



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Control de formación distribuido para una flotilla de
vehículos autónomos basado en sistemas multi-
agentes

presentada por

Ing. Jesus Avelino Vazquez Trejo

como requisito para la obtención del grado de
**Maestro en Ciencias en Ingeniería
Electrónica**

Director de tesis

Dr. Manuel Adam Medina

Codirector de tesis

Dr. Carlos Daniel García Beltrán

Cuernavaca, Morelos, México. Junio de 2020.



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., **22/mayo/2020**
No. de Oficio: **DIE/406/20**
Asunto: **Aceptación de documentos de tesis**

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Jesús Avelino Vázquez Trejo**, con número de control **M18CE045** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Control de Formación Distribuido para una Flotilla de Vehículos Autónomos Basado en Sistemas Multi-agentes"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Manuel Adam Medina
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 5992003

CODIRECTOR DE TESIS

Dr. Carlos Daniel García Beltrán
Doctor en Ciencias en Ingeniería Electrónica
Cédula profesional 8699605

REVISOR 1

Dr. Enrique Quintero-Mármol Márquez
Doctor en Ciencias en Ingeniería Química
Cédula profesional 10259407

REVISOR 2

Dr. Juan Reyes Reyes
Doctor en Ciencias en la Especialidad de Control Automático
Cédula profesional 4214833

C.p. M.E. Guadalupe Garrido Rivera- Jefa del Departamento de Servicios Escolares
Estudiante
Expediente
MPS/irr.



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos **28/mayo/2020**

OFICIO No. SAC/ 154/2020

Asunto: Autorización de impresión de tesis

JESÚS AVELINO VÁZQUEZ TREJO
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA
P R E S E N T E

Por este conducto tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "*Control de Formación Distribuido para una Flotilla de Vehículos Autónomos Basado en Sistemas Multi-agentes*", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E

Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



**CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA**

C.c.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares
Expediente
CVGR/CHG

Agradecimientos

Agradezco a mis padres, por todo el apoyo otorgado hasta el día de hoy, a mi hermano quien desde siempre me ha brindado su apoyo incondicional, a mis abuelos maternos quienes con cariño y paciencia siempre han sabido ser un gran ejemplo para toda la familia, al resto de mi familia quienes siempre han sabido dar palabras de aliento, comprensión y apoyo, a mi asesor de tesis el Dr. Manuel Adam Medina quien ha sabido ser paciente conmigo a lo largo de todo este tiempo, dándome su apoyo incondicional en todo momento, a Lorena Ruiz Ramírez por su paciencia, entrega y solidaridad con todo el grupo estudiantil del departamento de ingeniería electrónica, tanto maestría como doctorado muchas gracias, a mi codirector de tesis el Dr. Carlos Daniel García Beltrán quien me ha brindado su apoyo y me ha dado la libertad de desarrollar este trabajo con cierta libertad en la toma de decisiones. También agradezco abiertamente al Consejo Nacional de Ciencia y Tecnología (Conacyt) por la beca otorgada a lo largo del desarrollo del tema de tesis, así mismo, por el apoyo otorgado para realizar una estancia en Francia que fortaleció en gran manera el tema de tesis, al Tecnológico Nacional de México por las instalaciones brindadas para el desarrollo de mi formación, al profesor Mohammed Chadli, quien actualmente se encuentra laborando en la universidad de Evry Francia y quien me recibió con los brazos abiertos en su grupo de trabajo. Agradezco a mis profesores de primer semestre: Dra. Gloria Lilia Osorio Gordillo, Dr. Carlos Daniel García Beltrán, Dr. Juan Reyes Reyes, Dr. José Francisco Gómez Aguilar. A mis profesores de segundo semestre: Dr. Manuel Adam Medina, Dr. Ricardo Fabricio Escobar Jiménez, Dr. Carlos Manuel Astorga Zaragoza, Dr. Gerardo Vicente Guerrero Ramírez. En especial agradecimiento a mis revisores de tesis: Dr. Juan Reyes Reyes y al Dr. Enrique Quintero Mármol Márquez quienes a través de sus observaciones ayudaron a mejorar el trabajo de tesis.

Resumen

En este trabajo de tesis se presenta un esquema de control de formación por algoritmos de consenso aplicado a un equipo de cuadricópteros junto con las simulaciones pertinentes para validar el esquema, posteriormente al mismo se añade la perturbación del modelo de turbulencia de viento de Dryden y se comparan los resultados entre 3 algoritmos para control de formación por consenso.

Se diseñó un controlador interno para poder estabilizar los cuadricópteros y poder moverlos a libertad. La ley de control entregada por los diferentes algoritmos se adaptó por medio de unas transformaciones que relacionan la salida del algoritmo con los ángulos de Euler necesarios para mover los UAVs de forma adecuada. Se analizó y se añadió la teoría de grafos a los algoritmos de control de formación la cual permite realizar la forma objetivo comunicando únicamente a los drones entre sus vecinos cercanos. Se utilizó un algoritmo de control de formación tipo observador, el cual solo ocupa las posiciones relativas de los cuadricópteros y se comparó con otros que tienen disponibles todos los estados de posición y de velocidades primero sin perturbaciones y posteriormente añadiéndolas al esquema.

Índice general

Índice de figuras	XI
Índice de tablas	XVII
Nomenclatura	XIX
1. Introducción	1
1.1. Estado del arte	1
1.1.1. Control de Formación	2
1.1.2. <i>flocking</i>	7
1.1.3. <i>Rendezvous</i>	8
1.1.3.1. Resumen del estado del Arte	11
1.2. Hipótesis	12
1.3. Objetivos	12
1.3.1. Objetivo General	12
1.3.2. Objetivos Específicos	12
1.4. Planteamiento del problema	13
1.5. Propuesta de solución	15
2. Diseño de un esquema de control de formación para un sistema multi-agente.	19
2.1. Algoritmos de consenso	19
2.1.1. Consenso	20
2.1.2. Control de formación	20
2.1.3. Control basado en procesamiento centralizado y descentralizado.	21

2.2.	Diseño para resolver el problema de control de formación en sistemas multi-agentes con representación de sistemas doble integrador.	23
2.2.1.	Posicionamiento del problema	23
2.2.2.	Extensión o aplicación para control de formación	24
2.3.	Otros algoritmos de control de formación	24
2.3.1.	Algoritmo de consenso tipo observador modificado para control de formación	25
2.3.2.	Diseño de ganancias para realizar control de formación por medio del algoritmo tipo observador.	26
2.3.3.	Distancias relativas deseadas entre los agentes.	27
3.	Diseño del esquema de control de formación para sistemas multi-agentes.	29
3.1.	Algebra básica de teoría de grafos	29
3.1.1.	Ejemplos de grafos	30
3.1.2.	Matriz de adyacencias y matriz laplaciana	33
3.1.3.	Distancias entre los agentes en la formación final deseada	36
3.2.	Sistema doble integrador para el diseño del algoritmo de consenso.	37
3.3.	Cálculo de las matrices de retroalimentación para el algoritmo de control de formación tipo-observador.	38
4.	Resultados del esquema para control de formación aplicado a un sistema de multiples cuadricópteros.	41
4.1.	Modelo dinámico (Cuadricópteros)	41
4.2.	Transformaciones	44
4.2.1.	<i>Thrust</i> o empuje de los rotores del cuadricóptero modelo <i>crazyflie 2.0</i>	44
4.2.1.1.	<i>Thrust</i> máximo para el <i>crazyflie 2.0</i>	45
4.3.	Control PI para la parte rotacional del modelo dinámico del los UAVs	45
4.4.	Estructuración del esquema de control de formación para un equipo de cuadricópteros.	48
4.4.1.	Esquema de control de formación bajo el protocolo (2.6) para un equipo de cuadricópteros sin perturbaciones.	50

4.4.2.	Hexágono formado con el esquema (4.3) sin perturbaciones.	51
4.4.3.	Tetraedro formado con el esquema (4.3) sin perturbaciones.	55
4.4.4.	Esquema de control de formación bajo el protocolo (2.7) para un equipo de cuadricópteros sin perturbaciones.	59
4.4.5.	Hexágono formado con el esquema (4.12) sin perturbaciones.	60
4.4.6.	Tetraedro formado con el esquema (4.12) sin perturbaciones.	62
4.4.7.	Hexágono formado con el esquema (4.19) sin perturbaciones.	65
4.4.8.	Tetraedro formado con el esquema (4.19) sin perturbaciones.	67
4.5.	Perturbaciones (Modelo de turbulencias en el viento de Dryden)	69
4.5.1.	Esquema de control de formación bajo el protocolo (2.6) para un equipo de cuadricópteros con perturbaciones.	72
4.5.2.	Hexágono formado con el esquema (4.27) con perturbaciones.	73
4.5.3.	Tetraedro formado con el esquema (4.27) con perturbaciones.	75
4.5.4.	Esquema de control de formación bajo el protocolo (2.7) para un equipo de cuadricópteros con perturbaciones.	77
4.5.5.	Hexágono formado con el esquema (4.34) con perturbaciones.	78
4.5.6.	Tetraedro formado con el esquema (4.34) con perturbaciones.	80
4.5.7.	Esquema de control de formación bajo el protocolo (2.8) para un equipo de cuadricópteros con perturbaciones.	82
4.5.8.	Hexágono formado con el esquema (4.41) con perturbaciones.	83
4.5.9.	Tetraedro formado con el esquema (4.41) con perturbaciones.	85

5. Conclusiones	87
Bibliografía	89
A. Anexos	93
A.1. Cuadrado formado con el esquema (4.3) sin perturbaciones. . .	93
A.2. Pentágono formado con el esquema (4.3) sin perturbaciones. . .	98
A.3. Cuadrado formado con el esquema (4.12) sin perturbaciones. . .	102
A.4. Pentágono formado con el esquema (4.12) sin perturbaciones. . .	104
A.5. Cuadrado formado con el esquema (4.19) sin perturbaciones. . .	106
A.6. Pentágono formado con el esquema (4.19) sin perturbaciones. . .	108
A.7. Cuadrado formado con el esquema (4.27) con perturbaciones. . .	110
A.8. Pentágono formado con el esquema (4.27) con perturbaciones. . .	112
A.9. Cuadrado formado con el esquema (4.34) con perturbaciones. . .	114
A.10. Pentágono formado con el esquema (4.34) con perturbaciones. . .	116
A.11. Cuadrado formado con el esquema (4.41) con perturbaciones. . .	118
A.12. Pentágono formado con el esquema (4.41) con perturbaciones. . .	120

Índice de figuras

1.1.	Posiciones iniciales.	14
1.2.	Diagrama general para realizar control de formación con 3 agentes.	15
1.3.	Diagrama a bloques de la propuesta de solución.	17
2.1.	Ejemplo de una red con unidad de procesamiento centralizada.	21
2.2.	Ejemplo de una red con unidad de procesamiento descentralizada.	22
2.3.	Ejemplo de distancias relativas en una formación de 4 drones.	27
3.1.	Ejemplo de un grafo indirecto conectado.	31
3.2.	Ejemplo de un grafo fuertemente conectado.	31
3.3.	Ejemplo de un grafo balanceado y fuertemente conectado.	32
3.4.	Ejemplo de un árbol expandido directo.	32
3.5.	Topología de comunicación y matriz de adyacencias equivalente para el grafo indirectamente conectado.	33
3.6.	Topología de comunicación y matriz de adyacencias equivalente para el grafo fuertemente conectado.	34
3.7.	Topología de comunicación y matriz de adyacencias equivalente para el grafo balanceado y fuertemente conectado.	34
3.8.	Topología de comunicación y matriz de adyacencias equivalente para el grafo de árbol expandido directo.	35
3.9.	Formación deseada y las distancias deseadas de los agentes.	37
4.1.	Esquema expandido del diagrama a bloques de la propuesta de solución.	48
4.2.	Diagrama de flujo para la simulación del esquema de control de formación.	49

4.3. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	50
4.4. Topología de comunicación y matriz de adyacencias equivalente para la formación de un hexágono.	52
4.5. Hexágono formado con el algoritmo tipo observador (2.6) sin perturbaciones.	53
4.6. Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.	53
4.7. Thrust y velocidades lineales con respecto a los 3 ejes.	54
4.8. Topología de comunicación y matriz de adyacencias equivalente para la formación de un tetraedro.	56
4.10. Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.	57
4.9. Tetraedro formado con el algoritmo tipo observador (2.6) sin perturbaciones.	57
4.11. Thrust y velocidades lineales con respecto a los 3 ejes.	58
4.12. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	59
4.13. Hexágono formado con el algoritmo tipo observador (2.7) sin perturbaciones.	60
4.14. Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.	60
4.15. Thrust y velocidades lineales con respecto a los 3 ejes.	61
4.16. Tetraedro formado con el algoritmo tipo observador (2.7) sin perturbaciones.	62
4.17. Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.	62
4.18. Thrust y velocidades lineales con respecto a los 3 ejes.	63
4.19. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	64
4.20. Hexágono formado con el algoritmo tipo observador (2.8) sin perturbaciones.	65
4.21. Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.	65
4.22. Thrust y velocidades lineales con respecto a los 3 ejes.	66
4.23. Tetraedro formado con el algoritmo tipo observador (2.8) sin perturbaciones.	67

4.24. Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.	67
4.25. Thrust y velocidades lineales con respecto a los 3 ejes.	68
4.26. Perturbaciones con respecto a los 3 ejes.	71
4.27. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	72
4.28. Hexágono formado con el algoritmo tipo observador (2.6) con perturbaciones.	73
4.29. Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.	73
4.30. Thrust y velocidades lineales con respecto a los 3 ejes.	74
4.31. Tetraedro formado con el algoritmo tipo observador (2.6) con perturbaciones.	75
4.32. Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.	75
4.33. Thrust y velocidades lineales con respecto a los 3 ejes.	76
4.34. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	77
4.35. Hexágono formado con el algoritmo tipo observador (2.7) con perturbaciones.	78
4.36. Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.	78
4.37. Thrust y velocidades lineales con respecto a los 3 ejes.	79
4.38. Tetraedro formado con el algoritmo tipo observador (2.7) con perturbaciones.	80
4.39. Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.	80
4.40. Thrust y velocidades lineales con respecto a los 3 ejes.	81
4.41. Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones	82
4.42. Hexágono formado con el algoritmo tipo observador (2.8) con perturbaciones.	83
4.43. Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.	83
4.44. Thrust y velocidades lineales con respecto a los 3 ejes.	84
4.45. Tetraedro formado con el algoritmo tipo observador (2.8) con perturbaciones.	85

4.46. Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.	85
4.47. Thrust y velocidades lineales con respecto a los 3 ejes.	86
A.1. Topología de comunicación y matriz de adyacencias equivalente para la formación de un cuadrado.	95
A.2. Cuadrado formado con el algoritmo tipo observador (2.6) sin perturbaciones.	96
A.3. Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.	96
A.4. Thrust y velocidades lineales con respecto a los 3 ejes.	97
A.5. Topología de comunicación y matriz de adyacencias equivalente para la formación de un pentágono.	99
A.6. Pentágono formado con el algoritmo tipo observador (2.6) sin perturbaciones.	100
A.7. Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.	100
A.8. Thrust y velocidades lineales con respecto a los 3 ejes.	101
A.9. Cuadrado formado con el algoritmo tipo observador (2.7) sin perturbaciones.	102
A.10. Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.	102
A.11. Thrust y velocidades lineales con respecto a los 3 ejes.	103
A.12. Pentágono formado con el algoritmo tipo observador (2.7) sin perturbaciones.	104
A.13. Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.	104
A.14. Thrust y velocidades lineales con respecto a los 3 ejes.	105
A.15. Cuadrado formado con el algoritmo tipo observador (2.8) sin perturbaciones.	106
A.16. Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.	106
A.17. Thrust y velocidades lineales con respecto a los 3 ejes.	107
A.18. Pentágono formado con el algoritmo tipo observador (2.8) sin perturbaciones.	108
A.19. Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.	108
A.20. Thrust y velocidades lineales con respecto a los 3 ejes.	109

A.21. Cuadrado formado con el algoritmo tipo observador (2.6) con perturbaciones.	110
A.22. Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.	110
A.23. Thrust y velocidades lineales con respecto a los 3 ejes.	111
A.24. Pentágono formado con el algoritmo tipo observador (2.6) con perturbaciones.	112
A.25. Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.	112
A.26. Thrust y velocidades lineales con respecto a los 3 ejes.	113
A.27. Cuadrado formado con el algoritmo tipo observador (2.7) con perturbaciones.	114
A.28. Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.	114
A.29. Thrust y velocidades lineales con respecto a los 3 ejes.	115
A.30. Pentágono formado con el algoritmo tipo observador (2.7) con perturbaciones.	116
A.31. Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.	116
A.32. Thrust y velocidades lineales con respecto a los 3 ejes.	117
A.33. Cuadrado formado con el algoritmo tipo observador (2.8) con perturbaciones.	118
A.34. Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.	118
A.35. Thrust y velocidades lineales con respecto a los 3 ejes.	119
A.36. Pentágono formado con el algoritmo tipo observador (2.8) con perturbaciones.	120
A.37. Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.	120
A.38. Thrust y velocidades lineales con respecto a los 3 ejes.	121

Índice de tablas

4.1. Parámetros del cuadricóptero crazyflie 2.0.	43
4.2. Condiciones iniciales para la formación final de un hexágono. .	51
4.3. Condiciones iniciales para la formación final de un tetraedro. .	55
4.4. Parámetros del modelo de turbulencias de Dryden	69
A.1. Condiciones iniciales para la formación final de un cuadrado. .	94
A.2. Condiciones iniciales para la formación final de un pentágono. .	98

Nomenclatura

x_i	Posición en el eje x del agente i -ésimo.
y_i	Posición en el eje y del agente i -ésimo.
z_i	Posición en el eje z del agente i -ésimo.
\dot{x}_i	Velocidad ($\frac{m}{s}$) en el eje x del agente i -ésimo.
\dot{y}_i	Velocidad ($\frac{m}{s}$) en el eje y del agente i -ésimo.
\dot{z}_i	Velocidad ($\frac{m}{s}$) en el eje z del agente i -ésimo.
\ddot{x}_i	Aceleración ($\frac{m}{s^2}$) en el eje x del agente i -ésimo.
\ddot{y}_i	Aceleración ($\frac{m}{s^2}$) en el eje y del agente i -ésimo.
\ddot{z}_i	Aceleración ($\frac{m}{s^2}$) en el eje z del agente i -ésimo.
ϕ_i	Posición angular (rad) con respecto al eje x del agente i -ésimo.
θ_i	Posición angular (rad) con respecto al eje y del agente i -ésimo.
ψ_i	Posición angular (rad) con respecto al eje z del agente i -ésimo.
$\dot{\phi}_i$	Velocidad angular ($\frac{rad}{s}$) con respecto al eje x del agente i -ésimo.
$\dot{\theta}_i$	Velocidad angular ($\frac{rad}{s}$) con respecto al eje y del agente i -ésimo.
$\dot{\psi}_i$	Velocidad angular ($\frac{rad}{s}$) con respecto al eje z del agente i -ésimo.
$\ddot{\phi}_i$	Aceleración angular ($\frac{rad}{s^2}$) con respecto al eje x del agente i -ésimo.
$\ddot{\theta}_i$	Aceleración angular ($\frac{rad}{s^2}$) con respecto al eje y del agente i -ésimo.
$\ddot{\psi}_i$	Aceleración angular ($\frac{rad}{s^2}$) con respecto al eje z del agente i -ésimo.
u_i	Ley de control de los protocolos de control de formación del agente i -ésimo.
H	Matriz que contiene las distancias deseadas entre los agentes.
Y_i	Vector de salidas del agente i -ésimo.
\mathcal{L}	Matriz Laplaciana.
λ_i	Representa los eigenvalores de la matriz Laplaciana para $i=2, \dots, N$
A, B, C	Matrices del sistema lineal que representa el sistema doble integrador.
A_r, B_r, C_r	Matrices del sistema rotacional linealizado.

L, F	Matrices de retroalimentación del protocolo tipo observador.
c	Ganancia de acoplamiento.
I	Matriz identidad.
N	Total de agentes en un sistema multi-agente.
\mathcal{A}	Matriz de adyacencias.
g	Gravedad ($\frac{m}{s^2}$).
m_i	Masa (Kg) del agente i-ésimo.
T_i	Empuje de los motores del i-ésimo vehículo volador.
w_{xi}	Velocidad del viento ($\frac{m}{s}$) en el eje x sobre el agente i-ésimo.
w_{yi}	Velocidad del viento ($\frac{m}{s}$) en el eje y sobre el agente i-ésimo.
w_{zi}	Velocidad del viento ($\frac{m}{s}$) en el eje z sobre el agente i-ésimo.
h	Altura en pies.
Lx	Turbulencia con respecto al eje x a escala de la altura.
Ly	Turbulencia con respecto al eje y a escala de la altura.
Lz	Turbulencia con respecto al eje z a escala de la altura.
$\Phi_x(\Omega), \Phi_y(\Omega), \Phi_z(\Omega)$	Funciones de la componente espectral.
σ_x	Intensidad de la turbulencia con respecto al eje x .
σ_y	Intensidad de la turbulencia con respecto al eje y .
σ_z	Intensidad de la turbulencia con respecto al eje z .
$w_{(.)_0}$	Vector de velocidad estática del viento con respecto a la altura.
$a_{(.)_n}$	Define la energía del poder espectral.
J_x	Es la inercia ($Kg \times m^2$) en el eje x .
J_y	Es la inercia ($Kg \times m^2$) en el eje y .
J_z	Es la inercia ($Kg \times m^2$) en el eje z .
R	Par rotacional de Roll ($N \cdot m$).
P	Par rotacional de Pitch ($N \cdot m$).
Y	Par rotacional de Yaw ($N \cdot m$).

Introducción

En el área de control cooperativo de sistemas multi-agentes existe una vertiente llamada control de formación, la cual tiene como objetivo fundamental estabilizar las distancias/posiciones relativas entre los agentes a valores predefinidos. En el campo de sistemas multi-agentes, control de formación tiene muchas aplicaciones, algunos ejemplos de estos son, el lanzamiento de misiles, exploración marina, transporte cooperativo, formación de satélites, redes de sensores, etc. Así mismo en la literatura se muestran diferentes métodos para cumplir el objetivo fundamental de control de formación, dichas estrategias de navegación son: la estructura líder-seguidor, estructura virtual, navegación basada en comportamiento y estrategias basadas en consenso. Siendo esta última la utilizada en el presente trabajo de tesis, debido a que es un enfoque no tan recurrente en la literatura, que aporta robustez, teoría de control y teoría de grafos.

1.1. Estado del arte

Antes de la revisión del estado del arte, es pertinente introducir al lector con unos conceptos básicos, primeramente entiéndase control cooperativo como a un conjunto de agentes que buscan alcanzar un objetivo en común bajo un mismo esquema de control, *rendezvous* es cuando el objetivo del control cooperativo es hacer que los agentes alcancen y permanezcan en un valor común, el cual puede ser que los agentes del equipo permanezcan en un área común o que las velocidades del equipo se estabilicen a un valor para que

se desplacen uniformemente, se hace llamar control de formación cuando el objetivo es estabilizar las distancias/posiciones relativas entre los agentes a valores predefinidos y *flocking* es un comportamiento colectivo biológicamente inspirado en cardúmenes, parvadas, manadas, etc. donde se busca evitar colisiones entre los agentes, mantener un valor común en las velocidades y mantener una distancia relativamente cercana entre los mismos.

1.1.1. Control de Formación

En [1] resuelven el problema de control de formación usando la técnica *Multi-Layer Control Scheme* (MLCS por sus siglas en inglés), el cual divide a los agentes en diferentes triángulos que permite controlarlos en pequeños grupos y guiarlos a una formación final, para ello ocupa una técnica de seguimiento de trayectoria en el baricentro de los triángulos.

Menciona en la introducción que la mayoría de los enfoques para resolver el problema de control de formación pueden ser clasificados en las siguientes estrategias de navegación: líder-seguidor, estructura virtual y navegación basada en comportamiento. Así mismo mencionan que el problema de control de formación es investigado con estrategias basados en consenso en el cual se incluyen la teoría de grafos y teoría de control.

También introduce las estructuras de control, las cuales pueden ser centralizadas o descentralizadas, en la primera una sola unidad es la encargada de calcular todas las leyes de control de todos los agentes y en la segunda cada agente es capaz de calcular su propia ley de control a través de la información que recibe de su alrededor.

Implementaron un esquema de MLCS para realizar una formación con 4 agentes, donde se forman 2 triángulos guiados individualmente por el MLCS, utilizan 3 estrategias de ponderación de la ley de control.

- *LCS: Last Control System.*
- *MCS: Mean Control Signal.*
- *WMCS: Weighted Mean Control Signal.*

Las cuales se evalúan con los índices de desempeño:

- *IAE: Integral of the Absolute Error.*

- ITAE: *Integral Time Absolute Error*.
- IASC: *Integral Absolute Signal Control*.

También propone que el siguiente paso a seguir en la investigación sea experimentar y proponer nuevas formas de ponderar la señal de control para mejorar el desempeño. Para reducir el error en el seguimiento de trayectoria, implementar un compensador dinámico en el escalón del robot y para una aplicación cercana a la realidad se debe incluir en el controlador estrategias de evasión de obstáculos.

En [2] diseñan un control por consenso adaptable basado con una red neuronal aplicado en un equipo de UAVs de una ala.

En la introducción menciona los beneficios en el campo civil y militar, menciona la aplicación de operaciones de búsqueda y rescate en un área extensa, transporte de cargas pesadas cuando un sólo UAV no puede cargarla de forma estable, seguimiento de objetivos en movimiento, observando lugares amplios en aplicaciones en movimiento, observando lugares amplios en aplicaciones de seguridad pública o cualquier área que necesite vigilancia.

Decidieron trabajar con la línea de investigación de control de formación basada en consenso debido a las propiedades de robustez y escalabilidad que añade al sistema.

Cada agente comparte su error de posición desde su posición deseada esta información es sintetizada en una ley de control que busca mantener el mismo error para la posición de todos los UAVs hasta que todos consigan su posición objetivo, si logran alcanzar y mantener el consenso en sus errores de posición, entonces se conseguirá formar y mantener la formación deseada concluye que se desarrolló con éxito el diseño de un control de formación basado en la estructura líder-seguidor y propone nuevos trabajos como un controlador híbrido adaptable así como incluir otros esquema como por ejemplo el de evitar obstáculos y regulación.

En [3] se concentran en trabajar en el consenso de movimiento y control de formación de UAVs, el algoritmo que diseñaron se rige bajo el esquema líder-seguidor, donde los líderes son integrados implícitamente en el enjambre y pueden ser afectados por la retroalimentación de navegación de otros agentes.

En la introducción menciona los campos de aplicaciones de un enjambre de

UAVs, como, por ejemplo, misiones de vigilancia y reconocimiento, monitoreo y manejo de tráfico, detección y supresión de incendios, y para estas tareas es mucho más efectivo realizarlas con un equipo de UAVs que con uno sólo, pero una tarea previa es necesaria la de desarrollar un esquema de control de formación para conseguir el trabajo cooperativo de todos los UAVs.

Menciona 3 tipos de algoritmos de control de formación, líder-seguidor, basados en estructura virtual y el método basado en comportamiento, de los cuales el más estudiado es la estructura líder-seguidor, para resolver problemas de control de formación, debido a su conveniencia y a su fácil análisis e implementación.

Proponen un algoritmo de control de formación por retroalimentación, aplicable a UAVs de alta velocidad.

También especifica que, no existen líderes explícitos en el algoritmo, ya que ningún agente en el enjambre conoce la designación de los líderes, y por ello denominan a los líderes como implícitos.

Consideran que en el enjambre existen N individuos denotados por $1, \dots, N$ y cada UAV es visto como un punto-masa desde la perspectiva de control de formación. Menciona que se ha probado anteriormente que el controlador de trayectoria horizontal tiene una constante de tiempo mucho mayor que la del controlador de altura, así que el control de formación de un enjambre puede separarse como problemas horizontal y vertical de *flocking* y ellos solamente trabajaron con consenso en movimiento horizontal.

Prueban la estabilidad de su algoritmo para control de formación de un enjambre de UAVs con líderes implícitos, y mencionan en la conclusión que el equipo de UAVs será capaz de realizar la formación siempre y cuando el grafo inicial sea conectado.

Ventajas del algoritmo, reduce el uso de comunicaciones, y hace al enjambre más adaptable y escalable.

Proponen en [4] el diseño de un control por modos deslizantes con la estrategia de líder-seguidor para múltiples UAVs, eligieron esta técnica debido a que es un método robusto para controlar sistemas no lineales y mantiene al sistema insensible a perturbaciones externas.

Introduce con las aplicaciones como detección de aéreas peligrosas, control de fronteras, inspección de líneas de transmisión eléctrica, fotografía aérea, erosión terrestre entre otros. Decidieron trabajar con cuadricóptero por las ventajas que tienen sobre las de una sola ala, despejes y aterrizajes verticales y *hovering* así como los vuelos en espacios cerrados.

Dividen el sistema dinámico en 2, unos actuando completamente y el otro sub-actuado, el controlador para ambos es diseñado como SMC (Sliding Mode Controller), la validación de su trabajo es en simulación en ROS (Robot Operating System) y la planta son 3 cuadricóptero modelo Bepop Parrot 2. Proponen implementar vuelos en espacios cerrados, duplicar los cuadricópteros e implementar la habilidad de evadir obstáculos como trabajos futuros.

En [5] proponen una solución para el problema de control de formación de un tele-operador con un sólo maestro-múltiples esclavos, donde cada robot es capaz de comunicar a sus vecinos sólo en instantes irregulares de tiempo discreto ya que no es posible garantizar la estabilidad del tele-operador en presencia de comunicaciones intermitentes.

En la introducción habla de las aplicaciones de tele-operadores, operaciones a distancia, como, manejo de tóxicos y materiales peligrosos, exploración marina y tele-cirugía.

Ellos comparan a los robots autónomos con los sistemas tele-operados marcando la principal diferencia, de que los segundos la toma de decisiones las toma un operador humano. Este tipo de robots se utilizan varios robots esclavos, dependiendo de las tareas específicas para los que sean diseñados, por ello es necesario diseñar un esquema de trabajo cooperativo donde ellos utilizaron la teoría de grafos para diseñar una topología de comunicación.

En [6] resuelven el problema de control de formación utilizando la estructura líder-seguidor, aplicado en un modelo con incertidumbres y perturbaciones externas, de vehículos de superficie sub-actuada.

Para resolver el problema de parámetros desconocidos y perturbaciones externas utilizaron un estimador basado en modos deslizantes la cual añade robustez al sistema en lazo cerrado, a las incertidumbres paramétricas y perturbaciones externas, desarrollan una función continua para sustituir la función signo en el diseño del esquema de modos deslizantes.

Como trabajos futuros proponen implementar un esquema de evasión de obstáculos, expandir la aplicación práctica a control de formación de barcos, así como formación de barcos de rescate y como planear una formación combinando la comunicación y las capacidades individuales de búsqueda.

[7] presenta el estudio de control de formación con algoritmos de consenso propuesto para 5 robots móviles. Utiliza la estructura de líder seguidor. Y el controlador es dividido en 2 capas. Uno de los *robots* es seleccionado líder, el

cual tiene acceso a la información de referencia y un controlador de modos deslizantes basado en un método de control cinemático común, el líder es diseñado para seguir una trayectoria de referencia. Los otros 4 robots son designados como seguidores, que están sujetos a errores de diseño en la formación, se propone una terminal con un control de modos deslizantes para realizar el control de formación con algoritmos de consenso. Se muestra la simulación en Matlab de dicho estudio.

1.1.2. *flocking*

En [8] solucionaron el problema de *flocking* de sistemas multi-agentes con múltiples dinámicas de segundo orden, proponen un protocolo para un sistema con agentes idénticos y lo extienden a sistemas heterogéneos, donde las dinámicas de los agentes son diferentes.

En la introducción menciona algunos problemas de *flocking* que se han trabajado anteriormente, como, *flocking* en sistemas multi-agentes con retardos, *flocking* en ambiente ruidoso y *flocking* con incertidumbres en su dinámica y menciona que para describir el movimiento de los agentes y del líder virtual trabajaron con el sistema doble integrador.

El objetivo es que los agentes lleguen al consenso en la velocidad de movimiento del líder virtual evitando colisiones y manteniendo la conexión en la red.

Para trabajos futuros proponen estudiar algoritmos de *flocking* con dinámica lineal idéntica sometida a incertidumbres paramétricas, ya que la presencia de estas incertidumbres lleva a pensar en un problema de *flocking* heterogéneo.

En [9] implementaron la metodología llamada *Brain Emotional Learning Based Intelligence Controller (BELBIC)* para un control de *flocking* en un sistema multi-agente, la metodología BELBIC es un controlador inteligente biológicamente inspirado basado en un modelo computacional de aprendizaje emocional en cerebros de mamíferos, las capacidades de aprendizaje del BELBIC mejora las capacidades del sistema multi-agente cuando este se encuentra en presencia de ruido y/o incertidumbres en el sistema.

En la introducción define *flocking* como un gran número de entidades autopulsadas en movimiento colectivo la cual tiene 3 reglas básicas separación, alineación y cohesión. Menciona que en la literatura se han centrado en trabajar en:

- La optimización de la fuerza de control.
- La robustez para tratar con ruidos ambientales.
- La capacidad de manejar incertidumbres en el modelo.

Pero la mayoría de los trabajos recientes no se diseñan teniendo en cuenta propiedades multiobjetivo, y que al tener en cuenta estas propiedades en

paralelo a resolver el problema de *flocking* aumenta la complejidad computacional de tal forma que no es posible una implementación en tiempo real, de allí por qué vieron la posibilidad de aplicar al sistema multi-agente el controlador BELBIC en *flocking* ya que tiene propiedades de resolver problemas con múltiples objetivos y de alto nivel computacional. La cual hace factible su implementación en tiempo real.

Presentan en [10] un algoritmo predictivo de un modelo distribuido para un sistema multi-agente con retraso en las comunicaciones entre los agentes, los cuales están dinámicamente desacoplados e individualmente calculan sus leyes de control utilizando la información de sus vecinos y la propia. La función costo de este método penaliza los estados, esfuerzos de control, el error entre los estados predecidos y los deseados. En la introducción menciona las leyes de Raynolds que propuso en 1987 en [11] :

- Cohesión: Mantener una distancia constante entre los *flockmates*.
- Separación: Evitar colisiones entre los *flockmates*.
- Alineación: Mantener consenso en la velocidad de los *flockmates*.

Este trabajo está inspirado en el hecho de que al momento de compartir información de los estados entre los agentes existen retrasos. Inspirador por el trabajo de [12] para quienes proponen una estrategia de *Distributed Model Predictive Control (DMPC)* de estabilizar agentes, retraso de comunicaciones limitados entre los agentes en un sistema multi-agente no lineal de tiempo continuo, utilizan un mecanismo de espera combinado con un DMPC para que la información retrasada pueda sustituirse con el estado predicho anterior.

1.1.3. *Rendezvous*

Rendezvous es cuando el objetivo del control cooperativo es hacer que los agentes alcancen y permanezcan en un valor en común, el cual puede ser un área predefinida con un radio r donde los diferentes agentes permanezcan allí, que las velocidades de los agentes permanezcan en un valor igual para todos

y se desplacen a esa velocidad hacia puntos de desplazamiento predefinidos, donde en ambos objetivos no se tiene una comunicación activa entre ellos, sino más bien, se definen áreas circulares para los agentes donde los demás agentes no pueden entrar, esto para evitar colisiones.

Cada agente es capaz de obtener las posiciones de todos los agentes dentro de su región de censado. El problema de *rendezvous* consiste en desarrollar estrategias de control locales para cada agente, sin tener comunicaciones activas entre los agentes. Existen 2 sub-intervalos en la metodología para resolver el problema de *rendezvous*, la primera es el periodo de censado, que es un periodo de tiempo donde el agente se encuentra en un estado estacionario adquiriendo la información de los demás agentes, el segundo intervalo es llamado el periodo de maniobra, donde el agente se mueve de su posición actual a su siguiente punto predefinido.

Aplican en [13] el control activado por eventos de sistemas multi-agentes al control de *rendezvous* de sistemas multi-agente.

Primero utilizaron el modelo multi-agente para que el punto de convergencia sea especificado por la transformación proporcional del punto de equilibrio del estado original de los agentes y con este algoritmo aseguran que la posición de todos los agentes puedan converger en la posición de *rendezvous* deseada. Posteriormente agregan la estrategia de control activada por eventos para el control de *rendezvous* multi-agente.

En la introducción menciona que el mecanismo activado por eventos reduce la cantidad de información procesada y el ancho de banda del sistema basado en una condición particular.

Resuelven el problema de control de *rendezvous* en un enjambre de agentes, donde agregan el mecanismo activado por eventos.

El objetivo de *rendezvous* de multi-agentes es que todos los agentes converjan en un punto específico.

Debido a la revisión del estado del arte se visualizó la oportunidad de trabajar con sistemas multi-agentes aplicado a un equipo de cuadricópteros incluyendo perturbaciones ambientales. En los artículos [6], [14], [7], [2], [3], [5], descritos más a detalle anteriormente, manejan el enfoque líder-seguidor utilizando diferentes técnicas de control y en su mayoría modelos no lineales, para solucionar el problema de control de formación, ya que como mencionan en [3] es una estructura conveniente por su sencillez y de más fácil implementación y quizá por ello es el enfoque más visto

en la literatura actual. Se observa la oportunidad de utilizar algoritmos de consenso modificados para realizar control de formación en sistemas con múltiples agentes debido a que es un enfoque no tan recurrente en la literatura, que aporta robustez, teoría de control y teoría de grafos como lo mencionan en, [1] y [2], por último, se decidió por añadir un esquema de perturbaciones al sistema de múltiples agentes, ya que en la consulta del estado del arte se observó que pocos artículos dirigen su tema de investigación hacia la solución de esa problemática. A continuación, se hace un pequeño resumen visual de lo tratado anteriormente en las consultas de la literatura actual, en la tabla siguiente podemos observar las diferentes topologías de comunicación, aquellos artículos que agregan perturbaciones a sus esquemas, las diferentes técnicas de control y el tipo de modelo matemático que utilizan en los artículos consultados.

1.1.3.1. Resumen del estado del Arte

Paper	Tipo de control	Modelo Lineal o no lineal	Perturbaciones	Simulación y/o experimental	Topología
En [Sun et. Al., 2017]	Modos deslizantes	No lineal	✓	Simulación	Lider-seguidor.
En [Du et. Al., 2017]	Controlador de altura y formación de tiempo finito	No lineal	✗	Simulación	Árbol directo y fuertemente mente conectado, indirecto.
En [Ilie and Namerikawa, 2017]	MPC Distribuido	Lineal	✗	Simulación	Lider-seguidor, con conexión indirecta.
En [Zhao et. Al., 2017]	Modos deslizantes	No lineal	✗	Simulación	Lider-seguidor, balanceado y fuertemente conectado, indirecto.
En [Wang et. Al., 2017]	Control Adaptativo	No lineal	✗	Simulación	Topologías cambiantes conexión indirecta.
En [Guzey, 2017]	Red Neuronal	No lineal	✗	Simulación	Lider-seguidor, árbol balanceado.
En [Lee and Ahn, 2016]	Control Distribuido	No lineal	✗	Simulación	Árbol, indirecto y fuertemente conectado.
En [He et al., 2018]	Control Distribuido	No lineal	✗	Simulación	Lider-seguidor.
En [Liu et al., 2017]	Redes Neuronales	No lineal	✓	Simulación	No se hace mención.
En [Nguyen and La, 2017]	Control Distribuido	No lineal	✗	Simulación	Interconexión indirecta.
En [Yang et al., 2012]	Control Optimo	No lineal	✓	Simulación	No se hace mención.
En [Redrovan and Kim, 2018]	Modos deslizantes	No lineal	✗	Experimental	No se hace mención.
En [Xian et al., 2018]	Control Adaptativo	No lineal	✗	Simulación	Lider-seguidor, conexión indirecta.
En [Pacheco et al., 2018]	Multi-Layer Control Scheme	No lineal	✗	Simulación	No se hace mención.
En [Guzey, 2018]	Red Neuronal	Lineal	✗	Simulación	Lider-seguidor
En [Yang et al., 2018]	Esquema de control Following	No lineal	✓	Simulación	Lider-seguidor, topología fuertemente conectado (intermitente)

Tabla 1.1: Puntos importantes en la revisión del estado del arte.

1.2. Hipótesis

Es posible resolver el problema de control de formación con perturbaciones en un sistema multi-agente de vehículos aéreos no tripulados bajo un esquema de control distribuido descentralizado.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar un control distribuido para resolver el problema de control de formación basado en sistemas multi-agentes con perturbaciones.

1.3.2. Objetivos Específicos

- Definir el sistema multi-agente de segundo orden.
- Diseñar un esquema de control interno para estabilizar a la planta multi-agente.
- Diseñar una metodología de sistemas multi-agentes con comunicación rígida.
- Comparar los resultados obtenidos con diferentes algoritmos de control de formación.

1.4. Planteamiento del problema

En la actualidad, con ayuda de las nuevas tecnologías, se tiene la posibilidad de realizar diversas tareas de modo remoto, como por ejemplo el lanzamiento de misiles, la vigilancia de áreas de interés, misiones de reconocimiento de grandes extensiones de terreno, la exploración marina, la fotografía aérea, el resguardo de fronteras o el transporte de cargas pesadas, por mencionar unos pocos, por lo general este tipo de tareas son desempeñadas por múltiples equipos, como pueden ser, un grupo de vehículos autónomos, ya sean aéreos, terrestres o acuáticos, que necesariamente deben trabajar entre sí para alcanzar un objetivo en común, el tópico del cual principal del tema de tesis es precisamente ese, diseñar un esquema de control que permita a un grupo de drones trabajar en conjunto con un objetivo en común, el cual deberá realizar y mantener formaciones prediseñadas utilizando algoritmos de consenso modificados para control de formación. Al momento de querer realizar control de formación desde el enfoque de sistemas multi-agentes, es importante tomar en cuenta ciertas consideraciones. La presente problemática plantea que la comunicación entre los agentes sea una comunicación rígida, esto quiere decir, que todos los agentes se encuentran mandando y recibiendo información de los estados de sus agentes vecinos sin interrupciones ni cambios en la topología a lo largo del tiempo, debido a esto, no es necesario utilizar el enfoque líder-seguidor, que es el enfoque que se encontró como el más utilizado al realizar el estudio del arte. Se consideran conocidas las posiciones iniciales de todo el equipo multi-agente y se plantea que, a través de una topología de comunicación fuertemente conectada y algoritmos de consenso, el sistema multi-agente establezca sus distancias/posiciones relativas entre los agentes a valores predefinidos.

Otro punto a considerar, es que, previo a realizar la formación o implementar los algoritmos de consenso para aplicarlos al sistema multi-agente, se debe poder estabilizar el sistema, en este caso se requiere hacer control de formación en un equipo de drones, por lo tanto, se debe comenzar por diseñar un control interno que establezca los cuadricópteros, y el cual, permita manipular la posición de los mismos en el espacio, posteriormente, se debe diseñar una transformación que permita considerar al sistema desde el enfoque de un doble integrador, y por último, implementar los algoritmos de consenso que permitan realizar las formaciones deseadas sin importar las posiciones

iniciales del equipo de drones. Finalmente se pretende agregar perturbaciones al esquema de control de formación que afecte a los agentes, ya que esto hace que el caso de estudio se aproxime a un ambiente experimental más realista.

Ya se hizo mención que el caso de estudio será en un equipo de drones, específicamente en un grupo de cuadricópteros, en la figura (1.1) se muestra el caso hipotético de que tres agentes se encuentren en sus posiciones iniciales como en una fila.



Fig. 1.1: Posiciones iniciales.

Y se requiere realizar con estos 3 cuadricópteros la formación de un triángulo, en la figura (1.2) se muestra en un diagrama a bloques la problemática previamente planteada, donde se busca que un algoritmo de consenso para control de formación genere internamente en cada cuadricóptero, una ley de control capaz de ubicar a cada dron en el punto/área deseada, utilizando la información de los estados de los agentes vecinos y los propio los cuales son las posiciones relativas, y velocidades de todo los cuadricópteros, esto desde el punto de vista del control basado en procesamiento descentralizado.

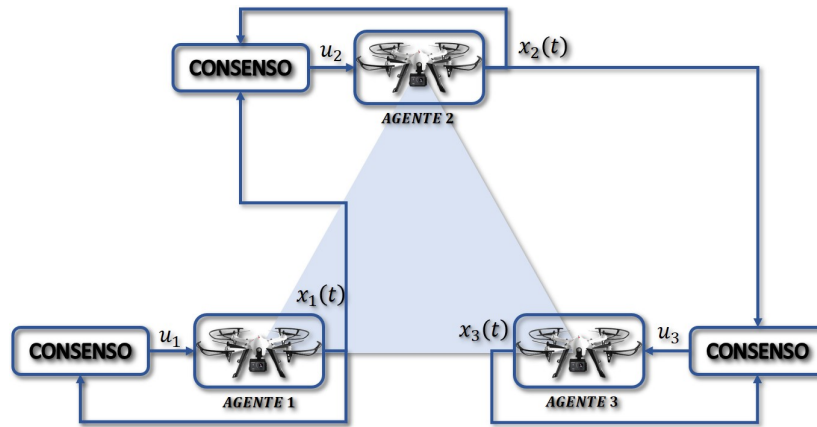


Fig. 1.2: Diagrama general para realizar control de formación con 3 agentes.

1.5. Propuesta de solución

Se propone resolver el problema de control de formación desde el enfoque de sistemas multi-agentes, el cual no depende de un líder, y aporta conceptos como teoría de control y teoría de grafos. En sistemas multi-agentes, cada agente se comunica con sus vecinos mediante una topología de comunicación, la cual está previamente diseñada, y se propone en este caso de estudio sea rígida y fuertemente conectada, más adelante en la sección pertinente se hablará más a fondo del diseño de las comunicaciones entre los agentes. El objetivo del algoritmo de control de formación es mantener distancias entre los agentes para formar figuras o formaciones y que éstas se mantengan a lo largo del tiempo, así mismo que, al realizar la figura/formación final, los agentes no colisionen.

Se asume que el sistema en el cual se va a trabajar es un equipo de cuadricópteros, los cuales son capaces de transmitir y recibir información de sus posiciones relativas y sus velocidades de desplazamiento, se propone trabajar bajo un esquema descentralizado, esto quiere decir, que cada dron es capaz de calcular su propia ley de control con la información recibida y alcanzar su posición requerida en la formación.

Un dron no es estable, así que previo a emplear algoritmos de consenso pa-

ra resolver el problema de realizar una formación en sistemas multi-agentes, se deben realizar 2 actividades previamente, la primera: se tiene que diseñar un controlador que permita estabilizar el sistema, de forma que se puedan controlar tanto los ángulos de los drones como su posición en el espacio y el segundo paso es relacionar las salidas de los algoritmos de consenso en entradas para dicho controlador.

La figura (1.3) resume lo anterior, el algoritmo de consenso necesita los estados del propio agente (Posiciones y velocidades), los estados de los agentes vecinos y las distancias necesarias para realizar la formación, posteriormente, se relaciona la salida del algoritmo para ser una entrada del controlador interno que permite estabilizar y mover el cuadricóptero, y el algoritmo se retroalimenta con los estados del propio agente para cerrar el lazo de control y a la vez esos mismos estados son enviados a sus agentes vecinos.

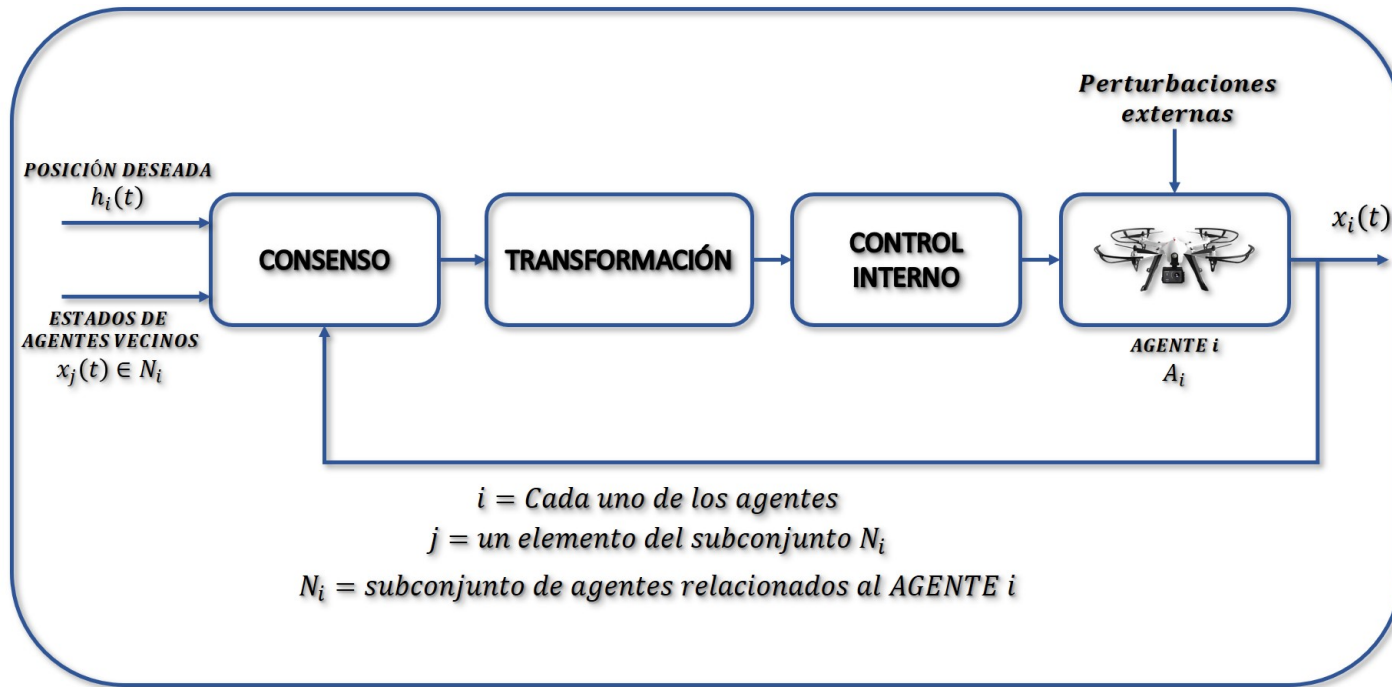


Fig. 1.3: Diagrama a bloques de la propuesta de solución.

Diseño de un esquema de control de formación para un sistema multi-agente.

2.1. Algoritmos de consenso

Los algoritmos de consenso aplicados en sistemas multi-agentes tienen como objetivo alcanzar y mantener un valor en común para todos los agentes del sistema. Los sistemas multi-agentes pueden conseguir un objetivo global usando sensores, una unidad de procesamiento centralizada o descentralizada, intercambio de información a través de topologías de comunicación y controladores. En el control cooperativo, los agentes comparten información entre cada uno de ellos. El control cooperativo de sistemas multi-agentes puede ser clasificado como se muestra a continuación:

- Problemas de control cooperativo sin formación: asignación de roles, entregas automáticas de paquetes pesados, tareas individuales o independientes, control de tráfico aéreo, búsquedas cooperativas y sincronizadas
- Problemas de control de formación: Operaciones de vigilancia y misiones de reconocimiento, el vuelo de UAVs, auto ensamble de conexiones

de redes móviles, vehículos submarinos autónomos, naves espaciales, aeronaves, satélites y sistemas de autopistas automáticas.

Un protocolo de consenso es una regla de comunicación que especifica el intercambio de información dentro de la red entre los agentes y sus vecinos.

2.1.1. Consenso

En el área de control cooperativo de sistemas multi-agentes, consenso es un problema fundamental, que está relacionado a control de formación, *flocking*, estimación distribuida, etc. Consenso significa que un equipo de agentes alcanza y coincide en un valor en común interactuando vía sensores y una red de comunicaciones. Los algoritmos de consenso existentes pueden ser categorizados dentro de 2 clases llamadas Consenso sin un líder y Consenso con líder. Este último también llamado consenso líder-seguidor o seguimiento distribuido. Cuando se habla del problema de líder-seguidor en consenso se trabaja bajo un solo líder. Sin embargo, algunas aplicaciones prácticas, es necesaria la existencia de más de un líder en el sistema. En presencia de múltiples líderes, se incrementa el problema de contención. El problema de contención quiere decir que los líderes conducen a sus seguidores a un espacio geométrico delimitado, un área donde los seguidores maniobren sin colisionar con los demás grupos.

2.1.2. Control de formación

Comparando control de formación con el problema de consenso, donde los estados finales de todos los agentes convergen a un valor único, el objetivo final esperado de los estados finales de todos los agentes pueden ser más diversos bajo el escenario de control de formación. El objetivo de control de formación es el estabilizar las distancias/posiciones relativas entre los agentes a valores predefinidos. Se puede categorizar control de formación de dos formas, en producir formaciones y en el seguimiento de formaciones. Producir formaciones es cuando el diseño del algoritmo para control de formación de un grupo de agentes se hace con el objetivo de realizar una figura geométrica pre definida en ausencia de una referencia para el equipo, y el seguimiento de formaciones se refiere al mismo objetivo, pero siguiendo una referencia

para el grupo pre diseñada. Cuando existe una referencia para el sistema multi-agente, es usualmente mucho más desafiante lograr el objetivo final en comparación de cuando se realiza control de formación sin seguimiento de trayectorias.

2.1.3. Control basado en procesamiento centralizado y descentralizado.

La forma en la cual la información de los estados de los diversos agentes en el sistema multi-agente es procesada, se clasifica como procesamiento centralizado o descentralizado. El concepto centralizado, quiere decir, que la información de las posiciones, velocidades y/o aceleraciones de los agentes es mandada a una unidad de procesamiento (Separada de los agentes), como se muestra en la figura (2.1), la cual recibe esta información, y con ella, calcula la actualización de la ley de control de todos los agentes y posteriormente la manda a cada uno de ellos, los aspectos positivos de este enfoque son que los agentes solo mandan y reciben información y no la procesan, lo cual hace que su complejidad a nivel de *hardware* sea mínima, pero esto hace que la unidad de procesamiento tenga una mayor carga informática, si a esto le agregamos un sistema multi-agente con una gran cantidad de elementos, entonces el procesamiento de la unidad centralizada puede verse comprometida, lo que haría necesario añadir más unidades de procesamiento o disminuir la cantidad de agentes en el sistema.

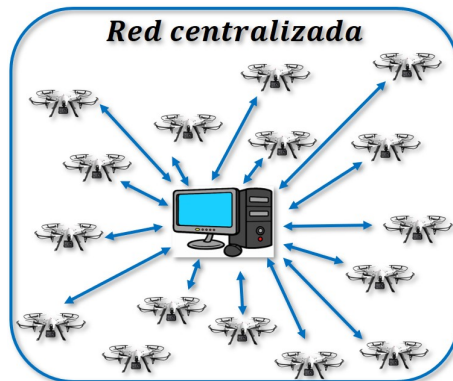


Fig. 2.1: Ejemplo de una red con unidad de procesamiento centralizada.

La red descentralizada fig. (2.2), se basa en que todos los agentes posean una unidad capaz de recibir la información de los estados de los agentes vecinos (Montada en los mismos), procesarla con la propia y generar la actualización de su ley de control, aunque puede parecer un procedimiento mejor que el de la unidad centralizada, la unidad descentralizada tiene un mayor costo económico, ya que se tiene que agregar a cada agente una unidad más compleja de procesamiento, con la principal ventaja de poder agregar más agentes al sistema de ser requerido.

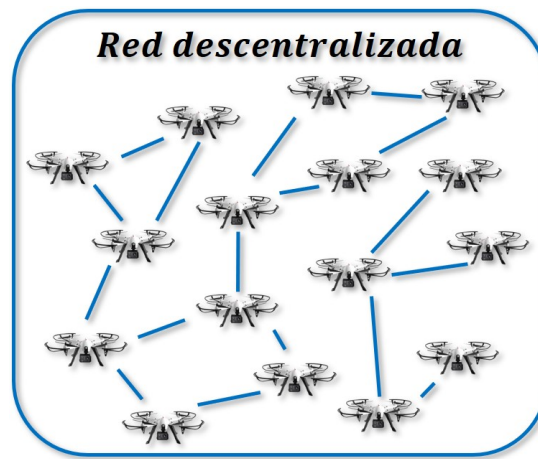


Fig. 2.2: Ejemplo de una red con unidad de procesamiento descentralizada.

2.2. Diseño para resolver el problema de control de formación en sistemas multi-agentes con representación de sistemas doble integrador.

2.2.1. Posicionamiento del problema

Considérese una red con un número N de agentes, con dinámicas iguales (Sistema multi-agente homogéneo). Las dinámicas de los i -ésimos agentes está descrita por:

$$\begin{aligned}\dot{x}_i(t) &= Ax_i(t) + Bu_i(t), \\ y_i(t) &= Cx_i(t), \\ i &= 1, \dots, N.\end{aligned}\tag{2.1}$$

donde x_i representa los estados, u_i la ley de control y y_i la salida del i -ésimo agente, A , B , C son matrices constantes con dimensiones compatibles, e igualmente el sistema debe ser controlable y observable.

Se ha definido consenso hasta ahora, a cuando un equipo de agentes alcanza y coincide en un valor en común, pero matemáticamente el objetivo de un protocolo de consenso es expresado en el siguiente límite:

$$\lim_{t \rightarrow \infty} \|x_i(t) - x_j(t)\| \rightarrow 0 \quad \forall i = 1, \dots, N.\tag{2.2}$$

La ley de control distribuida que cumple dicho objetivo se denomina como protocolo de consenso. El protocolo de consenso más simple se describe por:

$$u_i = - \sum_{j \in N_i} (x_i(t) - x_j(t)) \quad i = 1, \dots, N.\tag{2.3}$$

En otras palabras, para el sistema de N agentes descrito en (2.1) es posible implementar el protocolo de consenso (2.4) y así alcanzar un valor común cuando se cumpla (2.2) y cuando esto sucede se dice que el sistema ha alcanzado el consenso.

2.2.2. Extensión o aplicación para control de formación

La variación que ha de hacerse al protocolo (2.4) y este extienda para resolver el problema de control de formación en una red de agentes descrita como en (2.1), es incluir una matriz definida como

$H = (h_1, h_2, \dots, N) \in \mathbf{R}^{n \times N}$, que es una matriz constante que estructura la formación final (en (2.3.3) se describe su diseño), $h_i \in \mathbf{R}^n$ es la variable en la formación para el agente i . Se reestructura el protocolo de consenso (2.4) para resolver el problema de control de formación como:

$$u_i = - \sum_{j \in N_i} (x_i(t) - h_i - (x_j(t) - h_j)) \quad i, j = 1, \dots, N. \quad (2.4)$$

Entonces los agentes descritos en (2.1) alcanzaran y mantendrán formaciones deseadas dadas por una matriz $H = (h_1, h_2, \dots, N)$ cuando se cumpla el siguiente límite:

$$\lim_{t \rightarrow \infty} \|(x_i(t) - h_i) - (x_j(t) - h_j)\| \rightarrow 0 \quad \forall i, j = 1, \dots, N. \quad (2.5)$$

2.3. Otros algoritmos de control de formación

En el escenario donde se tengan disponibles tanto las velocidades como las posiciones de los agentes, es posible diseñar un algoritmo de control de formación como se muestra en [15], ellos presentan un algoritmo activado por eventos, pero si se reescribe el algoritmo como en (2.6) es posible resolver el problema de control de formación en un sistema del mismo tipo que (2.1)

$$u_i(t) = - \sum_{j \in N_i} [(p_i - p_j - C(h_i - h_j))] - \sum_{j \in N_i} (v_i - v_j) \quad i = 1, \dots, N. \quad (2.6)$$

Las restricciones que ocupa este algoritmo son conocer todos los estados del sistema, lo cual en una planta física es muy difícil o imposible de tener, este algoritmo no utiliza la teoría de grafos, lo que sucede es, que cada agente en la formación manda y recibe información de todos los demás agentes del sistema, y esto tampoco es práctico en un escenario experimental, ya que sí se incrementa el número de agentes, la transmisión de información puede complicar el escenario, o inclusive, puede ser imposible transmitir tanta información a la velocidad requerida por los agentes para realizar y mantener las formas deseadas.

Para disminuir la transmisión de información se incluye la teoría de grafos, donde si se define un grafo \mathcal{G} que incluya un árbol expandido directo se alcanzara una formación final deseada cuya configuración se encuentra en una matriz H y es una formación rígida, se modifica el algoritmo de control (2.6) de la siguiente forma:

$$u_i = -a_{ij} \sum_{j \in N_i} [p_i - p_j - C(h_i - h_j)] - a_{ij} \sum_{j \in N_i} (v_i - v_j) \quad (2.7)$$

2.3.1. Algoritmo de consenso tipo observador modificado para control de formación

Los protocolos de consenso aplicados a control de formación (2.6) y (2.7), utilizan los estados de posiciones y velocidades del sistema, asumiendo que se encuentran disponibles, pero en [16] muestran un protocolo tipo observador donde las velocidades de los diferentes agentes no son utilizadas para el cálculo de la ley de control.

$$\dot{r}_i = (A - BF)r_i + cL \sum_{j=1}^N a_{ij} [C(r_i - r_j) - ((Y_i - Y_j) - C(h_i - h_j))] \quad (2.8)$$

$$u_i = -Fr_i \quad (2.9)$$

$$H = [h_1, h_2, h_3, \dots, h_N]^T \quad (2.10)$$

Donde A , B y C son las matrices de la representación de espacio de estados del sistema doble integrador, F y L son ganancias de retroalimentación, c es la ganancia de acoplamiento, H es la matriz que contiene las distancias relativas deseadas para todos los agentes en la formación y h_i es el vector de distancias deseadas del i -ésimo agente, Y_i representa el vector que contiene las salidas del sistema, r_i es el vector de estados internos del observador, u_i es la ley de control.

Teorema 1 [16] *Se asume que el grafo de comunicación \mathcal{G} contiene un árbol expandido directo. Los agentes (2.1) alcanzan la formación H bajo el protocolo (2.8), si todas las matrices $A-BF$ y $A - c\lambda_i LC$, $i = 2, \dots, N$, son Hurwitz, y $Ah_i = 0 \forall i = 1, \dots, N$, donde λ_i , $i = 2, \dots, N$, son los eigenvalores diferentes de cero de la matriz laplaciana (\mathcal{L}).*

2.3.2. Diseño de ganancias para realizar control de formación por medio del algoritmo tipo observador.

A continuación, se muestra el diseño del protocolo (2.8), necesario para realizar control de formación en los sistemas de la forma mostrada en (2.1).

Se asume que el sistema lineal es controlable y observable, el protocolo tipo-observador puede ser construido bajo las siguientes condiciones.

- Los polos de $A + BF$ deben encontrarse al lado izquierdo de $x = -\alpha$, y entonces utilizando la fórmula de Ackermann se encuentra la matriz F .
- Resuelve la siguiente LMI.

$$A^T Q + QA - 2C^T C + 2\alpha Q < 0$$
para la solución donde $Q > 0$. Después se obtiene la matriz L de la siguiente forma

$$L = -Q^{-1}C^T$$
- Se puede seleccionar la ganancia de acoplamiento $c > c_{th}$ dada por

$$c_{th} = \frac{1}{\min \operatorname{Re}(\lambda_i(\mathcal{L}))} \quad i = 2, \dots, N.$$

2.3.3. Distancias relativas deseadas entre los agentes.

Como ya se mencionó anteriormente, se define una matriz H de la forma $H = [h_1, h_2, \dots, h_N]^T$ donde cada vector h_i contiene las coordenadas cartesianas del agente i -ésimo. Para formar la matriz H , primeramente, se define el número de agentes N , el número de dimensiones a considerar, que normalmente son de 2 o 3 dimensiones, y por último se definen las distancias deseadas que tendrán los agentes en la formación final deseada. A continuación, se muestra en la figura (2.3) un cuadrado trazado en 2 dimensiones con las coordenadas de cada agente.

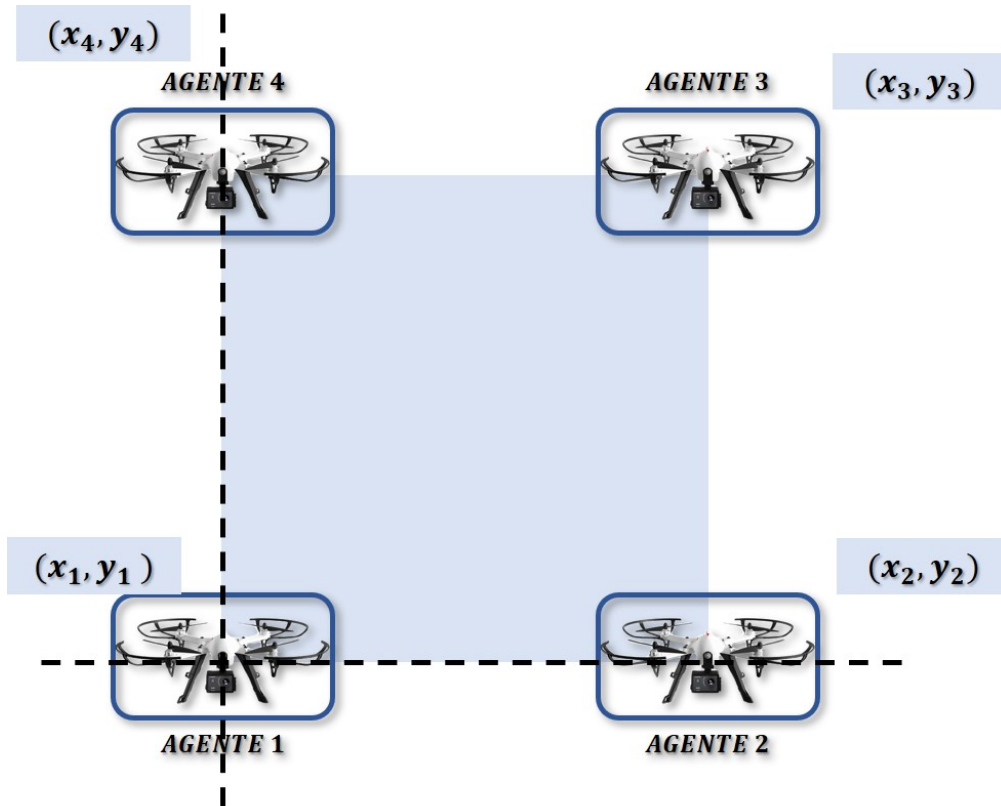


Fig. 2.3: Ejemplo de distancias relativas en una formación de 4 drones.

Considerando un número de agentes $N=4$, un espacio de dos dimensiones, con las coordenadas mostradas en la imagen (2.3), se puede definir la matriz H de la siguiente forma:

$$H = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.11)$$

sí en otro caso se manejarán 3 dimensiones, la matriz H quedaría de la forma:

$$H = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.12)$$

donde los ceros en ambos casos se añaden para permitir la compatibilidad de dimensiones en el algoritmo (2.8).

Para $H = [h_1, h_2, \dots, h_N]^T$ en un espacio de dos dimensiones, la matriz H de forma general es

$$H = \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (2.13)$$

y en un espacio de 3 dimensiones su forma general es

$$H = \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ z_1 & z_2 & \dots & z_N \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (2.14)$$

Diseño del esquema de control de formación para sistemas multi-agentes.

3.1. Algebra básica de teoría de grafos

Un equipo de agentes interactúa entre ellos por medio de comunicación o redes de sensores para conseguir objetivos en conjunto. Es conveniente modelar este intercambio de información entre los agentes, por medio de grafos directos o indirectos. Un grafo directo \mathcal{G} es un par $(\mathcal{V}, \mathcal{E})$, donde $\mathcal{V} = \{v_1, \dots, v_N\}$ es un conjunto no vacío de nodos y $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ es un conjunto de bordes de pares ordenados de nodos, llamados bordes. Un grafo ponderado asocia un peso con cada borde en el grafo. Un subgrafo $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ de \mathcal{G} es un grafo tal que $\mathcal{V}_s \subseteq \mathcal{V}$ y $\mathcal{E}_s \subseteq \mathcal{E}$. Bucles propios de la forma (v_i, v_i) no se aceptan a menos que se especifique. El borde (v_i, v_j) en el conjunto de bordes \mathcal{E} denota que el agente v_j puede obtener información desde el agente v_i , pero no necesariamente en viceversa. Para un borde (v_i, v_j) , el nodo v_i es llamado el nodo padre, v_j el nodo hijo, y se dice que v_i es un vecino de v_j . El conjunto de vecinos del nodo v_i está denotado como \mathcal{N}_i , cuya cardinalidad es llamada grado del nodo v_i .

Un grafo es definido como balanceado cuando tiene el mismo número de bordes de entrada y de salida para todos los nodos (El borde (v_i, v_j) se dice que sale con respecto del nodo v_i y entra con respecto a v_j). Un grafo con la propiedad de $(v_i, v_j) \in \mathcal{E}$ implica que $(v_j, v_i) \in \mathcal{E}$ para cualquier $v_i, v_j \in \mathcal{V}$ se les llama indirecto, cuando el borde (v_i, v_j) denota que los agentes v_i y v_j

pueden obtener información de cada uno.

Un camino directo del nodo v_{i1} al nodo v_{il} es una secuencia de bordes ordenados de la forma $(v_{i_k}, v_{i_{k+1}})$, $k = 1, \dots, l - 1$. Un camino indirecto en un grafo indirecto es definido de forma analógica. Un ciclo es un camino directo que comienza y termina en el mismo nodo. Nótese que, para un grafo indirecto, las conexiones fuertemente conectadas se les llama sencillamente conectadas. Un grafo directo es completo si existe un borde de cualquier nodo a todos los demás nodos. Un árbol indirecto es un grafo indirecto donde todos los nodos pueden conectarse por un solo camino indirecto. Un árbol directo, es un grafo directo en el cual cada nodo tiene un nodo padre exceptuando un nodo, llamado raíz, el cual no tiene nodo padre, y tiene caminos directos a todos los demás nodos. Un árbol directo es definido como expandido cuando se conecta con todos los nodos en el grafo. Puede ser demostrado que esto implica que existe al menos un nodo raíz conectado con un camino simple a todos los demás nodos. Se dice que un grafo tiene o contiene un árbol expandido directo si un subconjunto de bordes forma un árbol expandido directo. Esto es equivalente a decir que el grafo tiene al menos un nodo con un camino directo a todos los demás nodos. Para grafos indirectos, la existencia de un árbol expandido directo es equivalente a estar conectado. Sin embargo, en grafos indirectos, la existencia de un árbol expandido directo es una condición más débil que estar fuertemente conectado. Un grafo fuertemente conectado contiene al menos un árbol expandido directo [16].

3.1.1. Ejemplos de grafos

Un borde es denotado por una línea con una flecha direccional mientras un borde indirecto es denotado por una línea con puntas de flecha en ambos extremos, dicho en otras palabras, lo que denota la dirección en la cual fluye la transmisión de información de un agente a otro, es llamado borde y un borde indirecto denota 1 conexión donde 2 agentes mandan y reciben información entre ellos, mientras que el borde directo sólo manda información de un agente y otro la recibe. Como anteriormente se definió, el caso de estudio son los cuadricópteros, por ello en las figuras (3.1, 3.2, 3.3 y 3.4) pequeños cuadricópteros denotan los nodos o agentes en el sistema.

El primer ejemplo se muestra en la figura (3.1), un grafo indirectamente conectado, se le conoce de esta forma porque todas las conexiones en esta configuración se realizan con bordes bidireccionales, y como ya se hizo mención en el principio de este capítulo, para grafos indirectos, la existencia de un árbol expandido es equivalente a un grafo conectado.

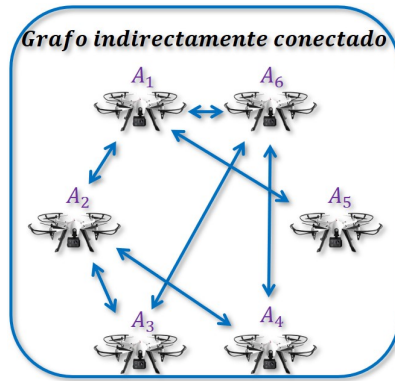


Fig. 3.1: Ejemplo de un grafo indirecto conectado.

En la figura (3.2), se muestra un grafo fuertemente conectado, donde fuertemente conectado quiere decir que el grafo es directo y que en él existe al menos un árbol expandido directo y nótese que la mayoría de conexiones son directas.

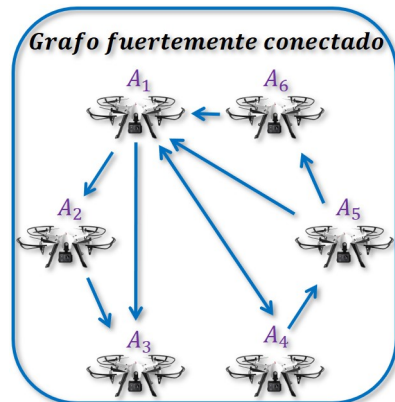


Fig. 3.2: Ejemplo de un grafo fuertemente conectado.

En la figura (3.3) se muestra un grafo balanceado y fuertemente conecta-

do, la diferencia que presenta este ejemplo a comparación del mostrado en la figura (3.2) es que es balanceado, esta característica se presenta cuando cada nodo o agente dentro del sistema tiene las mismas conexiones de salida que de entrada, nótese que en el ejemplo siguiente cada agente tiene 2 conexiones de salida y 2 de entrada de información.

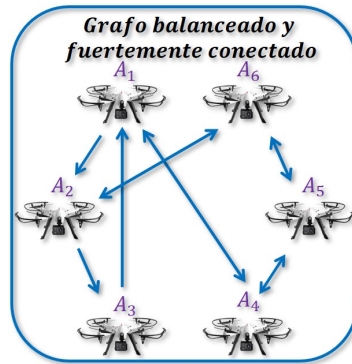


Fig. 3.3: Ejemplo de un grafo balanceado y fuertemente conectado.

Para el ejemplo siguiente es necesario definir primero el concepto de árbol directo, el cual es: un grafo directo donde todos los agentes tienen exactamente un nodo padre exceptuando uno, quien es llamado raíz y todo el grafo tiene conexiones directas. Un árbol directo se denomina como expandido, cuando conecta todos los agentes del sistema, como se muestra en la figura (3.4).

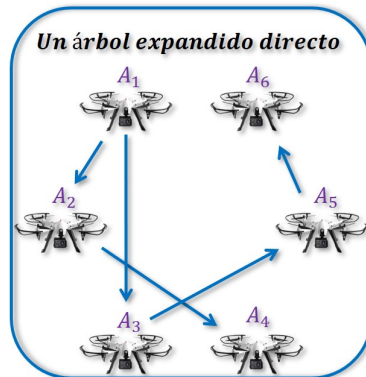


Fig. 3.4: Ejemplo de un árbol expandido directo.

3.1.2. Matriz de adyacencias y matriz laplaciana

La matriz de adyacencias está definida como $\mathcal{A} = [a_{ij}] \in \mathcal{R}^N$ asociada con el grafo directo \mathcal{G} , y los elementos de la matriz se determinan como, $a_{ii} = 0$, a_{ij} es un valor positivo si $(v_j, v_i) \in \mathcal{E}$, de otra forma $a_{ij} = 0$. La matriz de adyacencias de un grafo indirecto es definido de forma análoga excepto que $a_{ij} = a_{ji}$ para toda $i \neq j$. Nótese que a_{ij} denota el peso del borde $(v_j, v_i) \in \mathcal{E}$. Si los pesos no son relevantes, entonces, $a_{ij} = 1$ si $(v_j, v_i) \in \mathcal{E}$. La matriz Laplaciana $\mathcal{L} = [\mathcal{L}_{ij}] \in \mathcal{R}^{N \times N}$ del grafo \mathcal{G} y está definida como $\mathcal{L}_{ii} = \sum_{j \neq i} a_{ij}$ y $\mathcal{L}_{ij} = -a_{ij}$, $i \neq j$ [16].

Dicho en otras palabras, la matriz de adyacencias contiene solamente valores binarios, cada fila representa un agente, la matriz es cuadrada de dimensiones $N \times N$, la diagonal principal es una diagonal de ceros, cada cero representa que los agentes (v_i, v_j) no tienen vecindad, mientras que los 1 representan que si la tienen y que el agente i obtiene información de esos agentes j , las filas representan al agente i y las columnas al agente j .

Para los grafos de las figuras (3.1, 3.2, 3.3 y 3.4) las correspondientes matrices de adyacencias son las siguientes respectivamente.

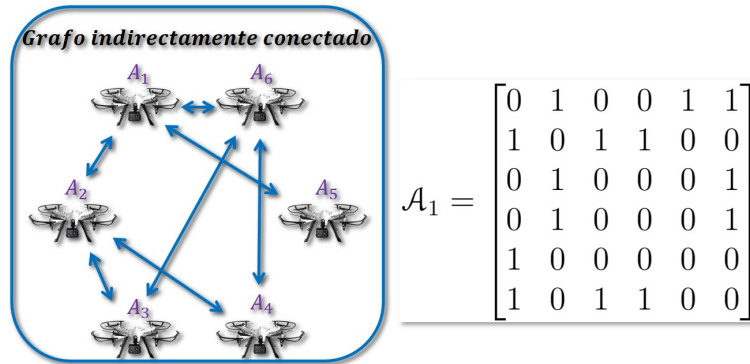


Fig. 3.5: Topología de comunicación y matriz de adyacencias equivalente para el grafo indirectamente conectado.

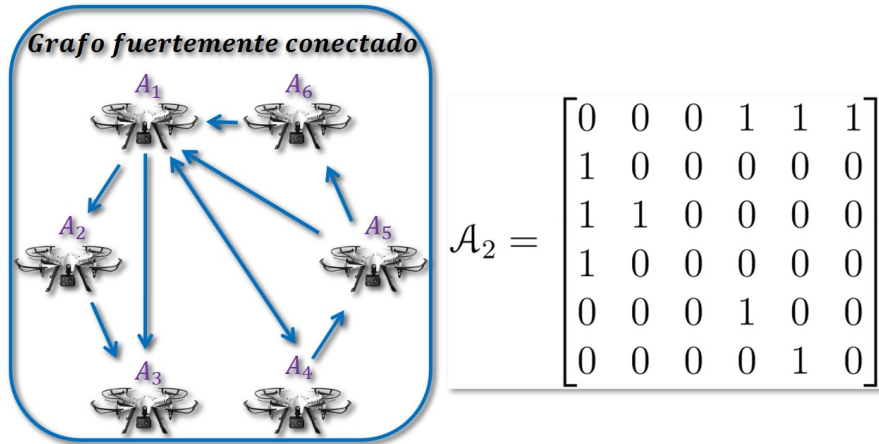


Fig. 3.6: Topología de comunicación y matriz de adyacencias equivalente para el grafo fuertemente conectado.

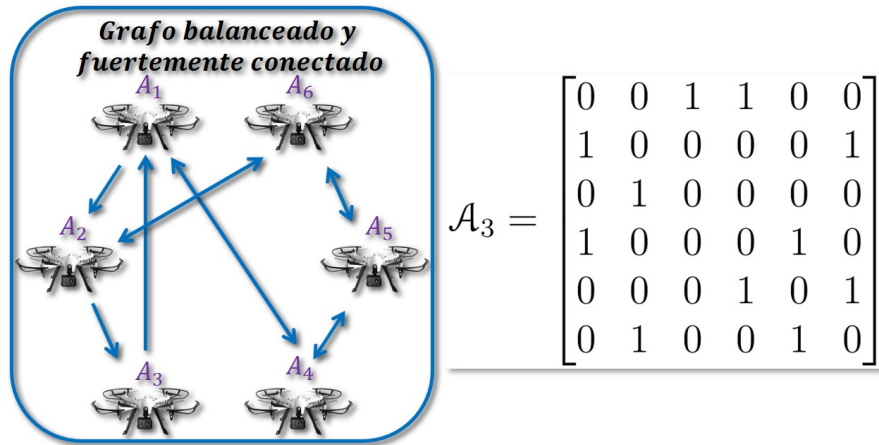


Fig. 3.7: Topología de comunicación y matriz de adyacencias equivalente para el grafo balanceado y fuertemente conectado.

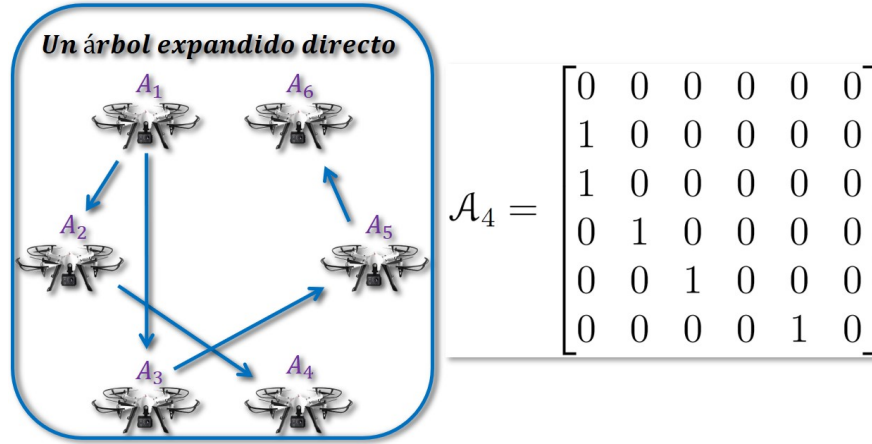


Fig. 3.8: Topología de comunicación y matriz de adyacencias equivalente para el grafo de árbol expandido directo.

y las matrices Laplacianas que describen las dinámicas de los sistemas multi-agentes de las figuras (3.1, 3.2, 3.3 y 3.4) son:

$$\mathcal{L}_1 = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 2 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & -1 & -1 & 0 & 3 \end{bmatrix}, \quad \mathcal{L}_2 = \begin{bmatrix} 3 & 0 & 0 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\mathcal{L}_3 = \begin{bmatrix} 2 & 0 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -1 & 2 \end{bmatrix}, \quad \mathcal{L}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

3.1.3. Distancias entre los agentes en la formación final deseada

La matriz H contiene las distancias relativas deseadas entre todos los agentes, por ejemplo, para realizar un hexágono se puede seleccionar la siguiente matriz H

$$H = \begin{bmatrix} 0 & 4 & 6 & 4 & 0 & -2 \\ 0 & 0 & 2\sqrt{3} & 4\sqrt{3} & 4\sqrt{3} & 2\sqrt{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.1)$$

Las primeras 3 filas en h_i son las distancias en las 3 coordenadas cartesianas, respectivamente $h_i = [x_i, y_i, z_i, 0, 0, 0]^T$, las últimas 3 filas se rellenan con ceros para ajustar dimensiones en las operaciones en el algoritmo 2.8. Se tendrán tantas columnas como agentes en el sistema, ya que cada columna de la matriz H contiene las posiciones relativas de cada agente, $h_1 = [0, 0, 0, 0, 0, 0]^T$ para el agente $i = 1$, $h_2 = [4, 0, 0, 0, 0, 0]^T$ para el agente $i = 2$, $h_3 = [6, 2\sqrt{3}, 0, 0, 0, 0]^T$ para el agente $i = 3$, $h_4 = [4, 4\sqrt{3}, 0, 0, 0, 0]^T$ para el agente $i = 4$, $h_5 = [0, 4\sqrt{3}, 0, 0, 0, 0]^T$ para el agente $i = 5$ y $h_6 = [-2, 2\sqrt{3}, 0, 0, 0, 0]^T$ para el agente $i = 6$.

En la imagen (3.9) se ilustra lo anterior

3.2 Sistema doble integrador para el diseño del algoritmo de consenso.

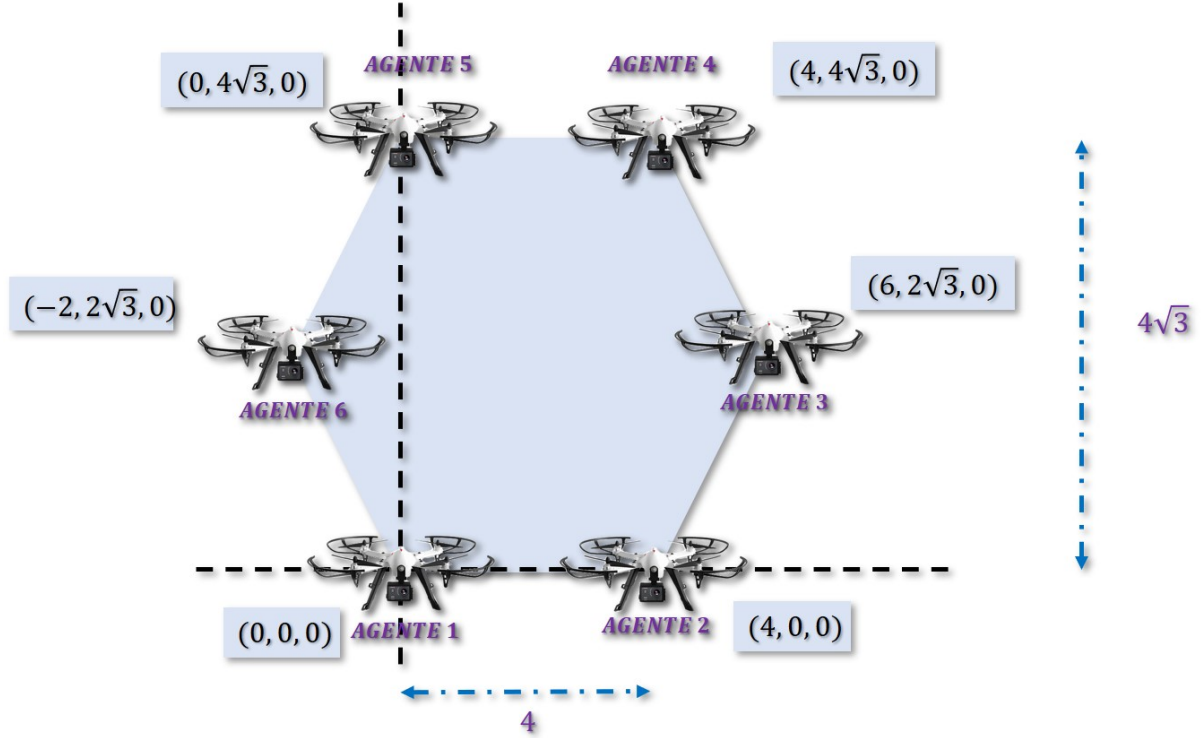


Fig. 3.9: Formación deseada y las distancias deseadas de los agentes.

3.2. Sistema doble integrador para el diseño del algoritmo de consenso.

Se considera una red de N sistemas doble integrador descritos por:

$$\begin{aligned}
 \dot{x}_i &= v_i, \\
 \dot{v}_i &= u_i, \\
 y_i &= Cx_i \\
 i &= 1, 2, \dots, N.
 \end{aligned}
 \tag{3.2}$$

Se reescribe el sistema doble integrador expandiéndolo a su forma de espacio de estados, ya que el algoritmo de consenso tipo observador modificado

para control de formación, utiliza las matrices A, B y C de esta representación lineal, para el cálculo de la ley de control del sistema multi-agente.

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ \ddot{x}_i \\ \ddot{y}_i \\ \ddot{z}_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1i} \\ u_{2i} \\ u_{3i} \end{bmatrix} \quad (3.3)$$

$$\begin{bmatrix} Y_{1i} \\ Y_{2i} \\ Y_{3i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} \quad (3.4)$$

esta representación es la misma planteada en (2.1) y que será utilizada para el cálculo de las ganancias de retroalimentación necesarias para el algoritmo de control de formación tipo observador (2.8).

3.3. Cálculo de las matrices de retroalimentación para el algoritmo de control de formación tipo-observador.

Utilizando las matrices A, B y C de la representación de espacio de estados del sistema doble integrador (3.3 y 3.4) mostrado anteriormente y definiendo los polos como $p = [-1 - 1 - 1 - 1.5 - 1.5 - 1.5]$, se hizo uso del comando “place” en el programa de simulación Matlab, para calcular la matriz F, como lo indica la primera restricción descrita en la sección (2.3.2).

$$F = \begin{bmatrix} 1.5 & 0 & 0 & 2.5 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 2.5 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 2.5 \end{bmatrix} \quad (3.5)$$

3.3 Cálculo de las matrices de retroalimentación para el algoritmo de control de formación tipo-observador.

Para la segunda restricción, se procedió a calcular la matriz Q , para ello se resuelve la desigualdad mostrada en el punto número dos de la sección (2.3.2), con el toolbox de Matlab llamado Yalmip definiendo $\alpha = 1$, primeramente se muestra el código utilizado para obtener la matriz (3.6).

```

Q = sdpvar(6,6,'symmetric')
des = A^T Q + Q A - 2C^T C + 2alpha Q
solvesdp([Q > 0, des < 0])
Q = double(Q)

```

$$Q = \begin{bmatrix} 0.7455 & 0 & 0 & -0.3548 & 0 & 0 \\ 0 & 0.7455 & 0 & 0 & -0.3548 & 0 \\ 0 & 0 & 0.7455 & 0 & 0 & -0.3548 \\ -0.3548 & 0 & 0 & 0.2651 & 0 & 0 \\ 0 & -0.3548 & 0 & 0 & 0.2651 & 0 \\ 0 & 0 & -0.3548 & 0 & 0 & 0.2651 \end{bmatrix}, \quad (3.6)$$

Una vez obtenida la matriz (3.6) se procede a calcular la matriz de retroalimentación L de la forma: $L = -Q^{-1}C^T$ obteniendo.

$$L = -Q^{-1}C^T = \begin{bmatrix} -3.6968 & 0 & 0 \\ 0 & -3.6968 & 0 \\ 0 & 0 & -3.6968 \\ -4.9486 & 0 & 0 \\ 0 & -4.9486 & 0 \\ 0 & 0 & -4.9486 \end{bmatrix} \quad (3.7)$$

Para el tercer y último punto de las restricciones mostradas en la sección (2.3.2) primero es necesario determinar la matriz Laplaciana:

$$\mathcal{L} = \begin{bmatrix} 3 & 0 & 0 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad (3.8)$$

Posteriormente se calculan los eigenvalores de \mathcal{L}

$$eig(\mathcal{L}) = \begin{bmatrix} 2 + 0i \\ 1 + 0i \\ 3.324 + 0i \\ 0 + 0i \\ 1.337 + 0.5622i \\ 1.337 - 0.5622i \end{bmatrix} \quad (3.9)$$

$$c_{th} = \frac{1}{\min Re(\lambda_i(\mathcal{L}))} \quad i = 2, \dots, N. \quad (3.10)$$

El término $\lambda_i(\mathcal{L})$ representa a los eigenvalores de la matriz laplaciana, donde no se toma en cuenta su primer eigenvalor, debido a que si se cumple con el Teorema 1 mostrado en la sección (2.3.1), entonces el grafo \mathcal{G} incluirá en su estructura un árbol expandido directo, y esto hace que siempre exista un cero en los eigenvalores de \mathcal{L} .

El mínimo de la parte real de los eigenvalores es 1, por lo tanto, el valor de $c_i h$ es:

$$c_{th} = \frac{1}{1} = 1. \quad (3.11)$$

Las matrices (3.5 y 3.7) se utilizarán en el capítulo siguiente para realizar ejemplos de simulación con el algoritmo de consenso tipo observador (2.8) que entregará las leyes de control para un equipo de cuadricópteros bajo diferentes circunstancias.

Resultados del esquema para control de formación aplicado a un sistema de multiples cuadricópteros.

4.1. Modelo dinámico (Cuadricópteros)

Como caso de estudio se consideró a un sistema multi-agente de tipo homogéneo, esto quiere decir que, se asume que todos los cuadricópteros son iguales, con los mismos parámetros, dimensiones y dinámica. En el caso contrario que en un mismo sistema se tengan dinámicas diferentes para multiples agentes, estos sistemas reciben el nombre de sistemas multi-agentes de tipo heterogéneo y el análisis para resolver el problema de control de formación es diferente para ambos tipos de sistemas.

Se muestra a continuación en las ecuaciones (4.1 - 4.6) el modelo dinámico de un dron tipo cuadricóptero, el cual como su nombre lo indica, posee 4 rotores que generan la fuerza necesaria para elevar y maniobrar el dron, este modelo matemático fue tomado de [17].

$$\ddot{x}_i = (\cos(\psi_i) \sin(\theta_i) \cos(\phi_i) + \sin(\psi_i) \sin(\phi_i)) \frac{1}{m_i} T_i, \quad (4.1)$$

$$\ddot{y}_i = (\sin(\psi_i) \sin(\theta_i) \cos(\phi_i) - \cos(\psi_i) \sin(\phi_i)) \frac{1}{m_i} T_i, \quad (4.2)$$

$$\ddot{z}_i = -g + (\cos(\theta_i)\cos(\phi_i))\frac{1}{m_i}T_i, \quad (4.3)$$

$$\ddot{\phi}_i = \dot{\theta}_i\dot{\psi}_i\left(\frac{J_y - J_z}{J_x}\right) + \frac{1}{J_x}R_i, \quad (4.4)$$

$$\ddot{\theta}_i = \dot{\phi}_i\dot{\psi}_i\left(\frac{J_z - J_x}{J_y}\right) + \frac{1}{J_y}P_i, \quad (4.5)$$

$$\ddot{\psi}_i = \dot{\theta}_i\dot{\phi}_i\left(\frac{J_x - J_y}{J_z}\right) + \frac{1}{J_z}Y_i. \quad (4.6)$$

Por practicidad de simulación, se reescribe el modelo dinámico de 6 ecuaciones de segundo orden, como un sistema de 12 ecuaciones de primer orden quedando de la forma:

$$\dot{x}_i = V_{xi}, \quad (4.7)$$

$$\dot{V}_{xi} = (\cos(\psi_i)\sin(\theta_i)\cos(\phi_i) + \sin(\psi_i)\sin(\phi_i))\frac{1}{m_i}T_i, \quad (4.8)$$

$$\dot{y}_i = V_{yi}, \quad (4.9)$$

$$\dot{V}_{yi} = (\sin(\psi_i)\sin(\theta_i)\cos(\phi_i) - \cos(\psi_i)\sin(\phi_i))\frac{1}{m_i}T_i, \quad (4.10)$$

$$\dot{z}_i = V_{zi}, \quad (4.11)$$

$$\dot{V}_{zi} = -g + (\cos(\theta_i)\cos(\phi_i))\frac{1}{m_i}T_i, \quad (4.12)$$

$$\dot{\phi}_i = w_{\phi i}, \quad (4.13)$$

$$w_{\phi i} = \dot{\theta}_i\dot{\psi}_i\left(\frac{J_y - J_z}{J_x}\right) + \frac{1}{J_x}R_i, \quad (4.14)$$

$$\dot{\theta}_i = w_{\theta i}, \quad (4.15)$$

4.1 Modelo dinámico (Cuadricópteros)

$$\dot{w}_{\theta_i} = \dot{\phi}_i \dot{\psi}_i \left(\frac{J_z - J_x}{J_y} \right) + \frac{1}{J_y} P_i, \quad (4.16)$$

$$\dot{\psi}_i = w_{\psi_i}, \quad (4.17)$$

$$\dot{w}_{\psi_i} = \dot{\theta}_i \dot{\phi}_i \left(\frac{J_x - J_y}{J_z} \right) + \frac{1}{J_z} Y_i. \quad (4.18)$$

Donde \dot{x}_i , \dot{y}_i y \dot{z}_i , son las velocidades lineales con respecto a los ejes cartesianos, \ddot{V}_{xi} , \ddot{V}_{yi} y \ddot{V}_{zi} , son las aceleraciones lineales, $\dot{\phi}_i$, $\dot{\theta}_i$ y $\dot{\psi}_i$, son las velocidades angulares de los ángulos de Euler *roll*, *pitch* y *yaw* respectivamente, \dot{w}_{ϕ_i} , \dot{w}_{θ_i} y \dot{w}_{ψ_i} son las aceleraciones angulares, T_i es el *thrust* o también llamado empuje, el cual es generado por los rotores y permite al dron elevarse, descender o mantener la posición de *hover*, R_i , P_i y Y_i son los torques rotacionales, m_i es la masa del dron, g es la aceleración de la gravedad y J_x , J_y y J_z son los momentos de inercia con respecto a los diferentes ejes.

Los parámetros utilizados en las simulaciones de este modelo dinámico se muestran en la tabla (4.1) que fueron tomados de [18], parámetros que pertenecen al cuadricóptero crazyflie 2.0.

Parametro	Descripción	Valor
m	Masa Total	0.33 [Kg]
g	Aceleración de la gravedad	9.81 [$\frac{m}{s^2}$]
d	Largo del brazo	$39.73 \times 10^{-3} [m]$
J_x	Momento de inercia en el eje x	$1.395 \times 10^{-5} [Kg \times m^2]$
J_y	Momento de inercia en el eje y	$1.436 \times 10^{-5} [Kg \times m^2]$
J_z	Momento de inercia en el eje z	$2.173 \times 10^{-5} [Kg \times m^2]$

Tabla 4.1: Parámetros del cuadricóptero crazyflie 2.0.

4.2. Transformaciones

Es necesario adaptar los protocolos de consenso aplicados a resolver control de formación (2.6), (2.7) y (2.8), (u_i) , para ser las entradas del modelo dinámico de los UAVs. En [15] se encuentran las relaciones que existen entre la ley de control u_i y los ángulos deseados ϕ_{di} , θ_{di} y ψ_{di} . Se calculan los ángulos deseados de todos los drones en el sistema como se muestra a continuación:

$$\phi_{di} = \arctan \left(\frac{-u_{2i}}{\sqrt{u_{1i}^2 + (u_{3i} + g)^2}} \right), \quad (4.19)$$

$$\theta_{di} = \arctan \left(\frac{u_{1i}}{u_{3i} + g} \right), \quad (4.20)$$

$$\psi_{di} = 0. \quad (4.21)$$

y para calcular el empuje o *thrust* de los rotores de los UAVs, la entrada de la parte traslacional del sistema, se utiliza la siguiente expresion mostrada en (4.22).

$$T_i = m_i \sqrt{u_{1i}^2 + u_{2i}^2 + (u_{3i} + g)^2} \quad (4.22)$$

4.2.1. *Thrust* o empuje de los rotores del cuadricóptero modelo *crazyflie 2.0*

Existen 3 movimientos verticales que un cuadricóptero es capaz de hacer, el primero se denomina como *hover*, la cual es una posición estática donde se permanece a la misma altura, para conseguir el *hover* es necesario que la fuerza de empuje o *thrust* de los 4 motores consiga igualar la fuerza aplicada por la gravedad sobre el dron, retomando la masa del cuadricóptero *crazyflie 2.0* mostrada en la tabla (4.1) y la aceleración de la gravedad, es posible calcular el *Thrust* que debe generar el dron para conseguir la posición de *hover* como:

4.3 Control PI para la parte rotacional del modelo dinámico del los UAVs

$$F = ma = 0.33Kg \left(9.81 \frac{m}{s^2} \right) = 3.2373Newtons \quad (4.23)$$

Los otros dos movimientos son la elevación y el descenso del cuadricóptero, por un lado para elevar el dron es necesario que el *thrust* sea mayor al mostrado en (4.23), es decir, se debe generar una fuerza mayor a la influenciada por la gravedad sobre el cuadricóptero, por el contrario para el descenso se necesita una fuerza menor.

4.2.1.1. *Thrust* máximo para el crazyflie 2.0

En [19] mencionan que el *Thrust* máximo que el cuadricóptero modelo *crazyflie* 2.0 es capaz de generar es de 61.16g pudiendo comprometer el vuelo, es debido a esto que especifican el valor de 42g como el valor máximo de *thrust* que recomiendan para no comprometer el vuelo del dron, ya que a 61.16g sólo se podría mantener la posición de *hover* y no es posible elevar el cuadricóptero. Para referenciar el *Thrust* máximo en las mismas unidades de fuerza que para la posición de *hover* mostrada en (4.23) se considera una masa adicional de 42g a la masa del dron, quedando el cálculo del *Thrust* máximo como:

$$F_{max} = m_{total}a = (0.33Kg + 0.042Kg) \left(9.81 \frac{m}{s^2} \right) = 3.64932Newtons \quad (4.24)$$

4.3. Control PI para la parte rotacional del modelo dinámico del los UAVs

El primer paso para la simulación del modelo dinámico de un UAV, es diseñar un esquema de control cuya finalidad sea cambiar los ángulos del dron de forma apropiada para mover su posición en el espacio, utilizando los protocolos de control de formación (2.6), (2.7) y (2.8), para alcanzar y mantener las distancias necesarias predefinidas en la matriz \mathcal{H} para cada dron y poder realizar las formaciones deseadas. Se asume que todos los UAVs en el equipo tienen exactamente la misma dinámica al igual que sus parámetros,

por lo tanto, solo es necesario calcular las ganancias Proporcional e Integral de un controlador PI y utilizar estas mismas ganancias para los N agentes en el sistema.

La representación lineal en espacio de estados de la parte rotacional del modelo dinámico de los UAVs se muestra a continuación:

$$\begin{bmatrix} \dot{\phi} \\ w_{\phi i} \\ \dot{\theta}_i \\ w_{\theta i} \\ \dot{\psi}_i \\ w_{\psi i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_i \\ w_{\phi i} \\ \theta_i \\ w_{\theta i} \\ \psi_i \\ w_{\psi i} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix} \begin{bmatrix} R_i \\ P_i \\ Y_i \end{bmatrix} \quad (4.25)$$

$$y_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \phi_i \\ w_{\phi i} \\ \theta_i \\ w_{\theta i} \\ \psi_i \\ w_{\psi i} \end{bmatrix} \quad (4.26)$$

para el cálculo de las ganancias proporcional e integral, se aumenta el sistema de la forma (4.27)

$$\begin{bmatrix} \dot{x}_i \\ \dot{x}_{ei} \end{bmatrix} = \begin{bmatrix} A_r & 0 \\ -C_r & 0 \end{bmatrix} \begin{bmatrix} x_i \\ x_{ei} \end{bmatrix} + \begin{bmatrix} B_r \\ 0 \end{bmatrix} u_i + \begin{bmatrix} 0 \\ R_i \end{bmatrix} \quad (4.27)$$

donde R es el vector de referencias para la parte rotacional del modelo dinámico del UAV.

Posteriormente se hizo uso del comando “place” en el programa Matlab, para calcular las ganancias Proporcional e Integral utilizando el modelo aumentado, las ganancias matriciales obtenidas son especificadas en (4.28) y (4.29), ganancias proporcional e integral respectivamente.

$$K_{pr} = \begin{bmatrix} 0.06511 & 0.00372 & -1.959 \times 10^{-06} & -1.781 \times 10^{-08} & 3.696 \times 10^{-06} & 3.340 \times 10^{-08} \\ -3.63263 \times 10^{-06} & -3.30178 \times 10^{-08} & 0.0625970 & 0.00357620 & -2.907 \times 10^{-06} & -2.643 \times 10^{-08} \\ 7.24445 \times 10^{-06} & 6.54818 \times 10^{-08} & -3.18690 \times 10^{-06} & -2.897 \times 10^{-08} & 0.1181 & 0.007320 \end{bmatrix} \quad (4.28)$$

4.3 Control PI para la parte rotacional del modelo dinámico del los UAVs

$$K_{I_r} = \begin{bmatrix} 0.31015 & -1.7821 \times 10^{-05} & 3.3627 \times 10^{-05} \\ -3.3033 \times 10^{-05} & 0.29815 & -2.6443 \times 10^{-05} \\ 6.5910 \times 10^{-05} & -2.8979 \times 10^{-05} & 0.610310 \end{bmatrix} \quad (4.29)$$

4.4. Estructuración del esquema de control de formación para un equipo de cuadricópteros.

Retomando el esquema a bloques de la propuesta de solución (1.3), se muestra en (4.1) el esquema expandido para resolver el problema de control de formación en un equipo de cuadricópteros modelo *crazyflie* 2.0.

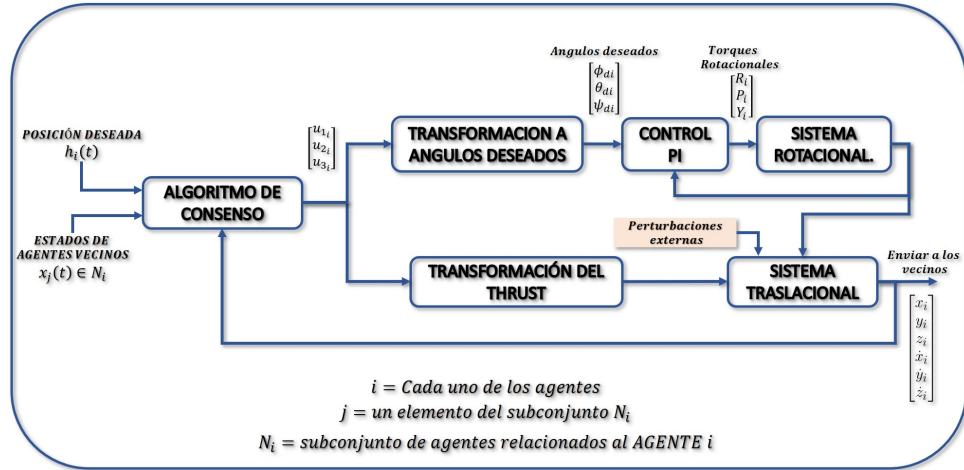


Fig. 4.1: Esquema expandido del diagrama a bloques de la propuesta de solución.

Donde el bloque de “Algoritmo de consenso” representa a (2.6), (2.7) y (2.8), el bloque de “Transformación a ángulos deseados” son las ecuaciones (4.19), (4.20) y (4.21), y la “Transformación del Thrust” se muestra en (4.22), el bloque mencionado en (1.3) llamado “Controlador interno” se sustituye por un bloque que representa un controlador PI, cuyas ganancias son (4.28) y (4.29) proporcional e integral respectivamente, el bloque que representa un dron en (1.3) se separa en dos bloques que representan el modelo dinámico de un cuadricóptero, primero el bloque de “Sistema Traslacional” denota a la parte del modelo dinámico de las ecuaciones (4.7) a la (4.12) y el bloque de “Sistema Rotacional” representa las ecuaciones de (4.13) a la (4.18).

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

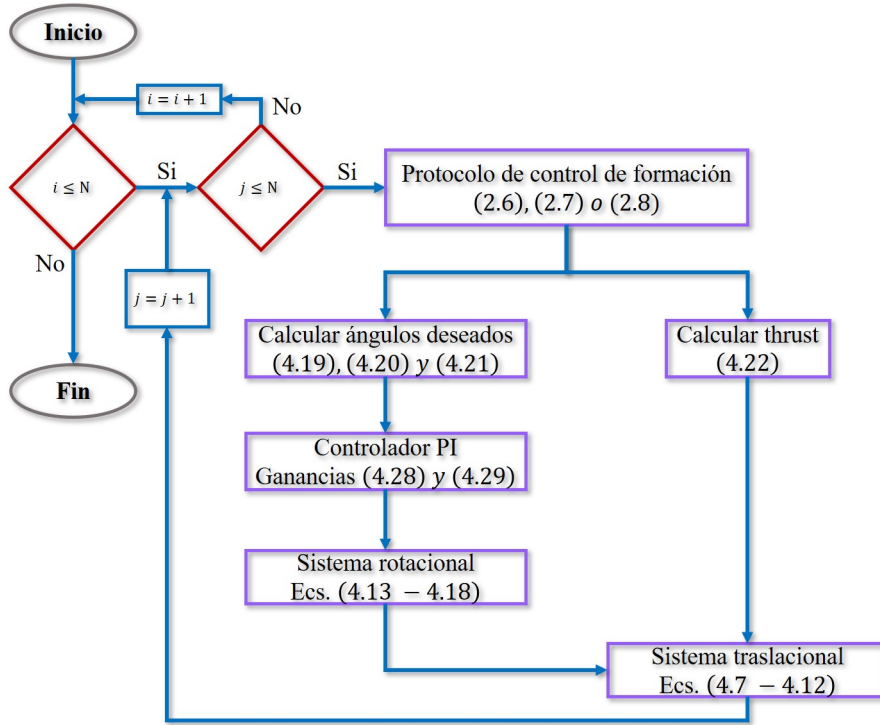


Fig. 4.2: Diagrama de flujo para la simulación del esquema de control de formación.

Para ejemplificar el orden para resolver las ecuaciones en las simulaciones que se mostraran en las paginas siguientes, se muestra un diagrama de flujo que describe la programación realizada, donde N varia sus valores con respecto a las diferentes formaciones.

En las paginas siguientes se mostraran los resultados obtenidos en las simulaciones, bajo los esquemas de los 3 protocolos para control de formación (2.6), (2.7) y (2.8), se presentan 2 formaciones finales, un hexágono y un tetraedro, primeramente se muestran los resultados de los algoritmos cuando no se presentan perturbaciones.

4.4.1. Esquema de control de formación bajo el protocolo (2.6) para un equipo de cuadricópteros sin perturbaciones.

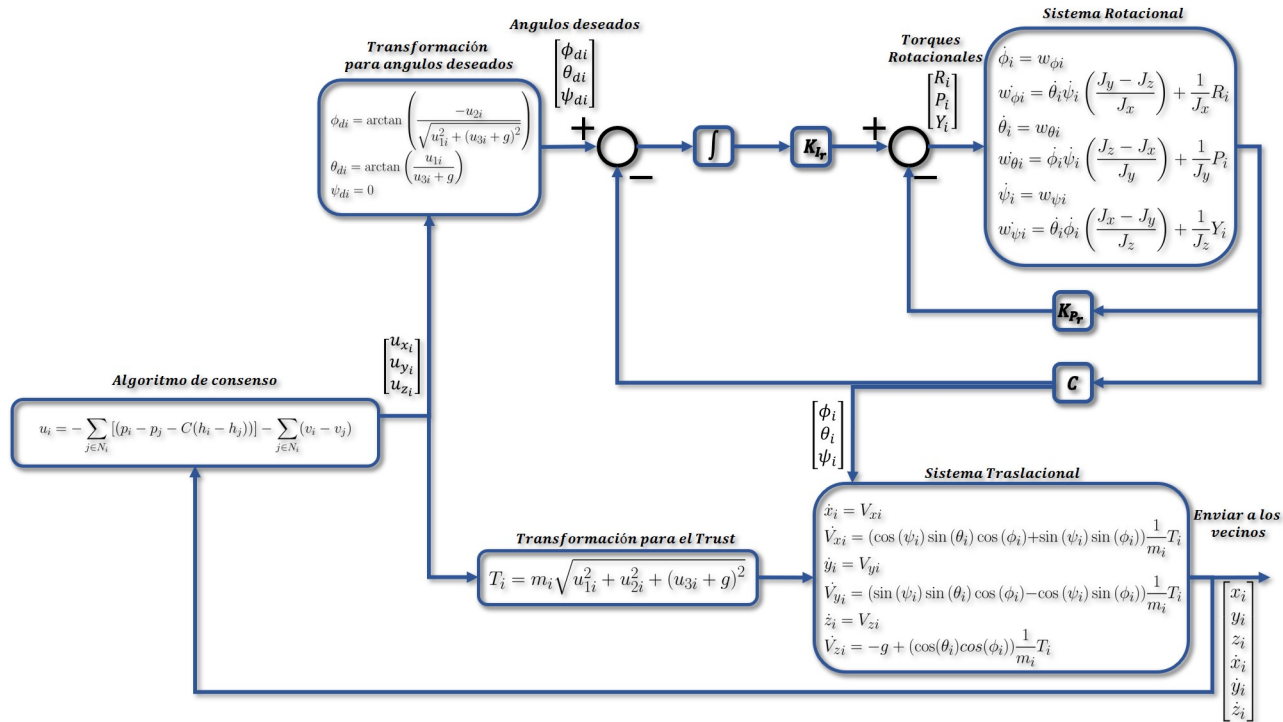


Fig. 4.3: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

4.4.2. Hexágono formado con el esquema (4.3) sin perturbaciones.

A continuación en la tabla (4.2) se muestran las condiciones iniciales para los estados de todos los cuadricópteros de un sistema multi-agente donde $N = 6$.

Condiciones iniciales del cuadricóptero						
	Agente 1	Agente 2	Agente 3	Agente 4	Agente 5	Agente 6
Posición en x	1	2	3	4	2	3
Posición en y	2	1	4	3	5	1
Posición en z	0	0	0	0	0	0
Velocidad \dot{x}	0	0	0	0	0	0
Velocidad \dot{y}	0	0	0	0	0	0
Velocidad \dot{z}	0	0	0	0	0	0
Posición angular ϕ	0	0	0	0	0	0
Posición angular θ	0	0	0	0	0	0
Posición angular ψ	0	0	0	0	0	0
Velocidad angular $\dot{\phi}$	0	0	0	0	0	0
Velocidad angular $\dot{\theta}$	0	0	0	0	0	0
Velocidad angular $\dot{\psi}$	0	0	0	0	0	0

Tabla 4.2: Condiciones iniciales para la formación final de un hexágono.

La matriz H que contiene la formación rígida para el sistema se muestra en (4.30), en este primer caso se busca realizar un hexágono.

$$H = \begin{bmatrix} 0 & 4 & 6 & 4 & 0 & -2 \\ 0 & 0 & 2\sqrt{3} & 4\sqrt{3} & 4\sqrt{3} & 2\sqrt{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.30)$$

La topología de comunicación está definida como se muestra en la figura (4.4) y se muestra así mismo su respectiva matriz de adyacencias.

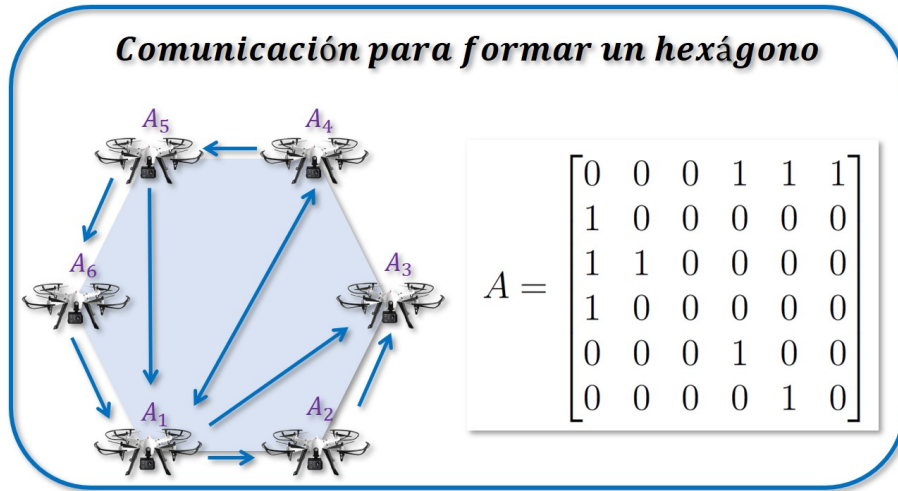


Fig. 4.4: Topología de comunicación y matriz de adyacencias equivalente para la formación de un hexágono.

Todas las simulaciones mostradas en las siguientes paginas, donde el objetivo sea formar un hexágono, se utilizaran, las condiciones iniciales (4.2), la matriz H (4.30) y la misma topología de comunicación (4.4).

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

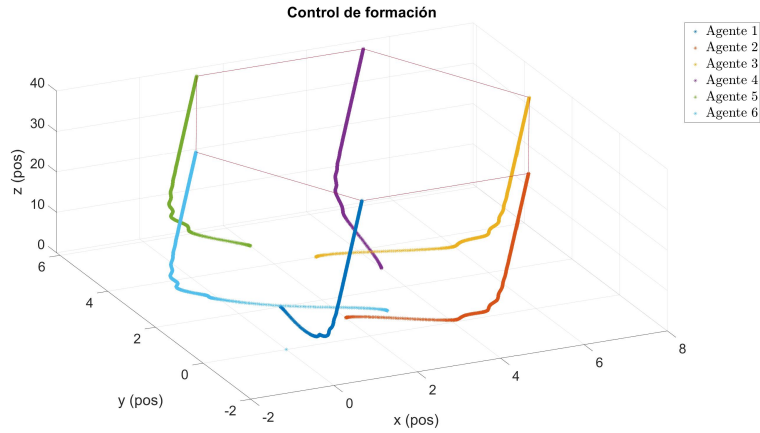


Fig. 4.5: Hexágono formado con el algoritmo tipo observador (2.6) sin perturbaciones.

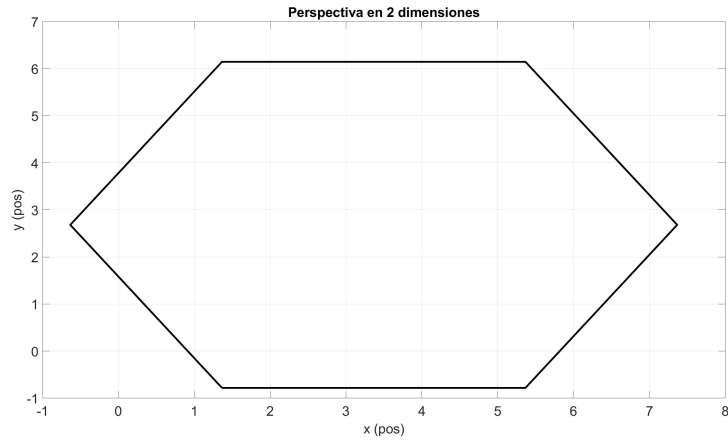


Fig. 4.6: Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.

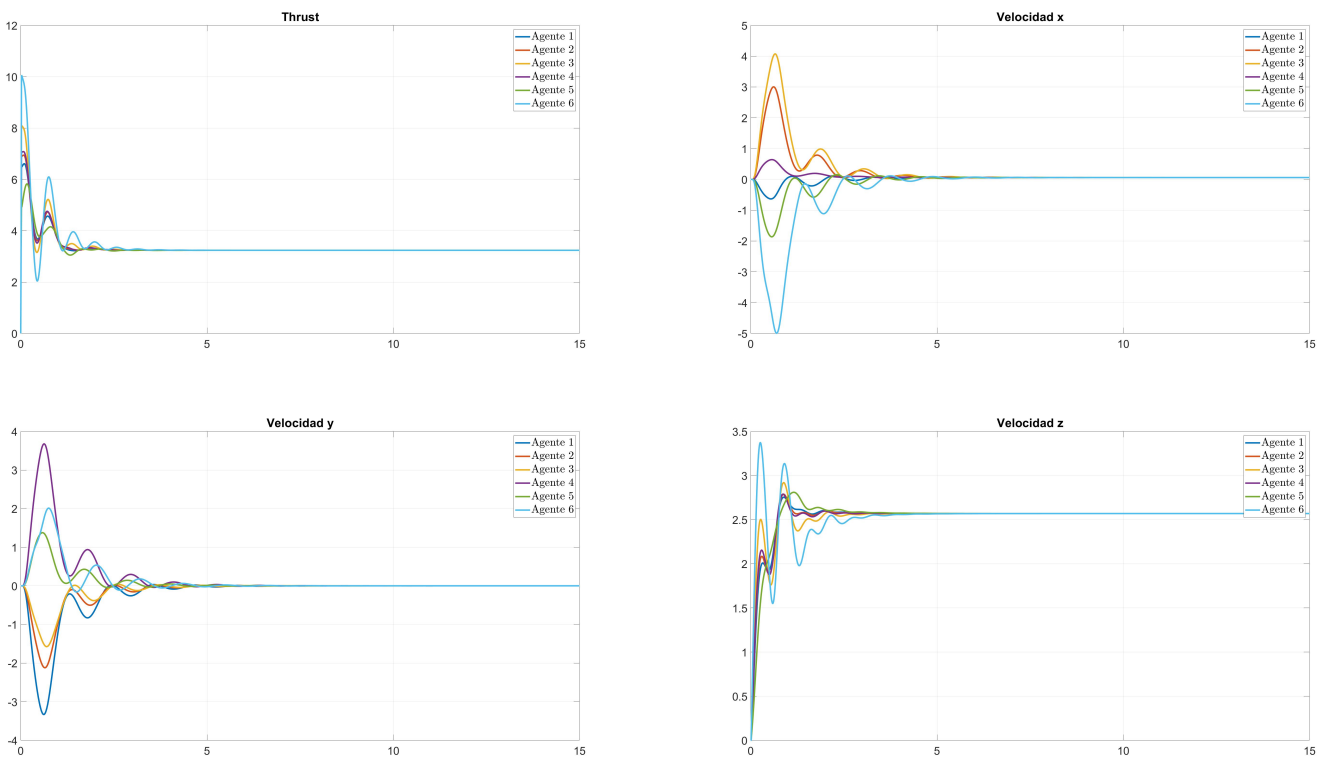


Fig. 4.7: Thrust y velocidades lineales con respecto a los 3 ejes.

4.4.3. Tetraedro formado con el esquema (4.3) sin perturbaciones.

A continuación en la tabla (4.3) se muestran las condiciones iniciales para los estados de todos los cuadricópteros de un sistema multi-agente donde $N = 4$.

Condiciones iniciales del cuadricóptero				
	Agente 1	Agente 2	Agente 3	Agente 4
Posición en x	1	2	3	4
Posición en y	2	1	4	3
Posición en z	0	0	0	0
Velocidad en x	0	0	0	0
Velocidad en y	0	0	0	0
Velocidad en z	0	0	0	0
Posición angular ϕ	0	0	0	0
Posición angular θ	0	0	0	0
Posición angular ψ	0	0	0	0
Velocidad angular $\dot{\phi}$	0	0	0	0
Velocidad angular $\dot{\theta}$	0	0	0	0
Velocidad angular $\dot{\psi}$	0	0	0	0

Tabla 4.3: Condiciones iniciales para la formación final de un tetraedro.

La matriz H que contiene la formación rígida para el sistema se muestra en (??), en este segundo caso se busca realizar un tetraedro.

$$H = \begin{bmatrix} 0 & 3 & \frac{3}{2} & \frac{3}{2} \\ 0 & 0 & \frac{3}{2}\sqrt{3} & \frac{2}{3}\sqrt{3} \\ 0 & 0 & 0 & \frac{3\sqrt{6}}{3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.31)$$

La topología de comunicación está definida como se muestra en la figura (4.8) y se muestra así mismo su respectiva matriz de adyacencias.

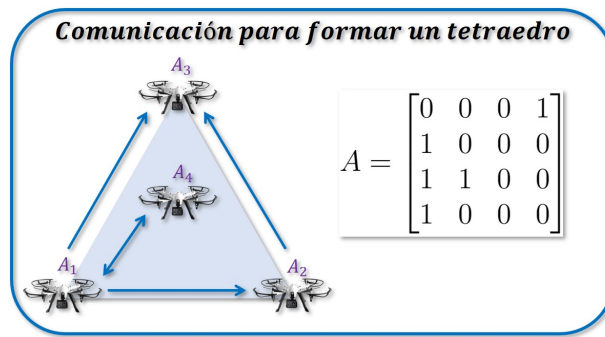


Fig. 4.8: Topología de comunicación y matriz de adyacencias equivalente para la formación de un tetraedro.

Todas las simulaciones mostradas en las siguientes paginas, donde el objetivo sea formar un tetraedro, se utilizaran, las condiciones iniciales (4.3), la matriz H (4.31) y la misma topología de comunicación (4.8).

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

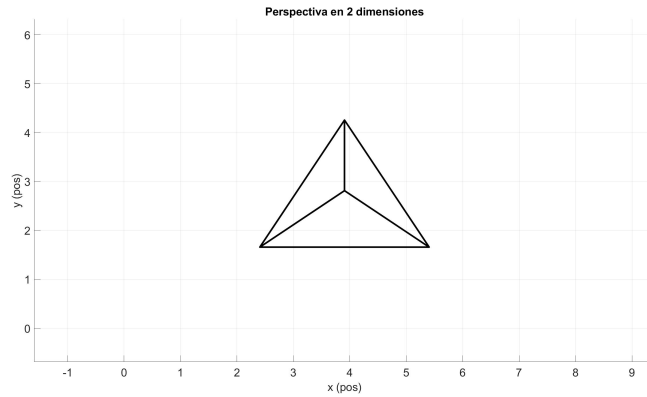


Fig. 4.10: Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.

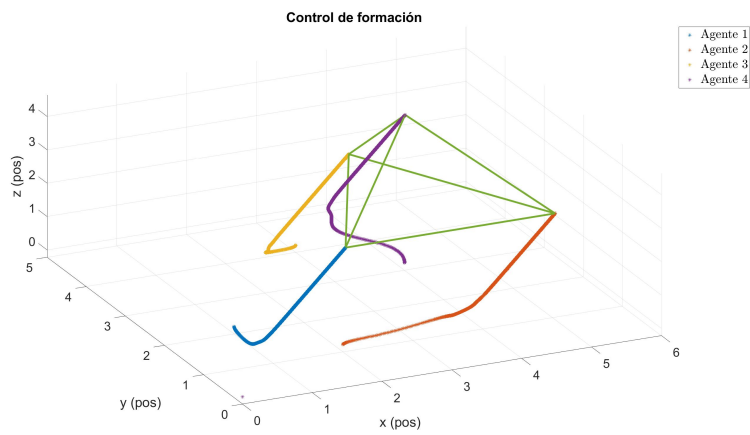


Fig. 4.9: Tetraedro formado con el algoritmo tipo observador (2.6) sin perturbaciones.

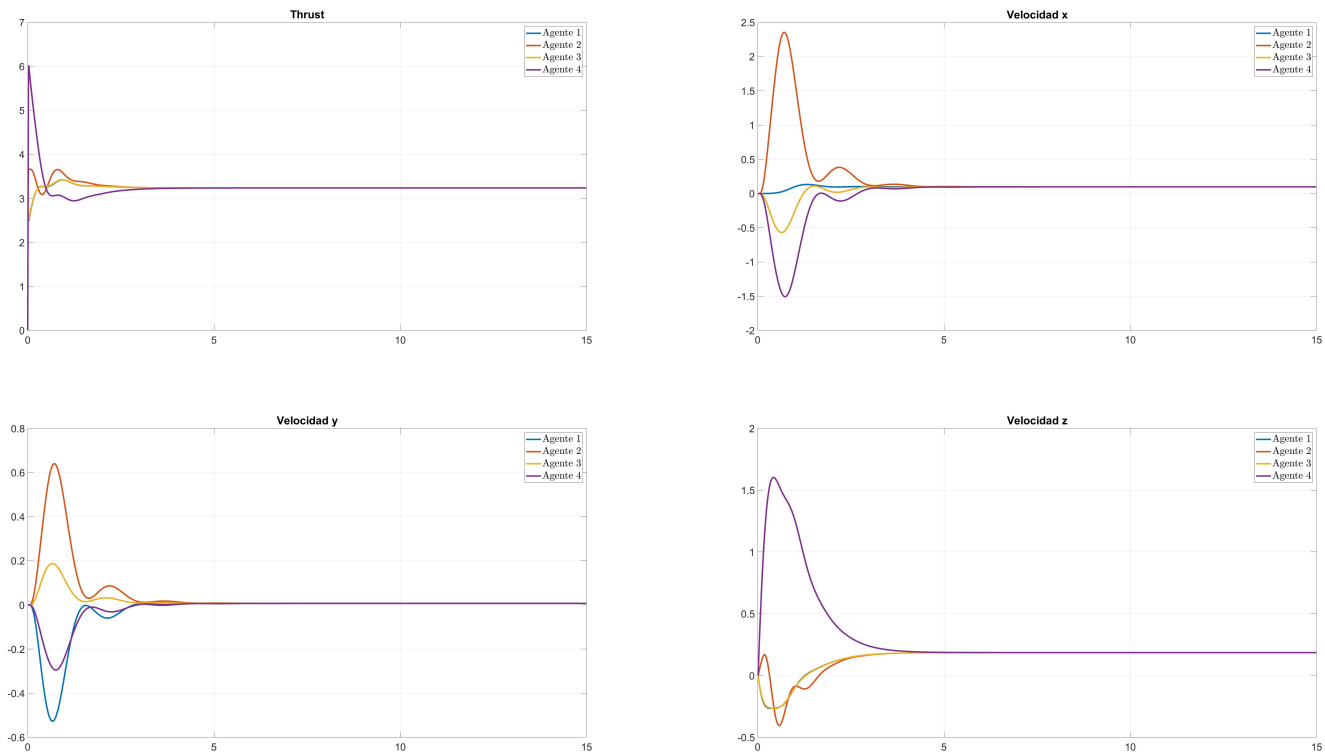


Fig. 4.11: Thrust y velocidades lineales con respecto a los 3 ejes.

4.4.4. Esquema de control de formación bajo el protocolo (2.7) para un equipo de cuadricópteros sin perturbaciones.

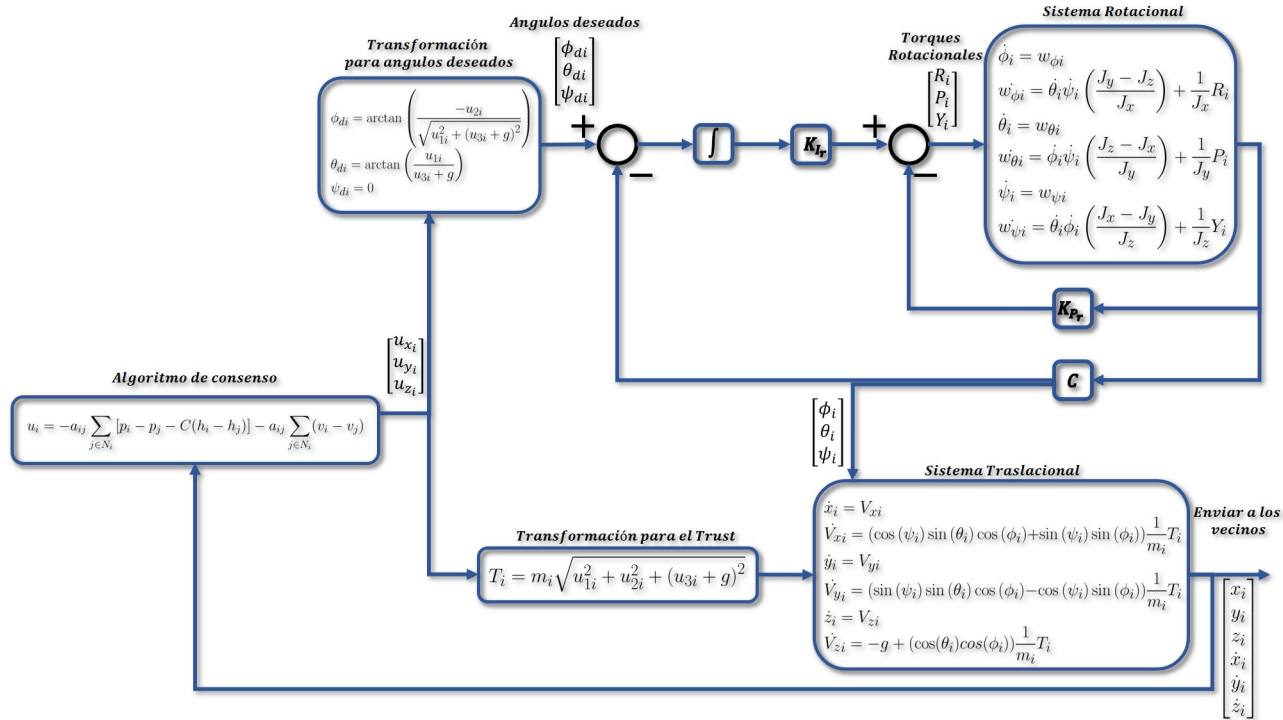


Fig. 4.12: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.4.5. Hexágono formado con el esquema (4.12) sin perturbaciones.

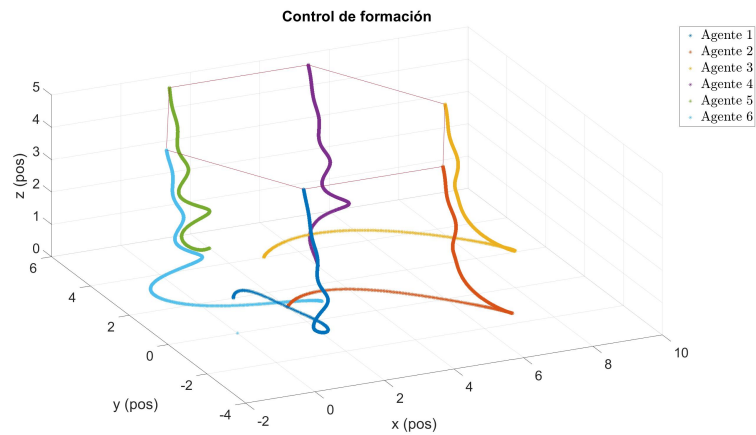


Fig. 4.13: Hexágono formado con el algoritmo tipo observador (2.7) sin perturbaciones.

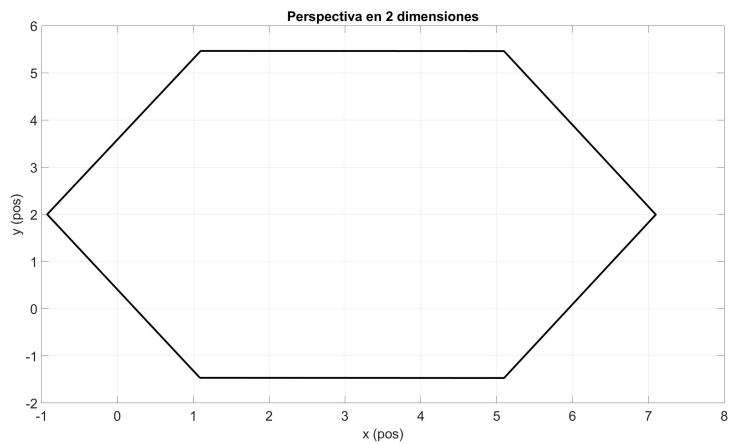


Fig. 4.14: Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.

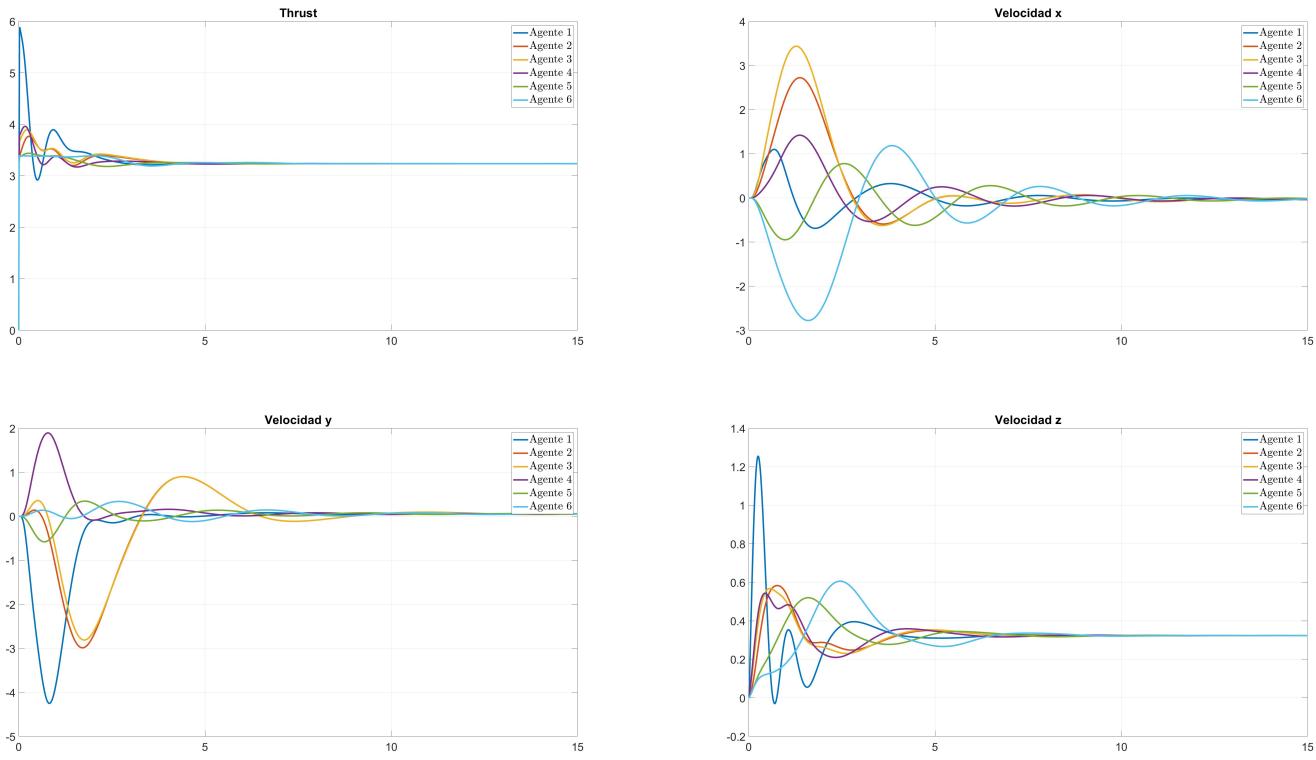


Fig. 4.15: Thrust y velocidades lineales con respecto a los 3 ejes.

4.4.6. Tetraedro formado con el esquema (4.12) sin perturbaciones.

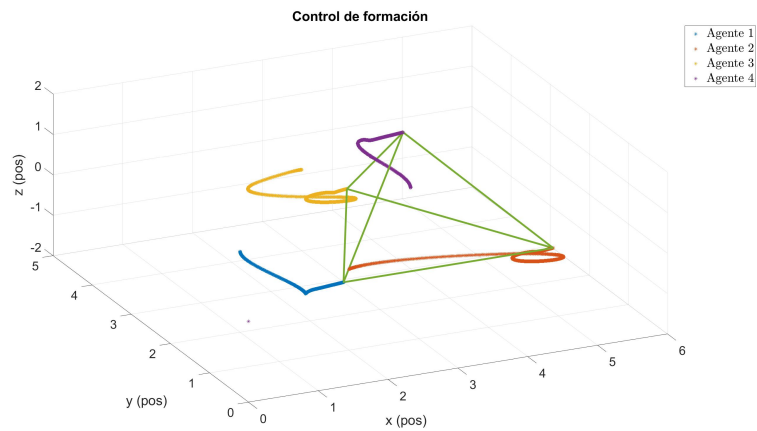


Fig. 4.16: Tetraedro formado con el algoritmo tipo observador (2.7) sin perturbaciones.

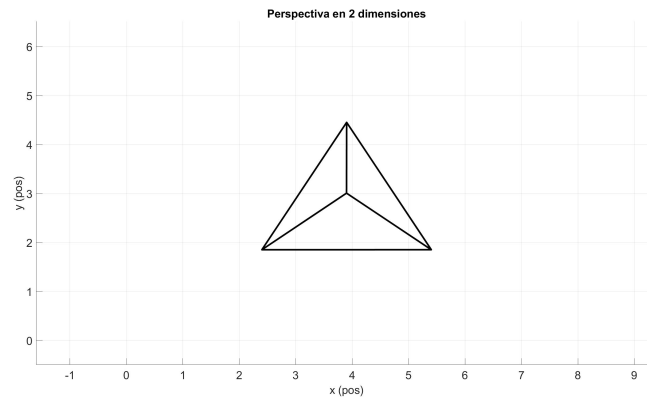


Fig. 4.17: Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.

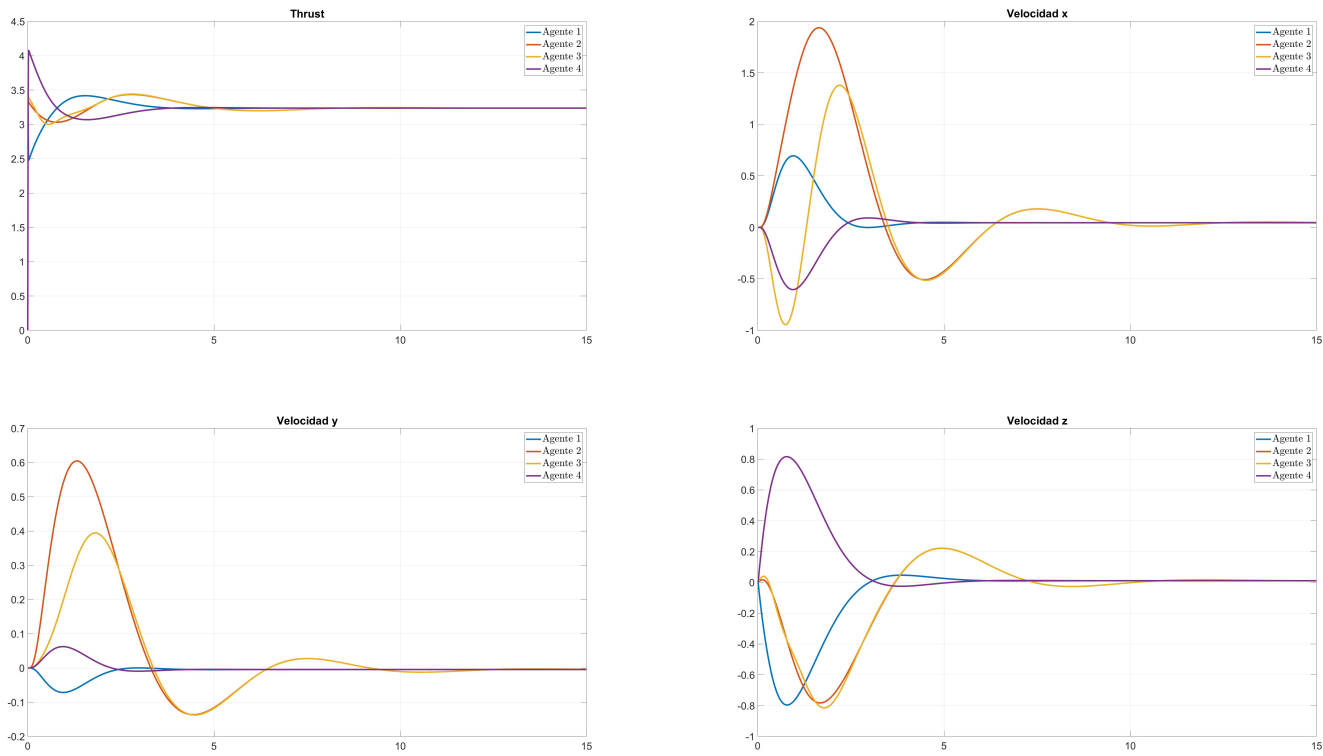


Fig. 4.18: Thrust y velocidades lineales con respecto a los 3 ejes.

subsectionEsquema de control de formación bajo el protocolo (2.8) para un equipo de cuadricópteros sin perturbaciones.

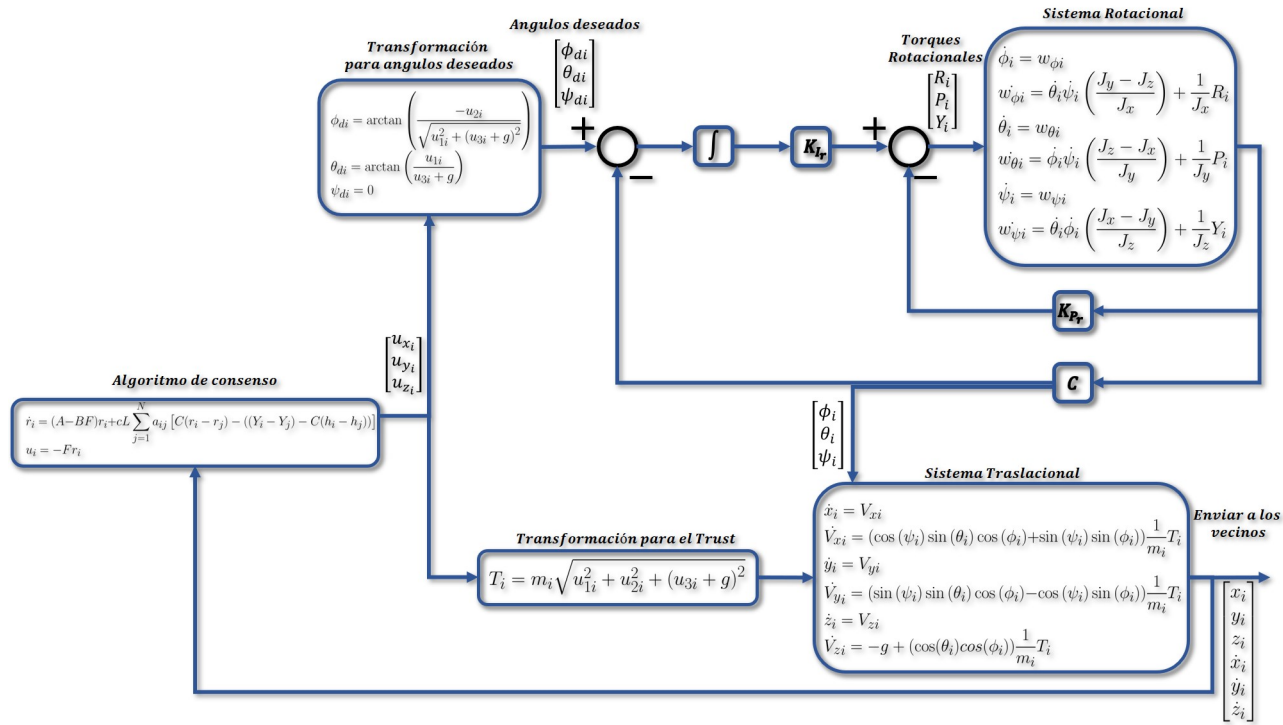


Fig. 4.19: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

4.4.7. Hexágono formado con el esquema (4.19) sin perturbaciones.

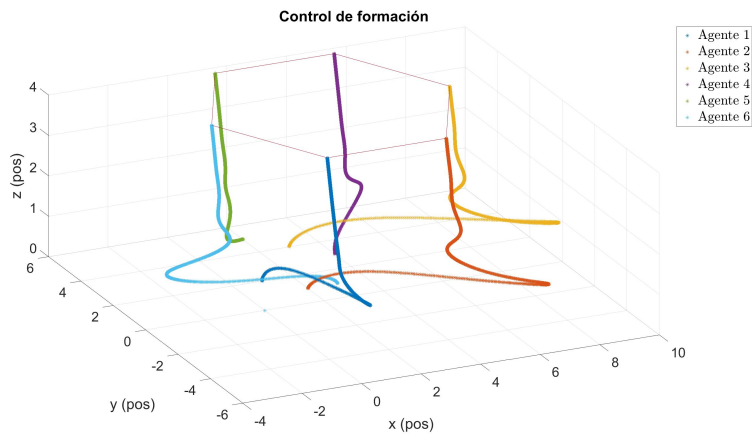


Fig. 4.20: Hexágono formado con el algoritmo tipo observador (2.8) sin perturbaciones.

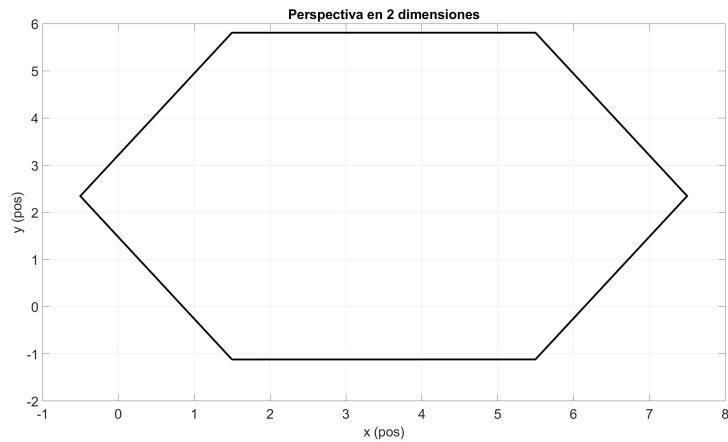


Fig. 4.21: Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.

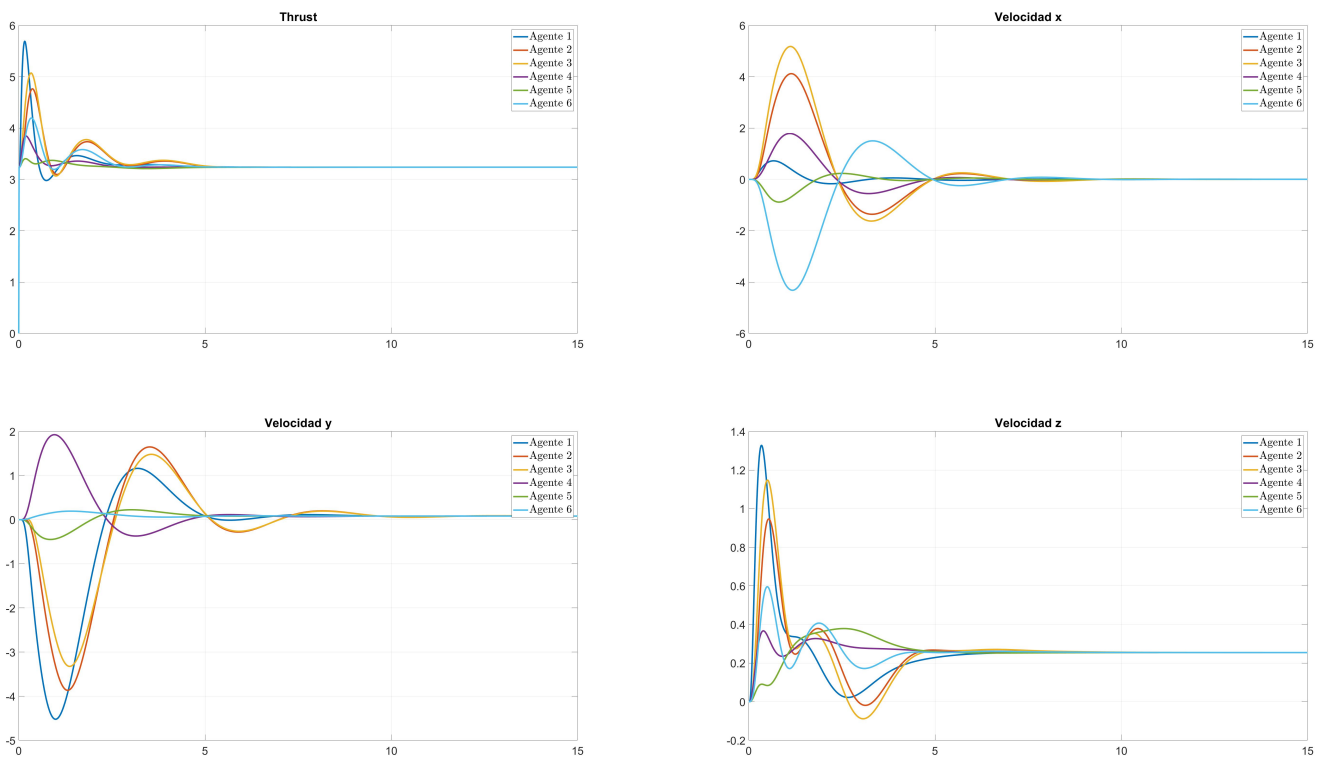


Fig. 4.22: Thrust y velocidades lineales con respecto a los 3 ejes.

4.4 Estructuración del esquema de control de formación para un equipo de cuadricópteros.

4.4.8. Tetraedro formado con el esquema (4.19) sin perturbaciones.

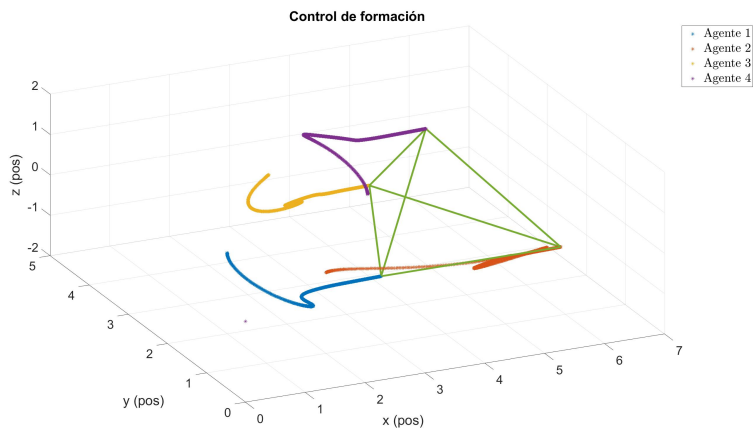


Fig. 4.23: Tetraedro formado con el algoritmo tipo observador (2.8) sin perturbaciones.

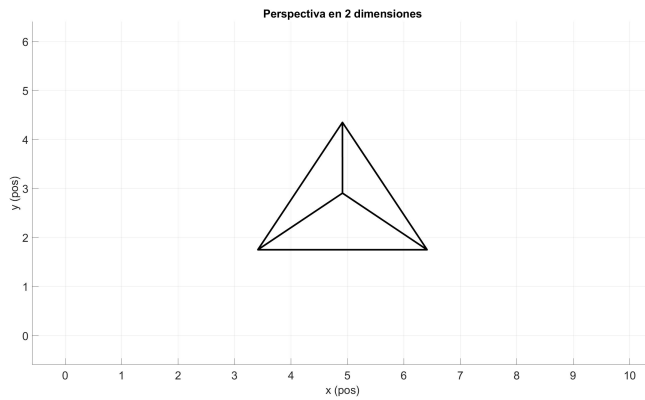


Fig. 4.24: Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.

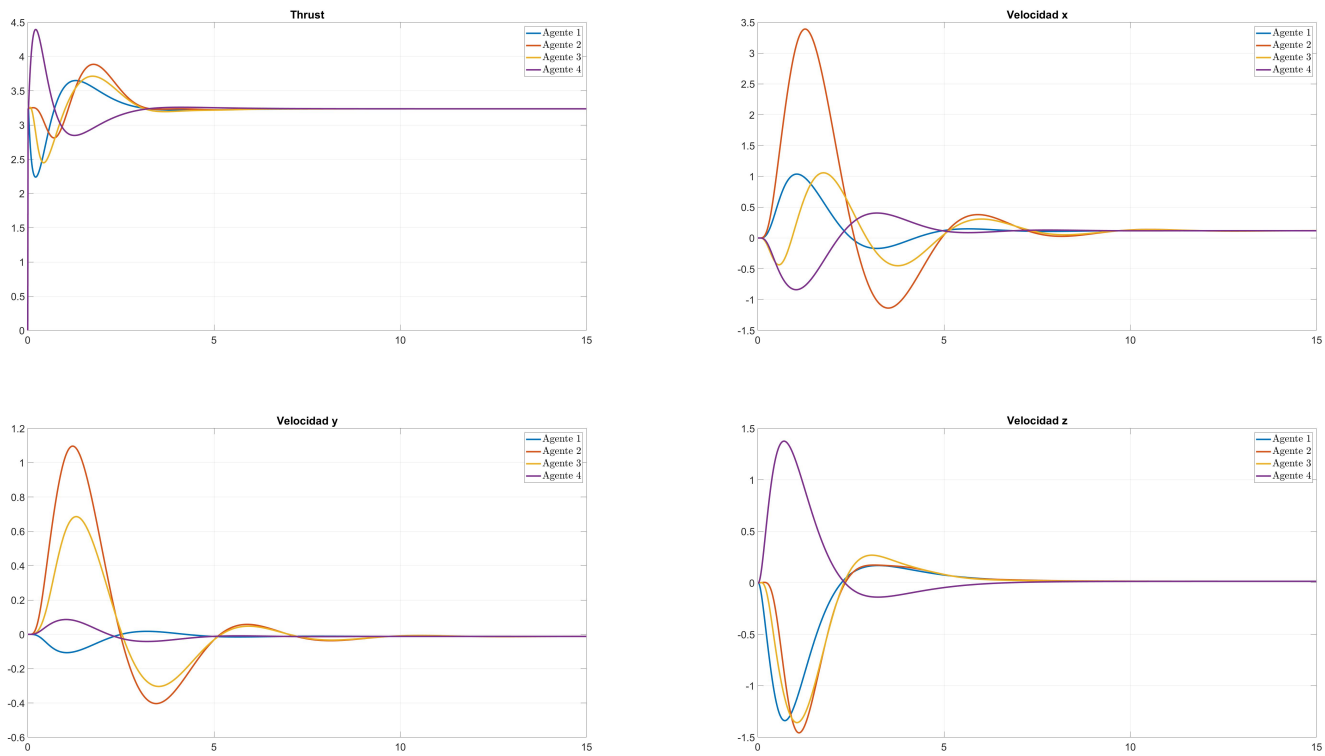


Fig. 4.25: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5. Perturbaciones (Modelo de turbulencias en el viento de Dryden)

En una aplicación real para control de formación de sistemas multi-agentes con cuadricópteros, es evidente que los drones se verán expuestos a diferentes anomalías ambientales, es por esto que se decidió agregar perturbaciones en al esquema de control de formación, con la finalidad de aproximar las simulaciones de forma más realista, y observar su comportamiento, es por esto que se incluyó el modelo de turbulencias en el viento de Dryden que fue tomado de [20].

$$w_{(\cdot)} = w_{(\cdot)_0} + \sum_{n=1}^N a_{(\cdot)_n} \sin(\Omega_{(\cdot)_n} t + \varphi_{(\cdot)_n}) \quad (4.32)$$

donde $\Omega_{(\cdot)_n}$ y $\varphi_{(\cdot)_n}$ se seleccionan de forma aleatoria, $w_{(\cdot)_0}$ es el vector de viento estático, $a_{(\cdot)_n}$ es la amplitud de la sinusoidal. En este caso de estudio se utiliza el modelo de baja altitud (menos de 1000 pies) cuyos parámetros son descritos en la tabla (4.4) tomados de [21].

x-axis	y-axis	z-axis
$\Phi_x(\Omega) = \sigma_x^2 \frac{2L_x}{\pi} \frac{1}{1+(L_x\Omega)^2}$	$\Phi_y(\Omega) = \sigma_y^2 \frac{L_y}{\pi} \frac{1+3(L_y\Omega)^2}{(1+(L_y\Omega)^2)^2}$	$\Phi_z(\Omega) = \sigma_z^2 \frac{L_z}{\pi} \frac{1+3(L_z\Omega)^2}{(1+(L_z\Omega)^2)^2}$
$\sigma_x = \frac{\sigma_w}{(0.1177+0.000823h)^{0.4}}$	$\sigma_y = \sigma_x$	$\sigma_z = 0.11w_0$
$L_x = \frac{h}{(0.1177+0.000823h)^{1.2}}$	$L_x = L_y$	$L_z = h$

Tabla 4.4: Parámetros del modelo de turbulencias de Dryden

Las magnitudes a_n estan dadas por

$$a_n = \sqrt{\Delta\Omega_n \Phi(\Omega_n)} \quad (4.33)$$

Las perturbaciones se implementaron como una adición a las velocidades del modelo dinámico de los UAVs (4.7 - 4.12) de la siguiente manera:

$$\dot{x}_i = V_{xi} + \mathbf{w}_{xi} \quad (4.34)$$

$$\dot{V}_{xi} = (\cos(\psi_i) \sin(\theta_i) \cos(\phi_i) + \sin(\psi_i) \sin(\phi_i)) \frac{1}{m_i} T_i \quad (4.35)$$

$$\dot{y}_i = V_{yi} + \mathbf{w}_{yi} \quad (4.36)$$

$$\dot{V}_{yi} = (\sin(\psi_i) \sin(\theta_i) \cos(\phi_i) - \cos(\psi_i) \sin(\phi_i)) \frac{1}{m_i} T_i \quad (4.37)$$

$$\dot{z}_i = V_{zi} + \mathbf{w}_{zi} \quad (4.38)$$

$$\dot{V}_{zi} = -g + (\cos(\theta_i) \cos(\phi_i)) \frac{1}{m_i} T_i \quad (4.39)$$

En las paginas siguientes se mostraran los resultados obtenidos al añadir las perturbaciones a los esquemas de control de formación, se mostraran las formaciones de un hexágono y un tetraedro, las condiciones iniciales y las topologías de comunicación permanecen iguales a como en las simulaciones sin perturbaciones.

A continuación, se muestran las perturbaciones añadidas al sistema, cabe resaltar que al principio el esquema se simula sin perturbaciones y no es hasta que se alcanza la formación cuando se añaden.

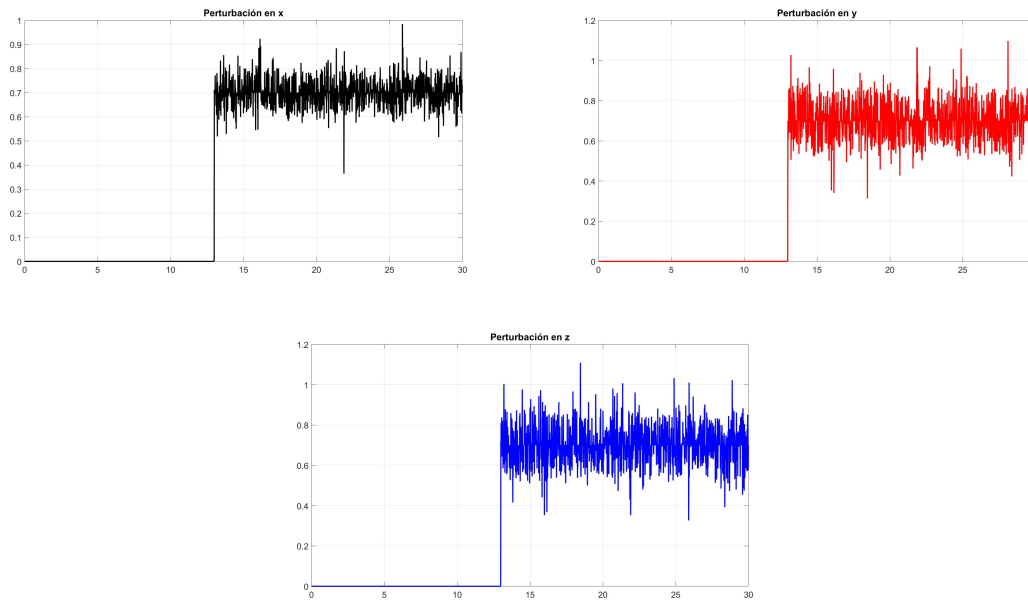


Fig. 4.26: Perturbaciones con respecto a los 3 ejes.

4.5.1. Esquema de control de formación bajo el protocolo (2.6) para un equipo de cuadricópteros con perturbaciones.

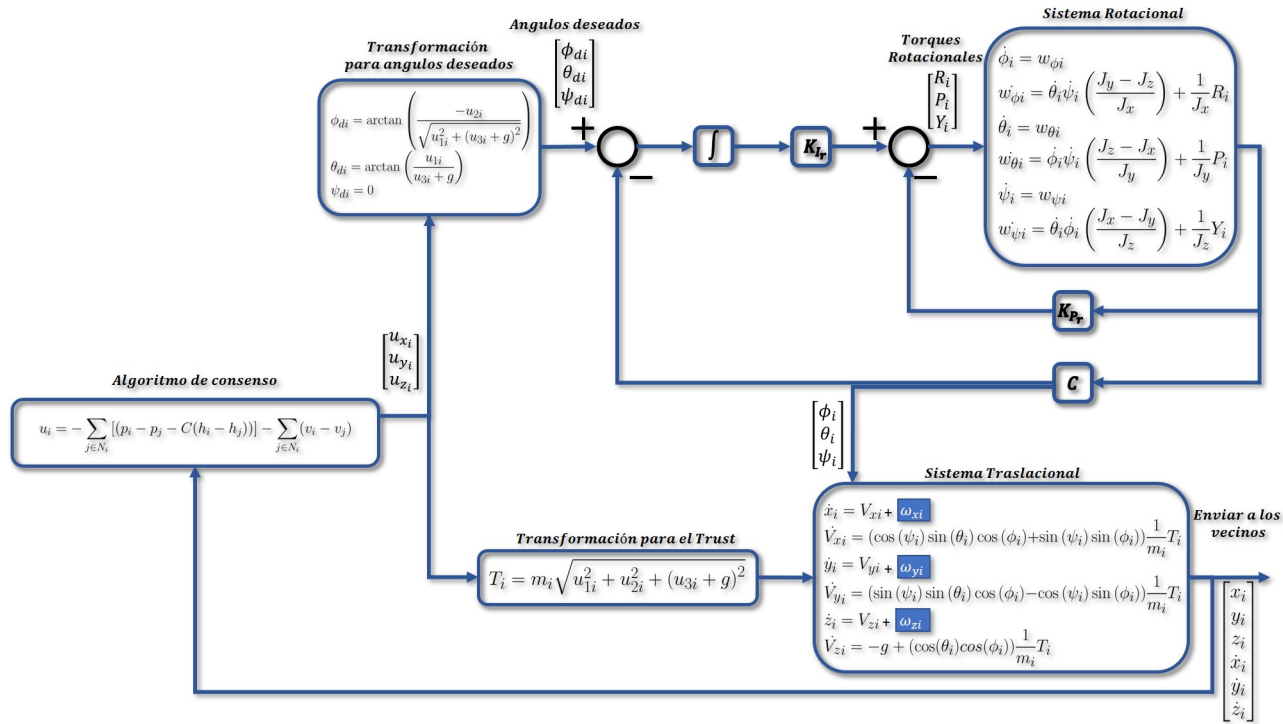


Fig. 4.27: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.5.2. Hexágono formado con el esquema (4.27) con perturbaciones.

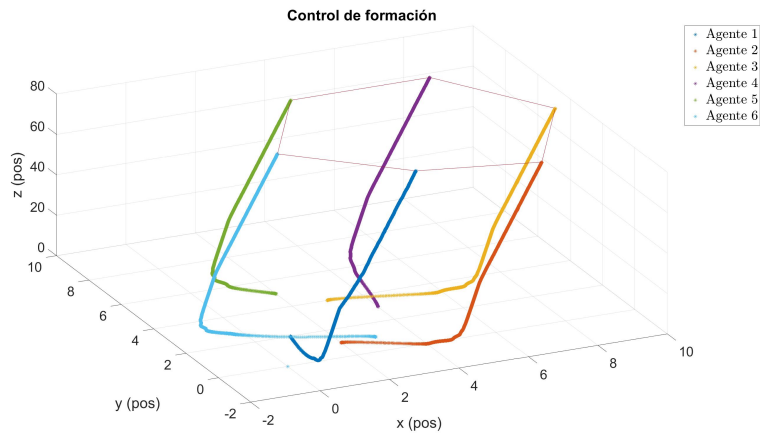


Fig. 4.28: Hexágono formado con el algoritmo tipo observador (2.6) con perturbaciones.

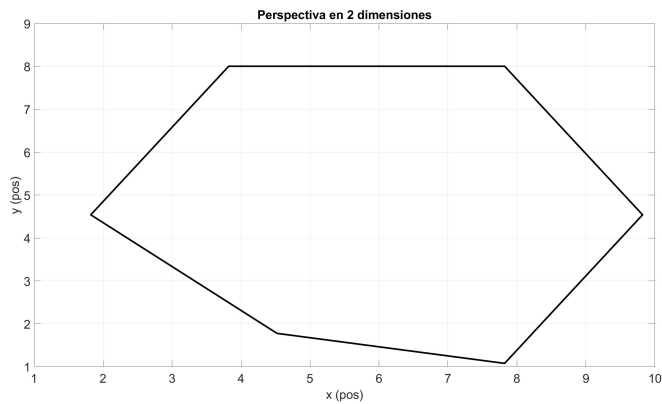


Fig. 4.29: Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.

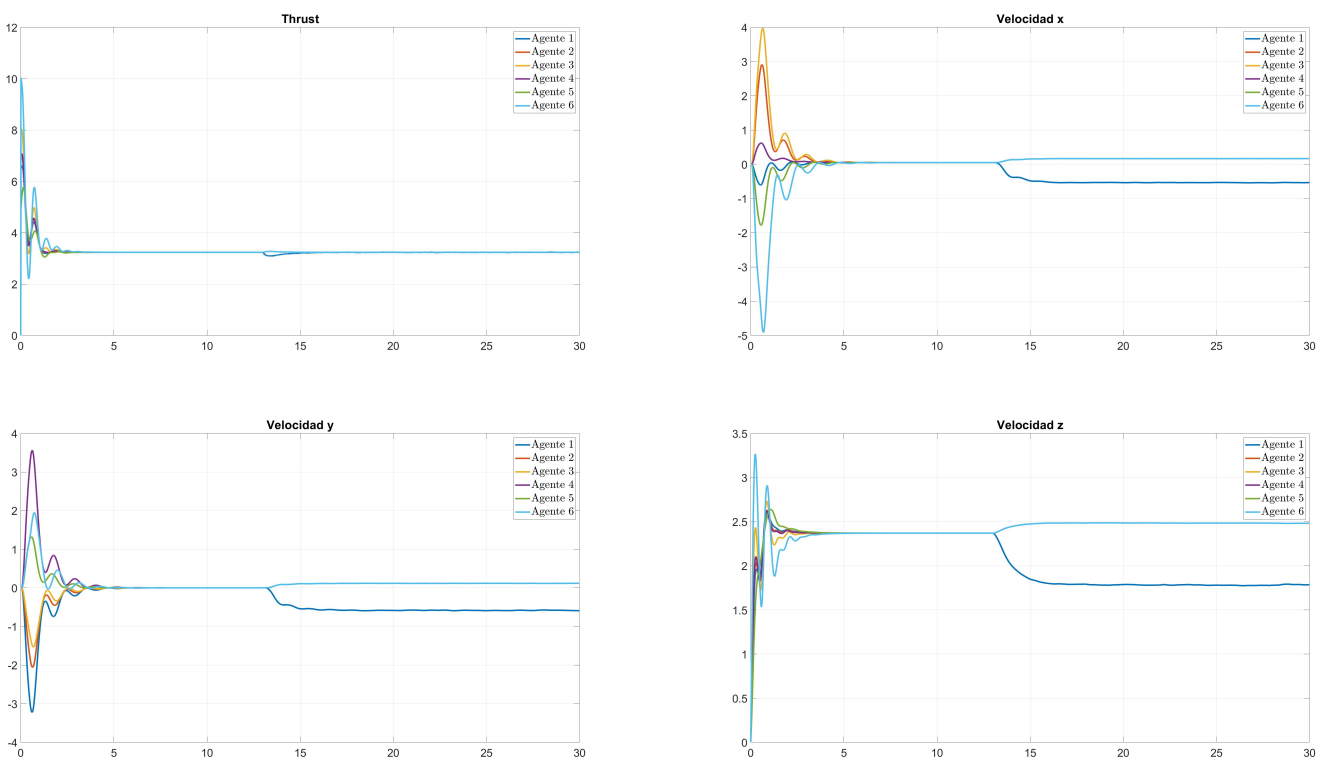


Fig. 4.30: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5.3. Tetraedro formado con el esquema (4.27) con perturbaciones.

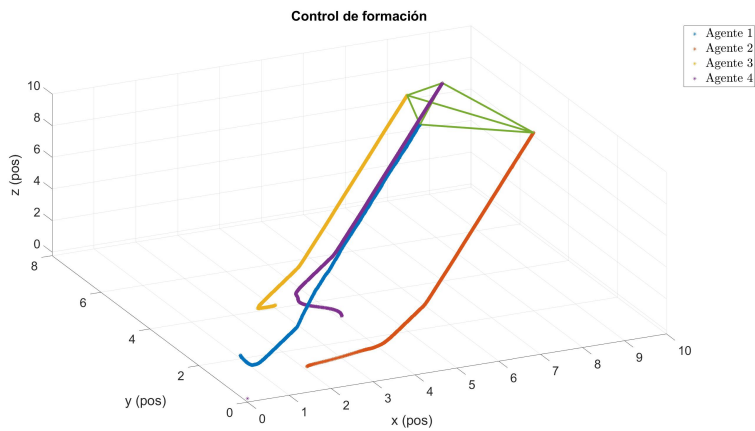


Fig. 4.31: Tetraedro formado con el algoritmo tipo observador (2.6) con perturbaciones.

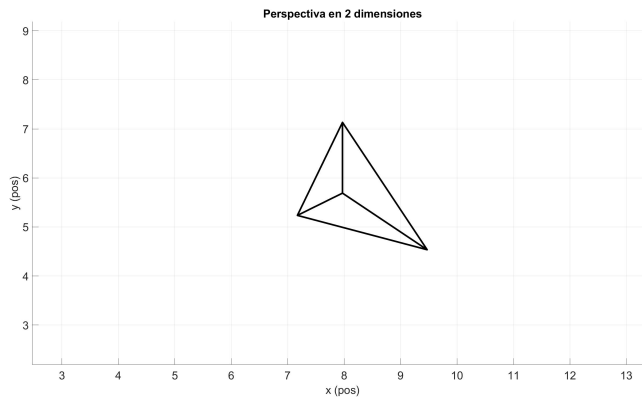


Fig. 4.32: Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.

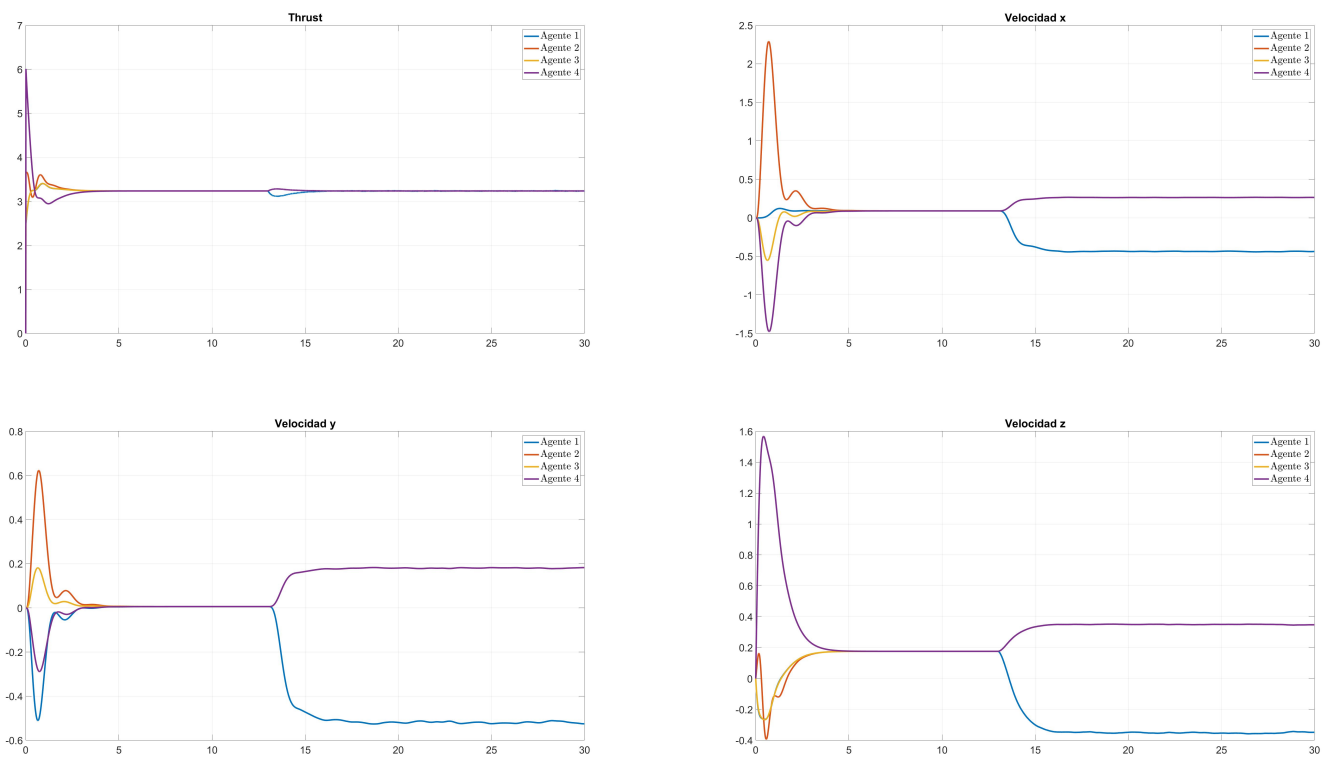


Fig. 4.33: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5.4. Esquema de control de formación bajo el protocolo (2.7) para un equipo de cuadricópteros con perturbaciones.

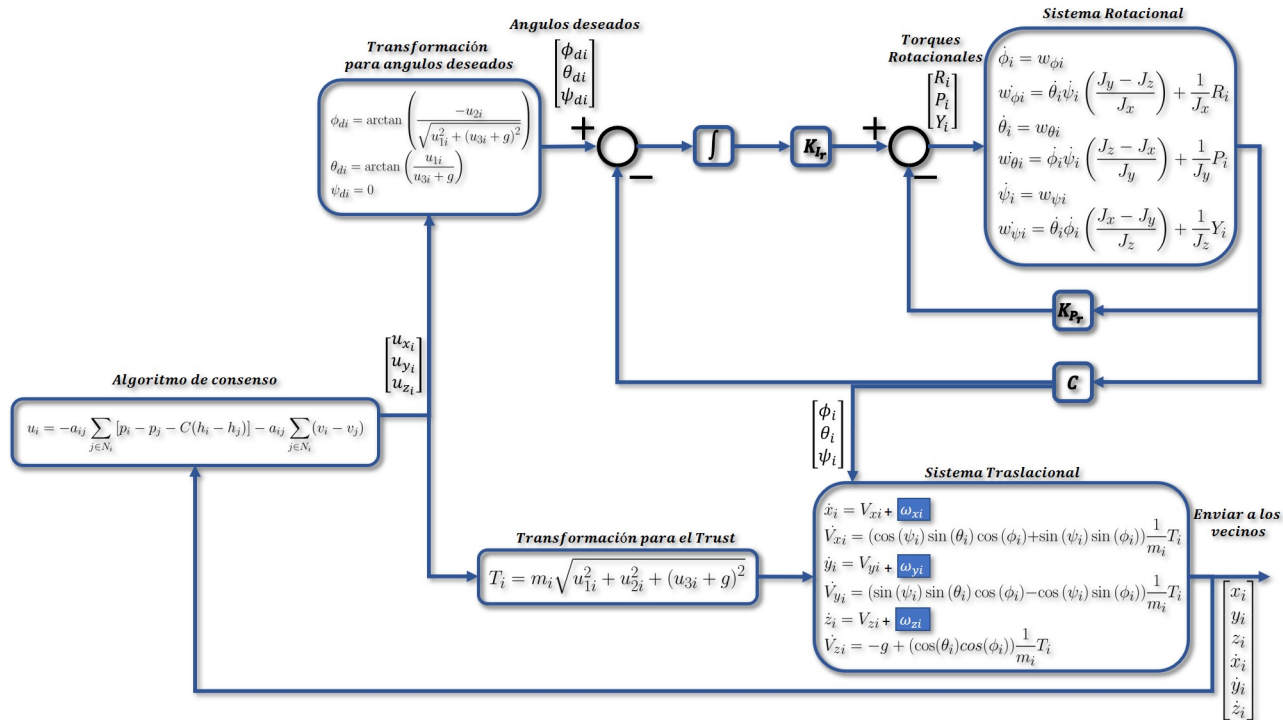


Fig. 4.34: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.5.5. Hexágono formado con el esquema (4.34) con perturbaciones.

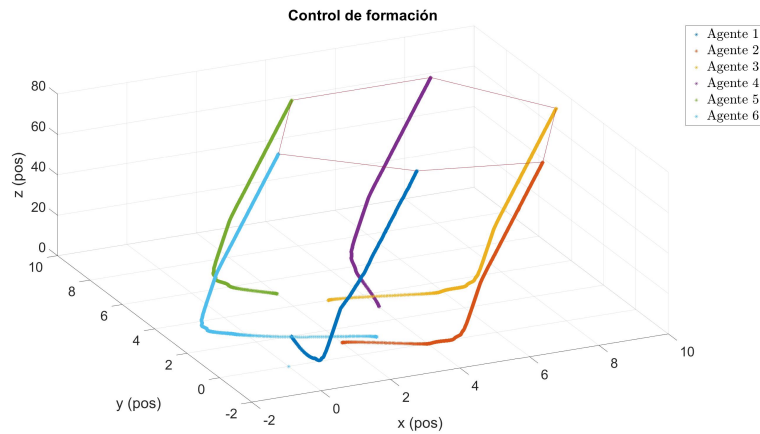


Fig. 4.35: Hexágono formado con el algoritmo tipo observador (2.7) con perturbaciones.

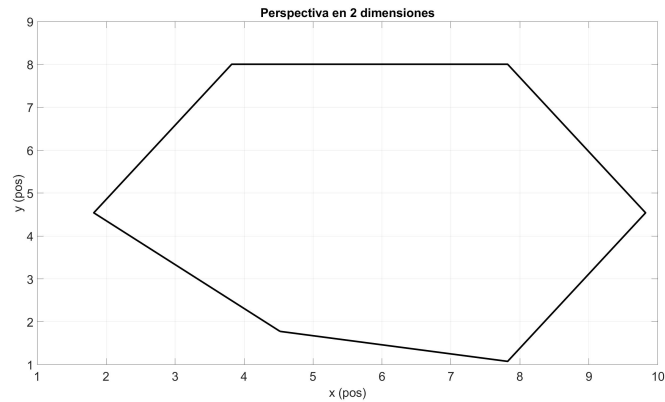


Fig. 4.36: Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.

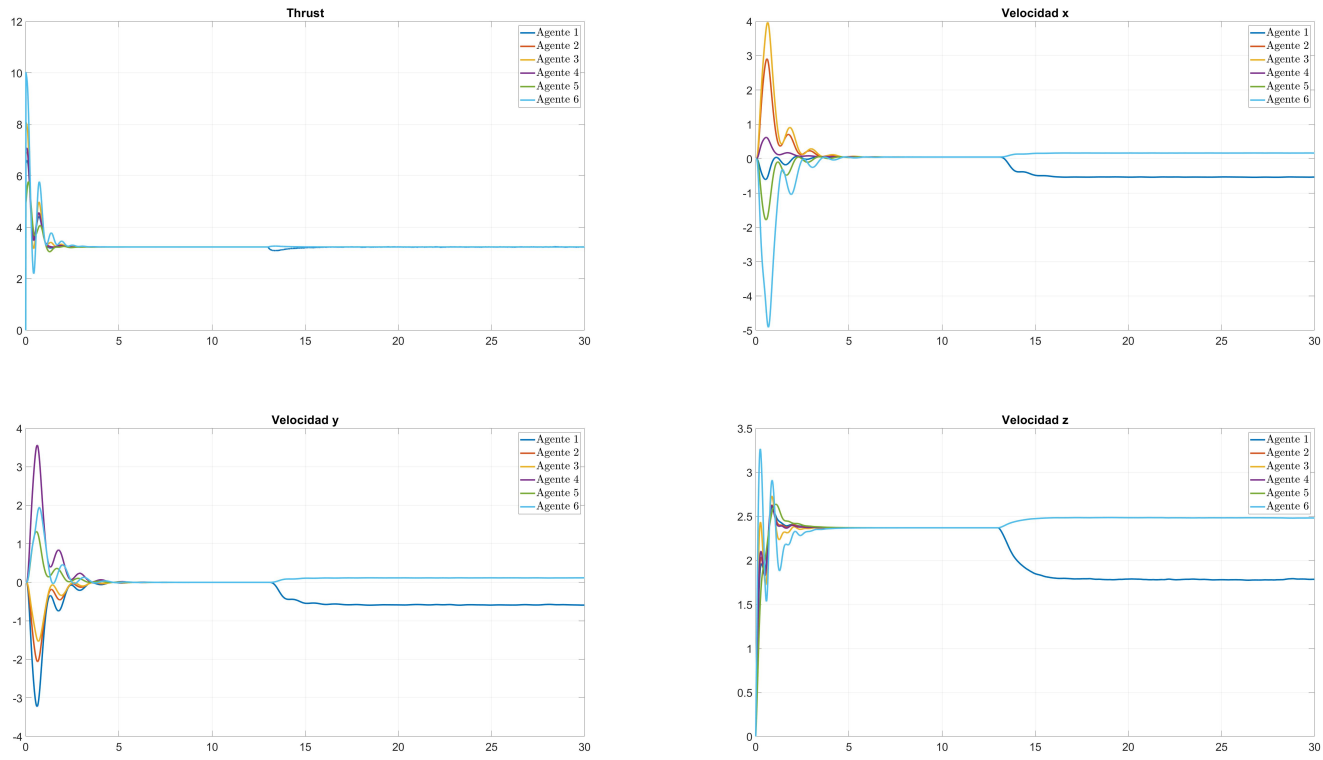


Fig. 4.37: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5.6. Tetraedro formado con el esquema (4.34) con perturbaciones.

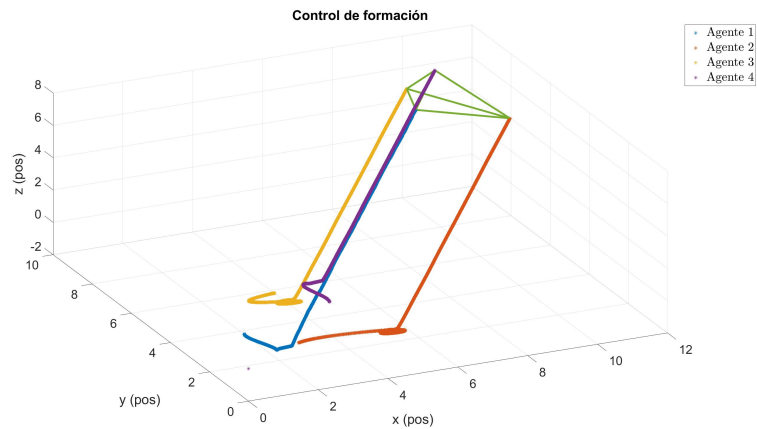


Fig. 4.38: Tetraedro formado con el algoritmo tipo observador (2.7) con perturbaciones.

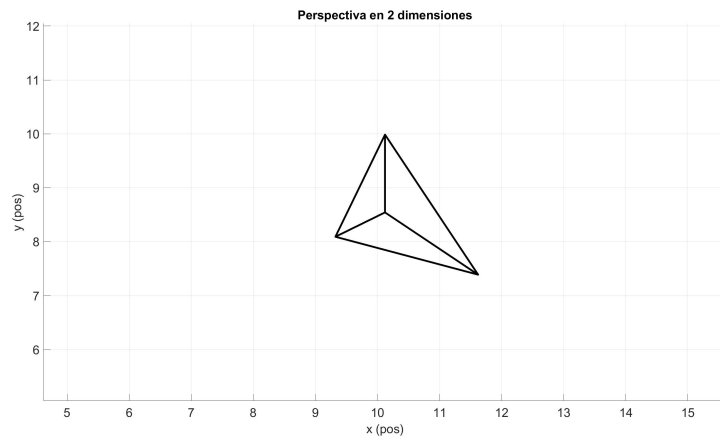


Fig. 4.39: Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.

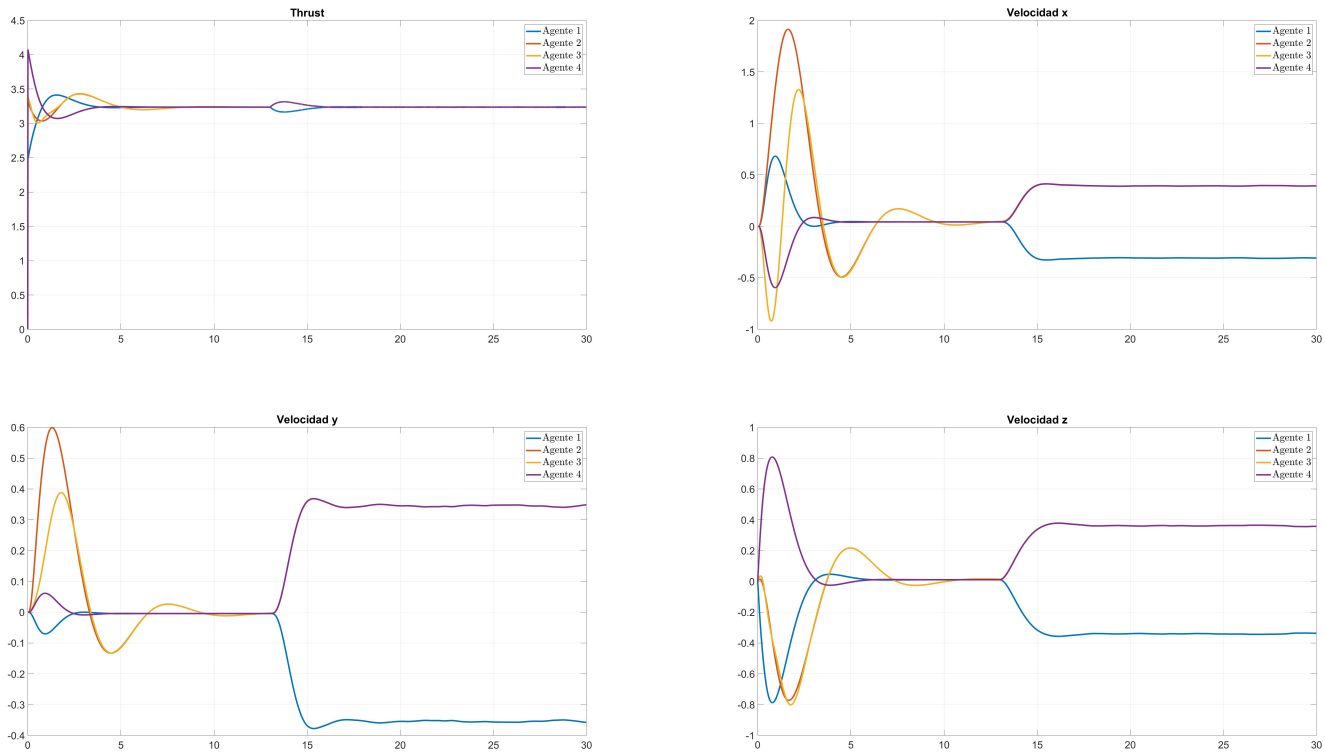


Fig. 4.40: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5.7. Esquema de control de formación bajo el protocolo (2.8) para un equipo de cuadricópteros con perturbaciones.

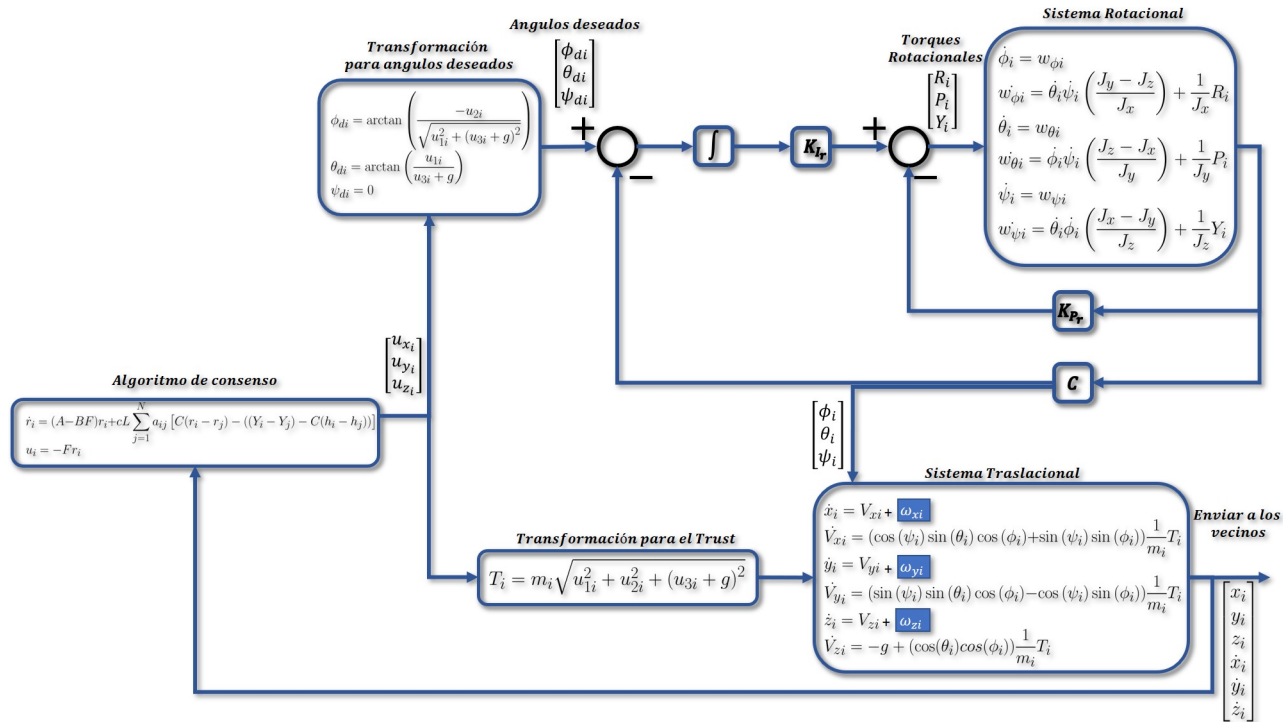


Fig. 4.41: Esquema de control de formación aplicado en un equipo de UAVs sin perturbaciones

4.5.8. Hexágono formado con el esquema (4.41) con perturbaciones.

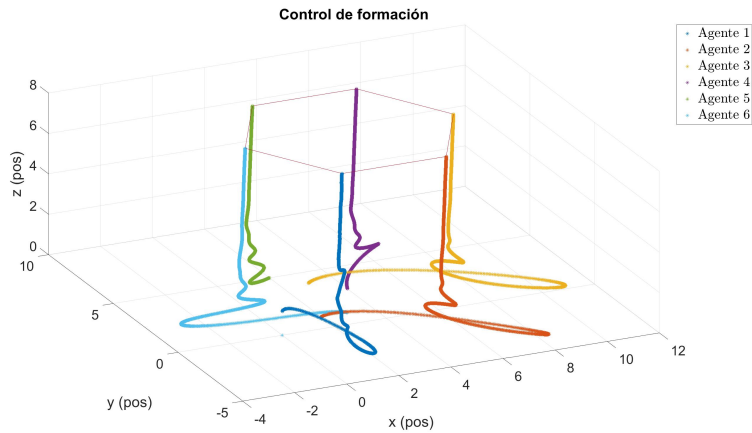


Fig. 4.42: Hexágono formado con el algoritmo tipo observador (2.8) con perturbaciones.

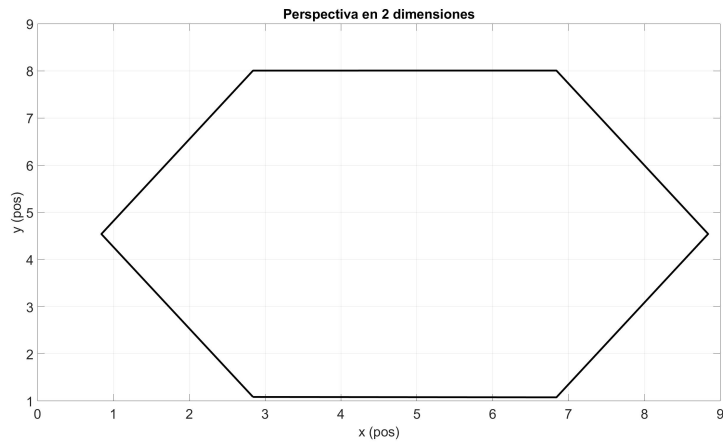


Fig. 4.43: Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.

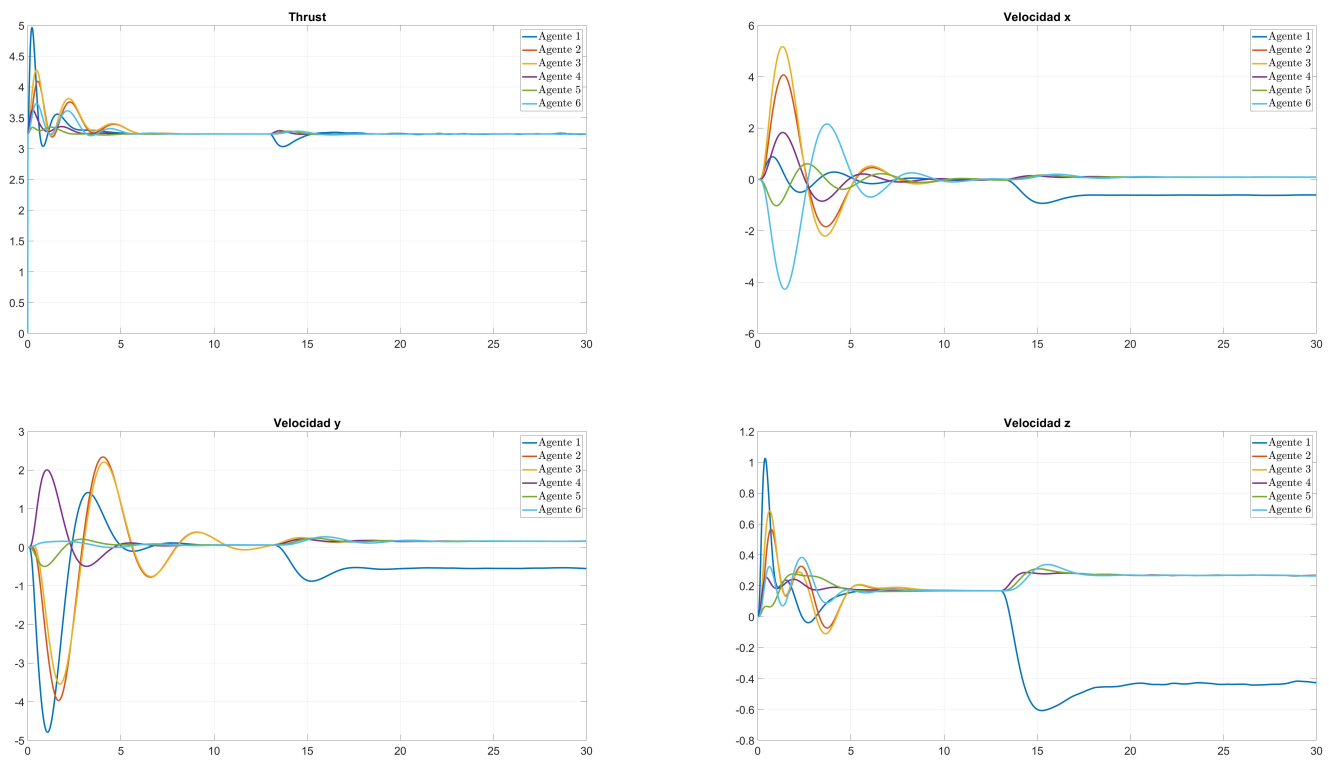


Fig. 4.44: Thrust y velocidades lineales con respecto a los 3 ejes.

4.5.9. Tetraedro formado con el esquema (4.41) con perturbaciones.

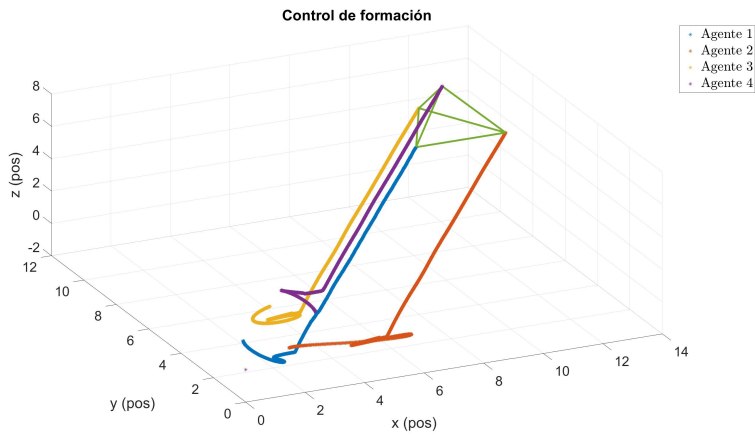


Fig. 4.45: Tetraedro formado con el algoritmo tipo observador (2.8) con perturbaciones.

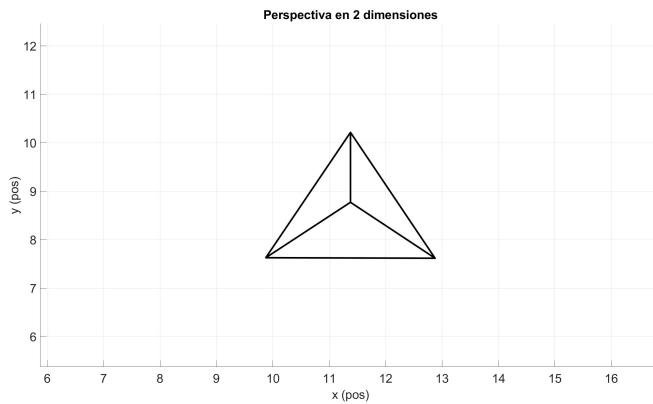


Fig. 4.46: Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.

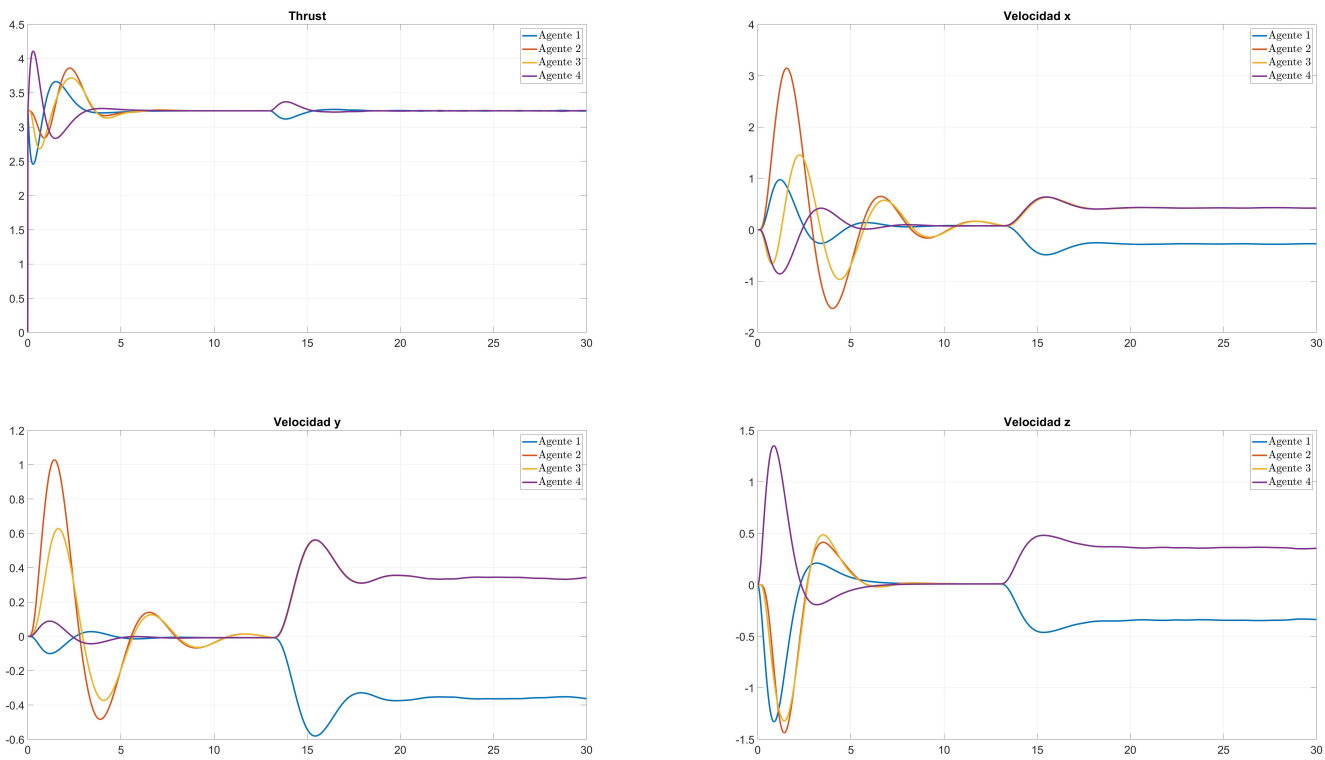


Fig. 4.47: Thrust y velocidades lineales con respecto a los 3 ejes.

Conclusiones

Se diseñó un esquema de control de formación basado en sistemas multi-agentes, se analizó y ejemplificó la teoría de grafos, se definió una topología de comunicación donde los agentes se comunican con sus vecinos más cercanos de forma constante y sin interrupciones, se decidió trabajar con un equipo de UAVs específicamente con un grupo de cuadricópteros. Para empezar a desarrollar el esquema de control de formación, fue necesario incluir un controlador PI interno que estabilizara la planta, definir los algoritmos de consenso para control de formación y transformar la ley de control calculada en estos protocolos, para considerar al dron como un sistema doble integrador y poder manipular sus ángulos y calcular el empuje necesario para realizar movimientos deseados, una vez obtenidas todas estas herramientas se prosiguió a desarrollar las simulaciones pertinentes en el software Matlab.

Con la lectura del estado del arte, se encontró la oportunidad de trabajar con sistemas multi-agentes que fuesen afectados por perturbaciones ambientales, así que se decidió añadir una entrada desconocida que afectara las velocidades de los UAVs en las simulaciones.

Se utilizaron 3 diferentes algoritmos para resolver la problemática de control de formación en sistemas multi-agentes con un equipo de cuadricópteros modelo *crazyfly* 2.0, se desarrollaron las simulaciones pertinentes, y se concluye que, los 3 algoritmos consiguen las formaciones cuando no se presentan perturbaciones en sus esquemas, pero solamente el algoritmo tipo observador (2.8) es capaz de mantener la formación en presencia de dichas perturbaciones, en el capítulo 4 de este documento de tesis se pueden visualizar

dichas simulaciones para un sistema de 6 agentes donde se busca formar un hexágono y un sistema con 4 agentes para formar un tetraedro, pero en la sección de anexos se presentan 2 formaciones más, la de un cuadrado y la de un pentágono.

Se concluye que los objetivos planteados al inicio de este trabajo de tesis fueron cumplidos y la hipótesis fue demostrada, se desarrolló un esquema de control de formación basado en sistemas multi-agentes capaz de compensar perturbaciones ambientales.

Como trabajos futuros se propone agregar un esquema para evitar colisiones y una estructura de control diseñada específicamente para compensar perturbaciones, por dar un ejemplo un esquema de control de modelo de seguimiento.

Bibliografía

- [1] G. V. Pacheco, R. H. Fonseca, V. E. Neto, and A. S. Brandao, “Control signal weighting for robot formation,” pp. 111–116, 2018. [2](#), [10](#)
- [2] H. Guzey, “Adaptive consensus-based formation control of fixed-wing muav’s,” pp. 184–187, 2017. [3](#), [9](#), [10](#)
- [3] L. He, P. Bai, X. Liang, J. Zhang, and W. Wang, “Feedback formation control of uav swarm with multiple implicit leaders,” *Aerospace Science and Technology*, vol. 72, pp. 327–334, 2018. [3](#), [9](#)
- [4] D. V. Redrovan and D. Kim, “Multiple quadrotors flight formation control based on sliding mode control and trajectory tracking,” in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2018, pp. 1–6. [4](#)
- [5] X. Yang, C.-C. Hua, J. Yan, and X.-P. Guan, “Adaptive formation control of cooperative teleoperators with intermittent communications,” *IEEE Transactions on Cybernetics*, no. 99, pp. 1–10, 2018. [5](#), [9](#)
- [6] Z. Sun, G. Zhang, Y. Lu, and W. Zhang, “Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation,” *ISA transactions*, vol. 72, pp. 15–24, 2018. [5](#), [9](#)
- [7] Y.-D. Zhao, D.-E. Kim, H.-N. Yoon, S.-I. Han, and J.-M. Lee, “Consensus formation control of multiple wheeled mobile robots,” pp. 1081–1086, 2017. [5](#), [9](#)

-
- [8] S. Yazdani and M. Haeri, “Flocking of multi-agent systems with multiple second-order uncoupled linear dynamics and virtual leader,” *IET Control Theory & Applications*, vol. 10, no. 8, pp. 853–860, 2016. [7](#)
- [9] Z. A. S. Dashti, M. Gholami, and M. Hajimani, “Brain emotional learning based intelligent controller for velocity control of an electro hydraulic servo system,” *IOSR J Elec Electron Eng*, vol. 12, no. 4, pp. 29–35, 2017. [7](#)
- [10] Z. Hou, W. Wang, G. Zhang, and C. Han, “A survey on the formation control of multiple quadrotors,” pp. 219–225, 2017. [8](#)
- [11] C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*. ACM, 1987, vol. 21, no. 4. [8](#)
- [12] H. Li and Y. Shi, “Distributed model predictive control of constrained nonlinear systems with communication delays,” *Systems & Control Letters*, vol. 62, no. 10, pp. 819–826, 2013. [8](#)
- [13] Y. Zhang, Y. Fan, C. Song, C. Dong, and L. Wang, “Multi-agent rendezvous control based on event-triggered mechanism,” pp. 780–784, 2017. [9](#)
- [14] M. Ille and T. Namerikawa, “Collision avoidance between multi-uav-systems considering formation control using mpc,” *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*, pp. 651–656, 2017. [9](#)
- [15] J. Guerrero-Castellanos, A. Vega-Alonzo, N. Marchand, S. Durand, J. Linares-Flores, and G. Mino-Aguilar, “Real-time event-based formation control of a group of vtol-uavs,” *2017 3rd International Conference on Event-Based Control, Communication and Signal Processing (EBCCSP)*, pp. 1–8, 2017. [24](#), [44](#)
- [16] Z. Li and Z. Duan, *Cooperative control of multi-agent systems: a consensus region approach*. CRC Press, 2014. [25](#), [26](#), [30](#), [33](#)
- [17] O. T. Gerardo, “Control tolerante a fallas aplicado a vehículos aéreos no tripulados.” Ph.D. dissertation, Tecnológico Nacional de México, 2018. [41](#)

- [18] C. Luis and J. L. Ny, “Design of a trajectory tracking controller for a nano quadcopter,” *Technical report, Mobile Robotics and Autonomous Systems Laboratory, Polytechnique Montreal.*, 2016. 43
- [19] J. Förster, “System identification of the crazyflie 2.0 nano quadrocopter,” B.S. thesis, ETH Zurich, 2015. 45
- [20] J. W. Langelaan, N. Alley, and J. Neidhoefer, “Wind field estimation for small unmanned aerial vehicles,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1016–1030, 2011. 69
- [21] MathWorks, “Dryden wind turbulence model (continuous),” <https://la.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>, 2020. 69

**A.1. Cuadrado formado con el esquema (4.3)
sin perturbaciones.**

A continuación en la tabla (A.1) se muestran las condiciones iniciales para los estados de todos los cuadricópteros de un sistema multi-agente donde $N = 4$.

Condiciones iniciales del cuadricóptero				
	Agente 1	Agente 2	Agente 3	Agente 4
Posición en x	1	2	3	4
Posición en y	2	1	4	3
Posición en z	0	0	0	0
Velocidad en x	0	0	0	0
Velocidad en y	0	0	0	0
Velocidad en z	0	0	0	0
Posición angular ϕ	0	0	0	0
Posición angular θ	0	0	0	0
Posición angular ψ	0	0	0	0
Velocidad angular $\dot{\phi}$	0	0	0	0
Velocidad angular $\dot{\theta}$	0	0	0	0
Velocidad angular $\dot{\psi}$	0	0	0	0

Tabla A.1: Condiciones iniciales para la formación final de un cuadrado.

La matriz H que contiene la formación rígida para el sistema se muestra en (A.1), como objetivo final se busca realizar un cuadrado.

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.1})$$

La topología de comunicación está definida como se muestra en la figura (A.1) y se muestra así mismo su respectiva matriz de adyacencias.

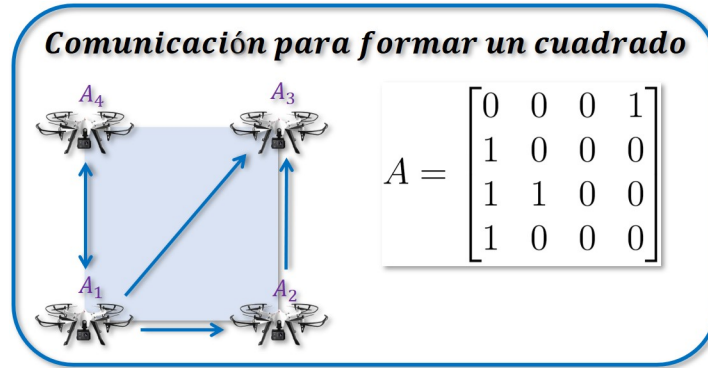


Fig. A.1: Topología de comunicación y matriz de adyacencias equivalente para la formación de un cuadrado.

Todas las simulaciones mostradas en las siguientes páginas, donde el objetivo sea formar un cuadrado, se utilizarán, las condiciones iniciales (A.1), la matriz H (A.1) y la misma topología de comunicación (A.1).

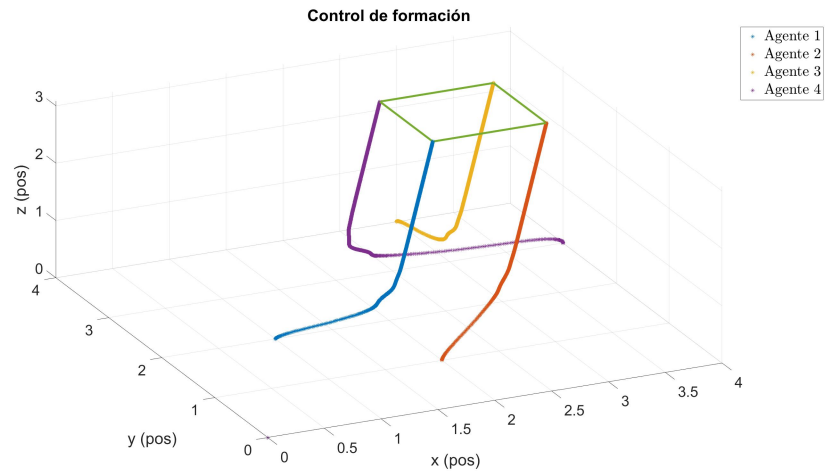


Fig. A.2: Cuadrado formado con el algoritmo tipo observador (2.6) sin perturbaciones.

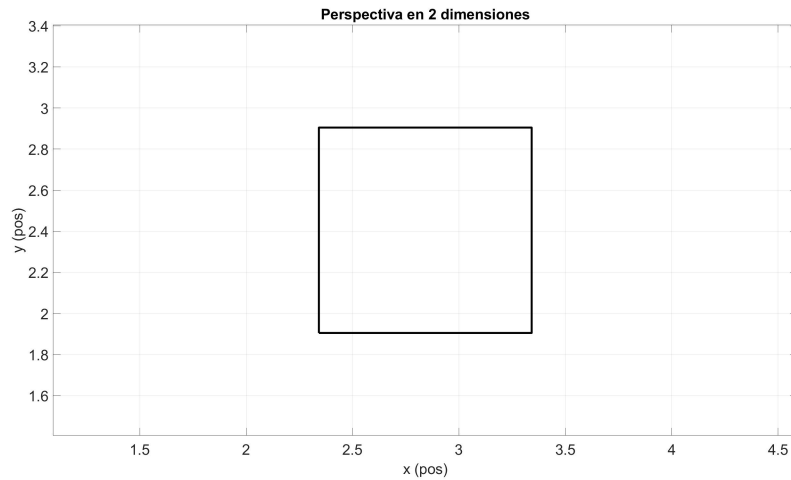


Fig. A.3: Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.

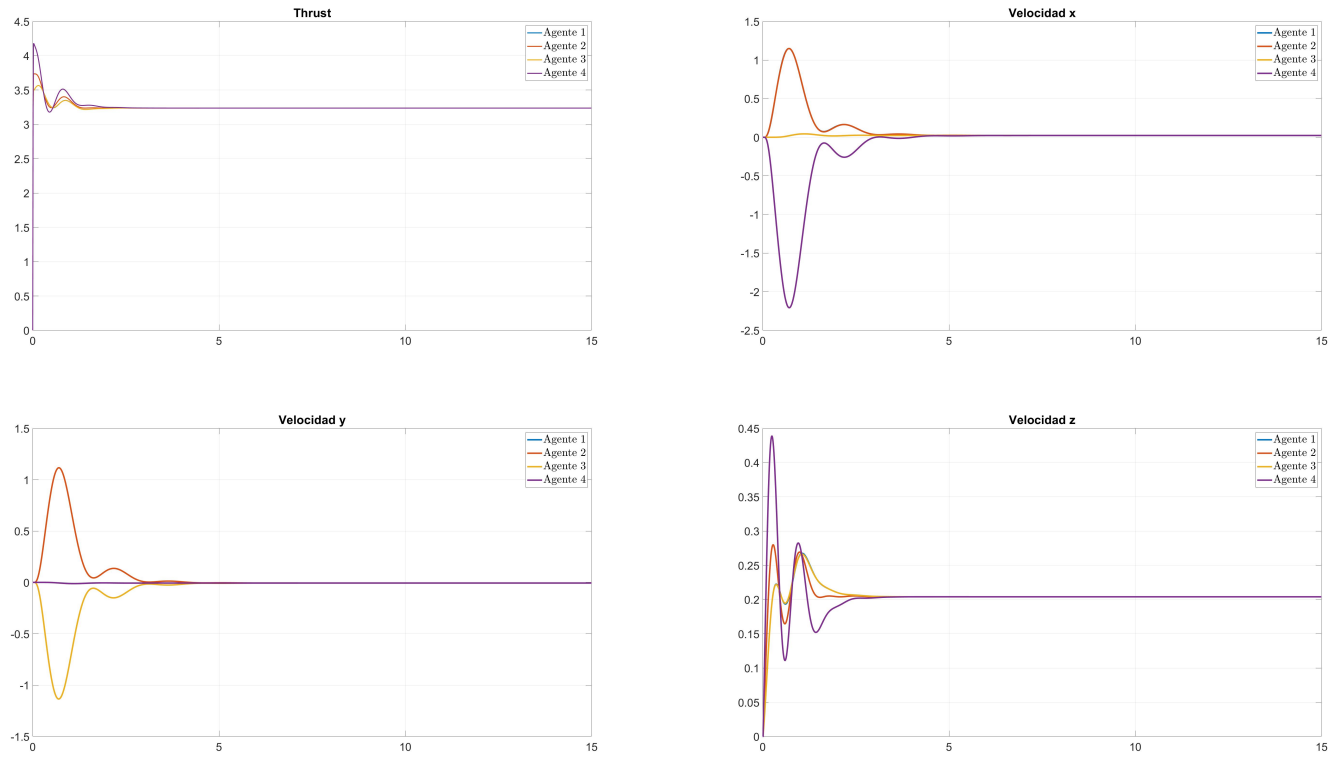


Fig. A.4: Thrust y velocidades lineales con respecto a los 3 ejes.

A.2. Pentágono formado con el esquema (4.3) sin perturbaciones.

A continuación en la tabla (A.2) se muestran las condiciones iniciales para los estados de todos los cuadricópteros de un sistema multi-agente donde $N = 5$.

Condiciones iniciales del cuadricóptero					
	Agente 1	Agente 2	Agente 3	Agente 4	Agente 5
Posición en x	1	2	3	4	2
Posición en y	2	1	4	3	5
Posición en z	0	0	0	0	0
Velocidad en x	0	0	0	0	0
Velocidad en y	0	0	0	0	0
Velocidad en z	0	0	0	0	0
Posición angular ϕ	0	0	0	0	0
Posición angular θ	0	0	0	0	0
Posición angular ψ	0	0	0	0	0
Velocidad angular $\dot{\phi}$	0	0	0	0	0
Velocidad angular $\dot{\theta}$	0	0	0	0	0
Velocidad angular $\dot{\psi}$	0	0	0	0	0

Tabla A.2: Condiciones iniciales para la formación final de un pentágono.

La matriz H que contiene la formación rígida para el sistema se muestra en (A.2), como objetivo final se busca realizar un pentágono.

$$H = \begin{bmatrix} 0 & 1 & 1 + \frac{\sin 18}{\sin 90} & \frac{1}{2} & -\frac{\sin 18}{\sin 90} \\ 0 & 0 & \frac{\sin 72}{\sin 90} & \frac{1}{2 \sin 36} + \frac{1}{2 \tan 36} & \frac{\sin 72}{\sin 90} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.2})$$

A.2 Pentágono formado con el esquema (4.3) sin perturbaciones.

La topología de comunicación está definida como se muestra en la figura (A.5) y se muestra así mismo su respectiva matriz de adyacencias.

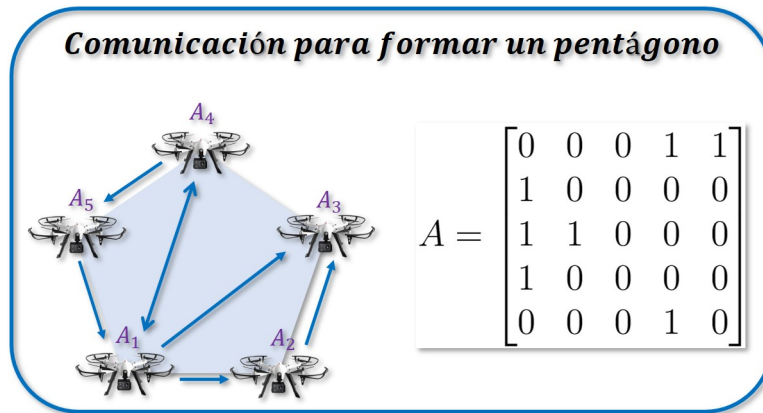


Fig. A.5: Topología de comunicación y matriz de adyacencias equivalente para la formación de un pentágono.

Todas las simulaciones mostradas en las siguientes páginas, donde el objetivo sea formar un pentágono, se utilizarán, las condiciones iniciales (A.2), la matriz H (A.2) y la misma topología de comunicación (A.5).

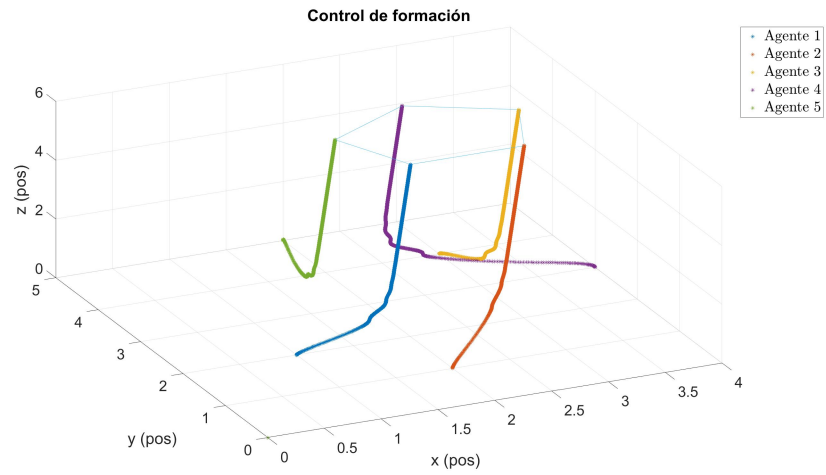


Fig. A.6: Pentágono formado con el algoritmo tipo observador (2.6) sin perturbaciones.

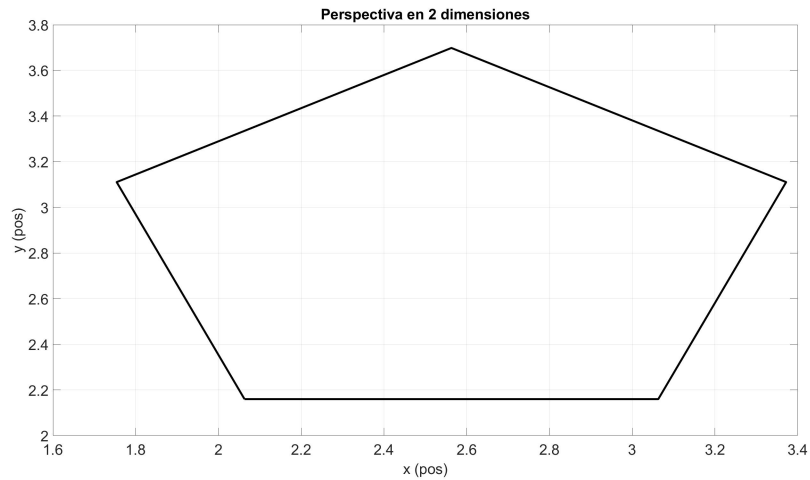


Fig. A.7: Perspectiva superior de la formación con el algoritmo (2.6) sin perturbaciones.

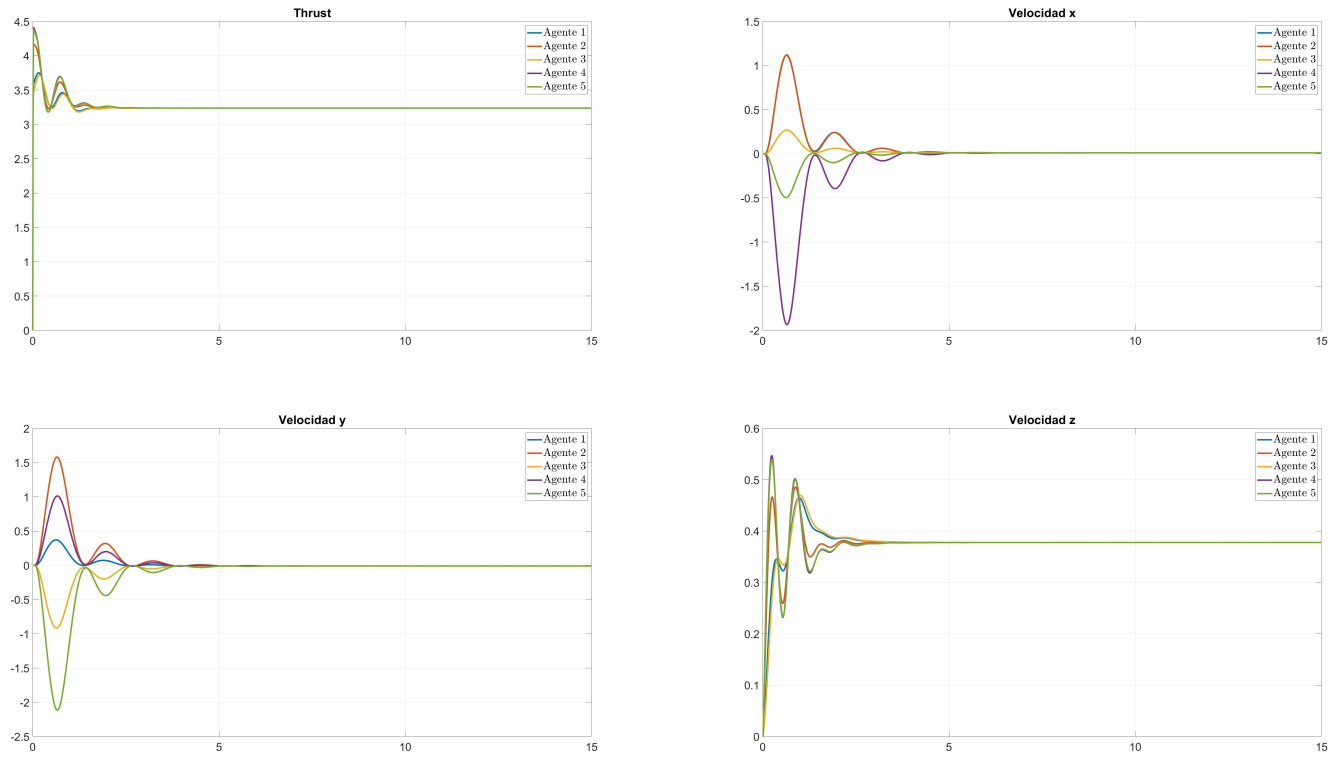


Fig. A.8: Thrust y velocidades lineales con respecto a los 3 ejes.

A.3. Cuadrado formado con el esquema (4.12) sin perturbaciones.

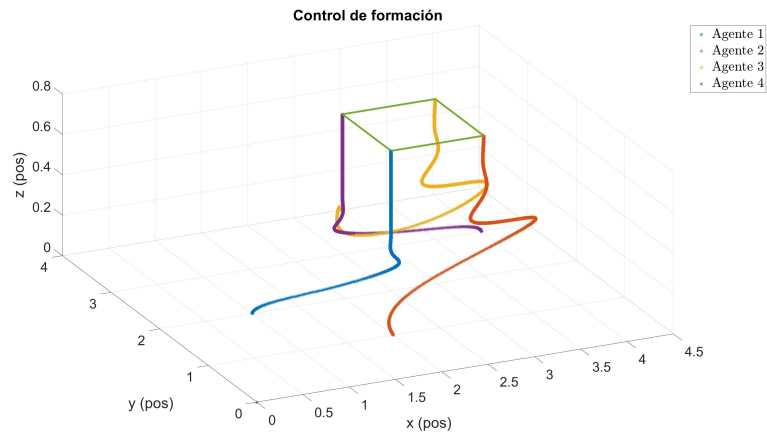


Fig. A.9: Cuadrado formado con el algoritmo tipo observador (2.7) sin perturbaciones.

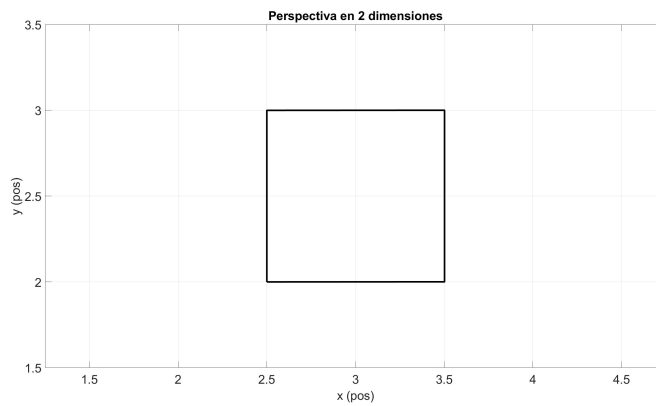


Fig. A.10: Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.

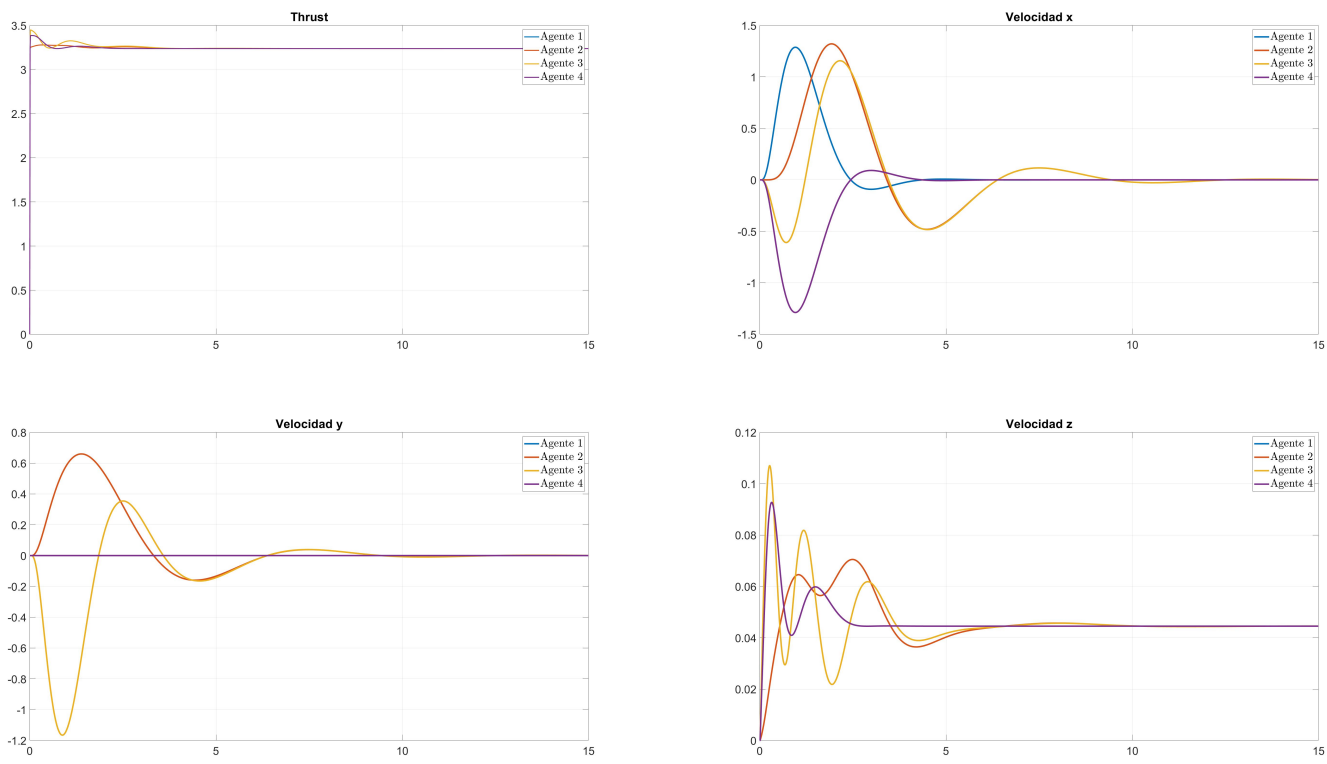


Fig. A.11: Thrust y velocidades lineales con respecto a los 3 ejes.

A.4. Pentágono formado con el esquema (4.12) sin perturbaciones.

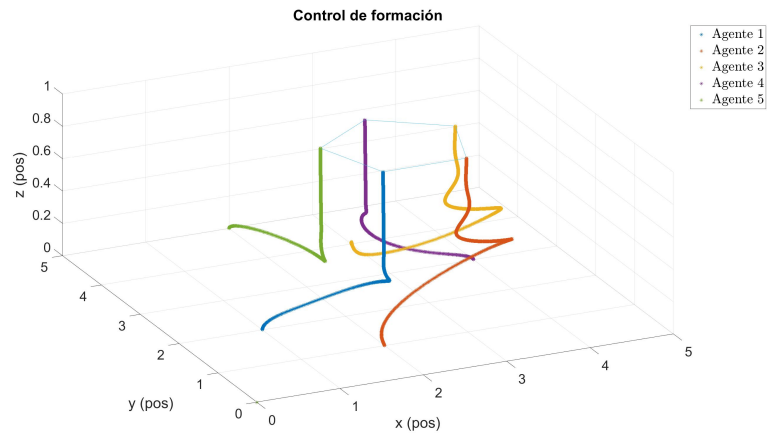


Fig. A.12: Pentágono formado con el algoritmo tipo observador (2.7) sin perturbaciones.

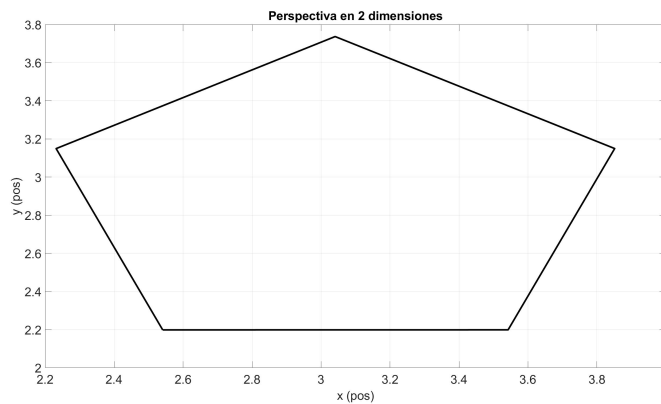


Fig. A.13: Perspectiva superior de la formación con el algoritmo (2.7) sin perturbaciones.

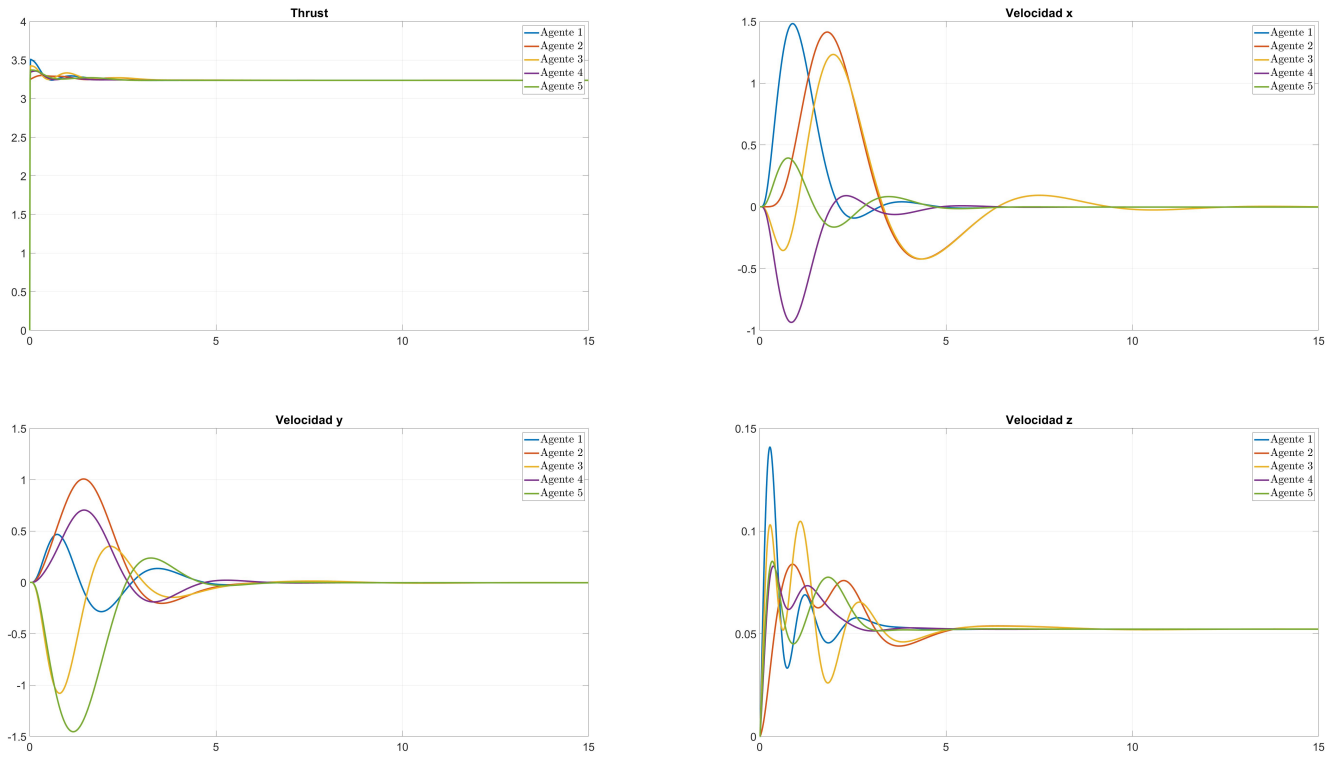


Fig. A.14: Thrust y velocidades lineales con respecto a los 3 ejes.

A.5. Cuadrado formado con el esquema (4.19) sin perturbaciones.

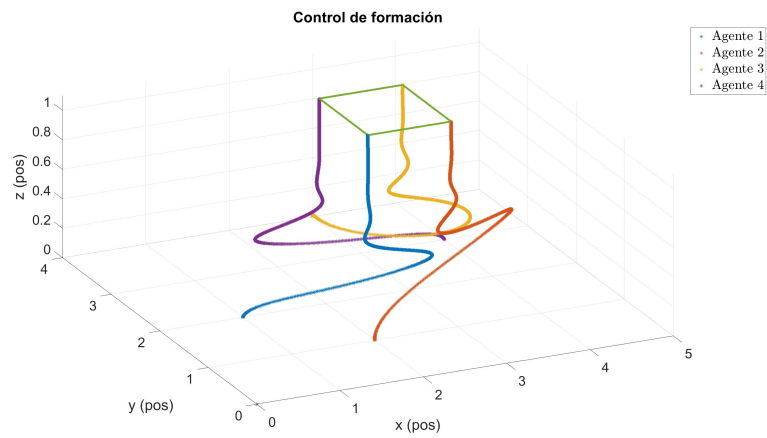


Fig. A.15: Cuadrado formado con el algoritmo tipo observador (2.8) sin perturbaciones.

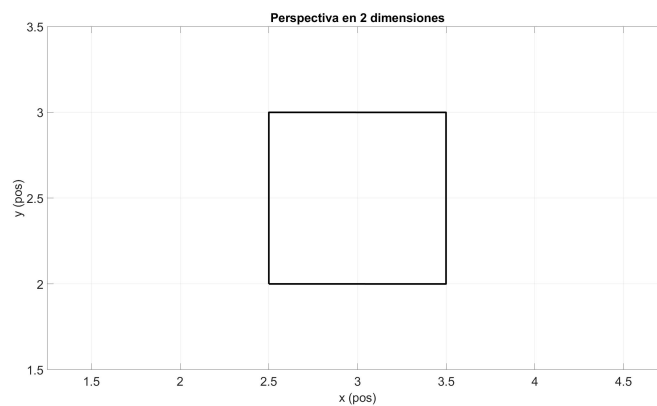


Fig. A.16: Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.

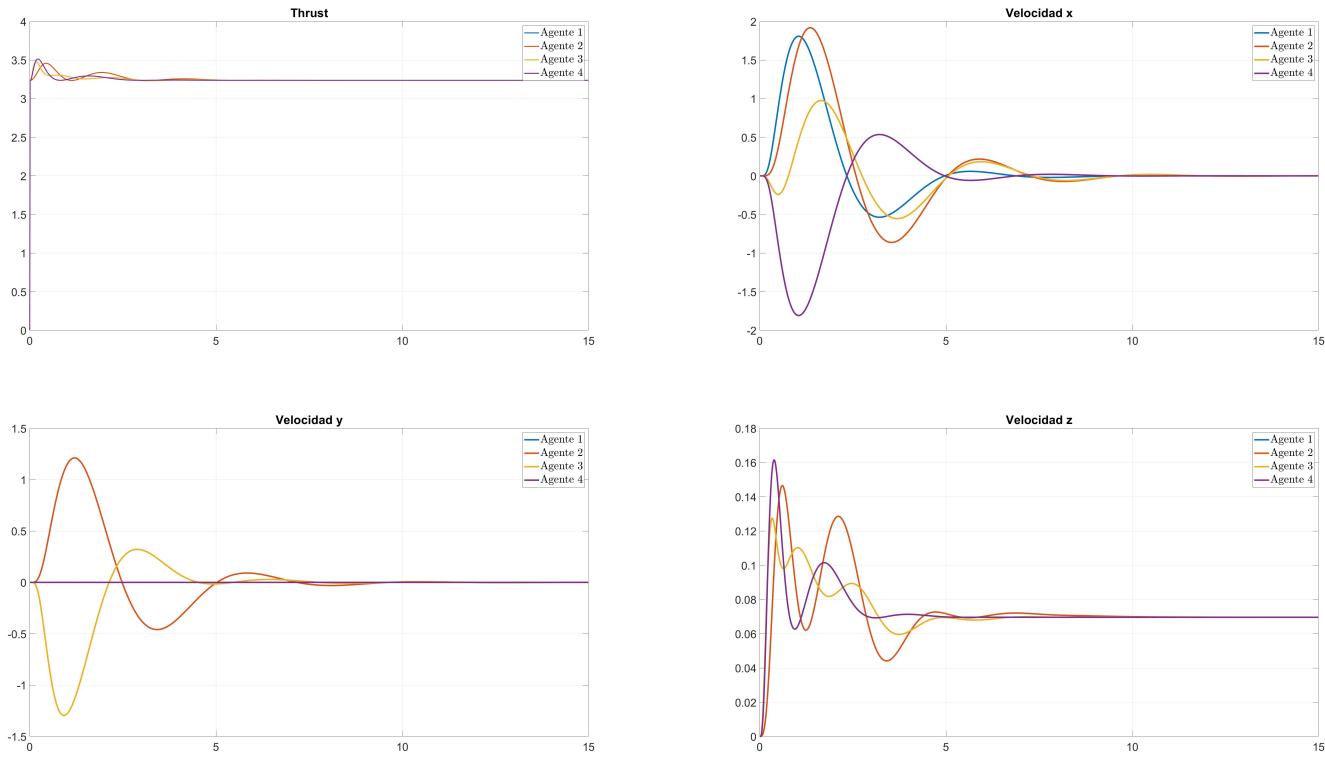


Fig. A.17: Thrust y velocidades lineales con respecto a los 3 ejes.

A.6. Pentágono formado con el esquema (4.19) sin perturbaciones.

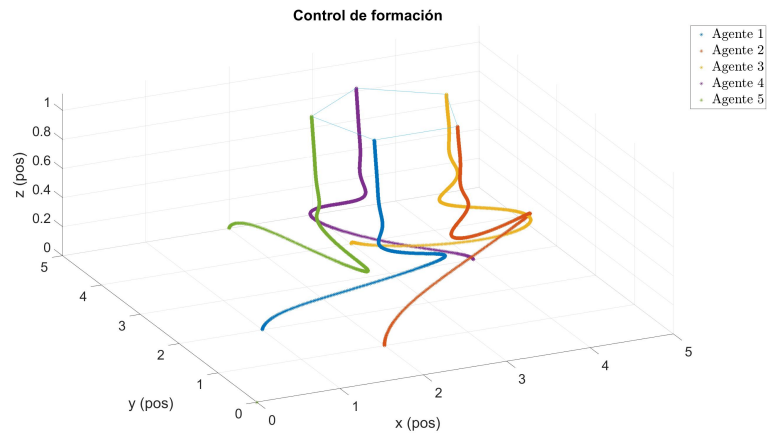


Fig. A.18: Pentágono formado con el algoritmo tipo observador (2.8) sin perturbaciones.

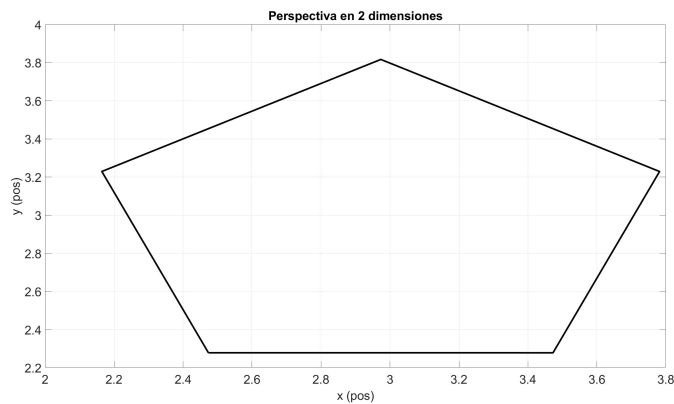


Fig. A.19: Perspectiva superior de la formación con el algoritmo (2.8) sin perturbaciones.

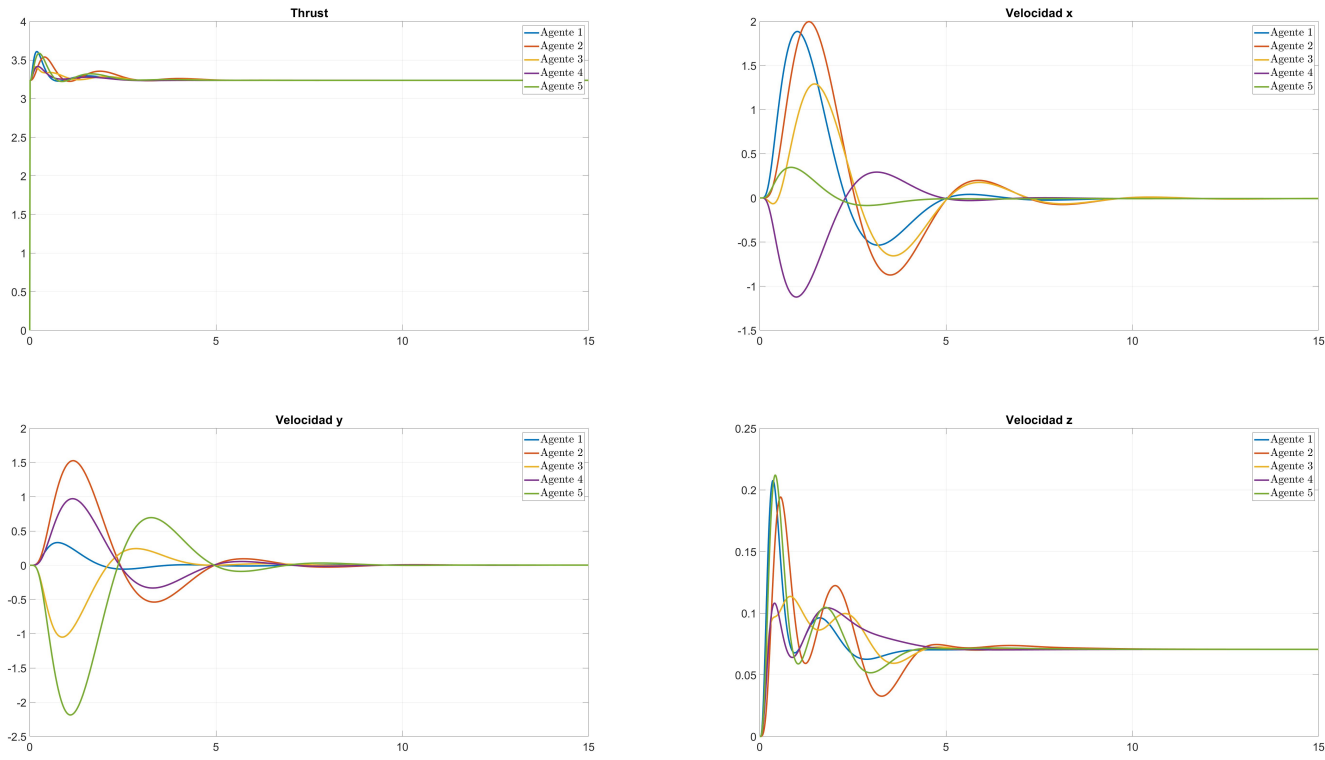


Fig. A.20: Thrust y velocidades lineales con respecto a los 3 ejes.

A.7. Cuadrado formado con el esquema (4.27) con perturbaciones.

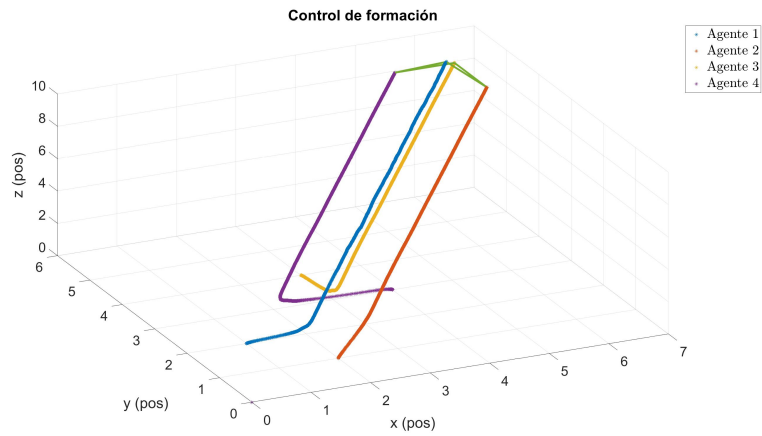


Fig. A.21: Cuadrado formado con el algoritmo tipo observador (2.6) con perturbaciones.

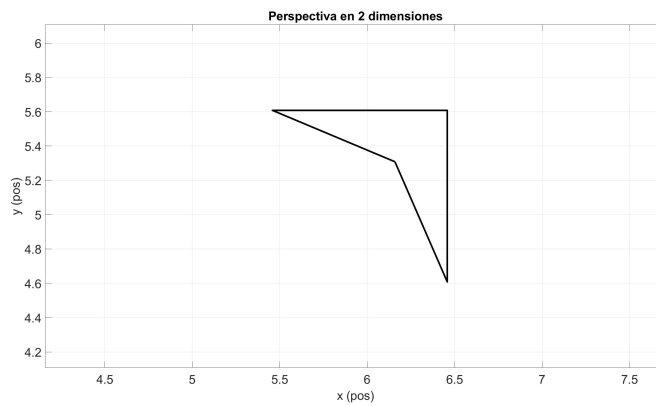


Fig. A.22: Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.

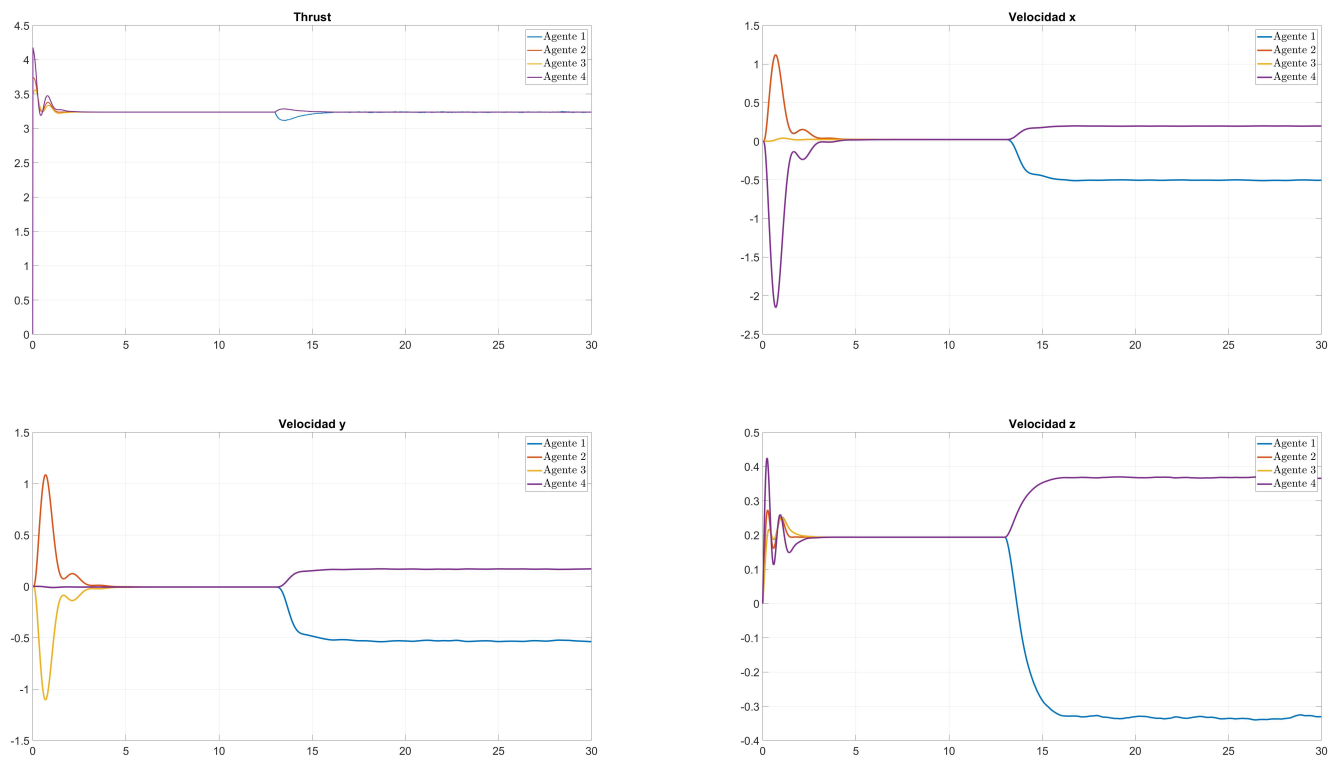


Fig. A.23: Thrust y velocidades lineales con respecto a los 3 ejes.

A.8. Pentágono formado con el esquema (4.27) con perturbaciones.

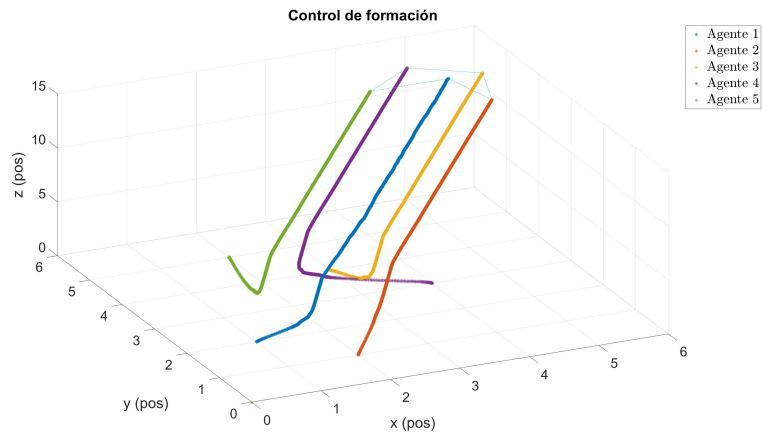


Fig. A.24: Pentágono formado con el algoritmo tipo observador (2.6) con perturbaciones.

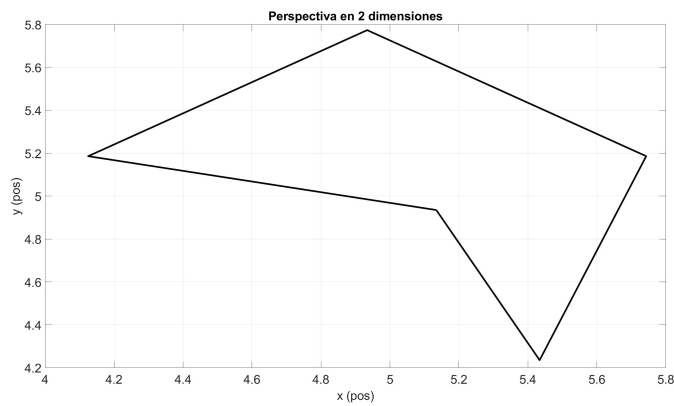


Fig. A.25: Perspectiva superior de la formación con el algoritmo (2.6) con perturbaciones.

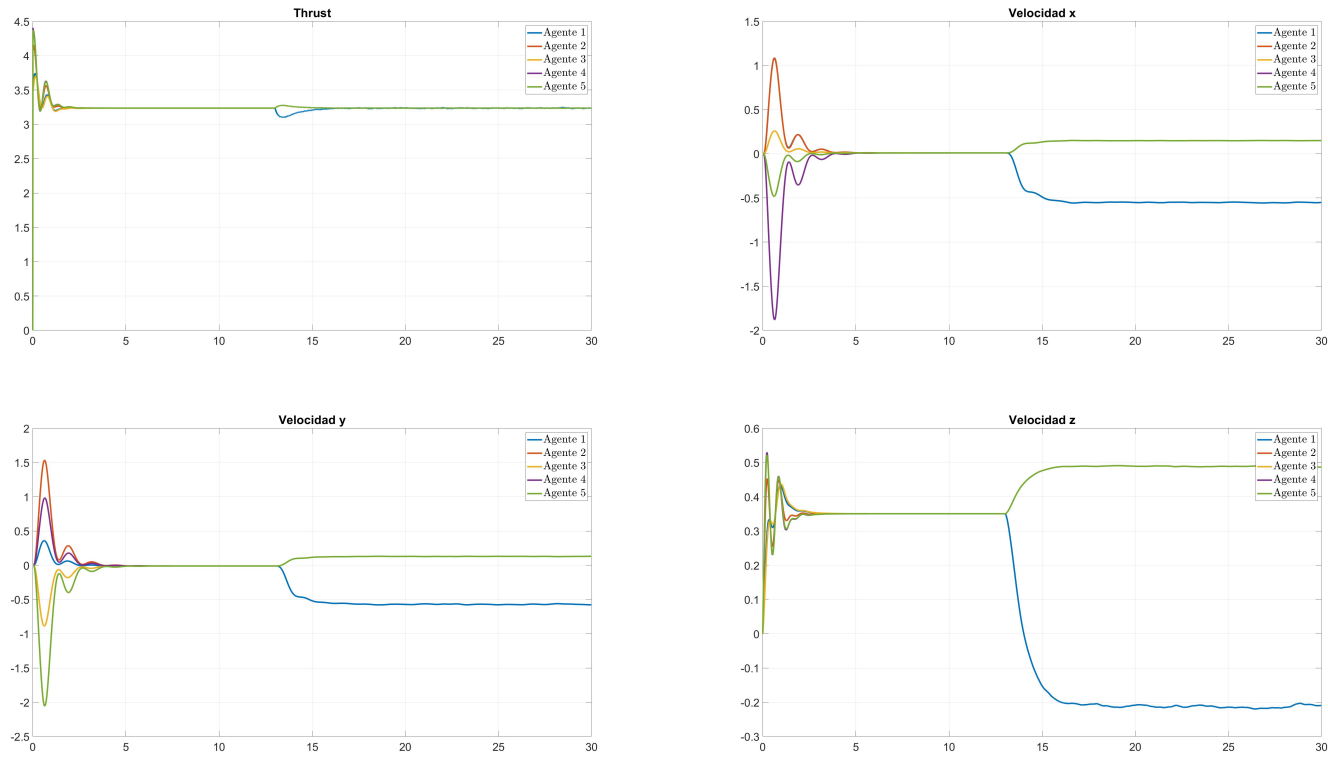


Fig. A.26: Thrust y velocidades lineales con respecto a los 3 ejes.

A.9. Cuadrado formado con el esquema (4.34) con perturbaciones.

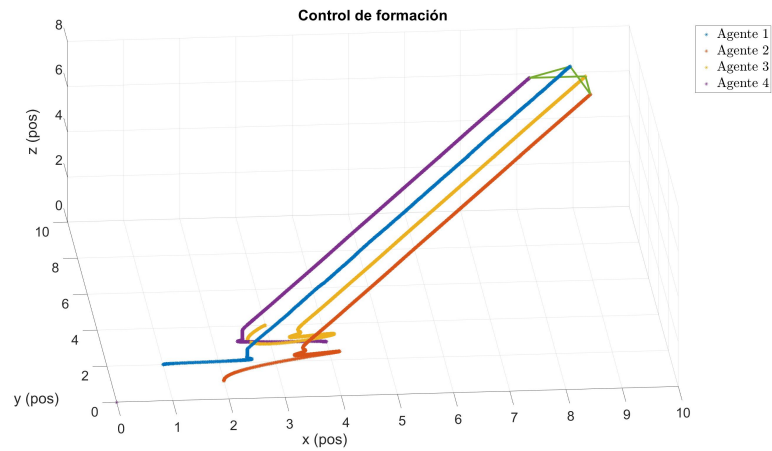


Fig. A.27: Cuadrado formado con el algoritmo tipo observador (2.7) con perturbaciones.

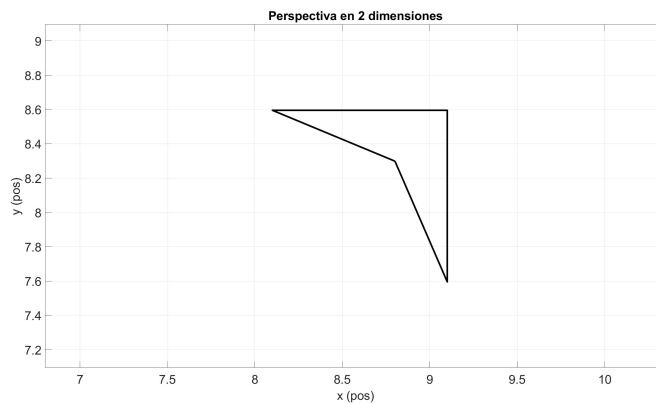


Fig. A.28: Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.

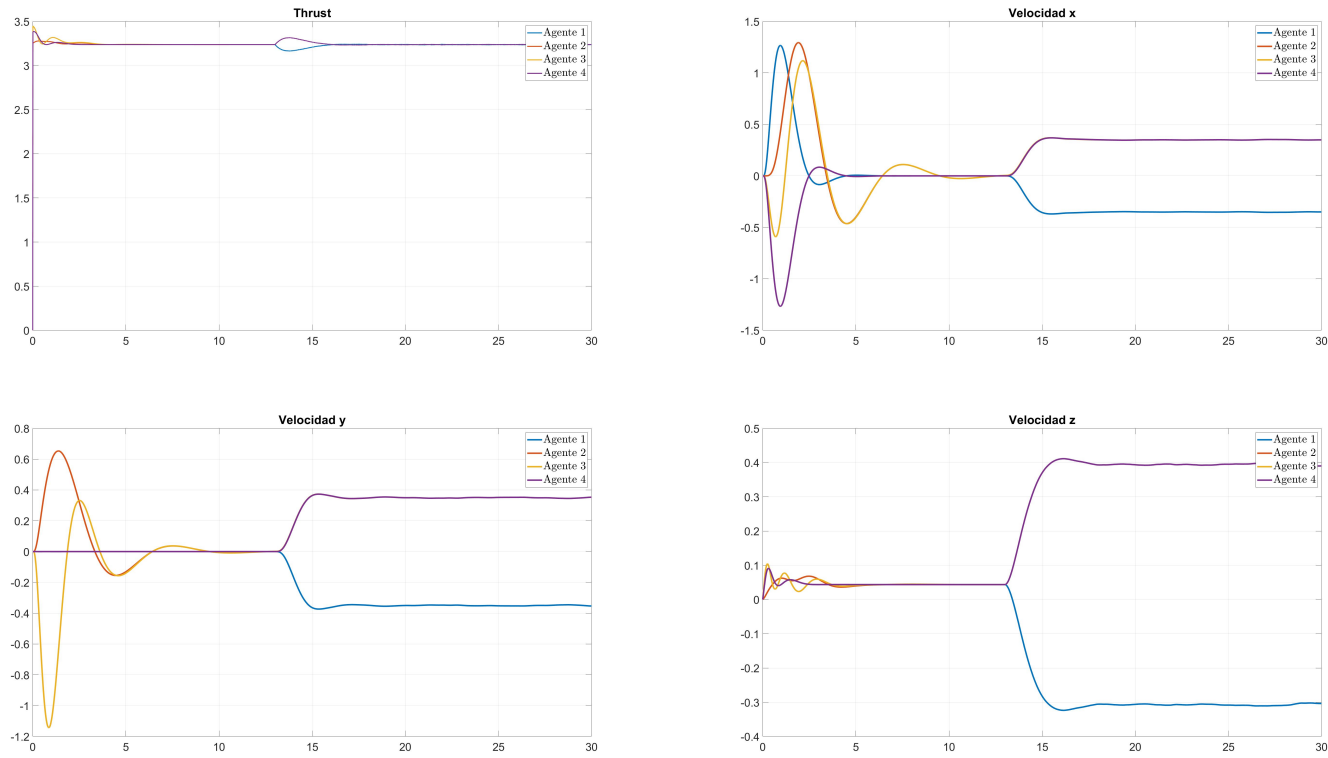


Fig. A.29: Thrust y velocidades lineales con respecto a los 3 ejes.

A.10. Pentágono formado con el esquema (4.34) con perturbaciones.

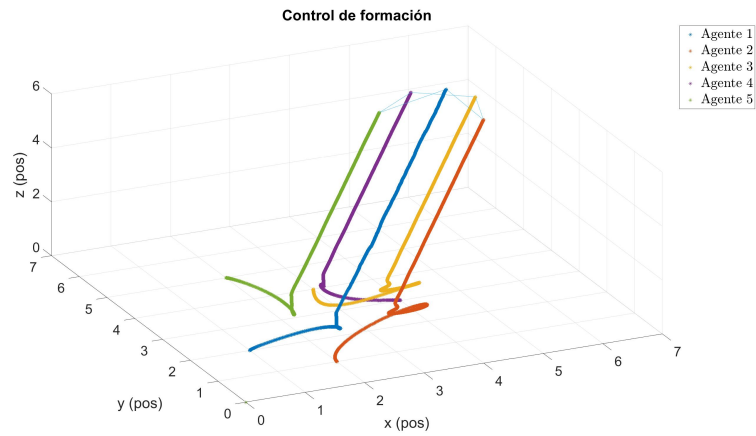


Fig. A.30: Pentágono formado con el algoritmo tipo observador (2.7) con perturbaciones.

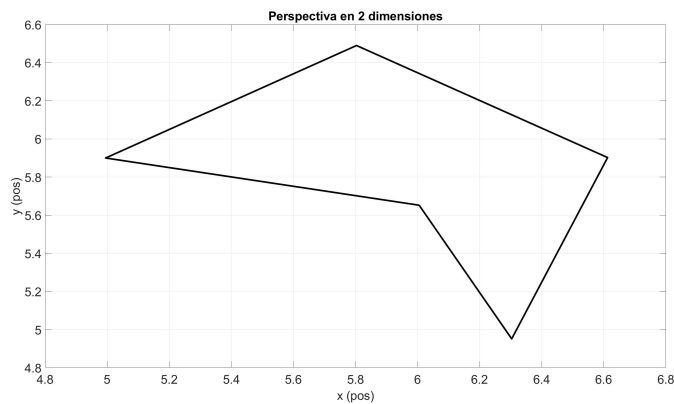


Fig. A.31: Perspectiva superior de la formación con el algoritmo (2.7) con perturbaciones.

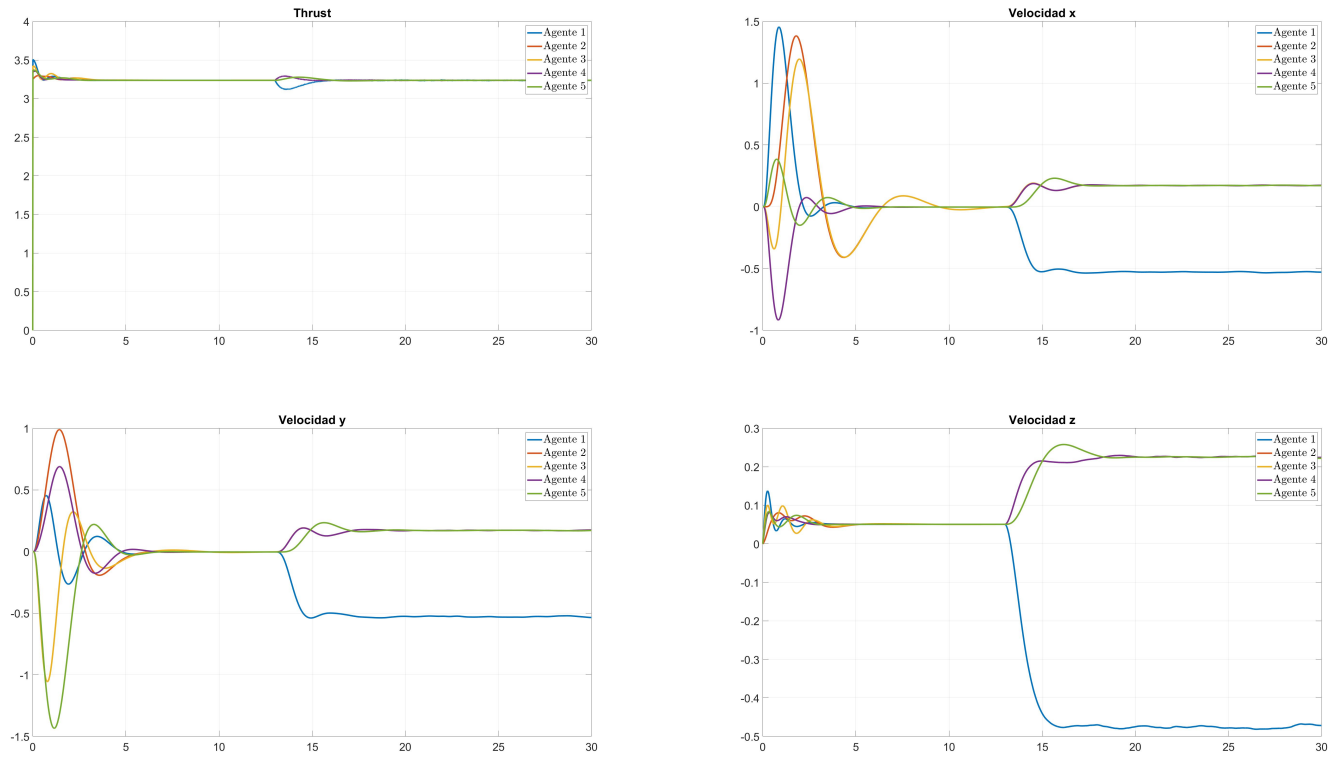


Fig. A.32: Thrust y velocidades lineales con respecto a los 3 ejes.

A.11. Cuadrado formado con el esquema (4.41) con perturbaciones.

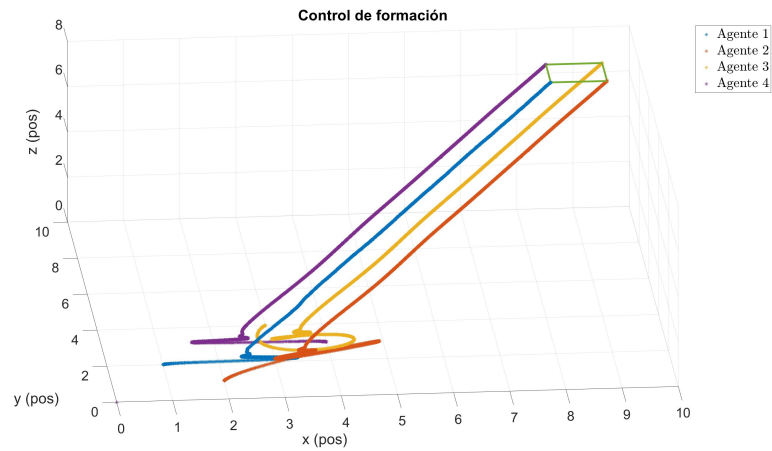


Fig. A.33: Cuadrado formado con el algoritmo tipo observador (2.8) con perturbaciones.

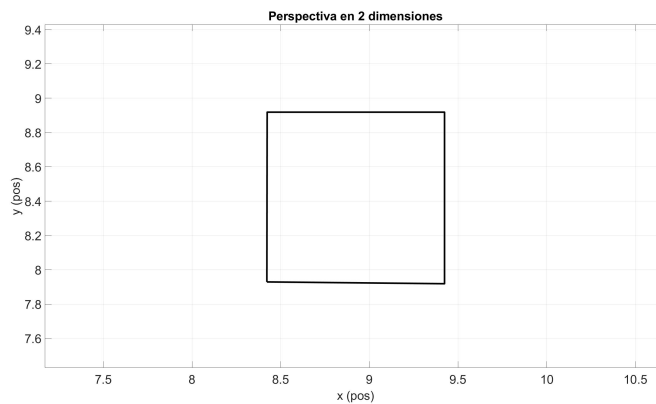


Fig. A.34: Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.

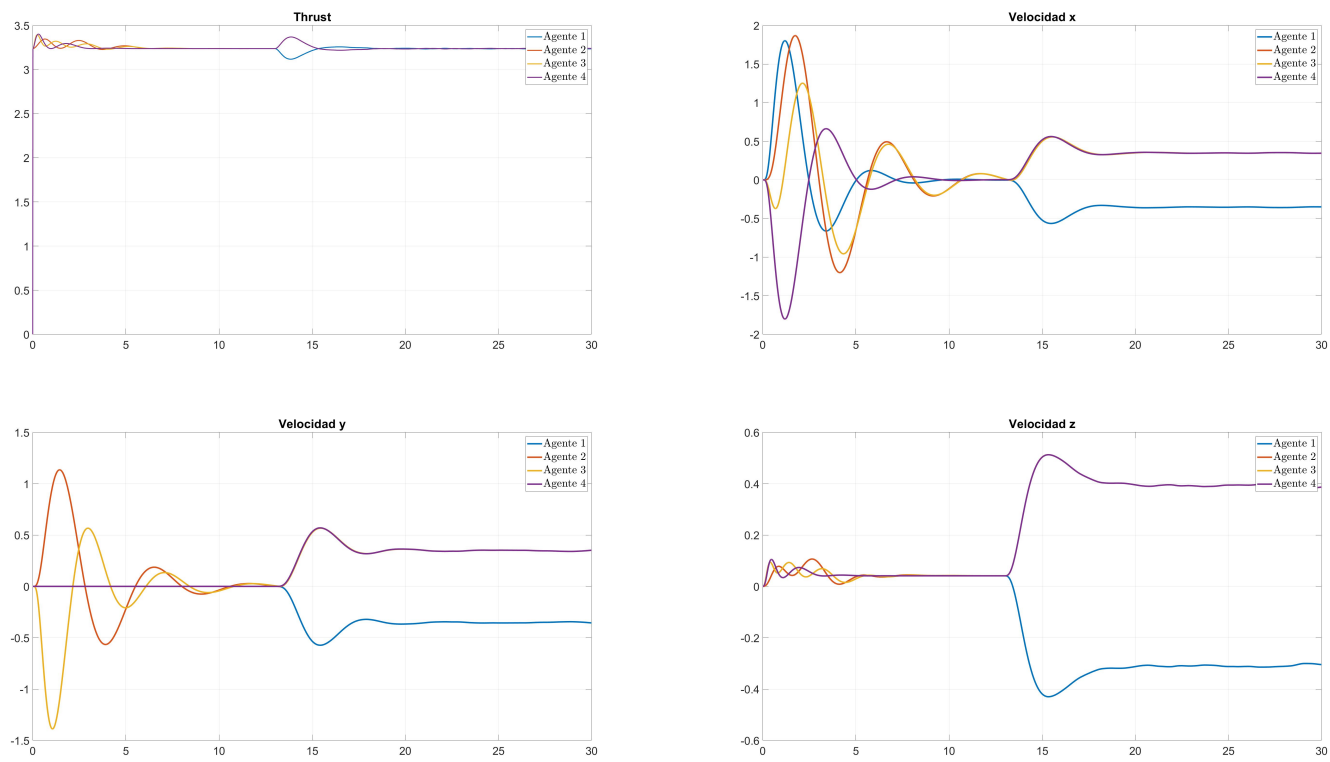


Fig. A.35: Thrust y velocidades lineales con respecto a los 3 ejes.

A.12. Pentágono formado con el esquema (4.41) con perturbaciones.

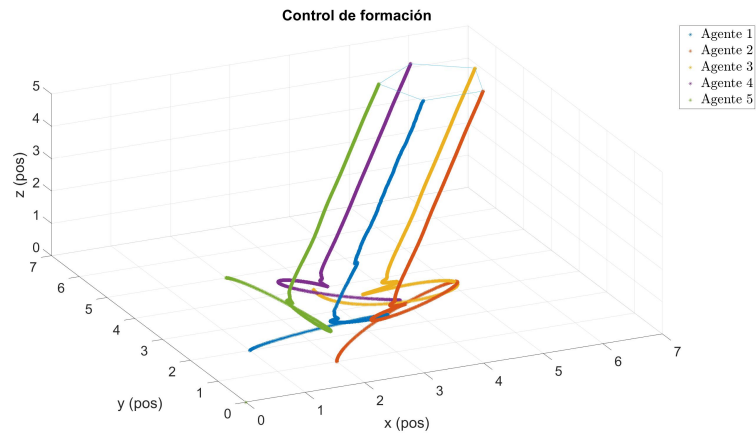


Fig. A.36: Pentágono formado con el algoritmo tipo observador (2.8) con perturbaciones.

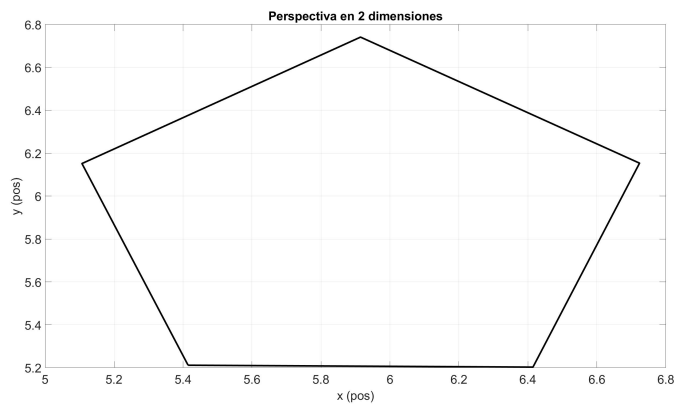


Fig. A.37: Perspectiva superior de la formación con el algoritmo (2.8) con perturbaciones.

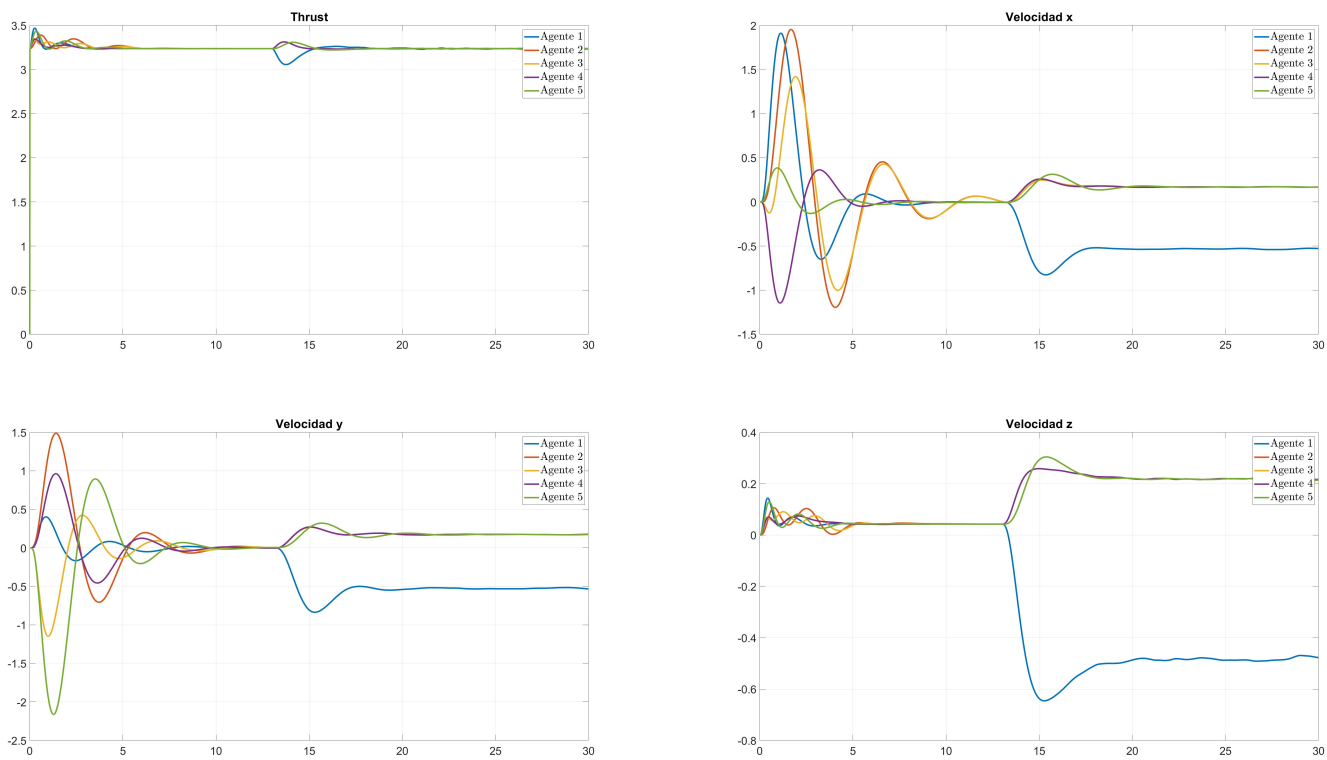


Fig. A.38: Thrust y velocidades lineales con respecto a los 3 ejes.