de particules (PSO). Plus précisément, nous avons effectué une étude comparative pour montrer les performances du contrôleur PD-flou en optimisant séparément ses gains de normalisation et ses paramètres structurels des fonctions d'appartenance. Il est à noter que le contrôleur flou optimisé se comporte mieux en présence de bruit.

Mots clés : la méthode d'optimisation par essaim de particules, bras manipulateur, PID, contrôleur flou.

Abstract

In this theses focuses on the modern heuristics approach of Particle Swarm Optimization (PSO). At the first case we have choose to apply this method for optimization of PID controller for attitude stabilization of a Quadrotor with experiments validation. However, manual tuning of these controllers is tedious and often leads to poor performance. The conventional Ziegler-Nichols (Z-N) method of PID tuning was done experimentally enables easy identification stable PID parameters in a short time, but is accompanied by overshoot, high steady-state error, and large rise time. Therefore, in this study, the modern heuristics approach of Particle Swarm Optimization (PSO) was employed to enhance the capabilities of the conventional Z-N technique. PSO with the constriction coefficient method experimentally demonstrated the ability to efficiently and effectively identify optimal PID controller parameters for attitude stabilization of a quadrotor. In second case, we conduct a comparative study to access the performance of the PD fuzzy controller by optimizing its gain and its structural parameters separately, the optimization will done offline; thus leading two distinct fuzzy controllers. The two controllers are applied to control a 3DOF PUMA560 robot manipulator. The optimized parameter controller performs better in the presence of noise in some joints.

Key words: Particle Swarm Optimization, Quadcopter, Manipulator robot, PID, Fuzzy logic controller

Dedication

إهداء

الْحَمْدُ لِلَّهِ الَّذِي أَرْسَلَ رَسُولَهُ بِالْهُدَىٰ وَدِينِ الْحَقِّ؛ لِيُظْهِرَهُ عَلَى الدِّينِ كُلِّهِ وَكَفَىٰ بِاللَّهِ شَهِيدًا. وَأَشْهَدُ أَنْ لَا إِلَهَ إِلَّا اللَّهُ وَحْدَهُ لَا شَرِيكَ لَهُ؛ إِقْرَارًا بِهِ وَتَوْحِيدًا. وَأَشْهَدُ أَنَّ مُحَمَّدًا عَبْدُهُ وَرَسُولُهُ. صَلَّى اللَّهُ عَلَيْهِ وَعَلَىٰ آلِهِ وَسَلَّمَ تَسْلِيمًا مَزِيدًا.

أما بعد:

أهدي هذا العمل المتواضع إلى أبي وأمي إلى من سهرنا وإجتهدنا في تربيته وتعليمي بفضل الله عز وجل.

حيي لربي والنبى محمدا	حيي إليهم لا يضاهى ماعدا
يكن الطريق إلى الجنان ممهدا	أبوايا لوجادوا علينا بالرضا
كي تجعلوني بين قومي سيدا	أبوايا كنتم على الدوام تناضلا
وكانكم أنجبتوني واحدا	فأخذت منكم ما يجب وزيادة
لم تبخلوا لم تجعلوه محمدا	وكنت أطلب مالكم تعطوني
وإذا شفيت يزول عنكم ما بدا	وبدا عليكم إذا مرضت كآبة
كنتم لأجلي تفرحان وتسعدا	وإن تسمعا أني أحقق مطلباً
حيي إليهم في الفؤاد ممدا	اليوم أخبر والدي بأنه
والقمر يشهد والسحاب مؤيدا	الشمس شهدت والسماء بعطفهم
هل مثل ربي في الشهادة شاهدا	والله يشهد لا أبالغ مطلقاً
واجعل لهم من حوض طه موردا	يا رب تحفظ والدي كالأههما
باب العبور إلى النعيم الخالدا	واكتب لهم حسن الختام لأنه

مجدي العوضي .شبكة الفصح

وإلى أختاي مريم ، سمية و أخي زكرياء.

وإلى زوجتي عريبي زويبة أم أميمة التي صبرت معي في كل المحن والصعاب وكانت بفضل الله عز وجل ثم بصبرها على غيابي وسفري بسبب الدراسة عوناً لي من مرحلة ماجستير إلى الدكتوراه.

يا مَنْ عَزَمْتَ عَلَى الزَّوْجِ وَلَسْتَ تَدْرِي مَا هِيَ وَتَعِيشَ حُرًّا مُسْتَقْرًّا فِي حَيَاةٍ رَاضِيَةٍ
فَاخْتَارَهَا لِدِينِهَا وَذَاتَ أَخْلَاقٍ عَالِيَةٍ وَرُوحَهَا جَمِيلَةً وَذَاتَ نَفْسٍ سَامِيَةٍ
قَصِيرَةً لِسَانِهَا لِلسَّرِّ لَيْسَتْ فَاشِيَةً قَوِيَّةً عَزْمُهَا فِي الدِّينِ لَيْسَتْ وَاهِيَةً
صَبُورَةً لِخَالَهَا عَنِ الْحَرَامِ لِأَهْيَةِ غَيُورَةٍ لِرُوجِهَا وَلِلْأُمُورِ دَارِيَةً
قَلِيلَةً خُرُوجِهَا وَلَا تَكُونُ شَاكِيَةً قَصِيرَةً طَرْفِهَا عَنِ التَّطَلُّعِ عَامِيَةٍ
رَقِيقَةً كَلَامِهَا لِلْمَشْكَلَاتِ نَاسِيَةً طَيِّبَةً سَمْتِهَا وَفِي الْحَدِيثِ ذَاهِيَةً
تَقِيَّةً لِرَبِّهَا فِي اللَّيْلِ دَائِمًا بِأَكْبَرِهِ مُعِينَةً لِرُوجِهَا حَتَّى يَكُونَ دَاعِيَةً
حَنُونَةً أُمُّهَا لَهَا نَصَاحَ غَالِيَةٍ قَلِيلَةً طَلِبَاتِهَا وَلَا تَكُونَ قَاسِيَةً
حَفِيظَةً لِبَعْدِكَ وَهِيَ لَكَ مُوَالِيَةً بَرِيرَةً لِأَمْرِكَ وَإِنْ حَزَنْتَ فَمُوَاسِيَةً
وَلُودَةً وَدُودَةً فِي الْحُكْمِ لَيْسَتْ قَاضِيَةً كَرِيمَةً عُدَّةً تَكُونُ أُمًَّ وَافِيَةً
فَحُجْرَةً صَغِيرَةً تَكُونُ عَيْشَةً هَانِيَةً وَمَهْنَةً بَسِيطَةً تَكْفِي الْحَيَاةَ الْحَالِيَةَ
وَلَقَمَةً هَنِيئَةً تَكُونُ لِلْجُوعِ كَافِيَةً وَشَرْبَةً مَرِيئَةً تَأْخُذُهَا بِيَدِ صَافِيَةٍ
وَزَوْجَةً مَطِيعَةً لِلْأَمْرِ لَيْسَتْ عَاصِيَةً وَابْنَةً بَرِينَةً لِلْجَنَّةِ صَارَتْ جَارِيَةً
وَنَظْرَةً قَرِيرَةً مِنْ زَوْجَةٍ حَانِيَةٍ وَبَسَمَةً لَطِيفَةً تُعِيدُ لَكَ الْعَافِيَةَ
فَهَذِي خَيْرُ مَتَاعِهَا وَهِيَ لَعْمَرِي فَإِنِّيهِ ذَاكَ لَمَنْ أَرَادَ أَنْ يَحْيِيَ حَيَاةً ثَانِيَةً

أبو حفص محمد بن خطاب. ملتيقى اهل الحديث

كما أهديه إلى فلذتي كبدي ابنتي أميمة وابني سعد واروى وعائشة
وإلى كل العائلة حفظهم الله جميعا وإلى كل أصدقائي.

سبحانك اللهم وبحمدك نشهد أن لا إله إلا أنت نستغفرك ونتوب إليك.

Acknowledgments

شكرو عرفان

أَبْدَأُ بِاسْمِ اللَّهِ مُسْتَعِينًا رَاضٍ بِهِ مَدِيرٌ مُعِينًا
وَالْحَمْدُ لِلَّهِ كَمَا هَدَانَا إِلَى سَبِيلِ الْحَقِّ وَاجْتَبَانَا وَمِنْ
أَحْمَدُهُ سُبْحَانَهُ وَأَشْكُرُهُ مَسَاوِي عَمَلِي أَسْتَغْفِرُهُ
وَأَسْتَعِينُهُ عَلَى نَيْلِ الرِّضَا وَأَسْتَمِدُّ لَطْفَهُ فِي مَا قَضَى
وَيُعَدُّ : إِنِّي بِالْيَقِينِ أَشْهَدُ شَهَادَةَ الْإِخْلَاصِ أَنْ لَا يُعْبَدُ
بِالْحَقِّ مَالُوهُ سِوَى الرَّحْمَنِ وَأَنَّ مَنْ جَلَّ عَنْ عَيْبٍ وَعَنْ نُقْصَانٍ
خَيْرَ خَلْقِهِ مُحَمَّدًا رَسُولَهُ مَنْ جَاءَنَا بِالْبَيِّنَاتِ وَالْهُدَى
إِلَى جَمِيعِ الْخَلْقِ صَلَّى عَلَيْهِ بِالنُّورِ وَالْهُدَى وَدِينِ الْحَقِّ
رُبُّنَا وَمَجْدًا وَالْآلِ وَالصَّحْبِ دَوَامًا سَرْمَدًا

أما بعد:

عَنْ أَبِي هُرَيْرَةَ، عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ: «لَا يَشْكُرُ اللَّهُ مَنْ لَا يَشْكُرُ النَّاسَ» رواه أحمد.
عَنْ عَائِشَةَ، أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ، قَالَ: «مَنْ أَتَى إِلَيْهِ مَعْرُوفٌ، فَلْيُكَاغِبْ بِهِ، وَمَنْ لَمْ يَسْتَطِعْ، فَلْيَذْكُرْهُ، فَمَنْ ذَكَرَهُ، فَقَدْ شَكَرَهُ»
رواه أحمد.

أشكر الله عز وجل وأحمده حمدا كثيرا طيبا مباركا أن وفقني لإتمام هذه المذكرة ثم أشكر من سخره الله عز وجل ليكون سببا في هذا الإنجاز وهما الأستاذ المشرف طجين محمد و الأستاذ بوشريط محمد اللذان كانا في نعمة النصح والتوجيه وفضلهما كتب ونحت في تاريخ مشوار تأطير كفاءات الوطن الجزائري فهما رمزان من رموز الارتقاء في الاخلاق وفي العمل وفي التعليم والبحث ونسال الله عزوجل ان يبارك فيهما وفي اهليهما وأولادهما.

لنا الشرف العظيم للقاء لجنة التحكيم التي تم تعيينها ونشكرهم على توجيهاتهم. نشكر الأستاذ ملوكي عزالدين بجامعة المسيلة الذي كان داعما لنا في المشوار التعليمي منذ اول سنة دراسية في الجامعة والى ايام عملنا معه بالجامعة كما أتقدم بجزيل الشكر للأستاذ بركات سعيد بجامعة المسيلة والأستاذ شماشمة محمد بجامعة قسنطينة الذي قدما لي يد العون, كما نشكر معلمة بختي سهيلة وخلاف سعاد على دعمهما في تصحيح الإنجليزية , كما أقدم الشكر الجزيل لسيد مقيدش نوردين الذي كان ناصحا ومعينا لنا في مشوارنا العلمي والعملية ندعوا له بالتوفيق لما فيه الخير والصلاح , كما أتقدم بجزيل الشكر إلى الأستاذ لارياص شريف الذي كان سببا في ابتعائنا الى بريطانيا وكل من كان في سبب في ابتعائنا اخص منهم الأستاذ حليلات محمد الطاهر مدير سابق لجامعة المسيلة و الدكتور لخضر خرشي نائب مدير الجامعة مكلف بالمسائل المتعلقة بالتنشيط و الترقية العلمية و التقنية و العلاقات الخارجية ومرحوم جميلة مسؤولة موارد بشرية الخاصة بالأساتذة بوزارة التعليم العالي.

شكري الخاص الى الزميل العلمي في الدكتوراه بنصحه وخبرته الاخ نوردين بوعروج.
وفي الأخير أقول لهم جميعا بارك الله فيكم وادعوا الله عز وجل أن يجازيهم خير الجزاء.
سبحانك اللهم وبحمدك نشهد أن لا إله إلا أنت نستغفرك ونتوب إليك.

أَخِي لَنْ نَنَالَ «الْعِلْمَ» إِلَّا بِسِنَّةٍ
ذَكَاءٌ، وَحِرْصٌ، وَاجْتِهَادٌ، وَبُلْغَةٌ
سَأُثْبِتُكَ عَنْ تَقْصِيْلِهَا بَيِّنَاتٍ:
وَصُحْبَةُ أَسْتَاذٍ، وَطَوْلُ زَمَانٍ

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

General Introduction.....	11
Chapter 1: Dynamic modeling of a Quadcopter	15
1.1 Introduction.....	15
1.2 The Quadcopter model	15
1.3 Dynamic model used for attitude stabilization.....	21
1.4 Dynamic model used for altitude stabilization.....	25
1.5 Conclusion.....	27
Chapter 2: Modeling rigid manipulator robot.....	29
2.1 Introduction.....	29
2.2 PUMA 560 robot Dynamics.....	29
2.3 Using PUMA Robot as 3-DOF Robot.....	32
3.4 Conclusion.....	37
Chapter 3: A Quadcopter hardware description.....	39
3.1 Introduction.....	39
3.2 Flight controller.....	39
3.3 Inertial Measurement Unit.....	40
3.4 Speed Controllers.....	41
3.5 Motors and Propellers.....	41
3.6 Data acquisition device.....	43
2.7 Radio Receiver.....	43
3.8 Lipo battery	44
3.9 UBEC power	44
3.10 Quadcopter frame	45
3.11 Hardware diagram	46
3.12 Conclusion	48

Chapter 4: Tuning PID Controllers for Stabilization of a Quadcopter.....	50
4.1 Introduction.....	50
4.2 Tuning the PID controller using the ZIEGLER-NICHOLS method	50
4.3 PID tuning by PARTICLE SWARM OPTIMIZATION	51
4.4 RESULTS AND DISCUSSION.....	52
4.5 PSO - PID gains optimization	57
4.6 Conclusion.....	62
Chapter 5: Fuzzy logic controller Optimization	64
5.1 Introduction	64
5.2 Fuzzy set theory.....	64
5.3 Basic fuzzy set operations.....	66
5.4 Fuzzy relations.....	68
5.5 Fuzzy logic control.....	69
5.5.1 The fuzzification process	70
5.5.2 The fuzzy rulebase	70
5.5.3 Fuzzy inference.....	71
5.5.4 The defuzzification	74
5.6 Fuzzy control of PUMA560 with 3DOF	77
5.7 Optimized Fuzzy PD controller.....	79
5.8 PARTICLE SWARM OPTIMIZATION.....	83
5.9 SIMULATION RESULTS.....	84
5.10 RESULTS AND DISCUSSION.....	87
5.11 Conclusion.....	89
General Conclusion.....	90
Bibliography.....	92
Appendix	98

LIST OF TABLES

Table 1.1 Quadrotor parameters.....	22
Table 1.2 some experiment results for thrust calculation.....	22
Table 1.3 Experimental thrust calculation.....	24
Table 2.1 Modified Denavit - Hartenberg Parameters.....	30
Table 2.2 Link Mass.....	30
Table 2.3 Centers of Gravity.....	31
Table 2.4 Diagonal Terms of the Inertia Dynamics and Effective Motor Inertia.....	31
Table 2.5 Motor and Drive Parameter.....	31
Table 2.6 Inertiel Constants.....	35
Table 2.7 Gravitational Constants.....	35
Table 4.1 Ziegler-Nichols Tuning rule based on critical gain K_c and critical period P_c	51
Table 4.2 Summary of PID parameters found by Z-N procedure.....	55
Table 4.3 Limited PSO search interval based on Z-N derived gains.....	58
Table 4.4 Summary of PID gains obtained by Z-N tuning and PSO.....	59
Table 4.5 Comparison of the Z-N and PSO gains for pitch angle stabilization.....	59
Table 4.6 Experimental comparative results for three optimized PID gains.....	61
Table 5.1 Tabular structure of a linguistic fuzzy rulebase.....	72
Table 5.2 Rule Base table.....	78
Table 5.3 Rule Base table; Negative Large (NL), negative Small (NS), zero (Ze), positive small (PS) and positive Large (PL).....	79
Table 5.4 Parameters of fuzzy controller to optimize.....	81
Table 5.5 Parameters for standard and symmetric fuzzy sets for gains optimization.....	82
Table 5.6 Optimized parameters for structure optimization.....	82
Table 5.7 ISE for manual (MT) and PSO gains optimization (GO), without and with noise.....	86
Table 5.8 The ISE without and with disturbance for Gains and structural PSO optimization.....	87
Table 5.9 Some differences simulation results between Gains and structural PSO optimization fuzzy controller.....	88

LIST OF FIGURES

Figure 1.1 A basic configuration of a quadcopter.....	22
Figure 1.2 The realized quadcopter.	23
Figure 1.3 PID control strategy for attitude stabilization.....	24
Figure 1.4 Hardware configuration for thrust calculation.....	25
Figure 1.5 The realized quadcopter based on Teensy 3.2.....	26
Figure 1.6 Teensy development board 3.2.....	26
Figure 1.7 PX4FLOW Smart Camera.....	27
Figure 2.1 The Puma 560 Robot.....	30
Figure 3.1 Teensy Development board.....	39
Figure 3.2 Arduino due.....	40
Figure 3.3 GY-86 IMU.....	40
Figure 3.4 Electronic Speed Controller.....	41
Figure 3.5 brushless DC motors (DYS BE1806-2300).....	42
Figure 3.6 GemFan Propellers.....	42
Figure 3.7 HC-05 Bluetooth device.....	43
Figure 3.8 Radio Receiver.....	43
Figure 3.9 Turnigy 2200mAh battery.....	44
Figure 3.10 Universal Battery Elimination Circuit.....	45
Figure 3.11 a HobbyKing™ Totem Q250 frame.....	45
Figure 3.12 hardware diagram of interconnection between electronics parts.....	46
Figure 3.13 The realized quadcopter insight view.....	47
Figure 3.14 Top view of the realized quadcopter.....	47
Figure 3.1 The Puma 560 Robot.....	52
Figure 4.1 The general strategy of optimization.....	52
Figure 4.2 Critical gain (K_c) variation in real time.....	53
Figure 4.3 Pitch error angle when K_c was varied in real time.....	53
Figure 4.4 Magnified black rectangle to calculate Critical Period P_c .in real time.....	54
Figure 4.5 Critical gain (K_c) change in real time.....	54
Figure 4.6 Altitude error while K_c was changed in real time.....	54
Figure 4.7 Critical Period P_c .in real time.....	55
Figure 4.8 Altitude control while ZN using.....	55
Figure 4.9 Altitude error produced by the calculated PID.....	56
Figure 4.10 The PID control response.....	56

Figure 4.11 The generated controlled thrust.....	56
Figure 4.12 altitude error for one second.....	56
Figure 4.13 PID output for one second.....	57
Figure 4.14 The generated thrust for each motor.....	57
Figure 4.15 Roll and Pitch errors tracking.....	57
Figure 4.16 Roll and Pitch controls.....	57
Figure 4.17 control of each motor.....	58
Figure 4.18 Change in particles' position during the search for the optimal PID controller gains.....	58
Figure 4.19 Pitch angles step responses for the three PID gains in simulation.....	59
Figure 4.20 Pitch error angle in steady-state for Z-N PID.....	60
Figure 4.21 Pitch error angle for Z-N PID.....	60
Figure 4.22 Pitch error angle in steady-state for PID PSO1.....	60
Figure 4.23 Pitch error angle for PID PSO1.....	60
Figure 4.24 Pitch error angle in steady-state for PSO2.....	61
Figure 4.25 Pitch error angle for PSO2.....	61
Figure 4.26 Errors angles of the roll, pitch and yaw implemented together in experiment.....	62
Figure 5.1 Crisp set (medium temperature).....	65
Figure 5.2 Fuzzy set 'medium temperature'.....	65
Figure 5.3 Overlapping sets 'low' and 'medium temperature'[31].....	67
Figure 5.4 'Union' and 'intersection' functions.[31].....	68
Figure 5.5 The complement of fuzzy set M.[31].....	68
Figure 2.6 Fuzzy Logic Control.....	71
Figure 5.7 Seven set fuzzy input windows for error (e) and rate of change of error (ce).....	72
Figure 5.8 Seven set fuzzy output window for control signal (u).....	73
Figure 5.9 Clipped fuzzy output window due to fuzzy inference.....	75
Figure 5.10 Fuzzy output window for Example 5.2.....	77
Figure 5.11 Structure type-1 FLC of PUMA560 3DOF.....	77
Figure 5.12 Fuzzy set for each articulation.....	78
Figure 5.13 General PSO scheme for Fuzzy PD with manipulator robot.....	79
Figure 5.14 Gain optimization fuzzy controller.....	80
Figure 5.15 Structural parameter optimization Fuzzy controller.....	80
Figure 5.16 Triangle membership function.....	81
Figure 5.17 Fuzzy sets for error in GO case.....	83
Figure 5.18 Fuzzy sets for derivative error in GO case.....	83

Figure 5.19 Fuzzy sets for control in GO case.....	83
Figure 5.20 Optimized Fuzzy sets for error in PO case.....	83
Figure 5.21 Optimized Fuzzy sets for derivative error in PO.....	83
Figure 5.22 Optimized Fuzzy sets for control in PO case.....	83
Figure 5.23 Third joint response in manual gain tuning with disturbance and its tracking error.	84
Figure 5.24 Third joint response in PSO gain tuning with disturbance and its tracking error.	85
Figure 5.25 Third joint control with and without disturbance manual gain tuning.	86
Figure 5.26 Third joint control with and without disturbance PSO gain tuning.	86

General Introduction

0.1 Introduction

In these last years, a growing interest has been shown in robotics. In fact, several industries (automotive, medical, manufacturing, space, . . .), require robots to replace human in dangerous, boring or onerous situations. For example we cannot replace welding robot by human in repetitive jobs that are boring, stressful, or labor intensive for humans. Also cleaning the main circulating pump in the nuclear power plant which this task is dangerous for human. Some important task need aerial robot or unmanned aerial vehicle, as wildfire mapping, agriculture monitoring, oil and gas exploration, ...etc. for those importance we are interest in this thesis by two type of robot the first Quadcopter which is kind of unmanned aerial vehicle. The second PUMA560 manipulator robot. Basic knowledge for robotics design are; dynamics modeling, sensors, actuators, microcontroller, microprocessor and programing language.

The advance in technology the use of the control theory more important, for that great research progress in control theory during last decades. Lot of approach and strategies proposed by researchers in most simple controller (PID) to most complex controller theory. The PID controller is simplest and oldest control theory and the most widely used controller in process control until today. A PID is widely used in control of linear or nonlinear systems. The PID it is an important' part in every engineering control tools.

The most popular of Unmanned Arial Vehicles is the quadrotor. Because of its ability to perform agile manoeuvres, take-off , land vertically, and hover. However, it is difficult to model [1][2] complex and suitable quadrotor control systems. Many control methods, such as fractional sliding mode [3], backstepping [4], nonlinear proportional-integral-derivative (PID) [5], sliding model controllers [6], and fuzzy PID [7], have been studied. The simplest control method among these is the PID controller [8] because of its low program complexity, low processing speed, and small program size, occupying little memory. Therefore, the PID can be implemented in a low-cost microcontroller.

In PID control design, the parameters of the controllers are tuned manually or by using the tuning rules found in the linear control-based literature, such as Ziegler-Nichols (Z-N) tuning [9][10]. Unfortunately, the manual tuning method has limitations of being time

consuming and very tedious. While the conventional Z-N method has been very useful and helpful in our research in terms of practicality, it is criticized by many other researchers.

After developing a quadrotor, we were met with the problem of how to choose PID gains in order for our quadrotor to achieve stability. The fastest method with the least required effort to experimentally select the appropriate PID gains was the Z-N tuning. Within a few minutes of implementation we could obtain the PID gain values required to stabilize the quadrotor. With respect to time and effort.

Many research show design and control strategies, in [11,12] full design and implementation of PID control system for a Quadcopter. Another design with experimental studies for switching model predictive control of a Quadcopter in [2]. In [8] present a control approach to obtain the better stabilization in attitude and altitude of quad-rotor using PID under different disturbance conditions. In [13] an adaptive PID controller is proposed for fault-tolerant control of a quadrotor helicopter system in the presence of actuator faults. In [14] presents a survey on publicly available open-source projects (OSPs) on quadrotor unmanned aerial vehicles (UAVs) and is based also on PID controllers. In [15] presents modeling, simulation and implementation of quadcopter, this work also based on PID controllers. From we notice no one present use of ZN for altitude and attitude PID parameters determination.

However, regarding robustness, overshoot, steady-state error, and rise time, the Z-N method falls short of the necessary standards. To overcome this, we instead employed a modern heuristics approach [16][17][18][19], specifically particle swarm optimization (PSO) with the constriction coefficient method unlike in[20] they use the basic PSO. PSO used to determine the optimal PID gains for attitude stabilization of the quadrotor experimentally counterpart in [20] they applied for position control and is very clear the optimization of the attitude stabilisation has more priority than position control optimization for quadrotor. This stands in contrast with many other research efforts whose results come only from simulations [21][22][23].

The second robotic system treated in this thesis, which is robot manipulator [24]. The control based on fuzzy logic have become [25][26] an appealing alternative solution for the control of robots manipulators as it does not require the knowledge of the dynamic model of the latter. In addition, as the mathematical model of robot arms depends on the mechanical parameters - subject to lifetime modifications (friction factors affected by the abuse of joints [24]), these considerations also give advantage to fuzzy control methods over other nonlinear control methods as a result of their robustness towards perturbations affecting the system and uncertainties of dynamic parameters. However, such fuzzy control [26] methods require appropriate tuning in order to work properly, particle swarm optimization [27] [28] [29] was

selected to optimize fuzzy PD controller [30]. In this context, two strategies can be adopted: i) either to tune the gains of the controller or to tune the structural parameters of the controllers.

In the last section, we propose a proportional derivative (PD) fuzzy controller [30] design for a three degree of free doom (3DOF) PUMA560 robot manipulator. We conduct a comparative study to assess the performance of the PD fuzzy controller by optimizing [28] [29] its gain [31][32] and its structural parameters[33][34] separately, the optimization will be done offline, it means after PSO finds the optimized gains and parameters will put them into the fuzzy PD controller; thus leading to two distinct optimized fuzzy controllers. As far as the authors are aware, such comparative study has not been carried out. The two controllers are applied to control a 3DOF PUMA560 robot manipulator. The Particle Swarm Optimization (PSO) was selected as the optimization method. Additionally, a comparative study in terms of integral of squared error (ISE) between the two proposed controllers is carried out.

0.2 Contribution and thesis organization

Our main contributions focused in this thesis are:

- Realization of a custom quadcopter autopilot based on Arduino Due and Teensy 3.2.
- Experimental PID identification using ZN for altitude and attitude stabilization of a quadcopter.
- Optimization of the obtained PID parameters using Particle Swarm Optimization with experiment validation.
- Parametric and structural optimization of a fuzzy controller using Particle Swarm Optimization for PUMA560 manipulator robot control.

The thesis is divided into four chapters:

The first chapter presents dynamic modeling of a quadcopter. The second chapter has PUMA560 manipulator robot modeling. The third chapter has a quadcopter and its main hardware used in our prototyping. The fourth chapter presents application of Ziegler-Nichols PID parameters tuning for stabilization of a quadcopter with optimization of the obtained parameters using Particle Swarm Optimization. The fifth chapter presents a comparison study between parametric and structural optimization of a fuzzy controller with PUMA560 manipulator robot.

In the last, general conclusion summarizes the main obtained results and future research can be obtained using this thesis.

Chapter 1

Dynamic modeling of a Quadcopter

1.1 INTRODUCTION

The most famous of Unmanned Aerial Vehicles is the quadcopter. Because of its ability to perform agile maneuvers, take-off and land vertically, and hover, it is an ideal choice for search and surveillance, rescue, monitoring, military, and agriculture applications in both outdoor and indoor environments.

In this chapter, provide identification, design, and dynamic model of a Quadcopter. These tools are very important because it describes how the Quadcopter fly according to its orientations and positions in space. These tools help to define and predict the orientations and positions reached by the Quadcopter by using just the four motor speeds. Two custom quadcopter prototyping will present the first one its flight controller use Arduino Due and the second use Teensy 3.2 development board.

1.2 THE QUADCOPTER MODEL

First, confirm that A quadrotor has four propellers mounted on the end of two perpendicular arms and is actuated by four brushless DC motors. A basic configuration of a quadrotor [35][44] is shown in Figure 1.1.

Each rotor pair of the same arm rotates in the same direction; one pair rotates clockwise, while the other rotates counter clockwise. The quadrotor moves by adjusting the angular velocity of each rotor. We used the literature to get dynamical model of a quadrotor same in [1][11][44][45].

To represent dynamic model we use Euler-Lagrange equations of motion. The generalized coordinates of Quadcopter are:

$$q = (\xi, \eta)^T \in R^6 \tag{1.1}$$

$$\xi = (x, y, z)^T \in R^3, \eta = (\psi, \theta, \phi)^T \in R^3 \tag{1.2}$$

Where ξ, η represent the position and orientation of Quadcopter in the inertial fixed frame respectively.

The translational and rotational kinetic energy of a quadcopter are:

$$T_{trans} = \frac{m}{2} \dot{\xi}^T \dot{\xi} = \frac{m}{2} (x^2 + y^2 + z^2) \quad (1.3)$$

$$T_{rot} = \frac{1}{2} \Omega^T I \Omega = \frac{1}{2} \dot{\eta}^T J \dot{\eta} \quad (1.4)$$

Where

$$J = W_{\eta}^T I W_{\eta} \quad (1.5)$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (1.6)$$

Where transformation matrix from inertial fixed frame to the inertial body frame is:

$$W_{\eta} = \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\phi & c\phi & 0 \\ c\theta c\phi & -s\phi & 0 \end{bmatrix} \quad (1.7)$$

Then

$$\dot{\eta} = W_{\eta}^{-1} \Omega \quad (1.8)$$

The potential energy of quadcopter consists of the gravitational potential energy which as:

$$U = mgz \quad (1.9)$$

The Lagrangian function defined as:

$$L = T_{trans} + T_{rot} - U = \frac{m}{2} (x^2 + y^2 + z^2) + \frac{1}{2} \Omega^T I \Omega - mgz \quad (1.10)$$

Satisfies the Euler Lagrange equation:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = F \quad (1.11)$$

Where F represents the forces and torques applied to quadcopter

After simplification calculations, we can obtain the following standards formulation:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = F \quad (1.12)$$

Where

$M(q) \in R^{6 \times 6}$ is the symmetric positive definite inertia matrix.

$C(q, \dot{q}) \in R^{6 \times 6}$ is the matrix of centrifugal and Coriolis forces.

$G(q) \in R^6$ is the gravity force vector. Moreover, the matrices M and C verify the passivity property necessary $\dot{M} - 2C = P$, Where P denotes an antisymmetric matrix.

The full dynamics model of a quadcopter by using Euler-Lagrange's equations is:

$$\begin{bmatrix} \frac{d}{dt} \left[\frac{\partial L_{trans}}{\partial \dot{\xi}} \right] - \frac{\partial L_{trans}}{\partial \xi} & 0 \\ 0 & \frac{d}{dt} \left[\frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \frac{\partial L_{rot}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} f \\ \tau \end{bmatrix} \quad (1.13)$$

$$f = RF \quad (1.14)$$

Where (2.14) is the translational force (thrust), τ represents the yaw, pitch and roll moments.

$$F = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix} \quad (1.15)$$

$$u = f_1 + f_2 + f_3 + f_4 \quad (1.16)$$

$$R = \begin{pmatrix} c\theta c\psi & s\psi s\theta & -s\theta \\ c\psi s\theta s\phi - s\psi c\phi & s\psi s\theta s\phi + c\psi c\phi & c\theta s\phi \\ c\psi s\theta c\phi + s\psi s\phi & s\psi s\theta c\phi - c\psi s\phi & c\theta c\phi \end{pmatrix} \quad (1.17)$$

Since no cross terms combining $\dot{\xi}, \dot{\eta}$ in (2.13) can be partitioned into dynamics separate.

The first is for translation motions are:

$$\frac{\partial L_{trans}}{\partial \dot{\xi}} = m\dot{\xi} \quad (1.18)$$

$$\frac{d}{dt} \left[\frac{\partial L_{trans}}{\partial \dot{\xi}} \right] = m\ddot{\xi} \quad (1.19)$$

$$\frac{\partial L_{trans}}{\partial \xi} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (1.20)$$

Finally, we obtain

$$f = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} + mg \end{bmatrix} \quad (1.21)$$

For rotational motions, we have

$$\frac{d}{dt} \left[\frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \frac{\partial L_{rot}}{\partial \eta} = \tau \quad (1.22)$$

$$\text{Or } \frac{d}{dt} \left[\dot{\eta}^T J \frac{\partial \dot{\eta}}{\partial \dot{\eta}} \right] - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T J \dot{\eta}) = \tau \quad (1.23)$$

We get

$$J\ddot{\eta} + \dot{J}\dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T J \dot{\eta}) = \tau \quad (1.24)$$

$$J\ddot{\eta} + \left(\dot{J} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T J) \right) \dot{\eta} = \tau \quad (1.25)$$

$$J\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau \quad (1.26)$$

We get transitional and rotational motions represent in this formulation:

$$f = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} + mg \end{bmatrix} \quad (1.27)$$

$$J\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau \quad (1.28)$$

Note that rotational motions we can be written in general form as

$$M\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau \quad (1.29)$$

$$\text{Where } M(\eta) = J(\eta) = W_\eta^T I W_\eta \quad (1.30)$$

$$\frac{d}{dt} \left[\Omega^T I \frac{\partial \Omega}{\partial \dot{\eta}} \right] - \Omega^T I \frac{\partial \Omega}{\partial \eta} = \tau \quad (1.31)$$

$$\frac{\partial \Omega}{\partial \dot{\eta}} = \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\phi & c\phi & 0 \\ c\theta c\phi & -s\phi & 0 \end{bmatrix} \quad (1.32)$$

However,

$$\begin{aligned} \Omega^T I \frac{\partial \Omega}{\partial \dot{\eta}} &= \begin{bmatrix} \dot{\phi} - \psi s\theta \\ \dot{\theta} c\phi + \psi c\theta s\phi \\ \psi c\theta c\phi - \dot{\theta} s\phi \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\phi & c\phi & 0 \\ c\theta c\phi & -s\phi & 0 \end{bmatrix} \\ &= [b_1 \quad b_2 \quad b_3] \end{aligned} \quad (1.33)$$

Where

$$b_1 = -I_{xx}(\dot{\phi}s\theta - \psi s^2\theta) + I_{yy}(\dot{\theta}c\theta s\phi c\phi + \psi c^2\theta s^2\phi) + I_{zz}(\psi c^2\theta c^2\phi - \dot{\theta}c\theta s\phi c\phi) \quad (1.34)$$

$$b_2 = I_{yy}(\dot{\theta}c^2\phi + \psi c\theta s\phi c\phi) - I_{zz}(\psi c\theta s\phi c\phi - \dot{\theta}s^2\phi) \quad (1.35)$$

$$b_3 = I_{xx}(\dot{\phi} - \psi s\theta) \quad (1.36)$$

Differentiating $\Omega^T I \frac{\partial \Omega}{\partial \dot{\eta}}$ we obtain:

$$\begin{aligned} \dot{b}_1 &= -I_{xx}(\ddot{\phi}s\theta + \dot{\phi}\dot{\theta}c\theta - \psi\dot{s}^2\theta - 2\psi\dot{\theta}s\theta c\theta) + I_{yy}(\ddot{\theta}c\theta s\phi c\phi - \dot{\theta}^2 s\theta s\phi c\phi - \dot{\theta}\dot{\phi}c\theta s^2\phi + \dot{\theta}\dot{\phi}c\theta c^2\phi \\ &+ \psi\dot{c}^2\theta s^2\phi - 2\psi\dot{\theta}s\theta c\theta s^2\phi + 2\psi\dot{\phi}c^2\theta s\phi c\phi) \\ &+ I_{zz}(\psi\dot{c}^2\theta c^2\phi - 2\psi\dot{\theta}s\theta c\theta c^2\phi - 2\psi\dot{\phi}c^2\theta s\phi c\phi - \ddot{\theta}c\theta s\phi c\phi + \dot{\theta}^2 s\theta s\phi c\phi + \dot{\theta}\dot{\phi}c\theta c^2\phi - \dot{\theta}\dot{\phi}c\theta c^2\phi) \end{aligned} \quad (1.37)$$

$$\begin{aligned} \dot{b}_2 &= I_{yy}(\ddot{\theta}c^2\phi - 2\dot{\theta}\dot{\phi}s\phi c\phi + \psi\dot{c}\theta s\phi c\phi - \psi\dot{\theta}s\theta s\phi c\phi + \psi\dot{\phi}c\theta c^2\phi - \psi\dot{\phi}c\theta s^2\phi) \\ &- I_{zz}(\psi\dot{c}\theta s\phi c\phi - \psi\dot{\theta}s\theta s\phi c\phi - \psi\dot{\phi}c\theta s^2\phi + \psi\dot{\phi}c\theta c^2\phi - \ddot{\theta}s^2\phi - 2\dot{\theta}\dot{\phi}s\phi c\phi) \end{aligned} \quad (1.38)$$

$$\dot{b}_3 = I_{xx}(\ddot{\phi} - \dot{\psi}s\theta - \dot{\psi}\dot{c}\theta) \quad (1.39)$$

On the other hand, we have:

$$\frac{\partial \Omega}{\partial \eta} = \begin{bmatrix} 0 & -\dot{\psi}c\theta & 0 \\ 0 & -\dot{\psi}s\theta c\phi & -\dot{\theta}s\phi + \dot{\psi}c\theta c\phi \\ 0 & -\dot{\psi}s\theta c\phi & -\dot{\psi}c\theta s\phi - \dot{\theta}c\phi \end{bmatrix} \quad (1.40)$$

$$\Omega^T I \frac{\partial \Omega}{\partial \eta} = [h_1 \quad h_2 \quad h_3] \quad (1.41)$$

Where

$$h_1 = 0 \quad (1.42)$$

$$h_2 = -I_{xx}(\dot{\psi}\dot{c}\theta - \dot{\psi}^2 s\theta c\theta) - I_{yy}(\dot{\psi}\dot{\theta}s\theta s\phi c\phi + \dot{\psi}^2 s\theta c\theta s^2\phi) - I_{zz}(\dot{\psi}^2 s\theta c\theta c^2\phi - \dot{\psi}\dot{\theta}s\theta s\phi c\phi) \quad (1.43)$$

$$h_3 = I_{yy}(-\dot{\theta}^2 s\phi c\phi - \dot{\psi}\dot{\theta}c\theta s^2\phi + \dot{\psi}\dot{\theta}c\theta c^2\phi + \dot{\psi}^2 c^2\theta s\phi c\phi) + I_{zz}(-\dot{\psi}^2 c^2\theta s\phi c\phi + \dot{\psi}\dot{\theta}c\theta s^2\phi - \dot{\psi}\dot{\theta}c\theta c^2\phi + \dot{\theta}^2 s\phi c\phi) \quad (1.44)$$

The Euler-Lagrange equation for torques is: first no conservative torques from the action of the thrust and second from the gyroscopic effect resulting from the propellers rotations:

$$\tau = \tau_\eta + \tau_g = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} + \begin{bmatrix} \tau_{g\phi} \\ \tau_{g\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{b}_1 - h_1 \\ \dot{b}_2 - h_2 \\ \dot{b}_3 - h_3 \end{bmatrix} \quad (1.45)$$

After mathematical simplification (2.27) and (2.45) we can represent the model same in [1][11][44][45] :

$$\left\{ \begin{array}{l} \ddot{x} = (c\phi s\theta c\psi + s\phi s\psi) \frac{1}{m} u \\ \ddot{y} = (c\phi s\theta s\psi - s\phi c\psi) \frac{1}{m} u \\ \ddot{z} = \frac{u}{m} (c\phi c\theta) - g \\ \ddot{\phi} = \dot{\theta}\dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \frac{J_r}{I_{xx}} \dot{\theta}\dot{\Omega}_r + \frac{\tau_\phi}{I_{xx}} \\ \ddot{\theta} = \dot{\phi}\dot{\psi} \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{J_r}{I_{yy}} \dot{\phi}\dot{\Omega}_r + \frac{\tau_\theta}{I_{yy}} \\ \ddot{\psi} = \dot{\phi}\dot{\theta} \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{\tau_\psi}{I_{zz}} \end{array} \right. \quad (1.46)$$

1.3 Dynamic model used for attitude stabilization

The two gyroscopic effects in dynamic model (1.46). Their influence in our case less important than the motor's action. Especially if we consider in hovering stabilization. In order to simplify of design PID controllers for our quadcopter, we can neglect these gyroscopic effects and thus remove the cross coupling [1][11][44][45]. The model (1.46) for attitude stabilization will be as in (1.47) and for altitude stabilization as in (1.48):

$$\left\{ \begin{array}{l} I_y \times \ddot{\theta} = \tau_\theta \\ I_x \times \ddot{\phi} = \tau_\phi \\ I_z \times \ddot{\psi} = \tau_\psi \end{array} \right\} \left\{ \begin{array}{l} \tau_\theta = (-f_1 - f_2 + f_3 + f_4) \times l_1 \\ \tau_\phi = (-f_1 + f_2 + f_3 - f_4) \times l_2 \\ \tau_\psi = (-\tau_{m1} + \tau_{m2} - \tau_{m3} + \tau_{m4}) \end{array} \right. \quad (1.47)$$

$$f_i = C_T \times \omega_i^2, \quad \tau_{mi} = C_D \times \omega_i^2 \quad i=1,2,3,4, \quad l_1 = l \times \cos(52), \quad l_2 = l \times \sin(52)$$

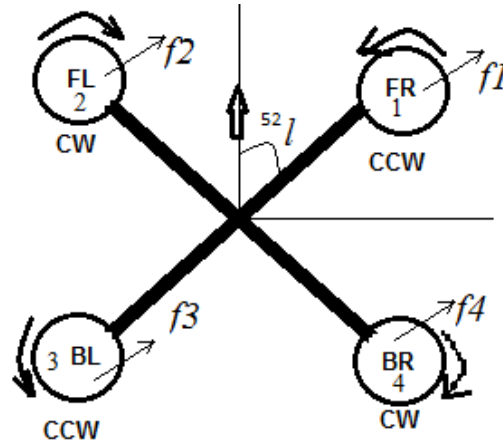


Figure 1.1 A basic configuration of a quadcopter.

Where f_i are the thrust forces generated by four motors, l is the length of the quad-rotor arm, and $\tau_\theta, \tau_\phi, \tau_\psi$ are the generated torques of roll and pitch and yaw plans. $\tau_{m1}, \tau_{m2}, \tau_{m3}, \tau_{m4}$ Are induced moments from there propeller rotation [35], inertial of quadcopter are I_x, I_y and I_z calculated by SOLIDWORKS, the thrust and drag coefficients data I found them in this technical data on this website [46] see Table 1.1.

Table 1.1, Quadrotor parameters

Calculated Thrust Coefficient	$8.5 \cdot 10^{-7}$	Mass	0.49Kg
Thrust Coefficient C_T	$8.8 \cdot 10^{-7}$	I_x	0.00102 Kg*m ²
Drag Coefficient C_D	$11.3 \cdot 10^{-8}$	I_y	0.00059 Kg*m ²
l	0.127 meter	I_z	0.00138 Kg*m ²

Table 1.2 some experiment results for thrust calculation

Duty (ms)	RPM(Round Per Minute)	Motor speed (rad/s)	Weight (kg)	Thrust (N)= Weight*gravity	Thrust coefficient= Thrust/(Motor speed ²)
1410	11225	1175.48	0.125	1.23	$8.12 \cdot 10^{-7}$
1900	20730	2170.84	0.39	3.83	$8.87 \cdot 10^{-7}$
Average=					$8.5 \cdot 10^{-7}$

The dynamic thrust of the motor-driven propeller was identified experimentally as described in references [2] [47] also the thrust coefficient calculated in same experiment is approximate to thrust coefficient in [46] see Table 1.1. The frequency response of the propeller

(GemFan 5030) speed with respect to the duty ratio of the motor speed controller (see Table 1.2) showed that dynamic thrust and duty ratio could be modelled as a first-order transfer function[47] configuration for this experiment showed in Figure 1.4.

Our platform was based on a 250 mm (QAV250) fiberglass frame similar to experiment in this work [12]. A homemade software, and the electronic components included an Arduino Due, an IMU, and an electronic speed controller. The real photo of quadrotor in Figure 1.2. The front adds part because Arduino Duo is long and the black thing below it is just polyester to reduce noise affect on Ultrasonic sensor. Since the software was mostly custom-developed, it could easily interface with any additional sensors and modify the control laws of the PID control strategy [22] for attitude stabilization in Figure 1.3.



Figure 1.2 The realized quadcopter.

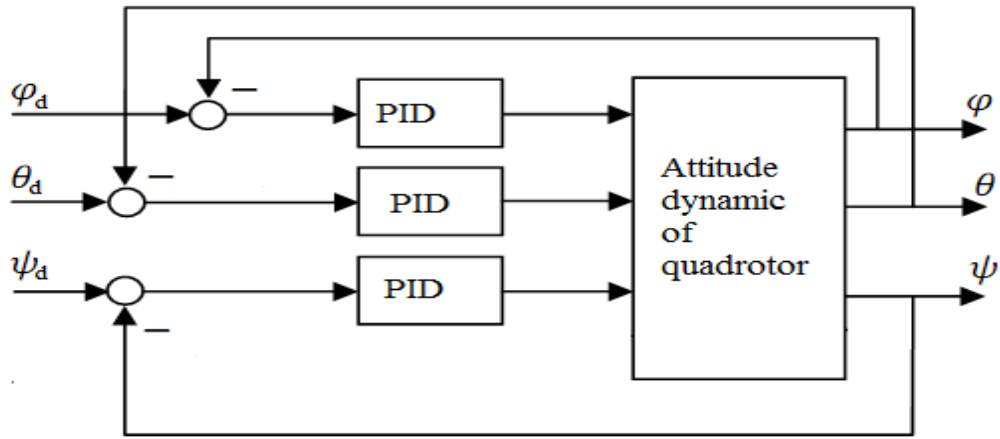


Figure 1.3 PID control strategy for attitude stabilization [22].

θ_d :is the desired roll Angle. φ_d : is the desired pitch Angle. Ψ_d :is the desired yaw angle.

Table 2.3 Experimental thrust calculation

Duty Cycle ms	1178.00	1214.00	1272.00	1333.00	1381.00	1410.00	1473.00	1512.86	1518.10	1556.13	1583.25	1612.83	1660.00	1900.00
RPM	3000.00	4300.00	6330.00	8535.00	10260.00	11225.00	13590.00	14944.29	15210.00	16533.75	17490.00	18695.00	20211.00	20730.00
rate Rad/s	314.16	450.29	662.88	893.78	1074.42	1175.48	1256.64	1564.96	1644.10	1731.41	1831.55	1957.74	2116.49	2170.84
T(N)	0.0869	0.1784	0.3867	0.7030	1.0159	1.2159	1.3896	1.4700	1.6224	1.7993	2.0135	2.3005	2.6887	2.8286

Note: the thrust in Table 1.2 calculated using balance in grams multiplied by gravity. This is clear that is approximately similar to thrust calculated by f_i in (1.47) based on angular velocity shown in Table 1.3 by Rate (Rad/s). Optical RPM give real round per minute. Rate(rad/s)= RPM* $2\pi/60$.

$$T(N)=C_T*Rate^2$$



Figure 1.4 Hardware configuration for thrust calculation.

1. Optical RPM meter, 2. Balancer in grams, 3. Wattmeter, 4. To control motor.

1.4 Dynamic model used for altitude stabilization

Our platform is the same which used for attitude stabilization in [38][48] but in this work for altitude stabilisation will use Teensy 3.2 as autopilot of our prototype[49]. A homemade program, and the electronic components involved an Teensy 3.2 [50] see Figure 1.6, an Inertial Measurement Unit(IMU), and PX4FLOW Smart Camera[51] see Figure 1.7. Reason of using this Smart camera for its good ultrasonic sensor and its sampling time is 0.01second. The real picture of quadcopter in Figure 1.5. Since the code of program was mostly self-developed, it could interface with any optional sensors and changing the control strategies for the studied case. We also based on some ready library of FreeIMU [52] [53] and this project[54].

$$\begin{cases} \ddot{z} = \frac{u}{m} \cos \theta \cos \varphi - g \\ u = f_1 + f_2 + f_3 + f_4 \\ \theta \approx \varphi \approx 0 \text{ in hovering} \end{cases} \quad (1.48)$$



Figure 1.5 The realized quadcopter based on Teensy.

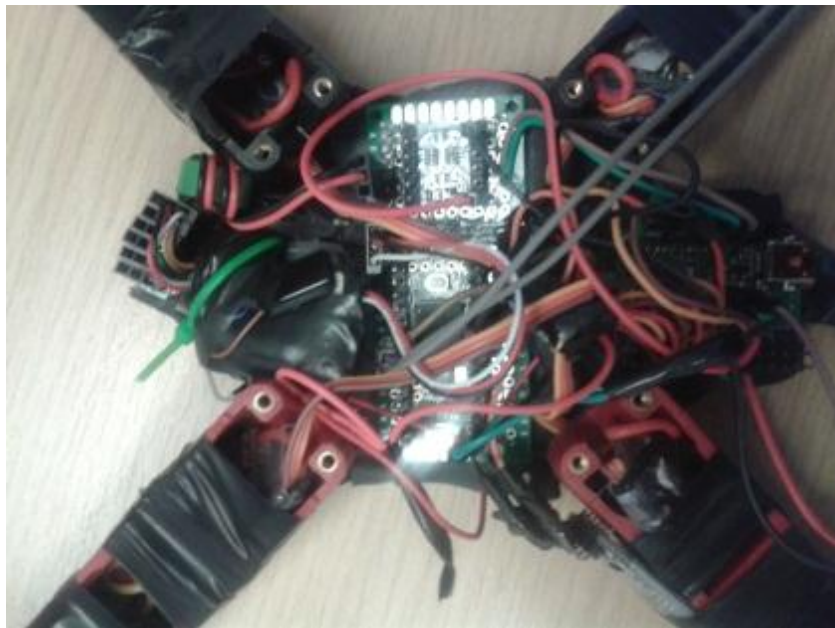


Figure 1.6 Teensy development board 3.2.



Figure 1.7 PX4FLOW Smart Camera.

1.5 CONCLUSION

In This chapter we present dynamic model of a Quadcopter. Also the basics knowledge of modelling the quadcopter as Euler-Lagragian equation presented to get dynamics equation of a quadcopter which will be used in simulation section. Provide experiment identification for quadcopter dynamics parameters. The obtained dynamics model using Euler-Lagrange equation in our case was simplified to use it with PID controllers by neglect gyroscopic torque effect and the cross coupling. We provide some pictures for the two realized quadcopter the first its flight controller based on Arduino Due and the second based on Teensy 3.2. We have used in the first Arduino Due because wide library used and more practitioner. After get good skills and more understand about flight controllers we have improve the prototype by using Teensy 3.2 development board. Teensy 3.2 development board is less size and weight ten times less than Arduine Due and faster program execution. The obtained models will used in simulation for design PID controllers for quadcopter stabilization.

Chapter 2

Modeling rigid manipulator robot

2.1 Introduction

The control and simulation of robots requires the development of different mathematical models. Several levels of modeling –geometric, kinematic and dynamic- are needed depending on the objectives, the constraints of the task and the desired performance.

Obtaining these models is not an easy task. The difficulty varies according to the complexity of the kinematics of the mechanical structure and its degrees of freedom. For that we take the dynamic model of PUMA560 from [24].

This chapter shows some mathematical tools witch using for modeling robot, the aim of modeling to simplifies and estimates the values of the geometric and dynamic parameters of the robot. Besides to find a control law on robot controller with reduced number of operations.

2.2 PUMA 560 robot Dynamics:

The dynamic model used for PUMA560 it is [24]:

$$M(q).\ddot{q} + B(q).[\dot{q}.\dot{q}] + C(q).[\dot{q}^2] + G(q) = \Gamma \quad (2.1)$$

Where,

$B(q)$: nxn(n-1)/2 matrix of Coriolis torques

$C(q)$: nxn matrix of Centrifugal torques

$[\dot{q}\dot{q}]$: n(n-1)/2x1 vector of joint velocity products given by:

$$[\dot{q}_1.\dot{q}_2, \dot{q}_1.\dot{q}_3, \dots, \dot{q}_1.\dot{q}_n, \dot{q}_2.\dot{q}_3, \dot{q}_2.\dot{q}_4, \dots, \dot{q}_{n-2}.\dot{q}_n, \dot{q}_{n-1}.\dot{q}_n]^T$$

$[\dot{q}^2]$: nx1 vector given by: $[\dot{q}_1^2, \dot{q}_2^2, \dots, \dot{q}_n^2]$

The position of zero joint angles and coordinate frame attachments to the PUMA arm are shown in Figure 2.1 above. The modified Denavit-Hartenberg parameters, assigned according to the method presented in [26] are listed in Table 2.1.

i	α_{i-1} (degrees)	θ_i	a_{i-1} (meters)	d_i (meters)
1	0	q_1	0	0
2	-90	q_2	0	0.2435
3	0	q_3	0.4318	-0.0934
4	90	q_4	-0.0203	0.4331
5	-90	q_5	0	0
6	90	q_6	0	0

Table 2.1 Modified Denavit - Hartenberg Parameters[24]

The mass of links 2 through 6 of the PUMA arm are reported in Table 2.2, the mass of link 1 is not included because that link was not removed from the base. Separately measured mass and inertia terms are not required for link one because that link rotates only about its own 2 axis.

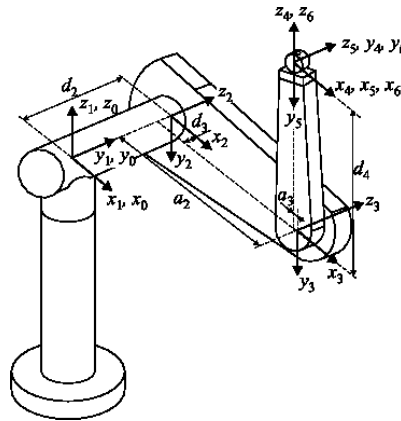


Figure 2.1. The Puma 560 Robot [55]

Link	Mass
Link2	17.40
Link3	4.80
Link4*	0.82
Link5*	0.34
Link6*	0.09
Link 3 with complete Wrist	6.04
Detached Wrist	2.24

Table 2.2 Link Mass[24]

The positions of the centers of gravity are reported in Table 2.3 the dimensions r_x, r_y and r_z refer to the x, y and z coordinate of the center of gravity in the coordinate frame attached to the link.

Link	r_x	r_y	r_z
Link 2	0.068	0.006	-0.016
Link 3	0	-0.070	0.014
Link 3 With Wrist	0	-0.143	0.014
Link 4*	0	0	-0.019
Link 5*	0	0	0
Link 6*	0	0	0.032
Wirst	0	0	-0.064

Table 2.3 Centers of Gravity[24].

Link	I_{xx}	I_{yy}	I_{zz}	I_{motor}
Link 1	-	-	0.35	1.14(± 0.27)
Link 2	0.130($\pm 3\%$)	0.524($\pm 5\%$)	0.539($\pm 3\%$)	4.71(± 0.54)
Link 3	0.066	0.0125	0.086	0.83(± 0.09)
Link 3 With Wrist	0.192($\pm 4\%$)	0.0154($\pm 5\%$)	0.212($\pm 4\%$)	-
Link 4*	1.8×10^{-3}	1.8×10^{-3}	1.3×10^{-3}	0.200(± 0.016)
Link 5*	0.30×10^{-3}	0.30×10^{-3}	0.40×10^{-3}	0.179(± 0.014)
Link 6*	0.15×10^{-3}	0.15×10^{-3}	0.04×10^{-3}	0.193(± 0.015)

Table 2.4 Diagonal Terms of the Inertia Dynamics and Effective Motor Inertia. [24]

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Gear Ration	62.61	107.36	53.69	76.01	71.91	76.73
Maximum Torque(N-m)	97.6	180.4	89.4	24.2	20.1	21.3
Break Away Torque (N-m)	6.3	5.5	2.6	1.3	1.0	1.2

Table 2.5 Motor and Drive Parameter[24]

2.3 Using PUMA Robot as 3-DOF Robot:

Recall that only three links of PUMA robot are used in this thesis, $q_4 = q_5 = q_6 = 0$.

The configuration space equation same in (2.1),

With,

Matrix A is a symmetric 6x6 matrix:

$$A(q) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \quad (2.2)$$

Where,

$$a_{11} = I_{m1} + I_1 + I_3 \cdot CC2 + I_7 \cdot SS23 + I_{10} \cdot SC23 + I_{11} \cdot SC2 + I_{21} \cdot SS23 + 2 \cdot [I_5 \cdot C2 \cdot S23 + I_{12} \cdot C2 \cdot C23 + I_{15} \cdot SS23 + I_{16} \cdot C2 \cdot S23 + I_{22} \cdot SC23] \quad (2.3)$$

$$a_{12} = I_4 \cdot S2 + I_8 \cdot C23 + I_9 \cdot C2 + I_{13} \cdot S23 - I_{18} \cdot C23 \quad (2.4)$$

$$a_{13} = I_8 \cdot C23 + I_{13} \cdot S23 - I_{18} \cdot C23 \quad (2.5)$$

$$a_{22} = I_{m2} + I_2 + I_6 + 2 \cdot [I_5 \cdot S3 + I_{12} \cdot C2 + I_{15} + I_{16} \cdot S3] \quad (2.6)$$

$$a_{23} = I_5 \cdot S3 + I_6 + I_{12} \cdot C3 + I_{16} \cdot S3 + 2 \cdot I_{15} \quad (2.7)$$

$$a_{33} = I_{m3} + I_6 + 2 \cdot I_{15} \quad (2.8)$$

$$a_{35} = I_{15} + I_{17} \quad (2.9)$$

$$a_{44} = I_{m4} + I_{14} \quad (2.10)$$

$$a_{46} = I_{23} \approx 0 \quad (2.11)$$

$$a_{55} = I_{m5} + I_{17} \quad (2.12)$$

$$a_{66} = I_{m6} + I_{23} \quad (3.13)$$

$$a_{21} = a_{12}, a_{31} = a_{13} \text{ and } a_{32} = a_{23}$$

While matrix B is:

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 \\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{412} & b_{413} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.14)$$

Where,

$$b_{112} = 2.[-I_3.SC2 + I_5.C223 + I_7.SC23 - I_{12}.S223 + I_{15}.2.SC23 + I_{16}.C223 + I_{21}.SC23 + I_{22}.(1 - 2.SS23)] + I_{10}.(1 - 2.SS23) + I_{11}.(1 - 2.SS2) \quad (2.15)$$

$$b_{113} = 2.[I_5.C2.C23 + I_7.SC23 - I_{12}.C2.S23 + I_{15}.2.SC23 + I_{16}.C2.C23 + I_{21}.SC23 + I_{22}.(1 - 2.SS23)] + I_{10}.(1 - 2.SS23) \quad (2.16)$$

$$b_{115} = 2.[-SC23 + I_{15}.SC23 + I_{16}.C2.C23 + I_{22}.CC23] \quad (2.17)$$

$$b_{123} = 2.[-I_8.S23 + I_{13}.C23 + I_{18}.S23] \quad (2.18)$$

$$b_{214} = I_{14}.S23 + I_{19}.S23 + 2.I_{20}.S23.(1 - 0.5) \quad (2.19)$$

$$b_{223} = 2.[-I_{12}.S3 + I_5.C3 + I_{16}.C3] \quad (2.20)$$

$$b_{225} = 2.[I_{16}.C3 + I_{22}] \quad (2.21)$$

$$b_{235} = 2.[I_{16}.C3 + I_{22}] \quad (2.22)$$

$$b_{314} = 2.[I_{20}.S23.(1 - 0.5)] + I_{14}.S23 + I_{19}.S23 \quad (2.23)$$

$$b_{412} = -b_{214} = -[I_{14}.S23 + I_{19}.S23 + 2.I_{20}.S23.(1 - 0.5)] \quad (2.24)$$

$$b_{413} = -b_{314} = -2.[I_{20}.S23.(1 - 0.5)] + I_{14}.S23 + I_{19}.S23 \quad (2.25)$$

$$b_{415} = -I_{20}.S23 - I_{17}.S23 \quad (2.26)$$

$$b_{514} = -b_{415} = I_{20}.S23 + I_{17}.S23 \quad (2.27)$$

Matrix C is:

$$C(q) = \begin{bmatrix} 0 & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{21} & 0 & c_{23} & 0 & 0 & 0 \\ c_{31} & c_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c_{51} & c_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.28)$$

Where,

$$c_{12} = I_4.C2 - I_8.S23 - I_9.S2 + I_{13}.C23 + I_{18}.S23 \quad (2.29)$$

$$c_{13} = 0.5.b_{123} = -I_8.S23 + I_{13}.C23 + I_{18}.S23 \quad (2.30)$$

$$c_{21} = -0.5.b_{112} = I_3.SC2 - I_5.C223 - I_7.SC23 + I_{12}.S223 - I_{15}.2.SC23 - I_{16}.C223 - I_{21}.SC23 - I_{22}.(1-2.SS23) - 0.5.I_{10}.(1-2.SS23) - 0.5.I_{11}.(1-2.SS2) \quad (2.31)$$

$$c_{23} = 0.5.b_{223} = -I_{12}.S3 + I_5.C3 + I_{16}.C3 \quad (2.32)$$

$$c_{31} = -0.5.b_{113} = -I_5.C2.C23 - I_7.SC23 + I_{12}.C2.S23 - I_{15}.2.SC23 - I_{16}.C2.C23 - I_{21}.SC23 - I_{22}.(1-2.SS23) - 0.5.I_{10}.(1-2.SS23) \quad (2.33)$$

$$c_{32} = -c_{23} = I_{12}.S3 - I_5.C3 - I_{16}.C3 \quad (2.34)$$

$$c_{51} = -0.5.b_{115} = SC23 - I_{15}.SC23 - I_{16}.C2.C23 - I_{22}.CC23 \quad (2.35)$$

$$c_{52} = -0.5.b_{225} = -I_{16}.C3 - I_{22} \quad (2.36)$$

And matrix G is:

$$g(q) = \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \quad (2.37)$$

$$g_2 = g_1.C2 + g_2.S23 + g_3.S2 + g_4.C23 + g_5.S23 \quad (2.38)$$

$$g_3 = g_2.S23 + g_4.C23 + g_5.S23 \quad (2.39)$$

$$g_5 = g_5.S23 \quad (2.40)$$

Where,

$$Si = \sin(\theta_i), Ci = \cos(\theta_i), Cij = \cos(\theta_i + \theta_j), Sijk = \sin(\theta_i + \theta_j + \theta_k),$$

$$CCi = \cos(\theta_i).cos(\theta_i) \text{ and } Csi = \cos(\theta_i).sin(\theta_i)$$

Tables 2.7 and 2.8 contain the computed values for the constants appearing in the equations of forces of motion,

$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

Table 2.6 Inertiel Constants ($kg.m^2$)[24]

$g_1 = -37.2 \pm 0.5$	$g_2 = -8.44 \pm 0.20$
$g_3 = 1.02 \pm 0.50$	$g_4 = 0.249 \pm 0.025$
$g_5 = -0.0282 \pm 0.0056$	

Table 2.7 Gravitational Constants (N.m) [1]

The three degree of freedom PUMA robot has the same configuration space equation general form as its 6-DOF convenient. In this type, the last three joints are blocked so they keep their initial states while the robot is moving. Using the configuration equation of the robot, and by setting the last joints as zero always, we can define a general equation that allows us to use PUMA robot as a 3-DOF robot.

However, the 3-DOF PUMA will have the same kinematics of its 6-DOF convenient with q_4, q_5 and q_6 set to zero.

For the configuration space equation of the robot in (2.1),

We set $q_4 = q_5 = q_6 = 0$, this yields:

$$\ddot{q} = [\ddot{q}_1 \dots \ddot{q}_2 \dots \ddot{q}_3 \dots 0 \dots 0 \dots 0]^T, \quad (2.41)$$

$$[\dot{q}\dot{q}] = [\dot{q}_1 \dot{q}_2 \dots \dot{q}_1 \dot{q}_3 \dots 0 \dots 0 \dots 0 \dots \dot{q}_2 \dot{q}_3 \dots 0 \dots 0 \dots 0 \dots 0 \dots 0 \dots 0 \dots 0]^T, \quad (2.42)$$

$$[\dot{q}^2] = [\dot{q}_1^2 \dots \dot{q}_2^2 \dots \dot{q}_3^2 \dots 0 \dots 0 \dots 0]^T, \quad (2.43)$$

$$B(q) \cdot \dot{q}\dot{q} = [b_{112} \cdot \dot{q}_1 \dot{q}_2 + b_{113} \cdot \dot{q}_1 \dot{q}_3 + b_{123} \cdot \dot{q}_2 \dot{q}_3 \dots b_{223} \cdot \dot{q}_2 \dot{q}_3 \dots 0 \dots b_{412} \cdot \dot{q}_1 \dot{q}_2 + b_{413} \cdot \dot{q}_1 \dot{q}_3 \dots 0 \dots 0]^T \quad (2.44)$$

And

$$C(q) \cdot \dot{q}^2 = [c_{12} \cdot \dot{q}_2^2 + c_{13} \cdot \dot{q}_3^2 \dots c_{21} \cdot \dot{q}_1^2 + c_{23} \cdot \dot{q}_3^2 \dots c_{31} \cdot \dot{q}_1^2 + c_{32} \cdot \dot{q}_2^2 \dots 0 \dots c_{51} \cdot \dot{q}_1^2 + c_{52} \cdot \dot{q}_2^2 \dots 0]^T \quad (2.45)$$

The angular acceleration is found as to be:

$$\ddot{q} = A^{-1}(q) \cdot \left\{ \Gamma - [B(q) \cdot \dot{q}\dot{q} + C(q) \cdot \dot{q}^2 + g(q)] \right\} \quad (2.46)$$

$$\text{Now let } I = \left\{ \Gamma - [B(q) \cdot \dot{q}\dot{q} + C(q) \cdot \dot{q}^2 + g(q)] \right\} \Rightarrow \ddot{q} = A^{-1}(q) \cdot I \quad (2.47)$$

$$I_1 = \Gamma_1 - [b_{112} \cdot \dot{q}_1 \dot{q}_2 + b_{113} \cdot \dot{q}_1 \dot{q}_3 + b_{123} \cdot \dot{q}_2 \dot{q}_3] - [c_{12} \cdot \dot{q}_2^2 + c_{13} \cdot \dot{q}_3^2] \quad (2.48)$$

$$I_2 = \Gamma_2 - [b_{223} \cdot \dot{q}_2 \dot{q}_3] - [c_{21} \cdot \dot{q}_1^2 + c_{23} \cdot \dot{q}_3^2] - g_2 \quad (2.49)$$

$$I_3 = \Gamma_3 - [c_{31} \cdot \dot{q}_1^2 + c_{32} \cdot \dot{q}_2^2] - g_3 \quad (2.50)$$

$$I_4 = \Gamma_4 - [b_{412} \cdot \dot{q}_1 \dot{q}_2 + b_{413} \cdot \dot{q}_1 \dot{q}_3] \quad (2.51)$$

$$I_5 = \Gamma_5 - [c_{51} \cdot \dot{q}_1^2 + c_{52} \cdot \dot{q}_2^2] - g_5 \quad (2.52)$$

$$I_6 = \Gamma_6 \quad (2.53)$$

These equations tell us that in order to ensure that \ddot{q}_4, \ddot{q}_5 and \ddot{q}_6 keep their zero values, it is better to set $I_4 = I_5 = I_6 = 0$; so by holding the control torques of the last three joints as

$$\Gamma_4 = [b_{412} \cdot \dot{q}_1 \dot{q}_2 + b_{413} \cdot \dot{q}_1 \dot{q}_3] \quad (2.54)$$

$$\Gamma_5 = [c_{51} \cdot \dot{q}_1^2 + c_{52} \cdot \dot{q}_2^2] + g_5 \quad (2.55)$$

And $\Gamma_6 = 0$, the last three joints are blocked at their initial states.

2.4 Conclusion

In this chapter we give some background mathematical to modeling robot we talking about geometric description and dynamic modeling and about modifications to get 3DOF.

The dynamic model description for PUMA560 robot was presented, after we explain how can we using PUMA560 as 3DOF robot, the control for it will presented later which is optimized fuzzy logic controller.

Chapter 3

A Quadcopter hardware description

3.1 INTRODUCTION

In this chapter, the main quadcopter's component parts are presented. They are fundamental to identify the quadcopter's attitude and its height from the ground. Every quadcopter requires, at minimum, the following components: a frame, motors with propellers, motor speed controllers, a battery, a radio receiver, a flight controller, and an attitude sensor. This chapter will discuss the function of each component and general hardware connection between the electronic components. The real photo of the quadcopter shows in Figure 3.13 and Figure 3.14.

3.2 Flight controller

Flight controller is the brain of a quadcopter and it performs the necessary operations to keep the quadcopter stable and controllable. For that we used Teensy development board and Arduino due showed in Figure 3.1 and Figure 3.2.

Teensy v3.2 development board [65]:based on the 32 bit ARM Cortex-M4 72 MHz CPU .The Teensy is a breadboard-friendly development board with loads of features in a, well, teensy package. Each Teensy 3.2 comes pre-flashed with a bootloader so you can program it using the on-board USB connection: No external programmer needed! You can program for the Teensy in your favorite program editor using C *or* you can install the Teensyduino add-on for the Arduino IDE and write Arduino sketches for Teensy.



Figure 3.1: Teensy Development board[65].

Arduino due: The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller[66].

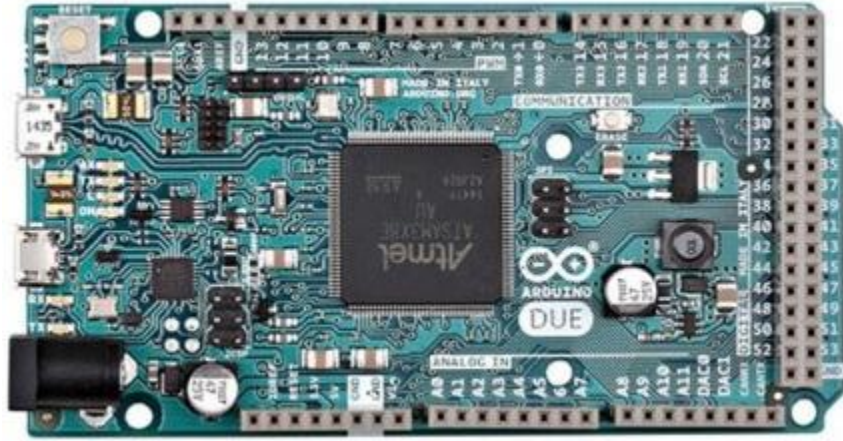


Figure 3.2: Arduino due[66].

3.3 Inertial Measurement Unit

iMU is the acronym of Inertial Measurement Unit, which is sensor capable of measuring the orientation of a body in 3D space[67]. For our project we used GY-86 see Figure 3.3.

GY-86 Description:

Small based on the sensor board of InvenSense MPU-6050 (3-axis accelerometer and 3-axis gyroscope), digital 3-axis compass HMC5883L Honeywell and the pressure sensor MS56.

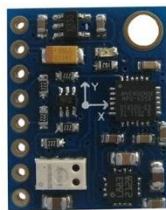


Figure 3.3 :GY-86 IMU[67].

The InvenSense **MPU-6050** sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino[68].

The Honeywell **HMC5883L** is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry[69].

The **MS5611-01BA**[70] is a new generation of high resolution altimeter sensors from MEAS Switzerland with SPI and I2C bus interface. This barometric pressure sensor is optimized for altimeters and variometers with an altitude resolution of 10 cm.

3.4 Speed Controllers

Every motor needs an individual electronic speed controller see Figure 3.4. These speed controllers accept commands in the form of Pulse Width Modulation signals and output the appropriate motor speed accordingly[71].



Figure 3.4 Electronic Speed Controller (ESC)[71].

3.5 Motors and Propellers

The motors spin the propellers to provide the quadcopter with lifting thrust. Quadcopters almost exclusively use brushless DC motors see Figure 3.5. For our project we used The DYS BE1806-2300kv Brushless Multirotor Motor and propeller reference is GemFan 5030 see Figure 3.6.

The DYS BE1806-13 (Figure 3.5) multirotor motor is ideal for small to medium sized multirotors and comes complete with mounting bolts, a prop adapter and 150mm power leads with 2mm bullet connectors[71].



Figure 3.5 brushless DC motors (DYS BE1806-2300)[71].

GemFan Propellers see Figure 3.6 have set the gold standard in the small class Multirotor community[72]. Known for their durability, great balancing and superb hub centering and expendable price. They come packaged as a set of 2, with one CW rotation and one CCW rotation. There is a wide range of color options that can be mixed and matched to help with multirotor orientation. This is the perfect upgrade for any small multirotor.



Figure 3.6: GemFan Propellers[72].

3.6 Data acquisition device

Data acquisition systems are used by most engineers and scientists for laboratory research, industrial control, test and measurement to input and output data to and from a computer. In our work the HC-05 Bluetooth device (see Figure 3.7) used for data acquisition. It is a great solution for wireless communication.



Figure 3.7: HC-05 Bluetooth device[73].

2.7 Radio Receiver

The radio receiver (Rx) receives radio signals from an RC transmitter and converts them into control signals for each control channel (throttle, yaw, roll & pitch). Modern RC receivers operate on a 2.4 GHz radio frequency, while older Rx units often used 72 MHz frequencies. Rx units may have as few as 4 channels, but many have more channels for additional control options. We selected a Turnigy 6 Channel Receiver for this project, seen in Figure 3.8[74].



Figure 3.8: Radio Receiver[74].

3.8 Lipo battery

The battery provides electrical power to the motors and all electronic components of the quadcopter. Lithium Polymer (LiPo) batteries are used almost exclusively, because they have high specific energy. For this project, we selected Turnigy 2200mAh batteries, seen in Figure 3.9.



Figure 3.9 Turnigy 2200mAh battery[74].

Turnigy batteries are known the world over for performance, reliability and price. It's no surprise to us that Turnigy Lipoly packs are the go-to pack for those in the know. Turnigy batteries deliver the full rated capacity at a price everyone can afford[74].

3.9 UBEC power

Power receiver and servos with UBEC power. The all new Turnigy 4A UBEC "Universal Battery Elimination Circuit" is an advanced switching DC-DC regulator which will supply a constant 5A or more with short bursts of up to 4amps. The Turnigy UBEC plugs into a 3 -10 Cell Lipoly pack and supplies a constant 5 or 6v to your receiver. The system include an anti-short circuit and overheat function with a thin metal shield cover, reducing noise Also included is an Output filter to reduce noise, an on/off switch and a step down regulator adapter see Figure 3.10[75].



Figure 3.10: Universal Battery Elimination Circuit(UBEC)[75].

3.10 Quadcopter frame

Quadcopter Frame The frame of the quadcopter provides the physical structure. It joins the motors to the rest of the quadcopter and houses all of the other components. The frame must be suitable to allow all four propellers to spin without collision, but must not be too large and therefore too heavy for the motors. For our quadcopter we chose a HobbyKing™ Totem Q250 Quadcopter Kit frame as seen in Figure 3.11, which measures at 250mm across opposite motors[76].



Figure 3.11: a HobbyKing™ Totem Q250 frame[76].

3.11 Hardware diagram

There are three main electronic units in our Quadcopter system. The Microcontroller Unit (MCU) for the implementation of controller, the Electronic Speed Controllers (ESCs) for the speed control of the brushless DC motors and finally the Inertial Measurement Unit (IMU) for providing the orientation data for state estimation. The selection and functionality of MCU and ESCs and their operation is described in this chapter. Figure 3.12 depicts the basic inter-connects between different electronic hardware modules.

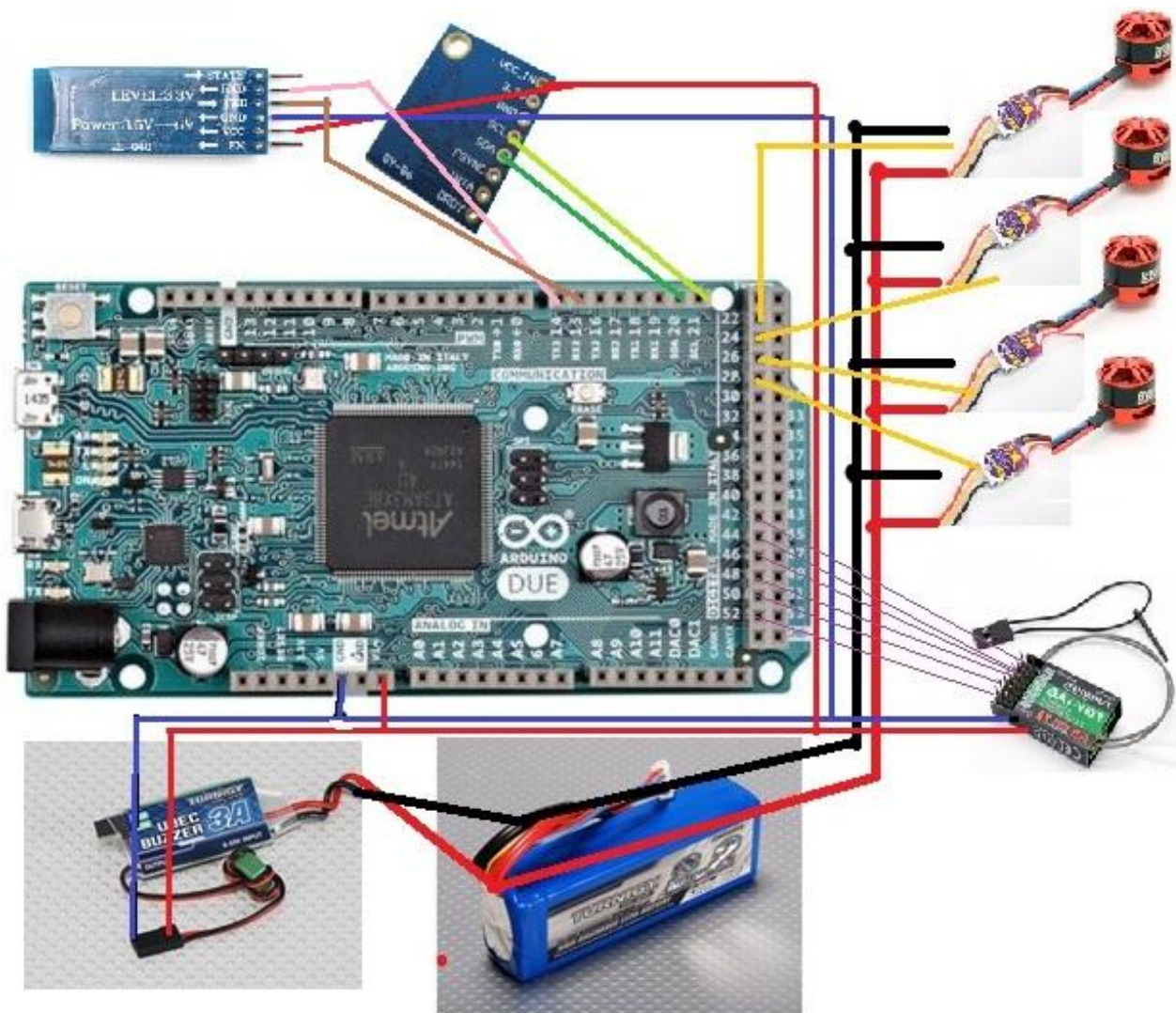


Figure 3.12: hardware diagram of interconnection between electronics parts.

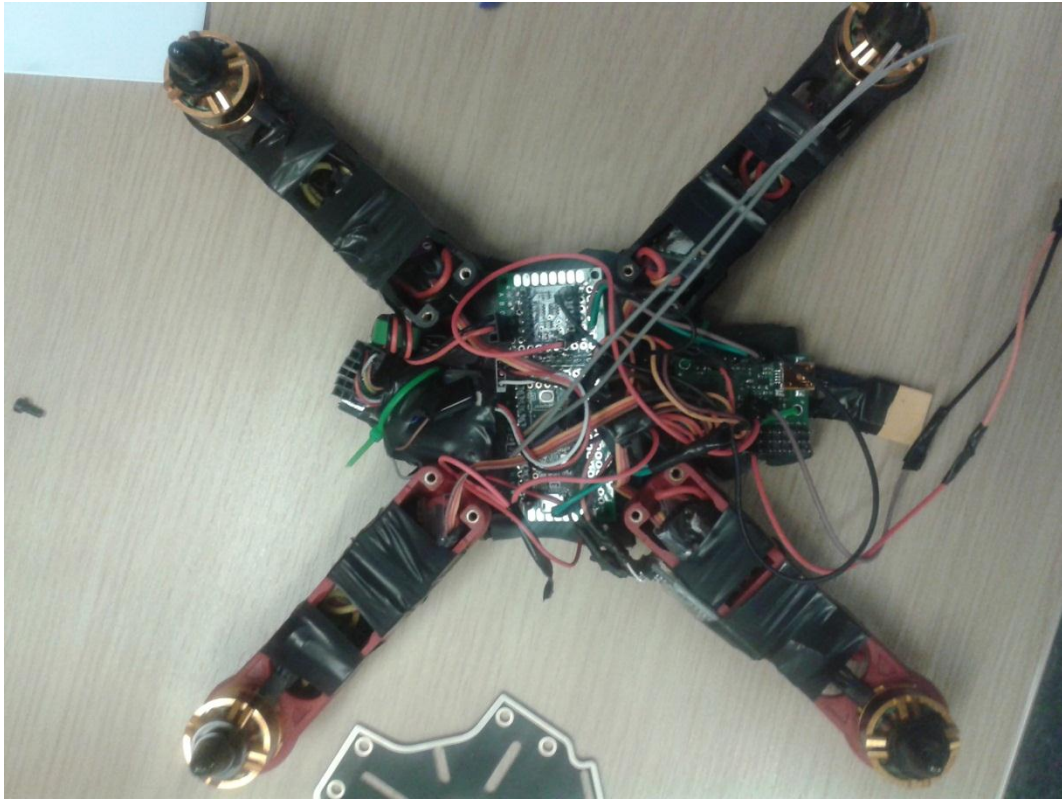


Figure 3.13: The realized quadcopter insight view.



Figure 3.14: Top view of the realized quadcopter.

3.12 CONCLUSION

The purpose of this chapter was to outline the development of the hardware that makes up the autopilot system. The chapter began with an explanation of the choice of airframe, batteries, actuators, sensors, communication links and propulsion system. The diagram of wire connection between all components parts also presented. The next chapter will present discussion about stabilization of a quadcopter based on the developed hardware.

Chapter 4

Tuning PID Controllers for Stabilization of a Quadcopter

4.1 INTRODUCTION

In PID control design, the parameters of the controllers are tuned manually or by using the tuning rules found in the linear control-based literature, such as Ziegler-Nichols (Z-N) tuning [9][10]. Unfortunately, the manual tuning method has limitations of being time consuming and very tedious. While the conventional Z-N method has been very useful and helpful in our research in terms of practicality.

In this chapter we give brief review of the conventional Z-N method. Present experimental results produced from the using ZN in both attitude and altitude stabilisation of a Quadcopter. Present strategy of the using PSO for PID controllers gains optimization for quadcopter stabilisation. Provide simulations and experimental results for quadcopter stabilization.

4.2 TUNING THE PID CONTROLLER USING THE ZIEGLER-NICHOLS METHOD

The Ziegler and Nichols proposed rules [9][10] for determining the values of proportional gain K_p , integral time K_i , and derivative time K_d . The determination of such parameters of PID controllers or tuning of PID controllers can be achieved on-site using experiments involving the quadrotor. There are two methods within the Z-N tuning approach. The first method of Z-N tuning is based on the open-loop step response of the system.

The second method of Z-N tuning, used in this work, is a closed-loop tuning method that requires finding the ultimate gain and period. This can be achieved by adjusting the controller gain (K_c) until the system undergoes sustained oscillations (at the ultimate gain or critical gain), while maintaining the integral time constant (T_i) at infinity and the derivative time constant (T_d) at zero. This work used the second method as shown in Table 4.1[9][10].

Table 4.1, Ziegler-Nichols Tuning rule based on critical gain K_c and critical period P_c .

Type of Controller	K_p	K_i	K_d
P	$0.5 \times K_c$	∞	0
PI	$0.45 \times K_c$	$1/1.2 \times P_c$	0
PID	$0.6 \times K_c$	$0.5 \times P_c$	$0.125 \times P_c$

After completing development of the quadrotor, we had to determine the PID parameters efficiently, and with little theoretical basis since initially, the quadrotor dynamics were not precisely determined. Therefore, the Z-N method was applied as the first step towards stabilizing the quadrotor because it enabled experiments to be used for real-time acquisition of responses that could be plotted to determine PID parameters. Our quadrotor was symmetric, so therefore we only applied this method to the pitch component of the PID controller.

To verify the performance we used the integral of square error (ISE) as comparison criteria, defined as follows:

$$ISE = \int_0^{\infty} e^2(t) dt \quad (4.1)$$

$$e = \text{The desired angle} - \text{The actual angle}$$

Therefore, for PSO-PID tuning, this criteria (4.1) was used as the objective function to seek a set of PID gains such that the control system met the minimum performance criteria.

4.3 PID TUNING BY PARTICLE SWARM OPTIMIZATION

There are a lot of heuristic optimization methods, and one of the most technical of these is PSO, first created by Kennedy and Eberhart in 1995 [18]. It is based on Swarm Intelligence methods, and aims to identify the optimal solution by imitating the movement of particles in a swarm similar to fish schooling or birds flocking.

There are several algorithms used for PSO [16][17][18]. In this work, the integral of square error (ISE) and maximum overshoot of error were fitness functions (multi-objective functions)[56], and the constriction coefficient [18][36] method was selected because method was selected because never used before with our case. The swarm is guaranteed to converge [18] and with simplest control of the convergence speed, whereby the speed in (4.2) and position in (4.3) of each particle changed according the following:

$$v_{ij}(t+1) = X[v_{ij}(t) + \phi_1(y_{ij}(t) - x_{ij}(t)) + \phi_2(\hat{y}_{ij} - x_{ij}(t))] \quad (4.2)$$

Where:

$$X = \frac{2k}{|2 - \phi - \sqrt{\phi(\phi - 4)}|}$$

With:

$$\phi = \phi_1 + \phi_2, \quad \phi_1 = c_1 r_1, \quad \phi_2 = c_2 r_2, \quad \phi \geq 4, \quad k \in [0 \ 1]$$

c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are random values in $[0 \ 1]$.

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (4.3)$$

PSO was used to tune PID gains (K_p , K_i , and K_d) offline using the model given in equation (1.47). PSO first produced an initial swarm of particles in the search space. Each particle represented a solution for PID gains, where their values were set within the range of the produced Z-N gains. Fifteen and twenty were chosen as the number of particles within the swarm and iterations within the search, respectively. A good set of PID controller gains yielded a good system response, and resulted in minimization of the performance criteria in listed in equation (4.1). The general strategy of the optimization problems is represented in Figure 4.1.

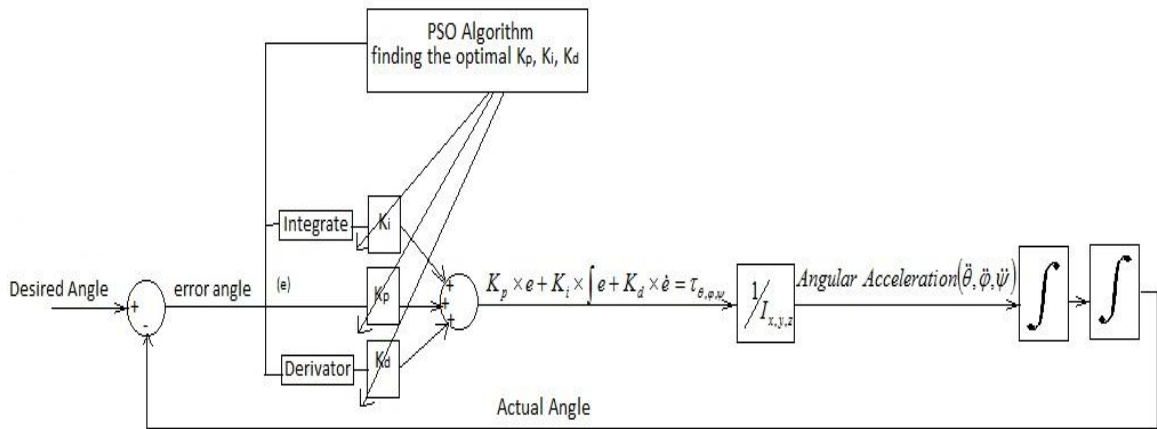


Figure 4.1 The general strategy of optimization

Note: Overshoot= MAX(Actual Angle)-Desired Angle

4.4 RESULTS AND DISCUSSION

In this section we describe the simulation and experimental results for three sets of PID gains. The first PID gains were obtained using the Z-N method. The second PID gains were

obtained using PSO, for which the objective function was equal to sum of the integral square of error and maximum of overshoot. The third PID gains were also obtained using the PSO method, but the objective function was replaced with the Integral Square of Error function.

4.4 1 Z-N PID tuning for attitude stabilization

This First, we employed Z-N tuning experimentally to determine PID controllers gains for attitude stabilization. Then, we varied the Z-N gain (K_c) until an oscillation appeared in our experiment as shown in Figure 4.2.

Figure 4.3 shows the pitch angle error when the Z-N gain was varied. Here, a decrease in error could be observed as K_c increased, until oscillation occurred. Based on the Z-N method, the critical gain is depicted in the black rectangular region in Figure 4.3 and its corresponding value is shown in Figure 4.2 ($K_c= 40$). A magnified rendition of the black rectangle shown in Figure 4.3 was used to calculate the critical period ($P_c=0.2$ s), as shown in Figure 4.4.

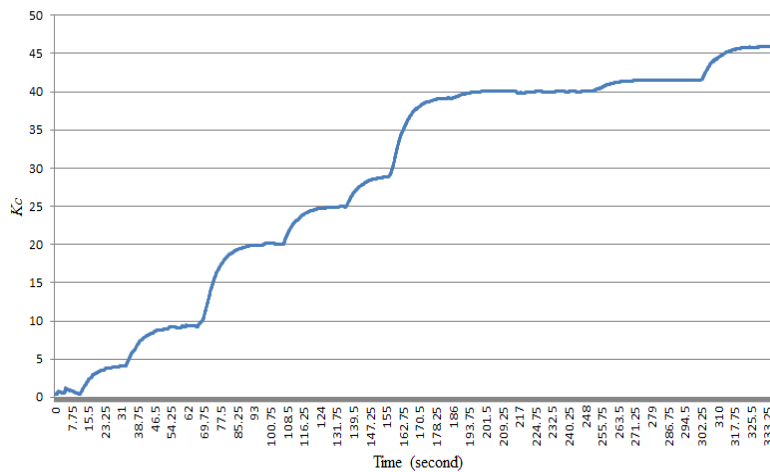


Figure 4.2 Critical gain (K_c) variation in real time.

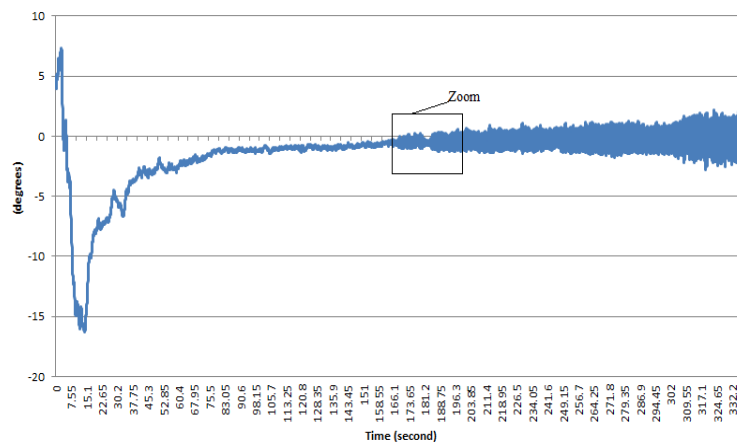


Figure 4.3 Pitch error angle when K_c was varied in real time.

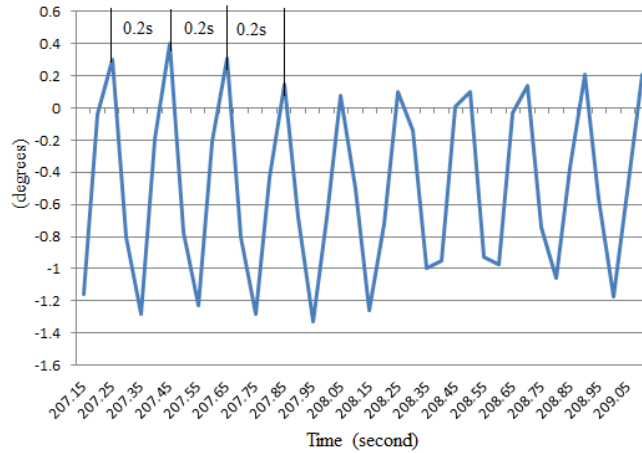


Figure 4.4 Magnified black rectangle to calculate Critical Period P_c in real time

Using the second Z-N method in Table 4.1, we found that:

$$K_c = 40, \quad P_c = 0.2 \text{ for PID rule : } K_p = 0.6 \times K_c = 24; \quad K_i = 0.5 \times P_c = 0.1; \\ K_d = 0.125 \times P_c = 0.025.$$

4.4.2 ZN PID for altitude stabilization

Same application which done in previous section will apply for tuning PID controller using ZN for altitude stabilization of a quadcopter.

Figure 4.6 presents the altitude error when the Z-N gain was changed. Here, a decrease in error could be observed as K_c increased, until oscillation come. Based on the Z-N procedure, the critical gain is in Figure 4.6 and its corresponding value is presented in Figure 4.5 ($K_c = 1.6$). Figure 4.7 presents the critical period ($P_c = 5$ s).

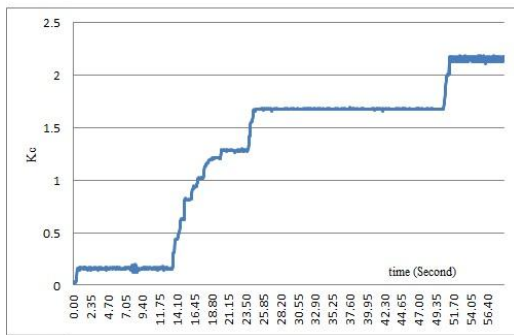


Figure 4.5 Critical gain (K_c) change in real time.

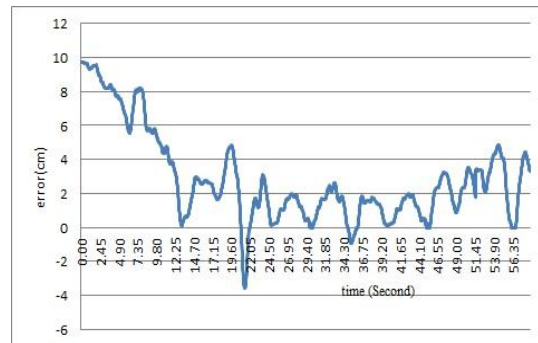


Figure 4.6 Altitude error while K_c was changed in real time.

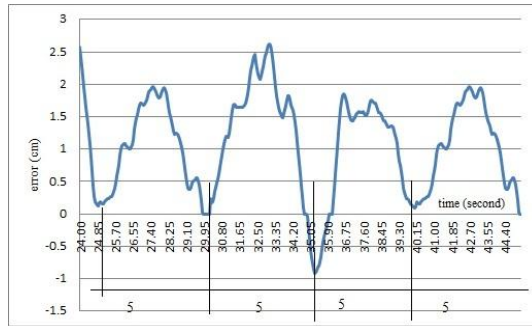


Figure 4.7 Critical Period P_c in real time

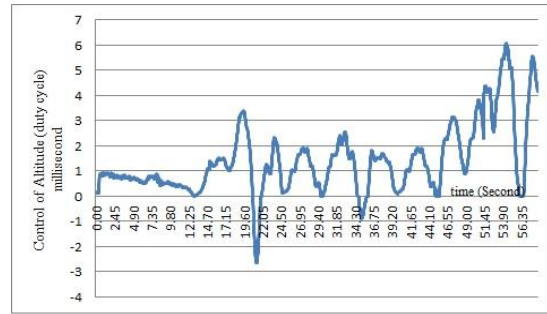


Figure 4.8 Altitude control while ZN using.

Using the second Z-N procedure in Table 4.1, we got that:

$$K_c = 1.6, P_c = 5 \text{ for PI rule : } K_p = 0.45 \times K_c = 0.72; K_i = 0.5 \times P_c = 2.5; K_d = 0.$$

The calculated PID parameters found using the Z-N procedure, is summarized in Table 4.2.

Table 4.2 Summary of PID parameters found by Z-N procedure

	P	I	D
Z-N gains	0.72	2.5	0

While ZN method in implementation also we record real time of altitude control see Figure 4.8 is very clear that has same behaviour of Fig 4.6 which has the real time changing of altitude error.

The PID parameters in Table 4.2, were validated experimentally, and the responses of the altitude error for PID Z-N, are presented in Figure 4.9, it is very clear the hovering error is between 1.5 to -1.5 centimetre.

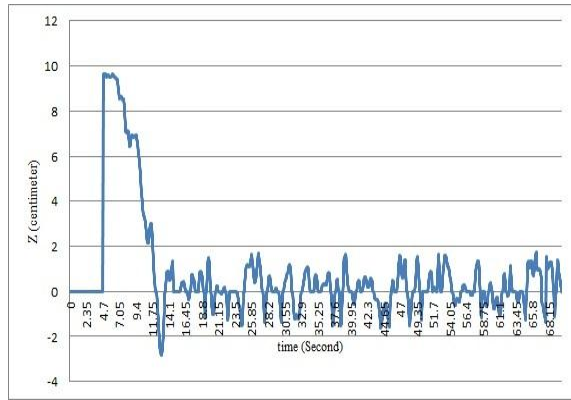


Figure 4.9 Altitude error produced by the calculated PID.



Figure 4.10 The PID control response.

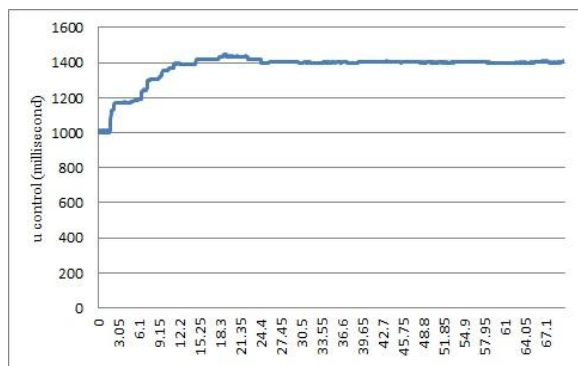


Figure 4.11 The generated controlled thrust.

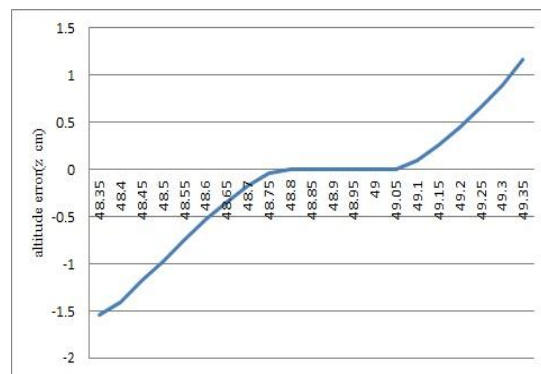


Figure 4.12 altitude error for one second.

Figure 4.10 presents the produced PID control by using Figure 4.10, this response is output of PID block but the control after adding the necessary constant for weight balancing is represented in Figure 4.11 which is the generated thrust for each motors.

The all previous presented figures are for 70 second but we choose to take snapshot for just one second to see our response system how work in real time. Figure 4.12 shows that our altitude errors are between -1.5 to 1.5 cm. the corresponding PID control presented in Figure 4.13. The PID control it must add amount which balance the weight of quadcopter its variation in Figure 4.14. While duration on second also we record Roll and Pitch response which are related with attitude stabilization. We add them just to demonstrate the real time application. Figure 4.15 shows the real time responses for Roll and Pitch angle error tracking is very clear while quadcopter flying that errors of the roll between -0.2 to 0.2 degrees and Pitch between -0.4 to 0.2 degrees. Figure 4.16 shows the corresponding control for Figure 4.15. The actuators of quadcopter

have driver and motor and propeller this system need Pulse Width Modulation (PWM) or we can say to it duty cycle which described in millisecond then the generated control law by PID described for each motors presented in Figure 4.17.

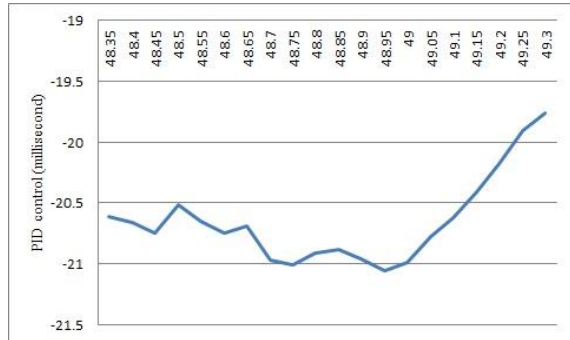


Figure 4.13 PID output for one second



Figure 4.14 The generated thrust for each motor.

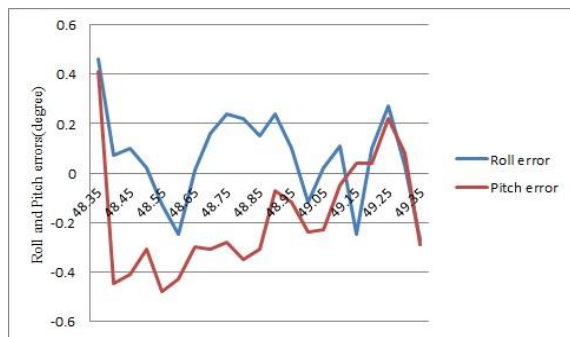


Figure 4.15 Roll and Pitch errors tracking.

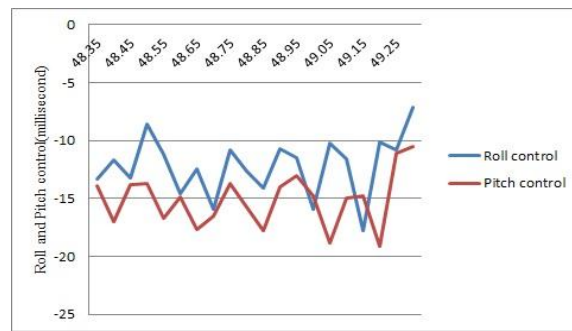


Figure 4.16 Roll and Pitch controls.

4.5 PSO - PID gains optimization

We established intervals for the resulting PID gains achieved using the Z-N method, as shown in Table 4.3 the PSO method then searched for the optimal PID gains depending on the objective function. The search for the optimal PID was conducted offline, meaning that all calculations were performed via simulation, for which the determined gains were verified both using the simulation and experimentally. We used two different objective functions: the first was the sum of the maximum of the overshoot and the Integral Square of Error (ISE), and the second was just the Integral Square of Error.

Table 4.3 Limited PSO search interval based on Z-N derived gains

	Z-N produced gains	Chosen interval for PSO
P	24	[13 30]
I	0.1	[0.01 1.5]
D	0.025	[0.01 1.5]

The PSO needed to determine number of particles and iterative number in order to calculate the PID gains. In our case the number, of particles was set to 15 and the iterative number was set to 20. Figure 4.18 shows changes in the particles' positions during the search for optimized PID gains according to the first objective function. The blue points represent the swarms, the black points represent the local best particles, and the red points represent the global best particles.

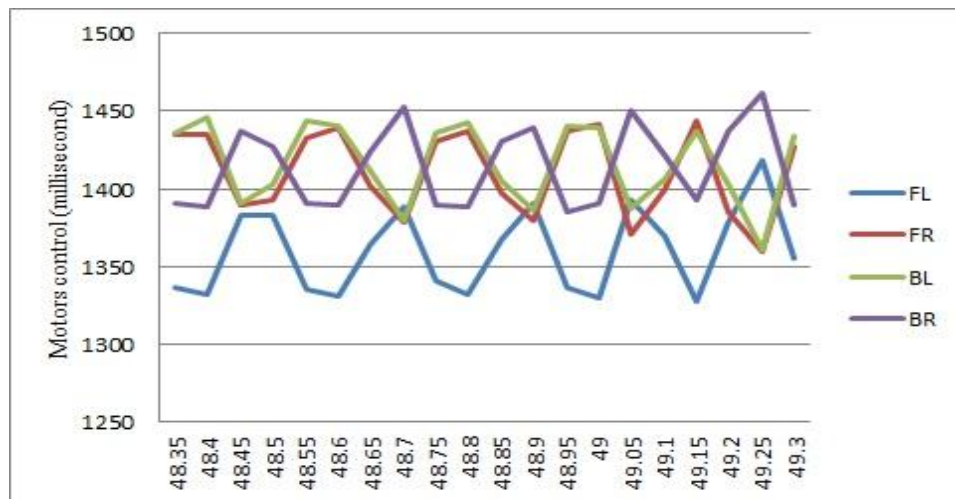


Fig 4.17 control of each motor.

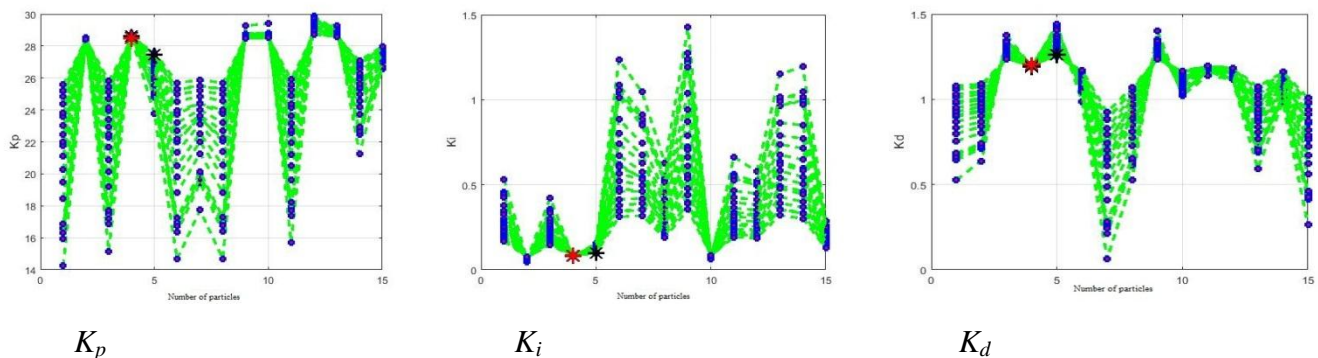


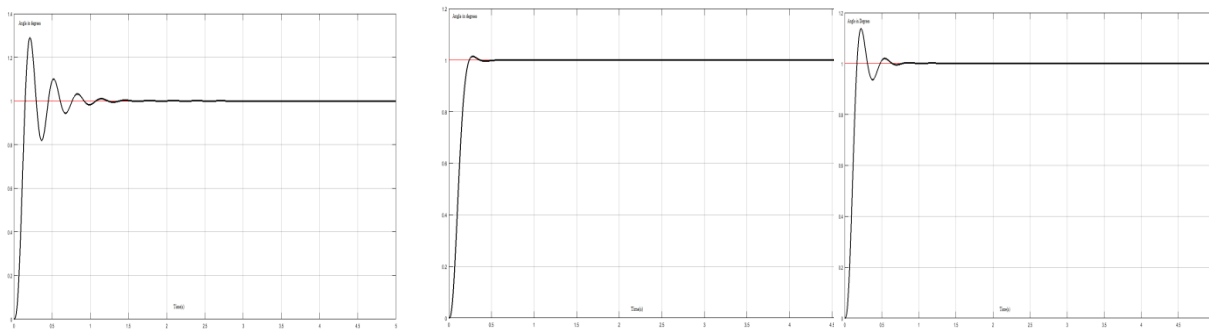
Figure 4.18 Change in particles' position during the search for the optimal PID controller gains.

The calculated PID gains obtained using two objective functions and the Z-N method, are summarized in Table 4.4.

Table 4.4 Summary of PID gains obtained by Z-N tuning and PSO

	Objective function	P	I	D
Z-Ngains		24	0.1	0.025
PSO1	Max(overshoot)+ ISE	28.5	0.0834	1.2
PSO2	ISE	25.69	0.4345	0.4325

The PID gains shown in Table 4.4, were tested in our simulation model, and the step responses for the pitch angles are shown in Figure 4.19. The comparative simulation results are given in Table 4.5, based on ISE, overshoot, and rise time.



With Z-N gains

With PSO1

With PSO2

Figure 4.19 Pitch angles step responses for the three PID gains in simulation.

Table 4.5 Comparison of the Z-N and PSO gains for pitch angle stabilization.

	Rise time(s)	Overshoot(degree)	ISE
Z-N	0.11	0.29	0.087
PSO1	0.132	0.0148	0.086
PSO2	0.107	0.23	0.08

The objective function of PSO2 was only the Integral Square of Error. From Table 4.5, it can be seen that the rise time was the same for PID Z-N and PSO2, given the existing overshoot, but with lower ISE in the latter. It is very clear that the rise time and overshoot in the PID Z-N

case was ameliorated by the PSO2. However, when we added the maximum of the overshoot as a second criterion to the objective function, it resulted in a lower overshoot, but with a smaller rise in ISE.

The PID controllers in Tab 4.4, were tested experimentally, and the plots of the pitch error angle in steady-state for PID Z-N, PID PSO1, and PID PSO2 are shown in Figure 4.20, Figure 4.22, and Figure 4.24, respectively. The pulse responses are shown in Figure 4.21, Figure 4.23, and Figure 4.25, respectively and the comparative results are listed in Table 4.6, based on ISE for steady-state and pulse response conditions.

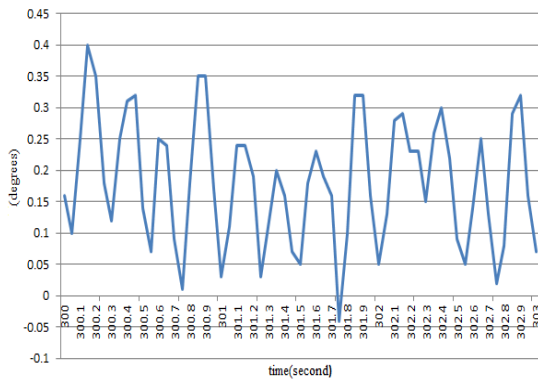


Figure 4.20 Pitch error angle in steady-state for Z-N PID

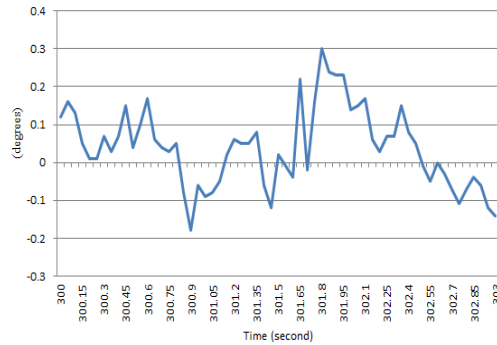


Figure 4.22 Pitch error angle in steady-state for PID PSO1

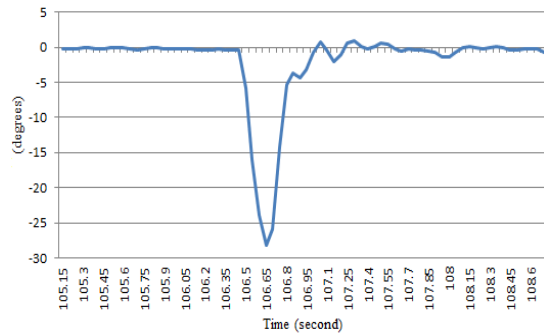


Figure 4.21 Pitch error angle for Z-N PID

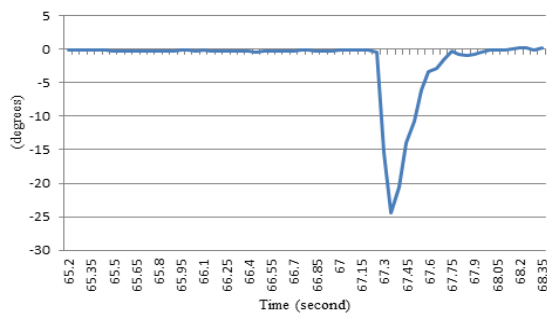


Figure 4.23 Pitch error angle for PID PSO1

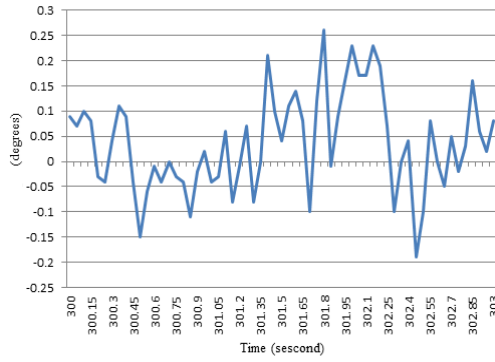


Figure 4.24 Pitch error angle in steady-state for PSO2

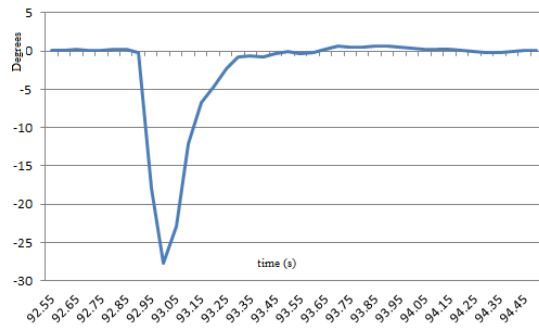


Figure 4.25 Pitch error angle for PSO2

Table 4.6 Experimental comparative results for three optimized PID gains.

	Rise time (s)	Overshoot (degree)	ISE_steady stat	ISE_pulse
Z-N	0.12	1.01	0.0433	65.506
PSO1	0.18	0.25	0.0122	40.554
PSO2	0.13	0.7	0.0107	45.87

The simulation results of the Z-N tuned PID show that the pitch angle error response produced high ISE, high overshoot, and low rise time, but a better PID performance was obtained by applying PSO. Different objective functions provided different PID controller gains. In case when the overshoot was taken into consideration by the objective function, the pitch error response produced the lowest overshoot with a low ISE, but the highest rise time. In the case when the overshoot was not taken into consideration by the objective function, the response produced the lowest ISE with an existing overshoot. The experimental results in Table 4.6, and Figure 4.20 through Figure 4.24 depict the corresponding simulation analyses. However, the Integral Square of Error of PSO1 was lower than PSO2 because the response of the latter had greater overshoot than PSO1. In actual tests, the Quadcopter is a noisy system and we cannot eliminate the gyroscopic torque and Coriolis-centripetal force that were neglected in the simulation results. For roll and the yaw motions, PID controllers were optimized using the same method presented in Figure 4.26, and demonstrated successful results and the error of three angles roll, pitch and yaw around 0.3 degree.

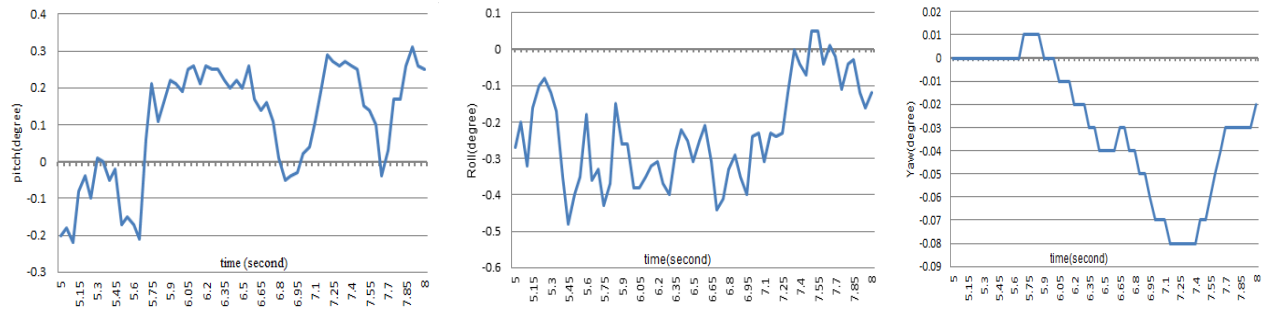


Figure 4.26 errors angles of the roll, pitch and yaw implemented together in experiment.

4.6 CONCLUSION

We provide experimental results of ZN method for PID controllers gains identifications for quadcopter stabilization. ZN tuning with fast way and least required effort to experimentally select the appropriate PID gains which stabilize our quadcopter in real time. To improve the performances is better use new recent optimization method, PSO selected for this reason. The second advantage of the Z-N method found in this study is that it produces PID gains that can be put into intervals, such that the PSO search within an interval by determining the sum of the Integral of Square Error and the maximum of the overshoot as a multi-objective function [19] or a fitness function.

We provide results of optimized PID controllers for attitude stabilization of a Quadcopter using PSO by the constriction coefficient method was never used before for attitude control of a Quadcopter. The main advantage of optimization process that find the optimal solution with scientific method not as manual finding depending on the luck, and let the user of it more confidence with their results. The proposed method was tested both using a MATLAB simulation and experimentally. The simulation and experimental results were also compared with a conventional Z-N tuning PID controller and with two different objective functions that the PSO depends on to search the optimal PID controller gains. The efficiency of PSO was verified through simulation and experimental results in our case.

Chapter 5

Fuzzy logic controller Optimization

5.1 Introduction

This chapter introduces the basic concepts, notation, and basic operations for the fuzzy sets that will be needed in the following chapters. Fuzzy sets as well as their operations will be discussed in this chapter. For this reason, in this chapter we will focus only on fuzzy logic. Since research on fuzzy set theory has been underway for over 30 years now, it is practically impossible to cover all aspects of current developments in this area. The main goal of this chapter is the optimization of a proportional derivative (PD) fuzzy controller using the particle swarm optimization (PSO) technique. More precisely, we conduct a comparative study to assess the performance of the PD fuzzy controller by optimizing its gain and its structural parameters separately, the optimization will be done offline; thus leading to two distinct fuzzy controllers. The two controllers are applied to control a 3DOF PUMA560 robot manipulator.

5.2 Fuzzy set theory

Fuzzy logic was first proposed by Zadeh (1965) and is based on the concept of fuzzy sets [31]. Fuzzy set theory provides a means for representing uncertainty. In general, probability theory is the primary tool for analyzing uncertainty, and assumes that the uncertainty is a random process. However, not all uncertainty is random, and fuzzy set theory is used to model the kind of uncertainty associated with imprecision, vagueness and lack of information.

Conventional set theory distinguishes between those elements that are members of a set and those that are not, there being very clear, or crisp boundaries. Figure 5.1 shows the crisp set 'medium temperature'. Temperatures between 20 and 30°C lie within the crisp set, and have a membership value of one.

The central concept of fuzzy set theory is that the membership function M , like probability theory, can have a value of between 0 and 1 [33] [57]. In Figure 5.2, the membership function has a linear relationship with the x -axis, called the universe of discourse U . This produces a triangular shaped fuzzy set. Fuzzy sets represented by symmetrical triangles are commonly used because they give good results and computation is simple. Other arrangements include non-symmetrical triangles, trapezoids, Gaussian and bell shaped curves [34] [58].

Let the fuzzy set 'medium temperature' be called fuzzy set M . If an element u of the universe of discourse U lies within fuzzy set M , it will have a value of between 0 and 1. This is expressed mathematically as [18][59]:

$$\mu_M(u) \in [0,1] \tag{5.1}$$

When the universe of discourse is discrete and finite, fuzzy set M may be expressed as [11]:

$$M = \sum_{i=1}^n \mu_M(u_i)/u_i \tag{5.2}$$

In equation (5.2) '/' is delimiter. Hence the numerator of each term is the membership value in fuzzy set M associated with the element of the universe indicated in the denominator.

When $n=11$, equation (5.2) can be written as:

$$M = 0/0 + 0/5 + 0/10 + 0.33/15 + 0.67/20 + 1/25 + 0.67/30 + 0.33/35 + 0/40 + 0/45 + 0/50 \tag{5.3}$$

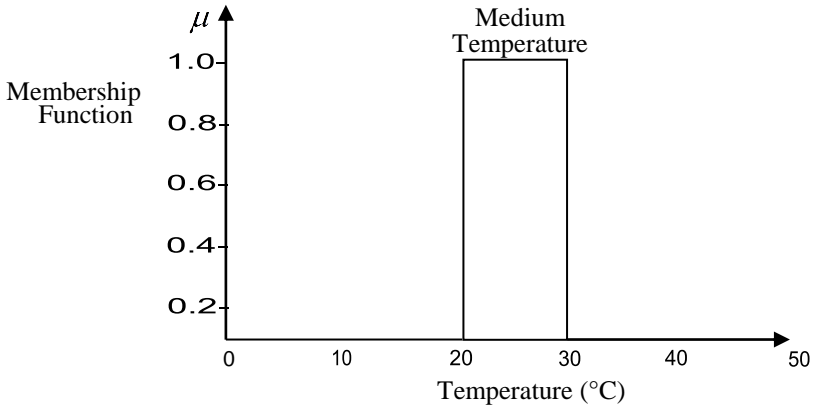


Figure 5.1 Crisp set (medium temperature) [18][59]:.

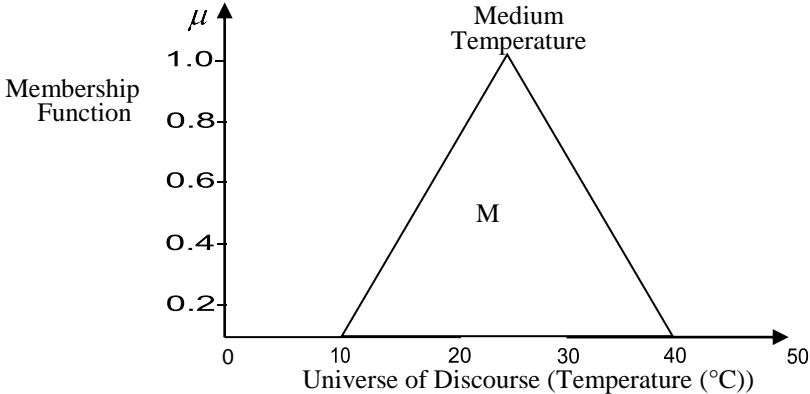


Figure 5.2 Fuzzy set 'medium temperature'[18][59].

5.3 Basic fuzzy set operations

Let A and B be two fuzzy sets within a universe of discourse U with membership functions μ_A and μ_B respectively. The following fuzzy set operations can be defined as [18][59]:

Equality: Two fuzzy sets A and B are equal if they have the same membership function within a universe of discourse U.

$$\mu_A(u) = \mu_B(u) \quad \text{for all } u \in U \quad (5.4)$$

Union: The union of two fuzzy sets A and B corresponds to the Boolean OR function and is given by:

$$\mu_{A \cup B}(u) = \mu_{A+B}(u) = \max\{\mu_A(u), \mu_B(u)\} \quad \text{for all } u \in U \quad (5.5)$$

Intersection: The intersection of two fuzzy sets A and B corresponds to the Boolean AND function and is given by:

$$\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\} \quad \text{for all } u \in U \quad (5.6)$$

Complement: The complement of fuzzy set A corresponds to the Boolean NOT function and is given by:

$$\mu_{\neg A}(u) = 1 - \mu_A(u) \quad \text{for all } u \in U \quad (5.7)$$

Example 5.1[31]

Find the union and intersection of fuzzy set low temperature L and medium temperature M shown in Figure 5.3. Find also the complement of fuzzy set M . using equation (5.2) the fuzzy sets for $n=11$ are:

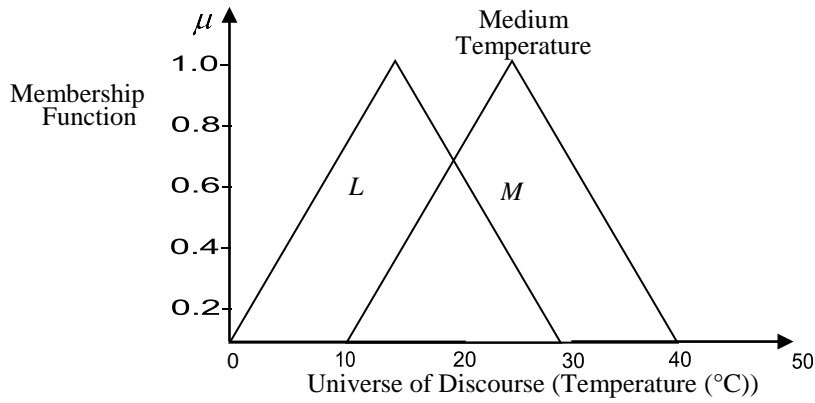


Figure 5.3 Overlapping sets 'low' and 'medium temperature[31]'.

$$\begin{aligned}
 L &= 0/0 + 0.33/5 + 0.67/10 + 1/15 + 0.67/20 + 0.33/25 \\
 &+ 0/30 + 0/35 + \dots + 0/50 \\
 M &= 0/0 + 0/5 + 0/10 + 0.33/15 + 0.67/20 + 1/25 + 0.67/30 \\
 &+ 0.33/35 + 0/40 + \dots + 0/50
 \end{aligned}
 \tag{5.8}$$

a - **Union:** Using equation (5.5)

$$\begin{aligned}
 \mu_{L+M}(u) &= \max(0,0)/0 + \max(0.33,0)/5 + \max(0.67,0)/10 \\
 &+ \max(1,0.33)/15 + \max(0.67,0.67)/20 + \max(0.33,1)/25 \\
 &+ \max(0,0.67)/30 + \max(0,0.33)/35 + \max(0,0)/40 + \dots \\
 &+ \max(0,0)/50
 \end{aligned}
 \tag{5.9}$$

$$\begin{aligned}
 \mu_{L+M}(u) &= 0/0 + 0.33/5 + 0.67/10 + 1/15 + 0.67/20 + 1/25 + 0.67/30 \\
 &+ 0.33/35 + 0/40 + \dots + 0/50
 \end{aligned}
 \tag{5.10}$$

b - **Intersection:** Using equation (5.6) and replacing 'max' by 'min' in equation (5.9) gives:

$$\begin{aligned}
 \mu_{L \cap M}(u) &= 0/0 + 0/5 + 0/10 + 0.33/15 + 0.67/20 + 0.33/25 \\
 &+ 0/30 + \dots + 0/50
 \end{aligned}
 \tag{5.11}$$

Equations (5.10) and (5.11) are shown in Figure 5. 4.

c- **Complement:** Using equation (5.7):

$$\begin{aligned}
 \mu_{-M}(u) &= (1-0)/0 + (1-0)/5 + (1-0)/10 + (1-0.33)/15 \\
 &+ (1-0.67)/20 + (1-1)/25 + (1-0.67)/30 + (1-0.33)/35 \\
 &+ (1-0)/40 + \dots + (1-0)/50
 \end{aligned}
 \tag{5.12}$$

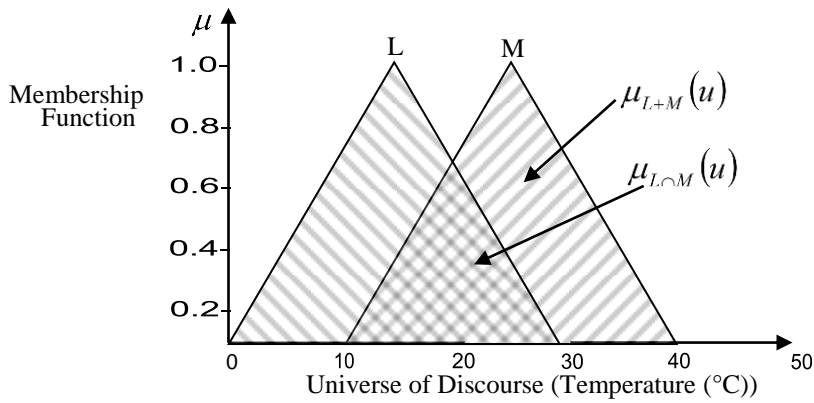


Figure 5.4 'Union' and 'intersection' functions.[31]

5.4 Fuzzy relations

An important aspect of fuzzy logic is the ability to relate sets with different universes of discourse. Consider the relationship [18][59]:

$$\text{IF } L \text{ THEN } M \quad (5.13)$$

In equation (5.13) L is known as the antecedent and M as the consequent. The relationship is denoted by:

$$A = L \times M \quad (5.14)$$

Or:

$$L \times M = \begin{bmatrix} \min\{\mu_L(u_1), \mu_M(v_1)\} & \cdots & \min\{\mu_L(u_1), \mu_M(v_k)\} \\ \min\{\mu_L(u_j), \mu_M(v_1)\} & \cdots & \min\{\mu_L(u_j), \mu_M(v_k)\} \end{bmatrix} \quad (5.15)$$

Where $u_1 \rightarrow u_j$ and $v_1 \rightarrow v_k$ are the discretized universe of discourse. Consider the statement:

$$L \times M = \begin{bmatrix} \min\{\mu_L(u_1), \mu_M(v_1)\} & \cdots & \min\{\mu_L(u_1), \mu_M(v_k)\} \\ \min\{\mu_L(u_j), \mu_M(v_1)\} & \cdots & \min\{\mu_L(u_j), \mu_M(v_k)\} \end{bmatrix} \quad (5.16)$$

Then for the fuzzy sets L and M defined by equation (5.8), for U from 5 to 35 in steps of 5

$$L \times M = \begin{bmatrix} \min(0.33,0) & \cdots & \min(0.33,1) & \cdots & \min(0.33,0.33) \\ \min(0.67,0) & \cdots & \min(0.67,1) & \cdots & \min(0.67,0.33) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \min(0,0) & \cdots & \min(0,1) & \cdots & \min(0,0.33) \end{bmatrix} \quad (5.17)$$

Which gives:

$$L \times M = \begin{bmatrix} 0 & 0 & 0.33 & 0.33 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0.33 & 0.67 & 0.67 & 0.67 & 0.33 \\ 0 & 0 & 0.33 & 0.67 & 1 & 0.67 & 0.33 \\ 0 & 0 & 0.33 & 0.67 & 0.67 & 0.67 & 0.33 \\ 0 & 0 & 0.33 & 0.33 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.18)$$

Several such statements would form a control strategy and would be linked by their union

$$A = A_1 + A_2 + A_3 + \cdots + A_n \quad (5.19)$$

5.5 Fuzzy logic control [31]

The basic structure of a Fuzzy Logic Control (FLC) system is shown in Figure 5.6. Fuzzy logic controller (FLC) which used in this thesis is Mamdani. The main parts are:

5.5.1 The fuzzification process

Fuzzification is the process of mapping inputs to the FLC into fuzzy set membership values in the various input universes of discourse. Decisions need to be made regarding [31] [18][59]:

- (a) Number of inputs
- (b) Size of universes of discourse
- (c) Number and shape of fuzzy sets.

A FLC that emulates a PD controller will be required to minimize the error $e(t)$ and the rate of change of error de/dt , or ce .

The size of the universes of discourse will depend upon the expected range (usually up to the saturation level) of the input variables. Assume for the system about to be considered that e has a range of ± 6 and ce a range of ± 1 .

The number and shape of fuzzy sets in a particular universe of discourse is a trade-off between precision of control action and real-time computational complexity. In this example, seven triangular sets will be used.

Each set is given a linguistic label to identify it, such as Positive Big (PB), Positive Medium (PM), Positive Small (PS), About Zero (Z), Negative Small (NS), Negative Medium (NM) and Negative Big (NB). The seven set fuzzy input windows for e and ce are shown in Figure 5.7. If at a particular instant, $e(t) = 2.5$ and $de/dt = ce = -0.2$, then, from Figure 5.7, the input fuzzy set membership values are:

$$\begin{aligned} \mu_{PS}(e) &= 0.7 & \mu_{PM}(e) &= 0.4 \\ \mu_{NS}(ce) &= 0.6 & \mu_z(ce) &= 0.3 \end{aligned} \quad (5.20)$$

5.5.2 The fuzzy rulebase

The fuzzy rule base consists of a set of antecedent-consequent linguistic rules of the form [18][59]:

$$IF \ e \text{ is } PS \ AND \ ce \text{ is } NS \ THEN \ u \text{ is } PS \quad (5.21)$$

This style of fuzzy conditional statement is often called a 'Mamdani'-type rule, after Mamdani (1976) who first used it in a fuzzy rulebase to control steam plant.

The rulebase is constructed using a priori knowledge from either one or all of the following sources:

- (a) Physical laws that govern the plant dynamics
- (b) Data from existing controllers.
- (c) Imprecise heuristic knowledge obtained from experienced experts.

If (c) above is used, then knowledge of the plant mathematical model is not required. The two seven set fuzzy input windows shown in Figure 5.7 gives a possible 7 x 7 set of control rules of the form given in equation (5.21). It is convenient to tabulate the two-dimensional rulebase as shown in Figure 5.8.

5.5.3 Fuzzy inference

Figure 5.8 assumes that the output contains seven fuzzy sets with the same linguistic labels as the input fuzzy sets[31] [18][59]. If the universe of discourse for the control signal $u(t)$ is ± 9 , then the output window is as shown in Figure 5.9.

Assume that a certain rule in the rulebase is given by equation (5.22)

$$IF\ e\ is\ A\ AND\ ce\ is\ B\ THEN\ u = C \tag{5.22}$$

From equation (5.5) the Boolean OR function becomes the fuzzy max operation, and from equation (5.6) the Boolean AND function becomes the fuzzy min operation. Hence equation (5.22) can be written as

$$\mu_C(u) = \max[\min(\mu_A(e), \mu_B(ce))] \tag{5.23}$$

Equation (5.23) is referred to as the max-min inference process or max-min fuzzy reasoning.

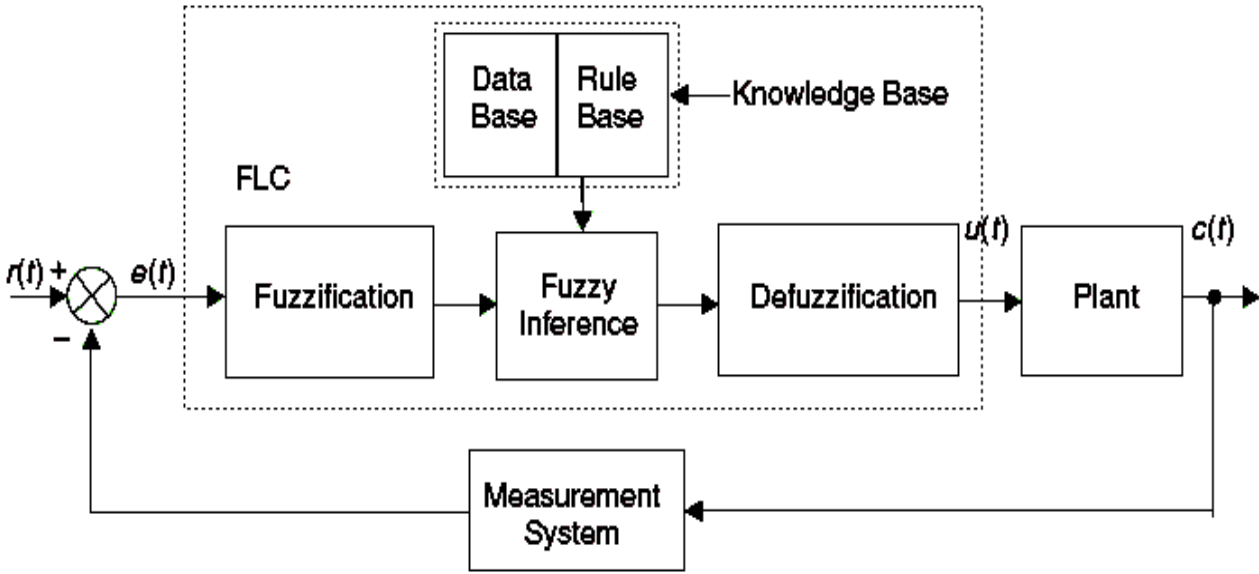


Figure 5.6 Fuzzy Logic Control[18][59]

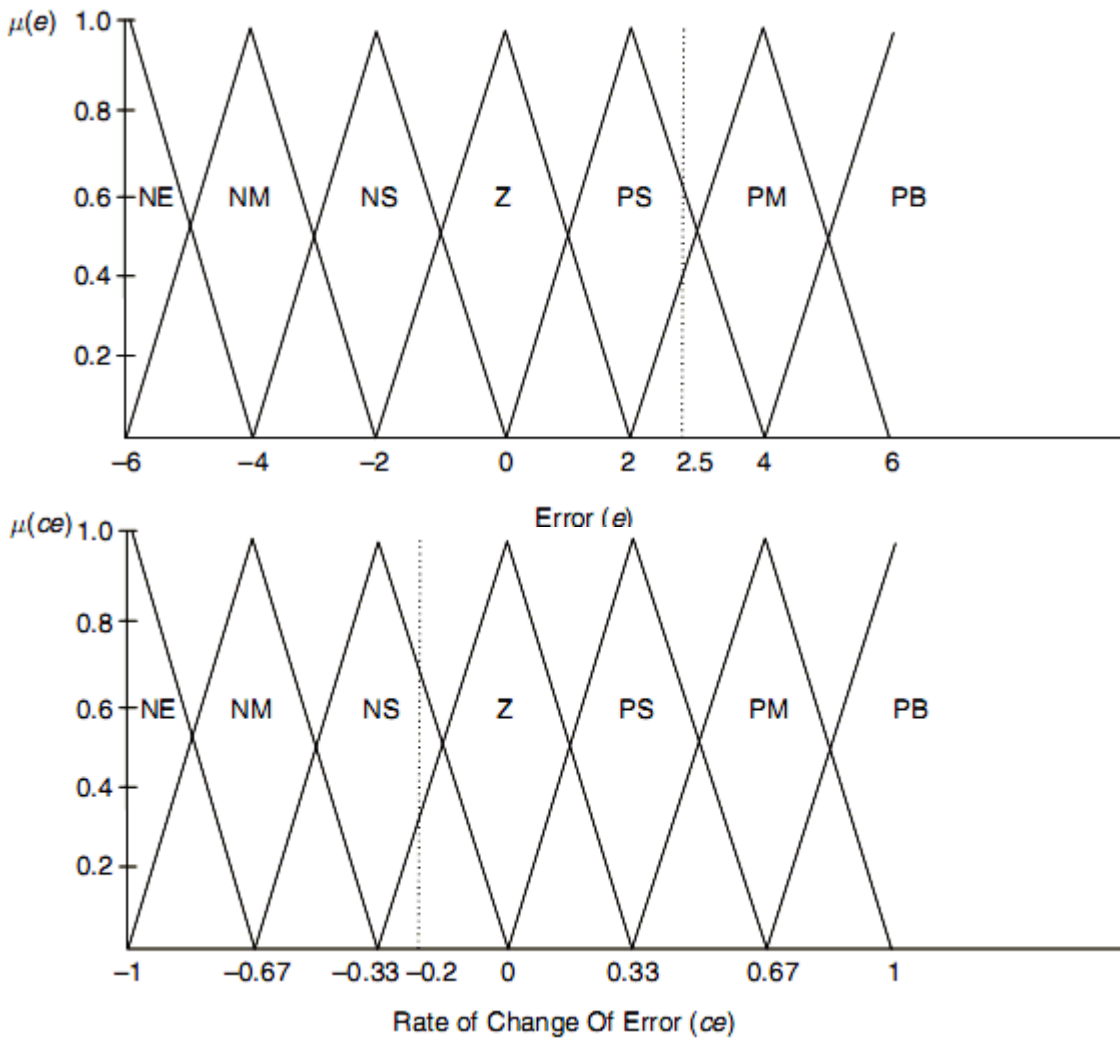


Figure 5.7 Seven set fuzzy input windows for error (e) and rate of change of error (ce). [18][59]:

$ce \backslash e$	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	Z	PM	PB
NM	NB	NB	NB	NM	PS	PM	PB
NS	NB	NB	NM	NS	PS	PM	PB
Z	NB	NM	NS	Z	PS	PM	PB
PS	NB	NM	NS	PS	PM	PB	PB
PM	NB	NM	NS	PM	PB	PB	PB
PB	NB	NM	Z	PM	PB	PB	PB

Table 5.1 Tabular structure of a linguistic fuzzy rulebase[18][59].

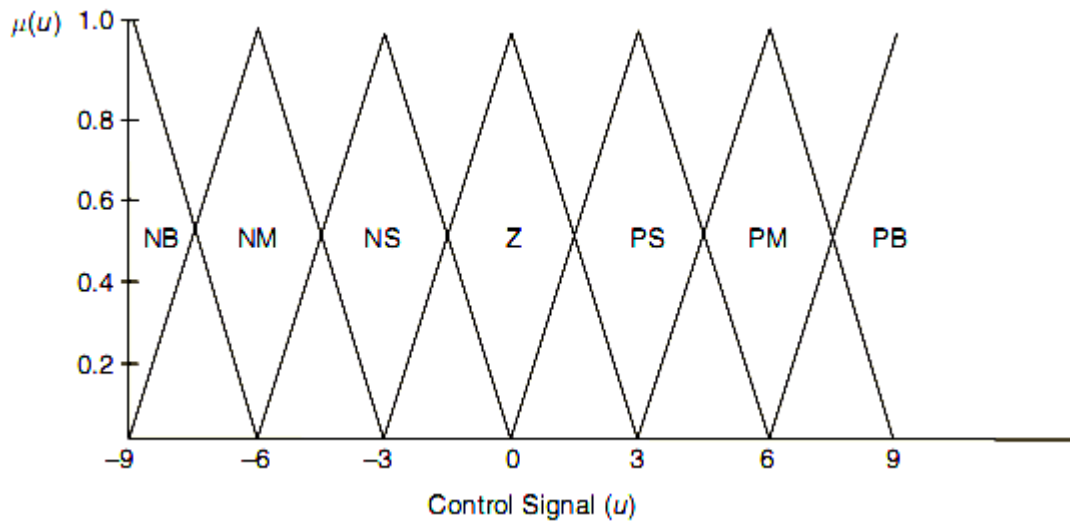


Figure 5.8 Seven set fuzzy output window for control signal (u).[18][59]

In Figure 5.7 and equation (5.20) the fuzzy sets that were 'hit' in the error input window when $e(t) = 2.5$ were PS and PM. In the rate of change input window when $ce = -0.2$, the fuzzy sets to be 'hit' were NS and Z. From Figure 5.8, the relevant rules that correspond to these 'hits' are:

$$\begin{aligned}
 &\dots \text{OR IF } e \text{ is PS AND } ce \text{ is NS} \\
 &\quad \text{OR IF } e \text{ is PS AND } ce \text{ is Z} \\
 &\quad \text{THEN } u = \text{PS}
 \end{aligned} \tag{5.24}$$

$$\begin{aligned}
 &\dots \text{OR IF } e \text{ is PM AND } ce \text{ is NS} \\
 &\quad \text{OR IF } e \text{ is PM AND } ce \text{ is Z} \\
 &\quad \text{THEN } u = \text{PM}
 \end{aligned} \tag{5.25}$$

Applying the max-min inference process to equation (5.24)

$$\mu_{PS}(u) = \max[\min(\mu_{PS}(e), \mu_{NS}(ce)), \min(\mu_{PS}(e), \mu_Z(ce))] \tag{5.26}$$

Inserting values from equation (5.20)

$$\begin{aligned}
 \mu_{PS}(u) &= \max[\min(0.7, 0.6), \min(0.7, 0.3)] \\
 &= \max[0.6, 0.3] = 0.6
 \end{aligned} \tag{5.27}$$

Applying the max-min inference process to equation (5.25)

$$\mu_{PM}(u) = \max[\min(\mu_{PM}(e), \mu_{NS}(ce)), \min(\mu_{PM}(e), \mu_Z(ce))] \quad (5.28)$$

Inserting values from equation (5.20)

$$\begin{aligned} \mu_{PS}(u) &= \max[\min(0.4, 0.6), \min(0.4, 0.3)] \\ &= \max[0.4, 0.3] = 0.4 \end{aligned} \quad (5.29)$$

Fuzzy inference is therefore the process of mapping membership values from the input windows, through the rulebase, to the output window(s).

5.5.4 The defuzzification

Defuzzification is the procedure for mapping from a set of inferred fuzzy control signals contained within a fuzzy output window to a non-fuzzy (crisp) control signal. The centre of area [60, 32, 59] [18] method is the most well known defuzzification technique, which in linguistic terms can be expressed as:

$$\text{Crisp control signal} = \frac{\text{Sum of first moments of area}}{\text{Sum of areas}} \quad (5.30)$$

For a continuous system, equation (5.30) becomes:

$$u(t) = \frac{\int u\mu(u)du}{\int \mu(u)du} \quad (5.31)$$

Or alternatively, for a discrete system, equation (5.30) can be expressed as:

$$u(kT) = \frac{\sum_{i=1}^n u_i \mu(u_i)}{\sum_{i=1}^n \mu(u_i)} \quad (5.32)$$

For the case when $e(t) = 2.5$ and $ce = -0.2$, as a result of the max-min inference process (equations (5.27) and (5.29)), the fuzzy output window in Figure 5.9 is 'clipped', and takes the form shown in Figure 5.10.

From Figure 5.10, using the equation for the area of a trapezoid:

$$\begin{aligned} Area_{PS} &= \frac{0.6(6+2.4)}{2} = 2.52 \\ Area_{PS} &= \frac{0.2(6+3.6)}{2} = 0.96 \end{aligned} \quad (5.33)$$

From equation (5.30)

$$u(t) = \frac{(2.52 \times 3) + (0.96 \times 6)}{2.52 + 0.96} = 3.83 \quad (5.34)$$

Hence, for given error of 5.5, and a rate of change of error of -0.2, the control signal from the fuzzy controller is 3.83.

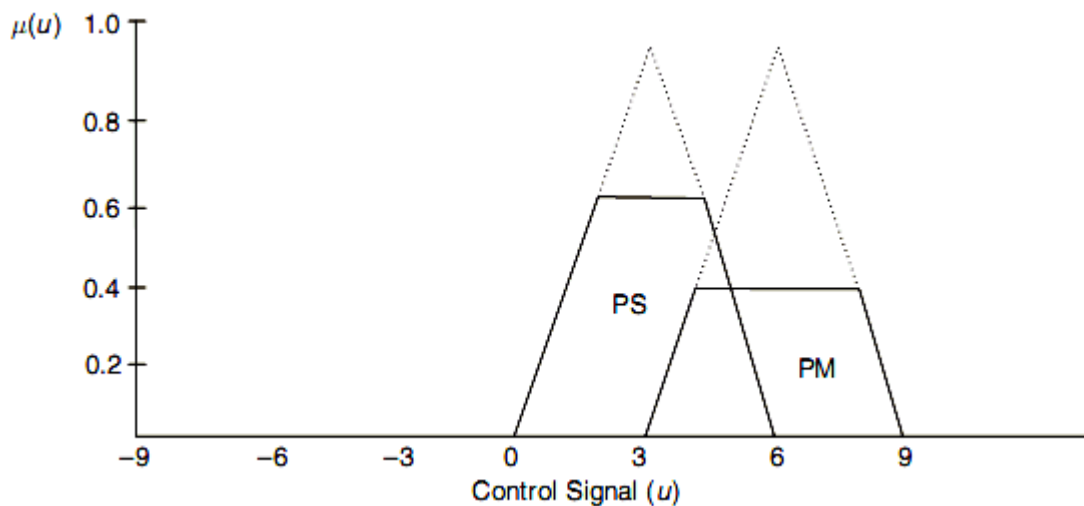


Figure 5.9 Clipped fuzzy output window due to fuzzy inference [31][18][59].

Example 5.2 [31]

For the input and output fuzzy windows given in Figure 5.7 and 5.9, together with the fuzzy rulebase shown in Figure 5.8, determine:

- (a) The membership values of the input windows e and ce .
- (b) The max-min fuzzy inference equations.
- (c) The crisp control signal $u(t)$.

when $e = -3$ and $ce = 0$

Solution

(a) When $e = -3$ and $ce = 0.3$ are mapped onto the input fuzzy windows, they are referred to as fuzzy singletons. From Figure 5.7

$$e = -3 \quad \mu_{NS}(e) = 0.5 \quad \mu_{NM}(e) = 0.5 \quad (5.35)$$

$ce = 0.3$, using similar triangles:

$$\frac{1}{0.33} = \frac{\mu_Z(ce)}{(0.33 - 0.3)} \quad (5.36)$$

$$\mu_Z(ce) = 0.09$$

And

$$\frac{1}{0.33} = \frac{\mu_{PS}(ce)}{0.3}$$

$$\mu_{PS}(ce) = 0.91 \quad (5.37)$$

(b) The rules that are 'hit' in the rulebase in Figure 5.8 are

$$\begin{aligned} &\dots \text{OR IF } e \text{ is NS AND } ce \text{ is Z} \\ &\quad \text{OR IF } e \text{ is NS AND } ce \text{ is PS} \\ &\quad \text{THEN } u = \text{NS} \end{aligned} \quad (5.38)$$

$$\begin{aligned} &\dots \text{OR IF } e \text{ is NM AND } ce \text{ is Z} \\ &\quad \text{OR IF } e \text{ is NM AND } ce \text{ is PS} \\ &\quad \text{THEN } u = \text{NM} \end{aligned} \quad (5.39)$$

Applying max-min inference to equation (5.38):

$$\mu_{NS}(u) = \max[\min(\mu_{NS}(e), \mu_Z(ce)), \min(\mu_{NS}(e), \mu_{PS}(ce))] \quad (5.40)$$

Inserting values into (5.40):

$$\begin{aligned} \mu_{NS}(u) &= \max[\min(0.5, 0.09), \min(0.5, 0.91)] \\ &= \max[0.09, 0.5] = 0.5 \end{aligned} \quad (5.41)$$

and similarly with equation (5.39)

$$\begin{aligned} \mu_{NM}(u) &= \max[\min(\mu_{NM}(e), \mu_Z(ce)), \min(\mu_{NM}(e), \mu_{PS}(ce))] \\ \mu_{PS}(u) &= \max[\min(0.5, 0.09), \min(0.5, 0.91)] \\ &= \max[0.09, 0.5] = 0.5 \end{aligned} \quad (5.42)$$

Using equations (5.41) and (5.42) to 'clip' the output window in Figure 5.9, the output window is now as illustrated in Figure 5.11.

(c) Due to the symmetry of the output window in Figure 5.11, from observation, the crisp control signal is:

$$u(t) = -4,5$$

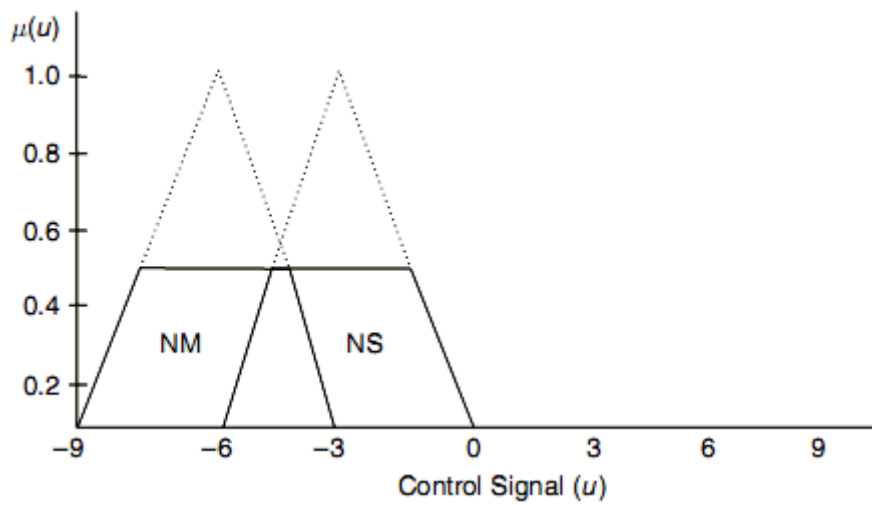


Figure 5.10 Fuzzy output window for Example 5.2[31][18][59].

5.6 Fuzzy control of PUMA560 with 3DOF

In Figure 5.11 we show the structure FLC type-1 of PUM560 with 3DOF, the regulator which we use is five classes, do mean has 25 rule bases, the rule base table in Table 5.2 and in Figure 5.12 fuzzy sets for error and change error and out put of control T . All the gains of type-1 fuzzy controller we do tuning until get good positions tracking with lower error in ideal case.

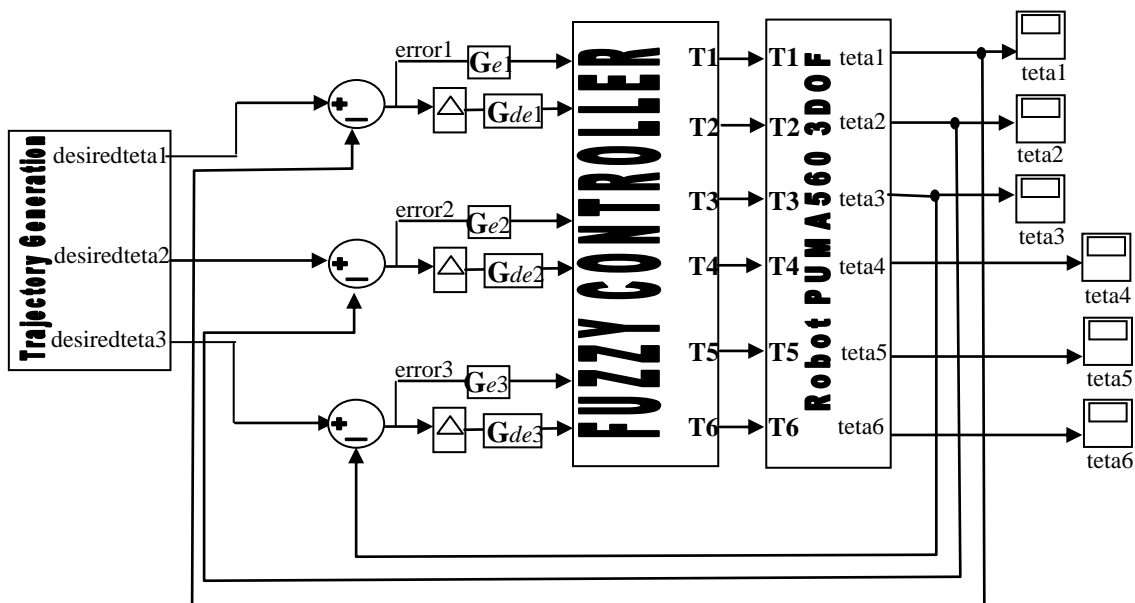


Figure 5.11 Structure type-1 FLC of PUMA560 3DOF.

		Velocity error				
Position error		LN	SN	Ze	SP	LP
	LN	LN	LN	LN	SN	Ze
	SN	LN	LN	SN	Ze	SP
	Ze	LN	SN	Ze	SP	LP
	SP	SN	Ze	SP	LP	LP
	LP	Ze	SP	LP	LP	LP

Table 5.2 Rule Base table[30]

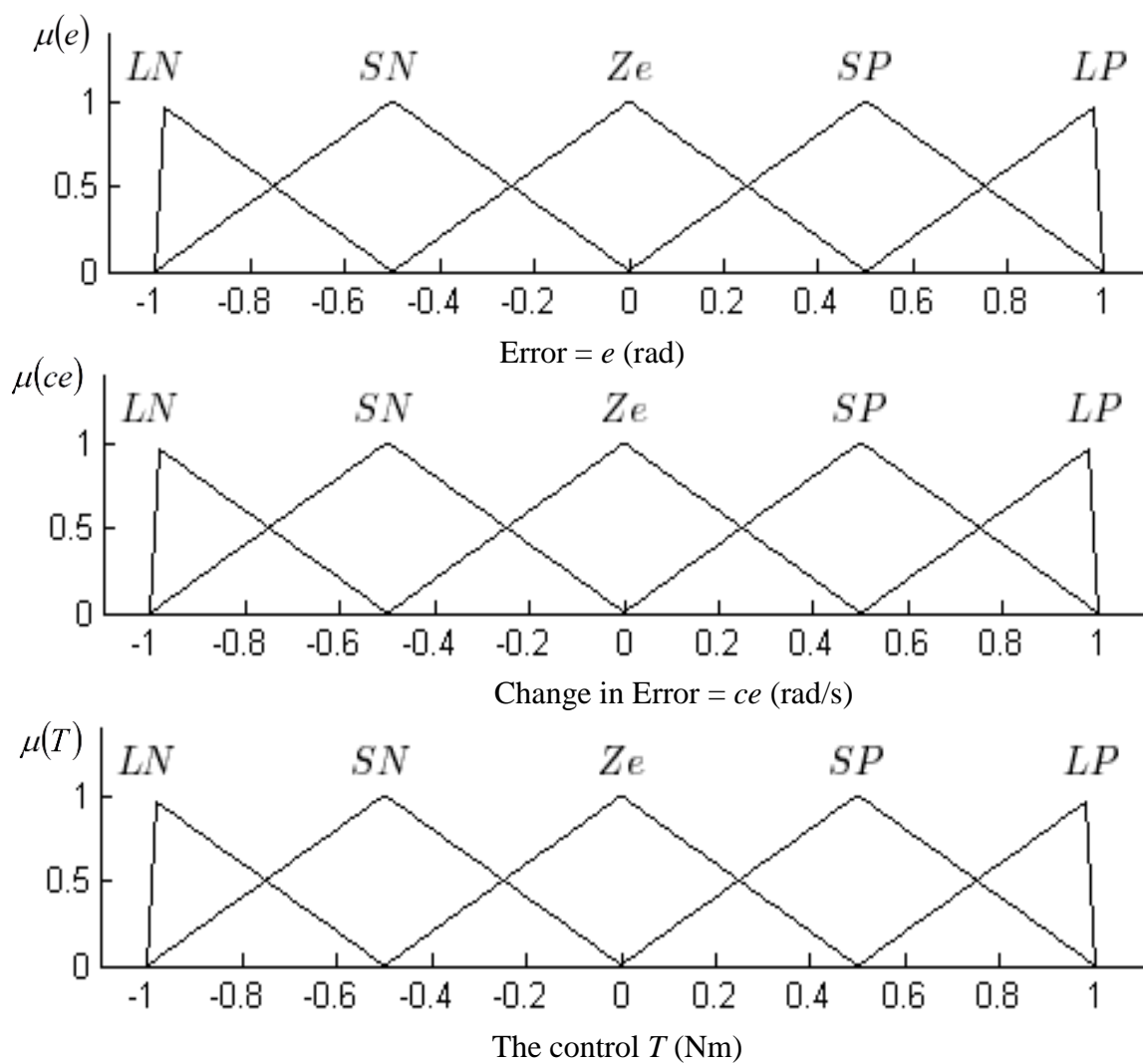


Figure 5.12 Fuzzy set for each articulation.

5.7 Optimized Fuzzy PD controller

The general PSO-based fuzzy PD control scheme [61] is depicted in Figure 5.13, below.

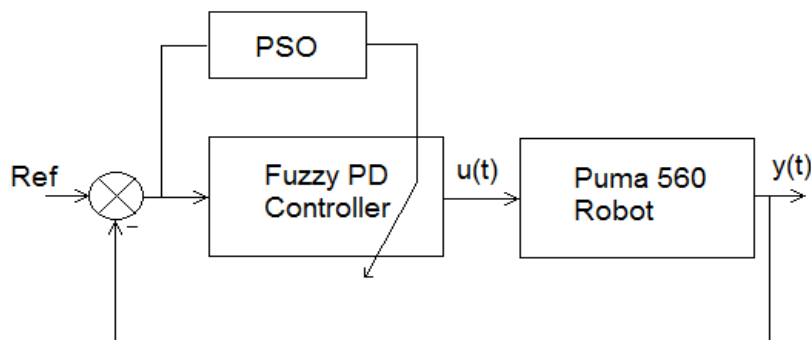


Figure 5.13 General PSO scheme for Fuzzy PD with manipulator robot [61].

The specific details on the PSO algorithm used for PUMA560 control are given in the next section. As for the fuzzy PD controller[7], two optimization strategies are employed: one for the gain as depicted in Figure 5.14, [31][32][60][59][62][57] and the other for the structure as depicted in Figure 5.15, [33][34][58][63][64]. Note that in the case of the gain optimization, the membership function of the fuzzy controller is symmetric. On the other hand, in the case of the structure optimization of the controller, the membership function is non-symmetric.

As usual the fuzzy controller consists of three main parts: i) the fuzzification ii) inference iii) defuzzification.

The fuzzification has as main role to change the crisp values to fuzzy values and determine the membership degrees.

The inference is the process of formulating the mapping from a given input to an output using fuzzy logic. It works on the rule base shown in Table 5.3 below:

		Derivative error				
		NL	NS	Ze	PS	PL
Position error	NL	NL	NL	NL	NS	Ze
	NS	NL	NL	NS	Ze	PS
	Ze	NL	NS	Ze	PS	PL
	PS	NS	Ze	PS	PL	PL
	PL	Ze	PS	PL	PL	PL

Table 5.3 Rule Base table; Negative Large (NL), negative Small (NS), zero (Ze), positive small (PS) and positive Large (PL)[30].

Here, the Mamdani fuzzy inference system [25][26][18] was used in the controller even though one can also use the max-min inference process. The defuzzification is the process where the fuzzy output from the inference stage is mapped into a crisp or real value. There are many methods for defuzzification in the literature. In this work the Centroid of area is chosen [25][26][18].

Note that in our case, as we have chosen to employ a PD controller, we have two gains K_p and K_d . The position errors and their derivative are quantized into five sets as shown in Figure 6, 7 and 8, represented by a set of linguistic variables with triangle membership function with universes of discourse in [-1,1].

In the case of the gain optimization, the three parameters K_p , K_d and K_{out} are optimized as shown in Figure 5.14.

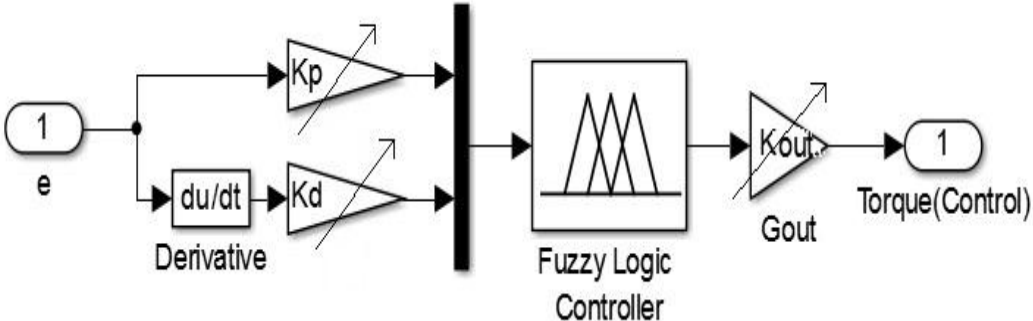


Figure 5.14 Gain optimization fuzzy controller [28] [30]

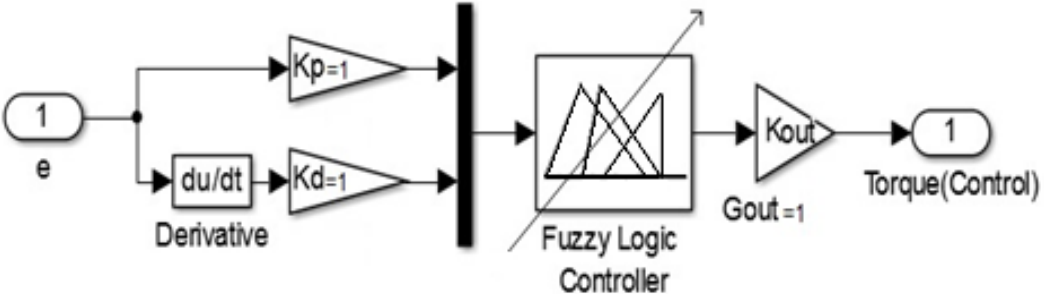


Figure 5.15 Structural parameter optimization Fuzzy controller[33][34]

In the case of the structure optimization, the membership function parameters are optimized while the gains K_p , and K_d are kept fixed at 1. Note also that the fuzzy sets in universal range are not symmetric. In this case we will optimize the range for two inputs and one output each one having its min and max value.

When I was do structure optimization of fuzzy controller, I was wondering witch better reprogram fuzzy controller with my code or just understand code of matlab tool box

fuzzy controller for doing optimization. I think understanding the code of matlab tool box fuzzy controller better and then optimize it to benefit from original matlab code fuzzy controller, the first step we should know mathematical triangle membership function Figure 5.16 from each input error or its derivative calculate the membership function degree equation(6) is very clear that triangle membership function needs three parameters(a,b,c) see Figure 5.16.

$$\mu = \text{trimf}(a,b,c) = \max(\min((x-a)/(b-a), (c-x)/(c-b)), 0) \quad (6)$$

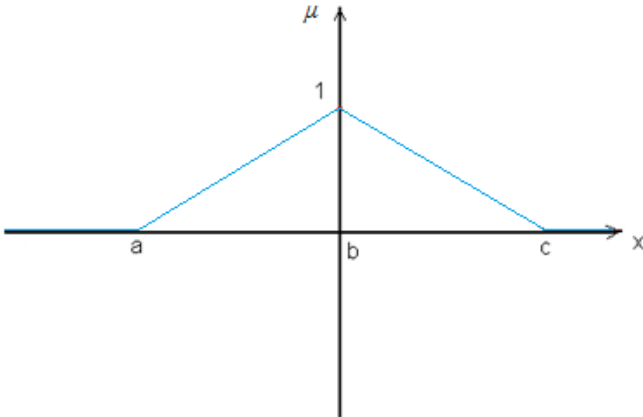


Figure 5.16 Triangle membership function.

Second step after take a look at fis file it should be declared in matlab workspace contain all needed parameter to generate fuzzy controller with matlab tool box. I was found the parameters for standard and symmetric fuzzy sets triangle in gains optimization case its code in first column in Table 5.4 below to get fuzzy sets like in Figure 5.17, and 5.18 and 5.19.

Table 5.4 parameters of fuzzy controller to optimize.

Standard and Symmetric fuzzy sets	To do parametric optimization
[Input1] it mean for error	[Input1] it mean for error
Name='e'	Name='e'
Range=[-1 1]	Range=[-1 1]
MF1='LN': 'trimf', [-1 -1 -0.5]	MF1='LN': 'trimf', [A A B]
MF2='SN': 'trimf', [-1 -0.5 0]	MF2='SN': 'trimf', [A B 0]
MF3='LP': 'trimf', [0.5 1 1]	MF3='LP': 'trimf', [C D D]
MF4='Ze': 'trimf', [-0.5 0 0.5]	MF4='Ze': 'trimf', [B 0 C]
MF5='SP': 'trimf', [0 0.5 1]	MF5='SP': 'trimf', [0 C D]
trimf name of the triangle function in matlab fuzzy tool box Figure 5.16	Where: $A < B < 0$ and $0 < C < D$

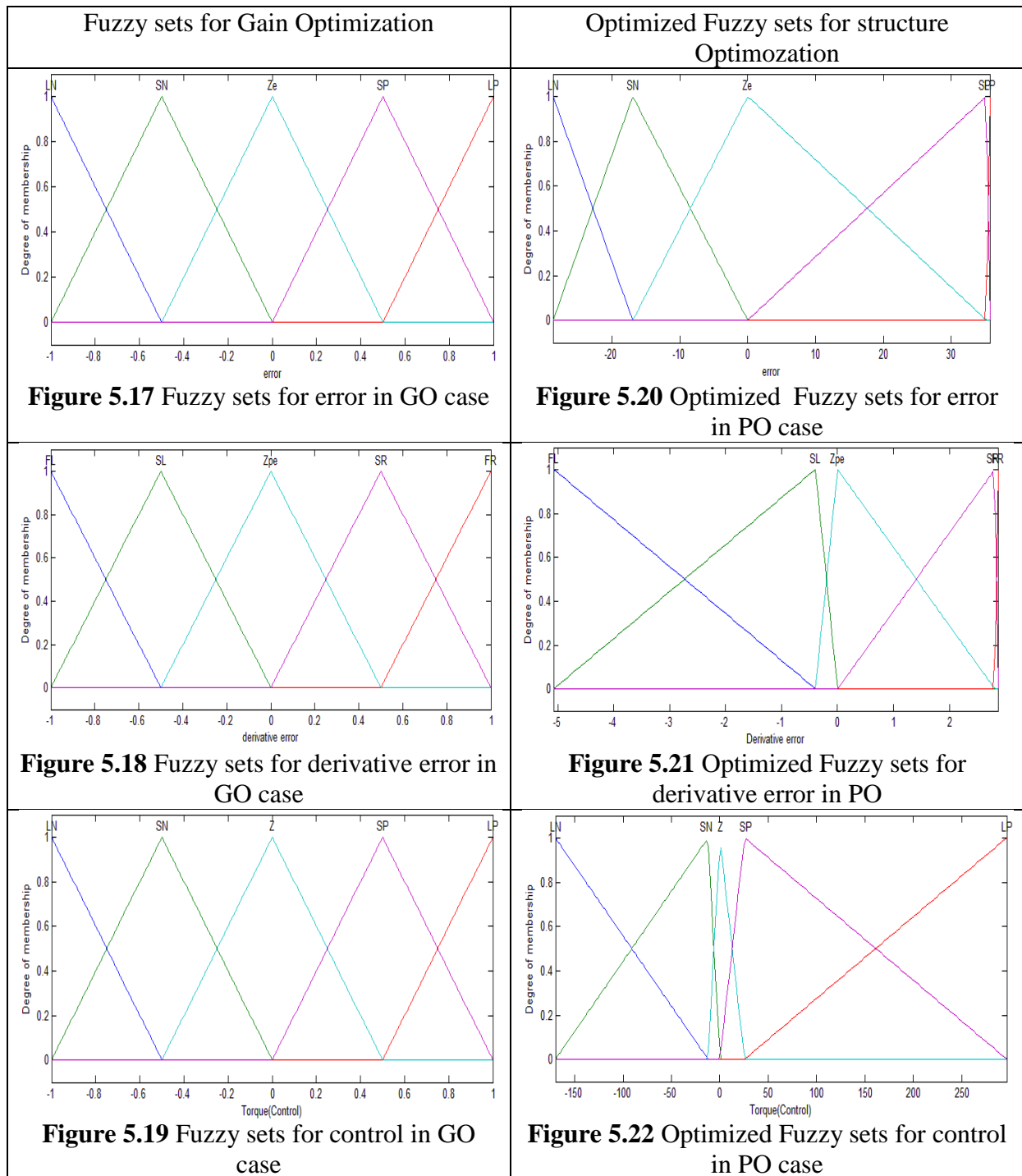
And to do structural optimization, after deep thinking to find solution how to change the values of this parameters to find another optimal parameters by using particle swarm optimization, that we need four parameters (A,B,C,D) and their code it is written in Table 5.4 second column, our controller has two inputs(error and its derivative) and one output then to get optimized controller in its parameters we need $4 \times 3 =$ twelve parameters, see that in gains optimization all parameters(A,B,C,D) are same see Table 5.5, it means same membership function for inputs and output, and all have the same shape and all are symmetric and there plot in Figure 5.17, and 5.18 and 5.19, but in parametric optimization case for input1(error), input2(derivative error) and output(control) each one has its optimal membership function with its range see Table 5.6 and its shape and all are not symmetric, the optimized parameters (A,B,C,D) are found by PSO in table 5.6 there plot in Figure 5.20, and 5.21 and 5.22.

	A	B	C	D
Input1(error)	-1	-0.5	0.5	1
Input2(derivative error)	-1	-0.5	0.5	1
Output(control)	-1	-0.5	0.5	1

Table 5.5 parameters for standard and symmetric fuzzy sets for gains optimization

	A	B	C	D
Input1(error)	-28.70	-16.96	35.19	35.70
Input2(derivative error)	-5.075	-0.40	2.81	2.87
Output(control)	-169.9	-12.26	25.43	296.66

Table 5.6 optimized parameters for structure optimization



5.8 PARTICLE SWARM OPTIMIZATION

In this section, we explain the PSO algorithm employed in the optimization of the aforementioned controllers. The PSO is a population based method for finding an optimal solution to an objective function. The so-called swarm consists of “ n ” particles, and the position of each particle stands for the potential solution in space. The particles change its condition according to the following three principles:

- control its inertia,
- control the position of the particles to its most optimal position,

- Control of the velocity of the particles

The position of each particle in the swarm is affected by the most optimal position based on its individual experience and on the global best position of the particle around its neighborhood. There are several algorithms of the PSO [4][5][6]. In this work integral of square error (ISE) is fitness function (objective function) and the constriction coefficient [4][5] method it was selected whereby the speed and position of each particle change according to the following equality:

$$v_{ij}(t+1) = X \left[v_{ij}(t) + \phi_1 \left(y_{ij}(t) - x_{ij}(t) \right) + \phi_2 \left(\hat{y}_j - x_{ij}(t) \right) \right] \quad (5.43)$$

c_1 and c_2 are the acceleration coefficients, r_1 and r_2 random values in [0 1]

5.9 SIMULATION RESULTS

We have used a cycloid trajectory test Figure 5.23. The different articulation movements, respectively from position $\{-50^\circ, -135^\circ, 135^\circ\}$ to the position $\{45^\circ, -85^\circ, 30^\circ\}$ with three end time of movement equal to $\{1.5, 4, 8\}$ seconds then will have three cycloid trajectory are differ in simulation time. The mathematical description of the trajectory is given by [1]:

$$\theta_{di} = \begin{cases} \theta_{di}(0) + 3 \times \left(\frac{t}{t_{end}} \right)^2 - 2 \times \left(\frac{t}{t_{end}} \right)^3 & \text{for } 0 \leq t \leq t_{end} \\ \theta_{di}(t_{end}) & \text{for } t_{end} \geq t \end{cases} \quad (5.44)$$

The Integral of Square Error (ISE) is used as performance indices in comparative between manual tuning and PSO tuning fuzzy controller see (5.45).

$$ISE = \int_0^{t_{end}} [e(t)]^2 dt \quad (9) \quad (5.45)$$

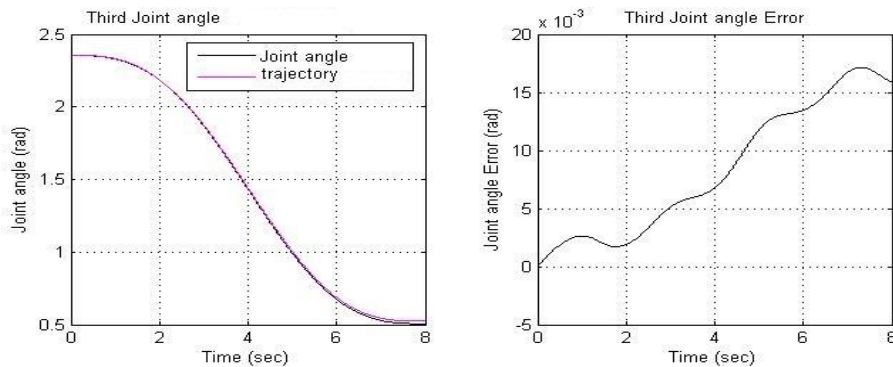


Figure 5.23. Third joint response in manual gain tuning with disturbance and its tracking joint error.

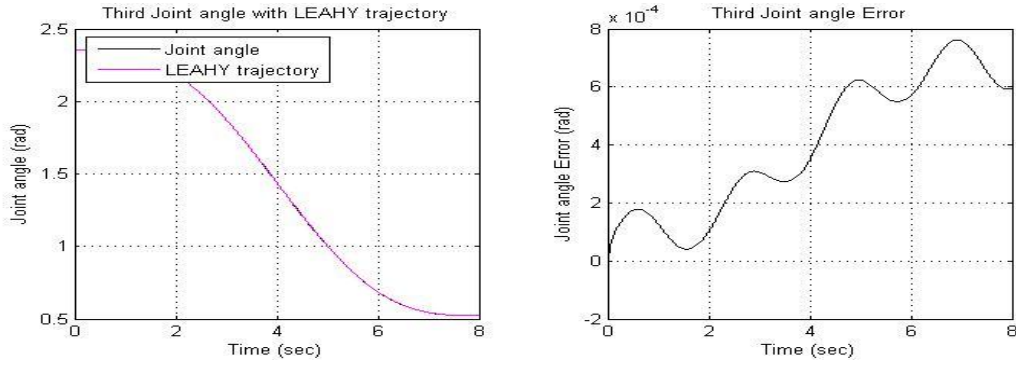


Figure 5.24. Third joint response in PSO gain tuning with disturbance and its tracking joint error.

Gains manual tuning fuzzy PD are chosen from manual tuning until get good positions with lower error and lower control energy and with smooth control signal without chattering in ideal case.

We have use noise (disturbance) for robustness test of controller to do comparison between manual tuning fuzzy controller and optimized fuzzy controller by use PSO, noise (disturbance) torque $d = \sin(3t) + 1$.

The simulation results are carried out to show comparisons in the system's response in a feedback controller when using a gains manual tuning fuzzy controller or PSO fuzzy controller. We started by manually tuning the gains of the fuzzy controller in ideal conditions without any kind of noise, we have shown ISE three joint angle in Table 5.7, in Figure 5.25, control signal or torque responses in ideal case (without noise) and with noise (disturbance) for gains manual tuning fuzzy controller, Figure 5.26, third joint control with and without noise PSO gains optimization. Comparison in terms of integral squared joint errors in ideal(without noise) and with noise cases $d = \sin(3t) + 1$ is shown in Table 5.7 last column is percentage of ISE integral square joint error for gains manual tuning fuzzy controller in noise case ($ISE_{MT}\%$) then ISE gains optimization (ISE_{GO}) disturbance case its calculated as(5.46):

$$ISE_{MT}\% = (ISE_{MT} \times 100) / (ISE_{MT} + ISE_{GO}) \quad (5.46)$$

Table 5.7 ISE for manual (*MT*) and PSO gains optimization (*GO*), without and with noise $d = \sin(3t) + 1$.

Joint	ISE Manual Tuning		ISE PSO GO		ISE_MT%	
	without noise	with noise	without noise	with noise	in noise	with noise
J1	4.36E-07	6.39E-06	9.19E-09	1.6E-07	97.56	
J2	8.35E-04	9.45E-04	1.97E-05	2.214E-05	97.71	
J3	5.40E-04	8.11E-04	1.19E-06	1.7E-06	99.79	

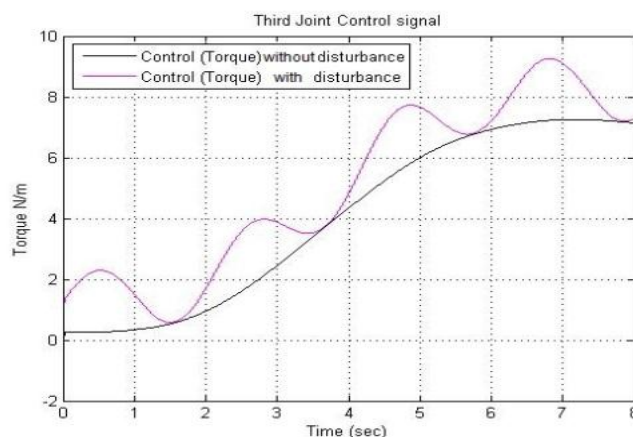


Figure 5.25. Third joint control with and without disturbance manual gain tuning.

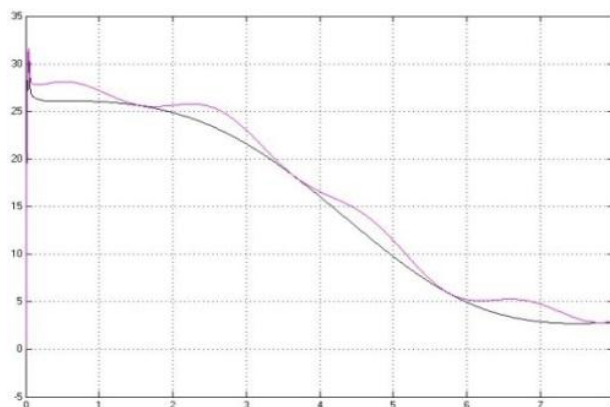


Figure 5.26. Third joint control with and without disturbance PSO gain tuning.

Table 5.8 represent comparison in terms of integral squared joint errors in ideal (without noise) and with noise (disturbance) between Gains and parametric optimization for Fuzzy PD controller, before last column in Table 5.8 percentage of ISE integral square joint error for Gains optimization (GO) fuzzy controller in with noise (disturbance) case (ISE_GO%) then

ISE_PO in with noise (disturbance) case its calculated as (5.47), at last column we have add same term but in ideal (without noise) case:

$$ISE_GO\% = (ISE_GO \times 100) / (ISE_GO + ISE_PO) \quad (5.47)$$

Joint	ISE Gain optimization (ISE_GO)		ISE parameter optimization (ISE_PO)		ISE_GO % in with noise	ISE_GO % in Without noise
	without noise	with noise	without noise	with noise		
J1	9.19E-09	1.6E-07	7.52E-06	7.52E-06	2.08	4.41E-07
J2	1.97E-05	2.2E-05	7.6E-06	8.73E-06	71.72	2.75E-05
J3	1.19E-06	1.7E-06	4.27E-07	5.21E-07	76.54	1.55E-06

Table 5.8 the ISE without and with disturbance for Gains and structural PSO optimization

NB. YE-0X stands for $Y \times 10^{-0X}$.

Table 5.9 shows some comparison and the initial parameters for gains and parametric PSO optimization for PD controller, with optimal parameters calculated from PSO algorithm.

5.10 RESULTS AND DISCUSSION

When comparing results, by visual comparison the actual and desired joint angles are overlapping Figure 5.23, and 5.24, and we can't see the differ, but in terms ISE we can see the differ to do comparison, we have three joints robot with noise (disturbance) torques to test robustness, all results divided in two tables, Table 5.7 manual gains and PSO gains optimization with/without noise $d = \sin(3t) + 1$ and Table 5.6 the ISE without and with noise (disturbance) for Gains and parametric PSO optimization, it is very clear that average of manual gains tuning ISE_MT% column around 98% this mean that tracking error angle for manual gains tuning fuzzy controller is big ninety-eight percent than error angle for PSO fuzzy controller in both case.

When comparing gains PSO and parametric optimization, is very clear ISE in PSO gains optimization ISE_GO% in ideal (without noise) case around zero percent but in with noise (disturbance) the average error is 50% $= (2+72+76)$.

Finally when comparing all results in all figures and Table 5.7 and Table 5.8, it is very clear that PSO gains optimization for fuzzy PD controller is better than PSO parametric optimization for fuzzy PD controller in ideal (without noise) case but in with noise (disturbance) case in ISE comparison terms will parametric optimization better, and in other terms shown in Table 5.9, we can see that parametric optimization have big time execution

than gains optimization to get optimal parameters and ability to programming more complicated and much parameters.

	Gain_PSO	Structural fuzzy_PSO
Initial	Min=[15,0.01,50],Max =[50,10,300]	MIN_In1=[-30,0.001,30,0.001] MAX_In1=[-40,1,40,1] Min_in2=[-3,0.001,3,0.001] Max_in2=[-12,1,12,1] Min_Out=[-150,0.001,150,0.001] Max_Out=[-300,1,300,1]
Particle number	10	10
Iterative	10	10
Optimized	Kp=42.8651 Kd=1.4932 Gout=202.1934	Rang_In1=[35.6957,-28.7037] Mid1=[0.9858, 0.5909] Rang_In2=[2.8687,-5.0750] Mid2=[0.9803, 0.0780] Rang_Out=[296.6617,-169.8966] Mid=[0.0857, 0.0721]
Fast	Fast five times than structural PSO.	Slow than Gain PSO because its instruction big five times.
programming	Easy and simple with few parameters	Hard and complicated with much parameters
ISE	Without noise same error in PO	Same error in GO
	With noise is big 50% then error in PO	Is low 50% then error in GO

Table 5.9 Some differences simulation results between Gains and structural PSO optimization fuzzy controller.

5.11 CONCLUSION

We have presented the particle swarm optimization for fuzzy PD with 3DOF PUMA560 using manual gains tuning and PSO gains and structure optimization of fuzzy PD controller after get our optimized gains and structure, in offline will put them on our fuzzy PD controller to compare them. We presented simulations for three joints where we simulated the systems' responses with and without noise (disturbance). In the simulations, a quantification of errors was achieved and documented in tables for ISE, it was shown that the lower errors were obtained using a PSO for parametric optimization of fuzzy PD controller in with noise (disturbance) but for few particle and low iterative will take long time then gains optimization to get optimal parameters, but in ideal case we find PSO with gains optimization is better in integral square error and in ability to programming it with few parameters and faster to get optimal parameters, our optimized intelligent control is better than classical control, and is applicable in real time.

General Conclusion

In this thesis focused on simulation and experimental control optimization using PSO for robotic system. The first is aeronautic robot which is quadcopter. We have started by dynamic modeling this system and simulate it on MATLAB. After we realized our two custom prototypes or homemade prototyping the first one based on Arduino due as autopilot controller. For weight, size and speed we have improve the first one to Teensy 3.2 autopilot controller based. All necessary dynamics parameters identified in an experiment like thrust calculations. For PID gains identifications, the manual tuning method has limitations of being time consuming and very tedious. While the conventional Z-N method has been very useful and helpful in our research in terms of practicality, it is criticized by many other researchers. In short time and with the least required effort to experimentally select the appropriate PID gains was the Z-N tuning. Within a few minutes of implementation we could obtain the PID gain values required to stabilize the quadrotor. With respect to time and effort, the Z-N method was the most advantageous. The second advantage of the Z-N method found in this study is that it produces PID gains that can be put into intervals, such that the PSO search within an interval by determining the sum of the Integral of Square Error and the maximum of the overshoot as a multi-objective function [19] or a fitness function. However, regarding robustness, overshoot, steady-state error, and rise time, the Z-N method falls short of the necessary standards. To overcome this, we have used PSO for PID gains optimization. We have proved in both simulation and experimental our optimized PID controller for stabilization of a quadrotor using PSO the constriction coefficient method was never used before for stabilization of a quadrotor. The main advantage of optimization process that find the optimal solution with scientific method not as manual finding depending on the luck, and let the user of it more confidence with their results. The proposed method was tested both using a MATLAB simulation and experimentally. The simulation and experimental results were also compared with a conventional Z-N tuning PID controller and with two different objective functions that the PSO depends on to search the optimal PID controller. The efficiency of PSO for finding the optimal values was verified through experimental results.

The second robotic system is PUMA560 manipulator robot. We have presented the particle swarm optimization for fuzzy PD with 3DOF PUMA560 using manual gains tuning and PSO gains and parametric optimization of fuzzy PD controller after get our optimized gains and parametric offline will put them on our fuzzy PD controller to compare them. We presented simulations for three joints where we simulated the systems' responses with and without noise (disturbance). In the simulations, a quantification of errors was achieved and documented in tables for ISE, it was shown that the lower errors were obtained using a PSO for parametric optimization of fuzzy PD controller in with noise (disturbance) but for few particle and low iterative will take long time then gains optimization to get optimal parameters, but in ideal case we find PSO with gains optimization is better in integral square error and in ability to programming it with few parameters and faster to get optimal parameters, our optimized intelligent control is better than classical control, and is applicable in real time. Future work implementation gains and structural optimized fuzzy PD controller manipulator robot, hybridization with other robust control.

The efficiency of PSO for PID and fuzzy controllers was verified in this thesis both in simulation and experiment. Because the quadcopter is our custom prototype and homemade. Can be good platform for future research. We can do modification both in software or material as research needed. Our prototype can test it with sliding mode, second order sliding mode, fractional controllers, and also as future work optimization Type-2 fuzzy controller using PSO..... etc.

Bibliography

- [1] S. Bouabdallah: design and control of quadrotors with application to autonomous flying. Autonomous Systems Laboratory, Swiss Federal Institute of Technology Lausanne, Switzerland, 2007.
- [2] K. Alexis, G. Nikolakopoulos, A. Tzes: 'Model predictive quadrotor control: attitude, altitude and position experimental studies', IET Control Theory & Applications, 2012, 6, (12), pp. 1812-1827
- [3] I. E. Carlos, J. M. V. Aldo, S. O. Anand: 'Attitude control of quadrotors based on fractional sliding modes: theory and experiments', IET Control Theory & Applications, 2016, 10, (7), pp. 825-832.
- [4] C. Fuyang, J. Rongqiang, Z. Kangkang: 'Robust Backstepping Sliding Mode Control and Observer-based Fault Estimation for a Quadrotor UAV', IEEE Transactions on Industrial Electronics, 2016, PP, (99), pp. 1
- [5] A. Benboualia, M. Tadjine, L. Chrifi-Alaoui: 'Robust Tracking Control Of Quadrotor Using A Nonlinear PID On Eulidean Group', The Mediterranean Journal of Measurement and Control, 2016, 12, (1), pp. 529-536
- [6] W. Haoping, Y. Xuefei, T. Yang, et al: 'Attitude Control of a Quadrotor Using Model Free Based Sliding Model Controller'. Proc. Int. Conf. Control Systems and Computer Science, Bucharest, 2015, pp. 149-154
- [7] A. S. Faiz, K. Kushsairy, A. B.M. Izhar, et al: 'Attitude stabilization of Quad-rotor (UAV) system using Fuzzy PID controller (an experimental test) '. Proc. Int. Conf. Computing Technology and Information Management, Johor, 2015, pp. 99-104
- [8] M. H. Tanveer, A. S. Faiz, D. Hazry, et al: 'Stabilized Controller Design For Attitude And Altitude Controlling Of Quad-Rotor Under Disturbance And Noisy Conditions', American Journal of Applied Sciences, 2013, 10, (8), pp. 819-831
- [9] K. Ogata: 'Modern Control Engineering', (5th Edn. 2010)
- [10] G. Malleshm, S. Mishra, A. N. Jha: 'Ziegler-Nichols based controller parameters tuning for load frequency control in a microgrid'. Proc. Int. Conf. Energy Automation and Signal, Bhubaneswar, Odisha, 2011, pp. 1-8
- [11] S. Bouabdallah, R. Siegwart: Full control of a quadrotor, in *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)* , pp. 153–138, Nov 2007.
- [12] C. Liu and S. D. Prior, "Design and implementation of a mini quadrotor control system in GPS denied environments," *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, Denver, CO, 2015*, pp. 462-469.

- [13] Amoozgar, Mohammad Hadi, Abbas Chamseddine, and Youmin Zhang. "Fault-tolerant fuzzy gain-scheduled PID for a quadrotor helicopter testbed in the presence of actuator faults." *IFAC Proceedings Volumes* 45.3 (2012): 282-287.
- [14] H. Lim, J. Park, D. Lee, H.J. Kim, "Build Your Own Quadrotor", *IEEE Robot. Automat. Mag.*, September 2012.
- [15] H. C. T. E. Fernando, A. T. A. De Silva, M. D. C. De Zoysa, K. A. D. C. Dilshan and S. R. Munasinghe, "Modelling, simulation and implementation of a quadrotor UAV," *2013 IEEE 8th International Conference on Industrial and Information Systems*, Peradeniya, 2013, pp. 207-212.
- [16] M. Federico, W. Beata: 'Particle swarm optimization (PSO). A tutorial', *Chemometrics and Intelligent Laboratory Systems*, 2015, 149, pp. 153-163
- [17] P. R. Dian, M. S.Siti, S. Y. Siti: 'Particle swarm optimization: Technique, system and challenges ', *International Journal of Computer Applications*, 2011, 14, (1), pp. 19-26
- [18] P. E. Andries:'Computational Intelligence: An Introduction, Second Edition', (John Wiley & Sons, 2007)
- [19] Majid Khodier: 'Optimisation of antenna arrays using the cuckoo search algorithm', *IET Microwaves Antennas & Propagation*, 2013, 7, (6), pp. 458-464
- [20] T. T. Mac, C. Copot, T. T. Duc, R. D. Keyser: AR.Drone UAV control parameters tuning based on particle swarm optimization algorithm, *Proc. Int. Conf. on Automation, Quality and Testing, Robotics - THETA 20th edition (AQTR)*, 2016
- [21] J. M. Mohammed, T. R. Mofeed, A. A. Abduladhem: 'Design Optimal PID Controller for Quad Rotor System', *International Journal of Computer Applications*, 2014, 106, (3), pp. 15-20
- [22] H. Boubertakh, S. Bencharef, S. Labiod: 'PSO-based PID control design for the stabilization of a quadrotor'. *Proc. Int. Conf. Systems and Control, Algeria, Algiers*, 2013, pp. 514-517
- [23] H. Bolandi, M. Rezaei, R. Mohsenipour, et al: 'Attitude Control of a Quadrotor with Optimized PID Controller', *Intelligent Control and Automation*, 2013, 4, (3), pp. 335-342
- [24] W. Khalil, E. Dombre, "Robot Manipulators: Modeling, Performance Analysis and Control", Wiley-ISTE, UK, 2007, ISBN: 9781905209101.
- [25] M.R.H.M Adnan, A. Sarkheyli, A.M. Zain, H. Haron, "Fuzzy logic for modeling machining process: a review", *Artificial Intelligence Review*, Vol. 43, No. 3, 2015, pp. 345-379.
- [26] E. Dadios, "Fuzzy Logic - Controls, Concepts, Theories and Applications", INTECH, Rijeka, Croatia, 2012, ISBN: 978-953-51-0396-7.

- [27] F. Marini, B. Walczak, "Particle swarm optimization (PSO). A tutorial", *Chemometrics and Intelligent Laboratory Systems*, Vol. 149, Part. B, 2015, pp. 153-163.
- [28] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, "Particle swarm optimization: Technique, system and challenges," *International Journal of Computer Applications*, Vol. 14, No. 1, pp. 19-26, 2011.
- [29] A.P. Engelbrecht, "Computational Intelligence: An Introduction, Second Edition", John Wiley & Sons, Ltd, Chichester, UK, 2007, ISBN: 9780470035610.
- [30] G.M. Khoury, M. Saad, H.Y. Kanaan, "Fuzzy PID Control of a Five DOF Robot Arm", *Journal of Intelligent and Robotic Systems*, Vol. 40, No. 3, 2004, pp. 299-320.
- [31] G. Štimac, S. Braut, R. Žigulić, "Comparative analysis of PSO algorithms for PID controller tuning", *Chinese Journal of Mechanical Engineering*, Vol. 27, No. 5, 2014, pp. 928-936.
- [32] J.S. Chiou, S.H. Tsai, M.T. Liua, "A PSO-based adaptive fuzzy PID-controllers", *Simulation Modelling Practice and Theory*, Vol. 26, 2012, pp. 49-59.
- [33] R. Martínez-Soto, O. Castillo, L.T. Aguilar, A. Rodriguez, "A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers", *International Journal of Machine Learning and Cybernetics*, Vol. 6, No. 2, 2015, pp. 175-196.
- [34] R. Martínez-Soto, O. Castillo, L. T. Aguilar, "Type-1 and Type-2 fuzzy logic controller design using a Hybrid PSO–GA optimization method", *Information Sciences*, Vol. 285, 2014, pp. 35-49.
- [35] L. Hao, B. Yongqiang, L. Geng, et al: 'Robust attitude control of uncertain quadrotors', *IET Control Theory and Applications*, 2012, 7, (11), pp. 1583–1589
- [36] Š. Goranka, B. Sanjin, Ž. Roberto,: 'Comparative analysis of PSO algorithms for PID controller tuning', *Chinese Journal of Mechanical Engineering*, 2014, 27, (5), pp. 928-936
- [37] **M.A. Khodja**, M. Tadjine, M.S. Boucherit, K. Busawon: Optimization of a Proportional Derivative (PD) Fuzzy controller using the Particle Swarm Optimization (PSO) Technique for a 3DOF Robot Manipulator, *The Mediterranean Journal of Measurement and Control*, Vol. 12, N. 4, 2016 , pp. 670-677.
- [38] **M.A. Khodja**, M. Tadjine, M.S. Boucherit, M. Benzaoui: Tuning PID Attitude Stabilization of a Quadrotor Using Particle Swarm Optimization (Experimental), *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 8, N. A8. 2017.
- [39] Bai, Qinghai. "Analysis of particle swarm optimization algorithm." *Computer and information science* 3.1 (2010): 180.
- [40] G.C. Onwubolu and M. Clerc, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", *International Journal of Production*

Research, vol. 42, issue. 3, pp. 473–491, February 2004.

[41] J. C. Bansal, K. Deep, "A Modified Binary Particle Swarm Optimization for Knapsack Problems", *Applied Mathematics and Computation*, vol. 218, issue. 22, pp. 11042–11061, 2012.

[42] A. Salman, I. Ahmad, S. Al-Madani, "Particle swarm optimization for task assignment problem", *Microprocessors and Microsystems*, vol. 26, issue. 8, pp. 363–371, 2002.

[43] R. Mendes, P. Cortez, M. Rocha, and J. Neves, "Particle swarms for feed forward neural networks training", *In International Joint Conference on Neural Networks*, pp. 1895–1899, Honolulu (Hawaii), USA, 2002.

[44] Tayebi, Abdelhamid, and Stephen McGilvray. "Attitude stabilization of a VTOL quadrotor aircraft." *IEEE Transactions on control systems technology* 14.3 (2006): 562-571.

[45] Lozano, Rogelio, ed. *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.

[46] <http://ecalch.ch/> 31-05-2017

[47] Y. Myunggon: 'A Transfer Function Model of Thrust Dynamics for Multi-Rotor Helicopters', *International Journal of Engineering Research & Technology*, 2016, 5, (1), ISSN. 2278-0181

[48] **M.A. Khodja**, M. Tadjine, M.S. Boucherit, M. Benzaoui: "Experimental dynamics identification and control of a quadcopter." *Systems and Control (ICSC), 2017 6th International Conference on*. IEEE, 2017.

[50] <https://www.pjrc.com/store/teensy32.html> , 31-05-2017.

[51] <https://pixhawk.org/modules/px4flow> , 31-05-2017

[52] <https://github.com/mjs513/FreeIMU-Updates>, 31-05-2017

[53] <http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/> 24/08/2017

[54] <https://github.com/DeveloperPaul123/Teensy-Quadcopter>,31-05-2017

[55] B. Armstrong, O. Khatib, J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 arm", *Proc. IEEE Conf. on Robotics and Automation*, 1986, pp. 510-518.

[56] F. F. Hélio, B. d. M. O. Paulo, J. S. P. Eduardo, et al: 'Many-Objective PSO PID Controller Tuning'. *Proc. 11th Portuguese. Conf. Automatic Control*, Portuguese, 2015, pp. 183-192

[57] J. Hurel, A. Mandow, A.G. Cerezo, "Tuning a fuzzy controller by particle swarm optimization for an active suspension system", *Proc 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, 2012, pp. 2524 – 2529.

- [58] L.K. Letting, J.L. Munda, Y. Hamam, “Optimization of a fuzzy logic controller for PV grid inverter control using S-function based PSO”, *Solar Energy*, Vol. 86, No. 6, 2012, pp. 1689-1700.
- [59] V. Jeyalakshmi, P. Subburaj, “PSO-scaled fuzzy logic to load frequency control in hydrothermal power system”, *Soft Computing*, 2015, pp. 1-18.
- [60] M. Sayed, S.M. Gharghory, H.A. Kamal, “Gain tuning PI controllers for boiler turbine unit using a new hybrid jump PSO”, *Journal of Electrical Systems and Information Technology*, Vol. 2, No. 1, 2015, pp. 99-110.
- [61] O. Zafer, K. Bingül, “A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control”, *Expert Systems with Applications*, Vol. 38, No. 1, 2011, pp. 1017–1031.
- [62] D.M. Wonohadidjojo, G. Kothapalli, M.Y. Hassan, “Position Control of Electro-hydraulic Actuator System Using Fuzzy Logic Controller Optimized by Particle Swarm Optimization”, *International Journal of Automation and Computing*, Vol. 10, No. 3, 2013, pp. 181-193.
- [63] Z. Bingül, O. Karahan, “A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control”, *Expert Systems with Applications*, Vol. 38, No. 1, 2011, pp. 1017-1031.
- [64] O. Castillo, R. Martinez-Marroquin, P. Melin, “Bio-inspired Optimization of Fuzzy Logic Controllers for Robotic Autonomous Systems with PSO and ACO”, *Fuzzy Information and Engineering*, Vol. 2, No. 2, 2010, pp. 119-143.
- [65] <http://www.hobbytronics.co.uk/teensy-v31> 03-01-2018.
- [66] <https://store.arduino.cc/arduino-due> 03-01-2018.
- [67] <https://store.open-electronics.org/GY-86> 03-01-2018.
- [68] <https://playground.arduino.cc/Main/MPU-6050> 03-01-2018.
- [69] <http://www.alldatasheet.com/datasheet-pdf/pdf/428790/HONEYWELL/HMC5883L.html> 03-01-2018.
- [70] http://www.alldatasheet.com/view_datasheet.jsp?Searchword=MS561101BA 03-01-2018.
- [71] https://hobbyking.com/en_us/dys-be1806-2300kv-brushless-multirotor-motor-2s3s.html?__store=en_us 03-01-2018.
- [72] https://hobbyking.com/en_us/gemfan-propeller-5x3-black-cw-ccw-2pcs.html 03-01-2018.
- [73] https://hobbyking.com/en_us/multiwii-mwc-fc-bluetooth-module-programmer-android-compatible-1.html?__store=en_us 03-01-2018.
- [74] <https://hobbyking.com/> 03-01-2018.

[75] https://hobbyking.com/en_us/turnigy-5-7-5a-8-42v-hv-ubec-for-lipoly.html 03-01-2018.

[76] https://hobbyking.com/en_us/hobbykingtm-totem-q250-quadcopter-kit.html?__store=en_us 03-01-2018.

Appendix

PARTICLE SWARM OPTIMIZATION Algorithm

In this section, we explain the PSO algorithm employed in the optimization of the aforementioned controllers. The PSO is a population based method for finding an optimal solution to an objective function. The so-called swarm consists of “ n ” particles, and the position of each particle stands for the potential solution in space. The particles change its condition according to the following three principles:

- control its inertia,
- control the position of the particles to its most optimal position,
- Control of the velocity of the particles

The position of each particle in the swarm is affected by the most optimal position based on its individual experience and on the global best position of the particle around its neighborhood.

There are several algorithms of the PSO [16][17][18]. In this work integral of square error (ISE) is fitness function (objective function) and the constriction coefficient [16][17] [39]method it was selected whereby the speed and position of each particle change according the following equality

$$v_{ij}(t+1) = X \left[v_{ij}(t) + \phi_1 \left(Pbest_{ij}(t) - x_{ij}(t) \right) + \phi_2 \left(Gbest_j - x_{ij}(t) \right) \right] \quad (A.1)$$

Where:

$$X = \frac{2k}{\left| 2 - \phi - \sqrt{\phi(\phi - 4)} \right|} \quad (A.2)$$

With:

$$\phi = \phi_1 + \phi_2, \quad \phi_1 = c_1 r_1, \phi_2 = c_2 r_2, \quad \phi \geq 4, k \in [0 \ 1] \quad (A.3)$$

c_1 and c_2 are the acceleration coefficients, r_1 and r_2 random values in[0 1]

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (\text{A.4})$$

The corresponding algorithm for this equality (1.1) and (1.4) are in Table 1.1 and 1.2.

Table A.1, Particle Swarm Optimization Algorithm

Algorithm 1.1
<ul style="list-style-type: none"> • Initialize number of particles (n), max of iterative (it), lower and upper bounds for position and speed ,($Pbest, Gbest$). • $it=0$; • While $it < max$ of iterative • Calculate φ_1, φ_2 are related with random values (r_1, r_2) using (1.3) • For $j=1$ to n do • Calculate speeds using (1.1) • Control that speeds are in searching space. Using (1.5) and (1.6) • Calculate new positions using (1.4) • Control that positions are in searching space. • Calculate fitness function $f(x_i)$.(1.7) • End for. • Calculate Pbest and Gbest using algorithm 1.2 • $it=it+1$ • end while.

A.3 Interval limitation

If a particle's velocity exceeds a specified maximum or minimum velocity, the particle's velocity is set to the maximum or minimum velocity. Let $V_{max,j}$ denote the maximum allowed velocity in dimension j , and $V_{min,j}$. Particle velocity is then adjusted before the position update using[18],

$$v_{ij}(t+1) = \begin{cases} v_{ij}(t+1) & \text{if } v_{ij}(t+1) \leq V_{max,j} \\ V_{max,j} & \text{if } v_{ij}(t+1) > V_{max,j} \end{cases} \quad (\text{A.5})$$

$$v_{ij}(t+1) = \begin{cases} V_{\min, j} & \text{if } v_{ij}(t+1) \leq V_{\min, j} \\ v_{ij}(t+1) & \text{if } v_{ij}(t+1) > V_{\min, j} \end{cases} \quad (\text{A.6})$$

Where $v_{ij}(t+1)$ is calculated using equation(1.1).

Table A.2. Algorithm for Global and Local best particles calculations

Algorithm 1.2
<ul style="list-style-type: none"> • For each particle i from n do • If $f(x_i) < f(P_{\text{best}})$ then $P_{\text{best}}=x_i$. • If $f(P_{\text{best}}) < f(G_{\text{best}})$ then $G_{\text{best}}=P_{\text{best}}$. • End for.

The Integral of Square Error (ISE) is used as performance indices:

$$ISE = \int_0^{t_{\text{end}}} [e(t)]^2 dt \quad (\text{A.7})$$

A.4 Example of PSO used for optimized PD Fuzzy Gains controller

In this example we have three parameters (PD and output gain) to optimize them. There limitation and initialize values are in Table A.3.

Table A.3 initializes parameters values

parameters	Min	Max
P	15	40
D	0.5	10
G_{out}	50	450
n	10	
iterative number	10	

The parameter, k in equation above controls the exploration and exploitation [18] abilities of the swarm. For $\kappa \approx 0$, fast convergence is obtained with local exploitation see Figure A.1, for PSO-

Fuzzy PD controller with $k \approx 0$. On the other hand, $\kappa \approx 1$ results in slow convergence with a high degree of exploration see Figure A.2.

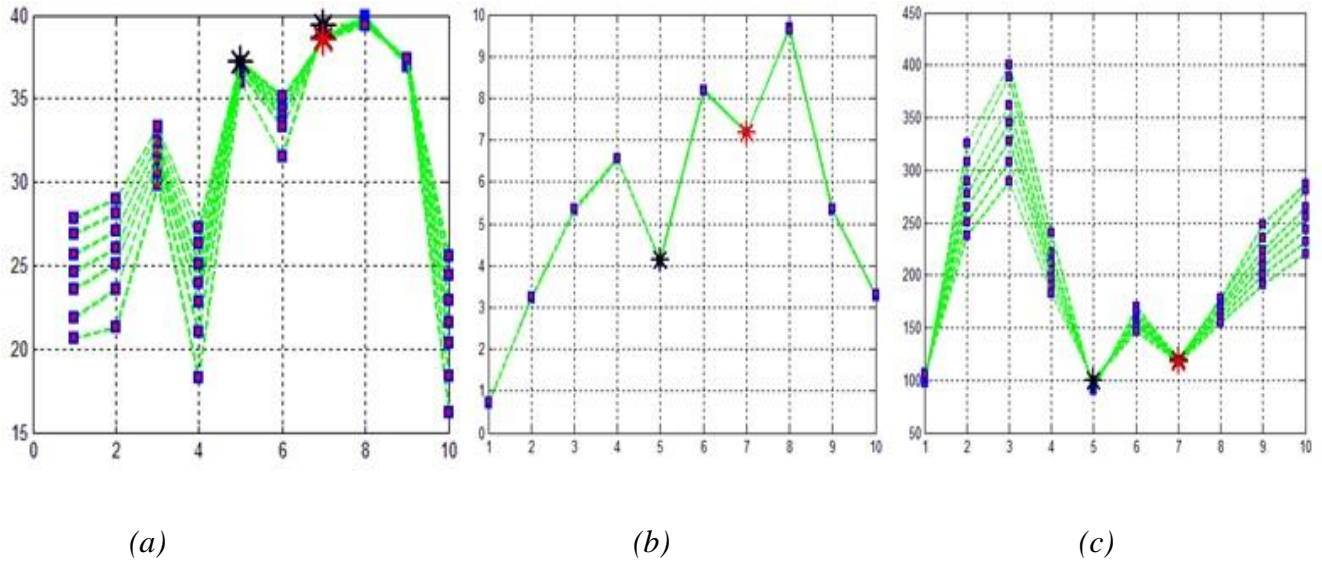


Figure A.1. Local exploitation for PSO-Fuzzy PD controller :(a) for P optimal gain search, (b) for D, (c) for gain out.

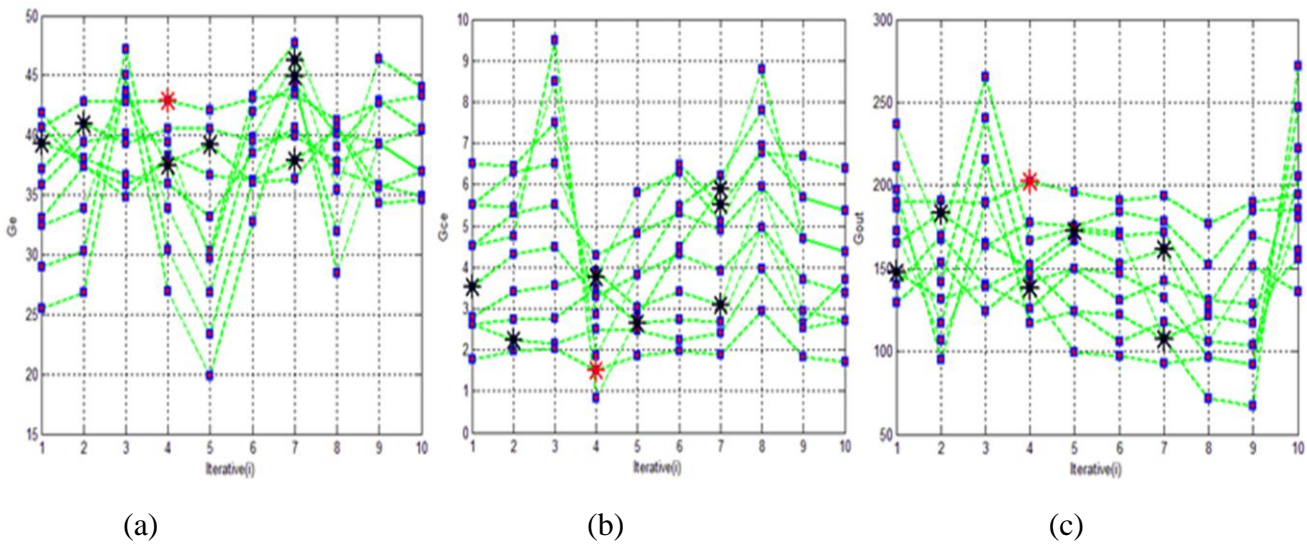


Figure A.2. High degree of exploration for PSO-Fuzzy PD controller :(a) for P optimal gain search, (b) for D, (c) for gain out

For Figure A.2, in $k \approx 1$, high degree of exploration that has many ten black points these points are local best PSO find and best one is red point it mean global best PSO. On other hand in Figure

A.1, in $k \approx 0$ local exploitation is very clear that particle positions are move locally because we can see just two black points are local best PSO find and the red is global best PSO or best one.

A.5 Advantages and Disadvantages of the Basic Particle Swarm Optimization Algorithm.

A.5.1 Advantages of the basic particle swarm optimization algorithm:

- PSO is based on the intelligence. It can be applied into both scientific research and engineering use.

-PSO has no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast.

-The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily.

- PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution.

A.5.2 Disadvantages of the basic particle swarm optimization algorithm:

-The method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.

-The method cannot work out the problems of scattering and optimization

-The method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

A.6 Conclusion

Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method. In past several years, PSO has been successfully applied in many research areas [40,41,42,43]. It is proved that PSO gets better results in a faster, simpler way compared to other methods [18]. Another reason that PSO is likable is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. We present its originality, formulation, algorithm and application example in our case. In the next chapters will present more applications details of Particle Swarm Optimization approach with linear and fuzzy controllers.