

Universidad Iberoamericana

Estudios con Reconocimiento de Validez Oficial por Decreto Presidencial
Del 3 de abril de 1981



“Modelado y coordinación grupal de robots móviles en ambientes ciberfísicos con retardos en la comunicación”

TESIS

Que para obtener el grado de

DOCTOR EN CIENCIAS DE LA INGENIERÍA

Presenta

HUBER GIRON NIETO

Director: Dr. Eduardo Gamaliel Hernández Martínez

Codirector: Dr. Enrique Dumas Ferreira Vázquez

Lectores: Dr. Andrés Guillermo Molano Jiménez

Dr. Mario Ramírez Neria

Dr. Jaime González Sierra

Ciudad de México, 2025

Resumen

Un sistema ciberfísico multi-robot integra agentes robóticos que comparten datos a través de redes de comunicación en tiempo real para alcanzar objetivos comunes, tomando decisiones de forma colectiva basadas en el conocimiento de su entorno. Este trabajo presenta una estrategia de control de formación para dos grupos de robots móviles ubicados en dos espacios de trabajo separados físicamente conectados mediante una red de comunicación. La técnica de control genera dos formaciones similares en cada espacio de trabajo mediante agentes virtuales que reflejan el comportamiento del robot físico correspondiente en el espacio de trabajo opuesto. Se derivan leyes de control para robots reales y virtuales de integrador simple y tipo unicycle que convergen a la formación deseada, incluso en presencia de retardos de comunicación. Las simulaciones numéricas realizadas demuestran la convergencia de la estrategia de control. Se desarrolló una plataforma ciberfísica de microrobots de bajo costo para ejecutar experimentos con robots reales. La configuración emplea una cámara como sensor de posición y orientación y se utiliza el protocolo MQTT para la comunicación con un servidor e intercambio de datos. Los resultados obtenidos en esta plataforma muestran la viabilidad del enfoque en un entorno físico real.

Declaración

Yo, Huber Giron Nieto, en calidad de autor de la tesis, “Modelado y coordinación grupal de robots móviles en ambientes ciberfísicos con retardos en la comunicación” autorizo para su utilización en la Biblioteca Francisco Xavier Clavigero en formato electrónico o impreso y sin fines de lucro, cuyo titular es la Universidad Iberoamericana Ciudad de México, por un período ilimitado a contar desde la fecha de la publicación de la presente, con expresa renuncia a los derechos derivados de su explotación en dichos medios y durante el mencionado período.

Índice general

1	Introducción	1
1.1	Planteamiento del problema	2
1.2	Objetivos	6
1.2.1	Objetivo general	6
1.2.2	Objetivos específicos	6
1.3	Originalidad y justificación	6
1.3.1	Originalidad	6
1.3.2	Justificación	7
1.4	Contribución	9
1.5	Trabajo previo en la IBERO	10
2	Antecedentes	13
2.1	Industria 4.0 y robótica	13
2.2	Robots móviles colaborativos en la industria 4.0	15
2.3	Control de formación y avance en formación	16
2.3.1	Temas recientes en el modelado y control de formación	20
3	Estado del arte	23
3.1	Formaciones ciberfísicas	23
3.2	Problemas actuales de formaciones ciberfísicas	24
3.2.1	Control de robots en sistemas ciberfísicos	25
3.2.2	Sistemas lineales con retardos	27
4	Estrategia de control	30
4.1	Formulación del problema	30
4.2	Estrategia de control	31
4.2.1	Control sin retardos en las comunicaciones	34

4.2.2	Control con retardos en las comunicaciones	36
4.3	Simulaciones numéricas	38
4.3.1	Control sin retardos en las comunicaciones	39
4.3.2	Control con retardos en las comunicaciones	49
5	Trabajo experimental	57
5.1	Robots tipo unicycle	59
5.2	Broker MQTT	62
5.3	Detección de posición con ArUco y visión por computadora . .	62
5.4	Resultados experimentales	64
6	Conclusiones	106
6.1	Contribuciones	106
6.2	Conclusión	107
6.3	Trabajo a futuro	108
6.4	Publicaciones y distinciones	109
	Bibliografía	110
A	Listado de empresas fabricantes de robots AGV	126
B	Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments	130
C	Cyber-Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach	131

Índice de figuras

1.1	Proyectos de robots móviles en la IBERO	11
4.1	Comunicación entre los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$	31
4.2	Ejemplo para $(n = 4, m = 2)$ de la relación entre el grafo de formación original $\mathcal{G}$ (arriba), los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$ y el grafo combinado $\hat{\mathcal{G}}$ (abajo).	32
4.3	Topología de la primera formación: formación en línea horizontal de 2 robots.	39
4.4	Topología de la segunda formación: formación en línea horizontal de 4 robots.	39
4.5	Topología de la tercera formación: formación en rombo de 4 robots.	40
4.6	Simulación numérica para formación en línea horizontal de 2 robots sin retardos.	41
4.7	Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en línea horizontal de 2 robots.	42
4.8	Simulación numérica de formación en línea horizontal de 4 robots sin retardos.	44
4.9	Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en línea horizontal de 4 robots.	45
4.10	Simulación numérica de formación en rombo de 4 robots sin retardos.	47
4.11	Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en rombo de 4 robots.	48
4.12	Simulación numérica de formación en línea horizontal de 2 robots con retardos.	50
4.13	Normas L_1 y L_2 con diferentes retardos en la simulación de formación en línea horizontal de 2 robots.	51

4.14	Simulación numérica de formación en línea horizontal de 4 robots con retardos.	53
4.15	Normas L_1 y L_2 con diferentes retardos en la simulación de formación en línea horizontal de 4 robots.	54
4.16	Simulación numérica de la formación en rombo de cuatro robots con un retardo de comunicación de $\tau = 5$ s.	55
4.17	Normas L_1 y L_2 con diferentes retardos en la simulación de formación en rombo de 4 robots.	56
5.1	Diagrama de los espacios de trabajo para los experimentos ciberfísicos.	58
5.2	Ejemplo de implementación del control en la plataforma experimental para la formación en rombo de cuatro robots.	58
5.3	Diagrama del sistema electrónico de la plataforma experimental de robots.	59
5.4	Modelo cinemático de un robot tipo unicycle.	61
5.5	Latencia de los brokers MQTT (media y desviación estándar).	63
5.6	Latencia y estabilidad del sistema de visión computacional.	64
5.7	V , W_R y W_L en el experimento de formación de 2 robots en línea horizontal.	66
5.8	Resultados experimentales para la formación de 2 robots en línea horizontal.	67
5.9	Normas L_1 y L_2 en el experimento de 2 robots en línea horizontal.	69
5.10	Error cuadrático instantáneo en el experimento de 2 robots en línea horizontal.	69
5.11	V , W_R y W_L en el experimento de formación de 4 robots en línea horizontal.	71
5.12	Resultados experimentales de la formación de 4 robots en línea horizontal.	72
5.13	Normas L_1 y L_2 en el experimento de formación de 4 robots en línea horizontal.	74
5.14	Error cuadrático instantáneo en el experimento de formación de 4 robots en línea horizontal.	75
5.15	Resultados experimentales para la formación en rombo de 4 robots en dos espacios.	76
5.16	V , W_R y W_L en el experimento de formación en rombo de 4 robots.	77
5.17	Normas L_1 y L_2 en el experimento en rombo de 4 robots.	79

5.18	Error cuadrático instantáneo en el experimento en rombo de 4 robots.	79
5.19	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2). .	82
5.20	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2). .	83
5.21	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2). .	84
5.22	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2). .	85
5.23	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).	86
5.24	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).	87
5.25	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).	88
5.26	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).	89
5.27	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2). .	90
5.28	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2). .	91
5.29	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2). .	92
5.30	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2). .	93
5.31	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).	94
5.32	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).	95

5.33	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).	96
5.34	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).	97
5.35	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (1/2).	98
5.36	Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).	99
5.37	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (1/2).	100
5.38	Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).	101
5.39	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (1/2).	102
5.40	Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).	103
5.41	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (1/2).	104
5.42	Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).	105

Capítulo 1

Introducción

El control y la coordinación de múltiples robots móviles [1] es un área de investigación importante para los avances tecnológicos actuales en el uso de robots industriales y de servicio [2]. En los últimos años, ha aumentado el uso de agentes móviles en logística [3], salud [4], tareas en campo [5] y defensa militar [6, 7].

La formulación del problema en robótica móvil se aborda inicialmente con respecto al modelo cinemático del tipo de robot móvil y la configuración mecánica. Algunas contribuciones se han centrado en robots con tracción diferencial o tipo uniciclo [8], robots omnidireccionales [9] y configuraciones tipo vehículo. Además del control individual de robots móviles, es posible coordinar un grupo de robots para tareas de logística y transporte [10, 11]. Los problemas fundamentales en la coordinación de robots móviles son el control de formación [12], el seguimiento de formación o control de marcha [13] y la evasión de colisiones entre robots [14, 15, 16].

Los esquemas de formación ciberfísica son un enfoque reciente para la comunidad de control multi-robot. Estos esquemas incluyen estrategias para coordinar robots móviles que se desplazan en diferentes espacios de trabajo físicos, compartiendo información a través de Internet y la nube [17]. Este nuevo esquema de formación es posible gracias a dispositivos de comunicación inalámbrica con alto ancho de banda [18] y arquitecturas de cómputo embebido robustas [19], lo que permite leyes de control de formación que pueden ejecutarse a bordo de los propios robots o en la nube. En estos sistemas ciberfísicos, los robots pueden lograr un movimiento coordinado a través de

espacios de trabajo remotos.

Los avances recientes en esquemas de formación ciberfísica [20], proponen una estrategia de programación jerárquica para robots móviles en entornos multi-almacén, integrando sistemas ciberfísicos industriales para optimizar operaciones logísticas. Algunos desarrollos se enfocan en sistemas de control ciberfísico diseñados con robots logísticos autónomos, centrado en la toma de decisiones en tiempo real e integración con plataformas de automatización industrial [21]. Otro avance reciente, introduce un sistema ciberfísico para la planificación óptima de trayectorias de robots móviles, considerando explícitamente el desgaste de los motores eléctricos para mejorar el rendimiento a largo plazo y la eficiencia energética [22].

Los retardos de comunicación afectan a los espacios de trabajo de sistemas ciberfísicos y pueden influir significativamente en su rendimiento si no se abordan adecuadamente. Los métodos basados en predictores se han utilizado extensamente para estimar estados futuros y compensar dichos retardos [23]. El análisis de estabilidad ha empleado funcionales de Lyapunov-Razumikhin y Lyapunov-Krasovskii, conduciendo a condiciones de desigualdades matriciales lineales que pueden resolverse eficientemente [24, 25], capaces de manejar múltiples retardos variables y obtener condiciones más estrictas [26]. Otros enfoques incluyen control basado en pasividad [27], back-stepping y control por modo deslizante [28], entre otros.

Los principales desafíos en este esquema a través de diferentes espacios de trabajo incluyen el intercambio de información a través de Internet [29, 30], nuevos retos para esquemas de control [31, 32, 33], efectos de retardos en comunicaciones por Internet [34, 35], el problema de pérdidas de comunicación entre espacios de trabajo [36, 37] y el impacto de la conectividad intermitente en enlaces por Internet [38, 39, 40].

1.1 Planteamiento del problema

Una vez abordada la importancia de la investigación de robots móviles conectados a internet y sus aplicaciones en la IBERO, identificamos la pertinencia de la aplicación de este nuevo conocimiento para la solución de problemas que permitan la mejora en la calidad y eficiencia del sector industrial y de servicios.

Algunos problemas, necesidades y retos tecnológicos referentes a la generación

y aplicación de los robots móviles en ambientes ciberfísicos son los siguientes:

- La mayoría de los procesos productivos requieren el traslado de objetos, lo cual representa un riesgo para el personal que realiza esta actividad.
- Desde la contingencia sanitaria del 2020 y ante el distanciamiento social experimentando durante esta, existe un impulso importante para el despliegue de robots móviles en servicios en el contexto rural y urbano, para el transporte de insumos y de diversos servicios de logística en sitio.
- Debido a la problemática de salud generada por el coronavirus SARS-CoV-2, el sector industrial y de servicios notó la importancia de reducir el contacto físico entre trabajadores, lo que impulsó la automatización de trabajos que puedan ser desempeñados por un robot.
- En la logística de una industria, normalmente se asigna personal para el transporte de material interno, es poco común el uso de sistemas automatizados y el costo de personal para transporte interno es bastante significativo con respecto al porcentaje de la mano de obra productiva.
- La organización adecuada del sector de transporte interno, representa no solo una de las áreas más promisorias para la reducción de costos de una empresa, evitar el congestionamiento de materiales, falta de abastecimiento y otras condiciones que reducen la eficiencia de la producción.
- El rendimiento de instalaciones industriales aumenta en razón del aumento de la eficiencia de las operaciones de transporte interno, ya que se reducen las horas pasivas y de flujo continuo de materiales, lo que significa una mejora del ritmo de trabajo.
- El nuevo reto tecnológico de la industria 4.0 demanda la innovación y el diseño de estrategias de control de robots móviles para su uso en los sistemas de transporte logístico de materiales de la industria.
- Los robots móviles actuales no se encuentran integrados eficientemente a los sistemas ciberfísicos de la industria 4.0 y no se encuentran preparados para interactuar de forma estrecha con máquinas y humanos.
- Existe en México un escaso volumen de investigación referente al diseño de estrategias de control basado en distancias con robots móviles en entornos ciberfísicos.

- México no cuenta con investigación enfocada en el desarrollo de robots móviles autónomos que trabajen en ambientes ciberfísicos con un estudio en los retardos que se puedan implementar en el sector productivo con el fin de modernizar y eficientizar el proceso hacia la industria 4.0

Con respecto a la problemática anterior se plantea un proyecto de investigación cuya relevancia se encuentra en la solución de problemáticas científicas y sociales específicas:

Problemática Científica: Se ha detectado en la literatura un bajo estudio del modelado de robots móviles y la coordinación y/o supervisión en ambientes ciberfísicos, cuya aplicación permita a futuro el desarrollo de plataformas de robots móviles implementables en sistemas industriales. La implementación de los conceptos teóricos y conceptuales al plano real a través de la experimentación e implementación, permitiría validar la teoría con experimentos, y perfeccionar la tecnología para que pueda ser utilizada.

Este proyecto plantea estudiar estos esquemas de coordinación para ambientes de la industria 4.0 con énfasis en los siguientes aspectos.

- Esquemas de formación centralizada donde la retroalimentación de señales para el control provengan de cámaras externas al robot. Con especial interés en mediciones de distancia, posición y ángulos de orientación, que pueden ser captadas por las cámaras mediante un sistema de visión por computadora.
- Esquemas de formación considerando posibles retardos a la entrada o salidas del control. Esto significa que se puede considerar un retardo entre la lectura de las cámaras al momento que es requerida por el control, e incluso retardos cuando un control en la nube envía señales al controlador del robot.
- Esquemas modulares e híbridos de control de coordinación de movimiento, donde el controlador local de cada robot resuelve los problemas posturales y de reacción inmediata como la evasión de obstáculos, mientras otro control general ubicado en la nube, coordina y supervisa la tarea de varios robots.
- Considerar en los esquemas de control que el espacio de trabajo los agentes pueden estar ubicados en distintas localizaciones geográficas pero ligados con comunicación por internet.

Reto de desarrollo tecnológico con responsabilidad social: Una nueva

tecnología emergente, no puede salir al mercado, sin antes haber estudiado los problemas a nivel investigación y en condiciones controladas de laboratorio. En México, los sistemas robóticos no está siendo desarrollados, y la tecnología existente se importa a precios altos. Es necesario implementar una plataforma propia que tenga componentes de altas prestaciones y más accesibles en el mercado. Para lograr un producto competitivo y con prestación adecuadas que cumplan incluso con aspectos de seguridad para las personas, como primer paso, es necesario integrar una plataforma de desarrollo experimental para robots ciberfísicos accesible y con las características necesarias para probar las leyes de control y condiciones especiales de un sistema de formación conectado a internet.

Las características generales de una plataforma de experimentación ciberfísica que cumpla con lo anterior planteado para este proyecto se enlistan a continuación:

- Desarrollo de 4 robots móviles de bajo costo, para permitir experimentar diferentes formaciones.
- Sistema de retroalimentación mediante el uso de camara web USB para computadora.
- Utilizar una computadora de escritorio de rendimiento promedio y lenguajes de programación educativos en el ambito de control y robótica como Matlab o Python.
- Comunicación inalámbrica vía protocolos de radiofrecuencia (RF), Bluetooth, Zigbee o WiFi entre robots y con estación de control con alcance máximo 10m.
- Comunicación de los robots a un servicio de nube para la supervisión del grupo de agentes móviles.
- Diseño mecánico y electrónico para el correcto funcionamiento de la dinámica del tipo de robot móvil seleccionado. Es indispensable, que además del cumplimiento de la parte estructural, también tenga buena presentación, con carcasa montada y con el etiquetado adecuado para su operación, entre otros aspectos relevantes de producto.

1.2 Objetivos

1.2.1 Objetivo general

Diseñar estrategias descentralizadas de coordinación grupal de movimiento de robots móviles multi-agentes considerando su operación y supervisión en sistemas ciberfísicos con retardos en la comunicación y un enfoque de agentes virtuales, así como su experimentación en una plataforma construida de robots móviles de bajo costo.

1.2.2 Objetivos específicos

- Actualizar la revisión del estado del arte sobre la coordinación de robots móviles y las recientes tecnologías para la implementación en robots móviles en ambientes ciberfísicos.
- Diseñar, analizar y simular estrategias de coordinación de robots móviles modelando la coordinación grupal en sistemas ciberfísicos con y sin retardos en las comunicaciones con un enfoque de agentes virtuales.
- Diseñar, analizar y experimentar en laboratorio estrategias de coordinación de robots móviles utilizando las plataformas de bajo costo de robots móviles disponibles en la IBERO.
- Diseñar e integrar tecnología de hardware, software y de comunicaciones para el desarrollo de al menos cuatro robots móviles de bajo costo para el desarrollo de experimentos.
- Aplicar y ajustar los esquemas de coordinación de movimiento a los robots móviles desarrollados.
- Extender los esquemas de coordinación en los robots móviles para un caso de aplicación multi-agente, en un ambiente ciberfísico con retardos.
- Publicación de resultados en congreso de investigación y revistas científicas indizadas, así como en reportes técnicos y tesis.

1.3 Originalidad y justificación

1.3.1 Originalidad

La originalidad del trabajo de investigación se centra en los siguientes puntos:

- Las investigaciones en el área de robótica colaborativa, normalmente se enfocan en el diseño de nuevos tipos de controladores para optimizar el seguimiento de trayectorias de un conjunto de robots coordinados. Este trabajo de investigación se enfocará en el control de robots coordinados mediante un sistema descentralizado en ambientes ciberfísicos.
- Este proyecto aportará en el desarrollo de estrategias que resuelvan los problemas básicos de coordinación de movimiento multi-robot, que son formación y avance en formación basado en posición, considerando los efectos de retardos del sistema cuando los robots operan en ambientes ciberfísicos, donde el control y monitoreo se encuentran en un espacio en la nube. Esto es oportuno dados los recientes intereses de la comunidad científica, así como de las nuevas tecnologías para robots móviles industriales y de manufactura digital de la Industria 4.0 que están surgiendo en el mercado.
- Se aportará a la comunidad científica, el desarrollo de la coordinación de robots móviles y sistemas ciberfísicos. Se evaluará y analizará cada modelo de control con y sin retardos por medio de simulaciones y de forma experimental para la validación.

1.3.2 Justificación

Este proyecto se justifica como investigación doctoral por los siguientes rubros:

- La naturaleza de este proyecto conlleva una aportación en el desarrollo de plataformas de experimentación de robots móviles con caracterización de sus parámetros físicos y trabajo práctico para aplicaciones reales.
- Este proyecto se asocia a las líneas de investigación de sistemas dinámicos y control de la IBERO, logrando así un enriquecimiento en sus laboratorios, infraestructura y generación del conocimiento de la comunidad universitaria.
- El desarrollo del proyecto se puede vincular con aplicaciones industriales y la generación de negocios, gracias a su futura escalabilidad en robots industriales.
- El problema coordinación en sistemas ciberfísicos, como un enfoque integral de coordinación multi-robot y los fenómenos dinámicos que se relacionan, no suele ser abordada en la literatura. Por lo tanto, el

enfoque de agentes virtuales del proyecto constituye una nueva definición del problema de coordinación planteando nuevos retos a la comunidad científica en aspectos teóricos y prácticos.

- El proyecto es complejo, dado que requiere de conocimientos especializados de sistemas dinámicos, robótica móvil, teoría de redes, sistemas de control lineal, así como de habilidades en el manejo de tecnología de manufactura, electrónica, cómputo, redes de telecomunicación, internet de las cosas, sensores, entre muchos otros. Estos conocimientos mezclados, además de la capacidad de identificar sistemas de nivel adecuado para la industria requiere de un perfil de doctorado.

Por otro lado, se resalta que este proyecto no es aislado del trabajo previo realizado en la IBERO, sino que se vincula con lo siguiente:

- Este proyecto se ligó al proyecto del grupo de investigación conformado por el Instituto de Investigación Aplicada y Tecnología (InIAT) que obtuvieron financiamiento de la IBERO con el proyecto *“Hacia nuevos paradigmas en la coordinación de movimiento de grupos de robots móviles”*
- El proyecto extenderá la investigación entre la IBERO y la Universidad Católica del Uruguay (UCU) en el tema de formación basada en distancias, favoreciendo la vinculación internacional del sistema AUSJAL y motivando la vinculación de estudiantes y profesores.
- Este proyecto está relacionado con las aplicaciones de tecnología que está impulsando el Instituto de Investigación Aplicada y Tecnologías (InIAT) de la IBERO. Con ello se buscarán posibles usuarios de la plataforma desarrollada, comenzando una línea de desarrollo tecnológico y aplicación del conocimiento para la Universidad.
- El proyecto vinculará la Universidad Iberoamericana ciudad de México con la Universidad Iberoamericana Puebla, lo que permitirá un trabajo colaborativo entre ambas universidades jesuitas de México, como un precedente en el desarrollo de proyectos de investigación y aplicación tecnológica.

Finalmente, se resalta que mi experiencia en el área de estudio es afín al desarrollo de la tesis doctoral por los siguientes aspectos:

- Formación en el área de electrónica, comunicaciones, mecatrónica, sistemas computacionales, prototipado rápido, innovación y desarrollo

tecnológico.

- Experiencia como investigador y coordinador del área de desarrollo de proyectos mecatrónicos del Instituto de Diseño e Innovación Tecnológica de la universidad Iberoamericana Puebla por cuatro años.
- Experiencia en el desarrollo de proyectos de mecatrónica con la incorporación de alumnos de licenciatura y maestría, especializado en sistemas embebidos, internet de la cosas, automatización, desarrollo de productos, diseño digital y prototipado rápido.
- Participación en proyectos de innovación y desarrollo de tecnología en la industria con recursos federales, el anterior CONACYT(Consejo Nacional de Ciencia y Tecnología) e INADEM (Instituto Nacional del Emprendedor).
- Experiencia en la compra e instalación de robots móviles de gama industrial en empresas de la región de Puebla, que abona al conocimiento de los requerimientos y normas que deben cumplir este tipo de desarrollos.

1.4 Contribución

Las principales contribuciones de esta tesis son las siguientes.

- El diseño e implementación de una estrategia de control para una formación ciberfísica de sistemas de robots móviles que se desplazan en dos espacios de trabajo.
- El enfoque de control que se base en el uso de agentes virtuales que convergen junto con los robots físicos que se mueven en el espacio de trabajo opuesto. La convergencia se analiza para el caso general de n robots. Bajo este esquema de control, los robots se comportan como si se desplazaran en un único espacio de trabajo.
- La formulación del problema se establece para robots de integrador simple en 2D. Sin embargo, tiene un alcance más amplio, ya que se sabe que diferentes robots móviles no holónomos pueden reducirse a esta representación simple mediante una transformación de linealización entrada-salida adecuada.
- El enfoque de control se desarrolla para robots de integrador simple en 2D, pero se extiende a robots tipo unicycle con dos ruedas, y se valida

mediante simulaciones numéricas.

- Finalmente, se construye una plataforma experimental de bajo costo. Se muestran experimentos con hasta cuatro robots físicos, distribuidos en dos espacios físicos diferentes conectados a través de internet mediante el protocolo MQTT; se prueban distintas topologías de formación, tasas de muestreo y brokers MQTT.

1.5 Trabajo previo en la IBERO

La IBERO ha realizado investigación en robótica móvil desde el año 2012, ha generado el estudio de estrategias de formación, marcha y evasión de colisiones basadas en posiciones relativas con experiencias de trabajo experimental como se ve en la Figura 1.1. Esta investigación ha contribuido al estudio de coordinación multi-robot tanto a nivel teórico como de desarrollo tecnológico. Esto se ha logrado a través de una serie de proyectos financiados por la Dirección de Investigación y Posgrado de la IBERO México a través de distintas convocatorias y con presupuesto del anterior CONACyT.

Se han realizado diferentes publicaciones representativas del trabajo teórico enfocado en la formación descentralizada de robots móviles en tiempo continuo [41], formación en tiempo discreto [42], para control de marcha [43] y marcha con velocidades aproximadas [44]. Se han diseñado arquitecturas híbridas para la emulación de tareas de transporte de objetos en sistemas automatizados de manufactura utilizando autómatas de estado finito [45] y redes de Petri, así como planeadores de tareas de traslado [46, 10, 47]. También se ha investigado la formación de robots y evasión de colisiones utilizando distancias y orientación [14] y la adición de ángulos internos en el patrón de formación [48, 12]. Para evitar proyectos aislados, el grupo de investigadores en robótica, ha enfocado sus esfuerzos en los problemas fundamentales de coordinación de robots móviles que son formación, evasión de obstáculos y avance en formación o marcha, generando nuevas estrategias de control y su validación en plataformas experimentales mediante robots móviles de aplicación real, como en el proyecto con fondos de la Dirección de Investigación (DINVP) del periodo 2012-2014, donde cinco robots móviles de tipo unicycle de la marca Irobot y una cámara monocromática instalada en el techo del laboratorio, que fungía como sensor de localización en 2D, en CONACyT 2014-2015, los robots de carga Atlas y Apolo de grado industrial 2017-2018 y la aprobación



Figura 1.1: Proyectos de robots móviles en la IBERO

del proyecto de drone y drone-puerto para entrega de medicamentos 2020-2021, mostrados en la Figura 1.1. Al incorporar drones, fue necesaria la instalación de un sistema de captación de movimiento en 3D de la marca VICON. Actualmente se trabaja en el vuelo autónomo de drones con la estrategia de control diseñada [49]. Este proyecto de tesis forma parte del proyecto DINVP 2021-2023 donde se estudian nuevas estrategias de sistemas de robots en contextos actuales como sistemas ciberfísicos entre otros.

La plataforma experimental fue premiada en el “*International Symposium on Experimental Robotics*” en el año 2018. Con la infraestructura desarrollada se ha beneficiado a cursos de posgrado, pregrado y la creación de un moderno Laboratorio de Análisis de Movimiento en alianza con el Departamento de Estudios de Ingeniería para la Innovación y el InIAT. El grupo de investigación de la IBERO ha sumado a investigadores de otras instituciones como de la Universidad Católica del Uruguay (UCU), el Instituto Politécnico Nacional (IPN), el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), el Tecnológico de La Laguna (TLa), La División Científica de la Policía federal y recientemente a la Universidad Veracruzana (UV), la Universidad Autónoma de Baja California (UABC), Universidad Politécnica de Pachuca y la Universidad EIA en Colombia (EIA). Las plataformas de experimentación han sido compartidas con la UCU, TLa, la UABC y la EIA, impulsando la colaboración en robótica a nivel docencia e investigación.

El presente proyecto es una continuación de todo este esfuerzo de desarrollar tecnología de robótica móvil en un paso importante, que es la de desarrollar estrategias de control para la coordinación grupal de robots móviles en ambientes ciberfísicos tomando en cuenta retardos de la comunicación y el nuevo enfoque de agentes virtuales. También pretender crear una alianza de trabajo colaborativo con la IBERO campus Puebla y juntar las experiencias de nuestras universidades del sistema SUJ (Sistema Universitario Jesuita) en el desarrollo de tecnología.

Capítulo 2

Antecedentes

Este capítulo se abordan la aplicación de los robots móviles en la industria 4.0 y los modelos de control para la formación, avance y coordinación de robots móviles.

2.1 Industria 4.0 y robótica

Actualmente una de las tendencias en la industria es la automatización, robótica y la implementación de sistemas ciberfísicos para el control y monitoreo de los procesos. Un elemento que cobra más fuerza en los últimos años es la robótica móvil industrial, la cual se enfoca en vehículos autónomos capaces de realizar tareas complejas de forma cooperativa con otros elementos de automatización, robots y usuarios. El desarrollo de nuevas tecnologías para la automatización y control industrial se enfocan en optimizar procesos productivos industriales.

La tecnología industrial actualmente se encuentra enfocada en la automatización y producción inteligente, lo cual es enmarcado en la llamada cuarta revolución industrial o Industria 4.0. El término industria 4.0 se refiere a un modelo de organización y de control de la cadena de valor a través del ciclo de vida del producto y a lo largo de los sistemas de fabricación apoyado por las tecnologías de información y los sistemas inteligentes. Los pilares clave de la industria 4.0 principalmente se enfocan en IIoT(Internet de las cosas-Industrial), Big Data, Cloud Computing (computo en la nube), Robótica y comunicación M2M (máquina a máquina) [50].

El IIoT aporta la sensorización masiva y la conectividad que vinculan activos físicos con sus réplicas digitales. A escala de planta, el paradigma se amplía mediante enlaces M2M de muy baja latencia (hoy impulsados por redes 5G y 6G), que permiten a máquinas y células de producción negociar tareas y balancear cargas de forma autónoma [51, 52] .

El big data se utiliza para analizar el gran volumen de datos industriales, mediante modelos de aprendizaje profundo sobre flujos históricos y en línea que alimentan gemelos digitales y algoritmos de mantenimiento predictivo, permitiendo reducir tiempos de inactividad y mejoras económicas [53]. La computación en la nube, complementada con entornos edge y fog, ofrece escalabilidad elástica para almacenar, procesar y orquestar datos sin inversiones locales. Algunos modelos de computo en la nube con IIoT permiten desplegar servicios avanzados con inteligencia artificial, gestión de activos y simulación con tiempos de puesta en marcha menores frente a arquitecturas tradicionales [54].

La robótica es un pilar importante para la transición a la industria 4.0. ya que esta permitirá que los robots se integren en tareas en las que antes no tenían participación ni coordinación entre las máquinas. La gran mayoría de robots en la industria participan en actividades aisladas, lo que limitaba la cantidad de tareas en las que pueden participar. Gracias a los nuevos sistemas inteligentes y ciberfísicos, esto permitirá trabajar directamente con personas y otros robots sin requerir de elementos de seguridad que aislen el trabajo de los robots. La robótica sustenta la automatización física: los manipuladores industriales, la tendencia de adopción de cobots (robots colaborativos) y robots de servicio capaces de colaborar con operarios, desplazarse con seguridad y reprogramarse fácilmente. La aplicación de robots en líneas de ensamblaje, logística y agricultura confirman incrementos de productividad y mejoras en ergonomía cuando se integran cobots con sensores de fuerza y visión inteligente [55].

Finalmente, la convergencia de estos pilares se consolida en arquitecturas en la nube para la AIoT (Inteligencia Artificial para el IoT) industrial, donde capas periféricas ejecutan inferencias locales y la nube se reserva para la optimización global. Este enfoque híbrido reduce la latencia, mejora la resiliencia ante fallos de red y facilita la escalabilidad de sistemas automatizados y procesos de producción distribuidos [56].

2.2 Robots móviles colaborativos en la industria 4.0

En plantas industriales, el trabajo de varios robots móviles, conocidos en este contexto como Automated Guided Vehicles (AGV's), están comenzando a suprimir los sistemas de manejo de materiales compuestos por bandas transportadoras o robots manipuladores sobre rieles. Esto incrementa la flexibilidad y productividad del sistema, apuntando hacia la llamada manufactura inteligente e industria 4.0 [50]. Ejemplo de múltiples robots trabajando en la industria se encuentran diseñados para el traslado de anaqueles en empresas de manejo de paquetería, robots con ruedas omnidireccionales, que se pegan a modo de vagones de tren, en fábricas de aviones, vehículos con brazos manipuladores a bordo, robots que transportan rodillos para cambios de bandas y diversos montacargas autónomos [57].

Los AGV's de última generación están dotados de sensores de auto localización y mapeo, conocidos como sistemas SLAM, que son capaces de ubicarse en un medio inicialmente desconocido y trazar rutas a través de la creación y actualización de un mapa de sitio [58]. Los sensores de ubicación son sistemas compuestos por cámaras de vídeo, cámaras 3D que miden profundidad muchas veces combinadas con sensores inerciales y de tipo lidar. La integración de estos sensores da redundancia de medición espacial, pero al mismo tiempo demandan un mayor procesamiento de cómputo [59]. Por lo tanto, el controlador de estos AGV's es una plataforma de cómputo embebida con un sistema operativo montado y capacidades de realizar procesamiento paralelo en tiempo real, tal es el caso de la plataforma Jetson de la marca Nvidia. El sistema operativo de robótica, típicamente conocido como sistema ROS, es una de las alternativas para la integración de sensores, actuadores y control de AGV's, debido a que permite un rápido desarrollo con la conexión de drivers disponibles de componentes industriales que los fabricantes proporcionan para acelerar la programación del control y la supervisión de los robots[60]. Finalmente, los AGV's son actuados por motores y sus etapas de potencia, que actualmente están disponibles en el mercado en una amplia gama, asegurando su regulación de torques, velocidad o posición angular con controladores clásicos, los cuales se enlazan al control mediante protocolos de comunicación estandarizados.

En el mercado actual de AGV's existen más de 98 marcas en el mundo, de las cuales la mayoría se encuentran Estados Unidos, China y Alemania. El

mercado de robots AGV se encuentra en crecimiento, de acuerdo con el portal Research in China [61], el volumen de ventas de robots móviles tipo AGV en México llegó a más de 40 mil unidades en 2019 y espera que en los próximos años las ventas continúen aumentando con una tasa de crecimiento constante aun dentro de los próximos años. Por otro lado, la aplicación de robots móviles y de servicios se encuentra creciendo en varios sectores, principalmente en el sector de logística industrial, sin embargo, se espera un gran crecimiento en robots médicos en los próximos años, además de otros sectores [62]. En el Apéndice A podemos ver un listado completo de las empresas manufactureras de AGV'S del mundo.

La principal aplicación de los robots AGV es la logística [3], cuya función principal es el movimiento de material, control de inventarios y apoyo en la cadena de producción. El principio de funcionamiento de los robots AGV, por mencionar una marca en específico como los robots MIR, basan su algoritmo de posicionamiento y control de trayectorias en sensores ubicados de forma local en el mismo robot, todo el procesamiento se ejecuta directamente en la computadora del robot, este solo posee un módulo de comunicación vía wifi para comunicarse remotamente con un PLC o computadora para recibir órdenes.

Todo lo anterior, plantea que el reto del desarrollo de AGV's está en la integración de componentes de software y hardware disponibles en el mercado y su reto principal no radica en el control de bajo nivel de sus actuadores, sino más bien en la coordinación de movimiento de varios AGV's para tareas específicas, su interacción con el ser humano siguiendo normas de seguridad, así como su interrelación con otros procesos automatizados ahora ligados con las tecnologías de la Industria 4.0 [63].

2.3 Control de formación y avance en formación

La coordinación de robots móviles puede servir para el traslado de personas u objetos. Desde el punto de vista teórico, las tareas de transporte se resuelven combinando el control de formación, evasión de colisiones (para colocarse equidistantes debajo del objeto, sin choques entre robots) y control de marcha (para mover al objeto en una trayectoria deseada).

Para que los robots logren implementarse de manera autónoma y cooperativa en la nueva industria 4.0, será necesario que estos dispositivos cuenten con sistemas de control que permitan un funcionamiento preciso en tareas de posicionamiento. Uno de los puntos más importantes de los sistemas de control en los robots es la posición ya que esta permitirá definir las estrategias de trayectoria y formación para el movimiento de forma coordinada y la cooperación en tareas.

La problemática base de los robots móviles, es la dificultad para encontrar su ubicación en el espacio, ya que pueden tener muchos problemas al tener únicamente como punto de referencia los sensores locales, lo que puede generar un error de ubicación y de mal trayectoria al no tener una retroalimentación externa. La localización precisa es muy importante en la robótica móvil, ya que es necesaria para planificar trayectorias y permitir sistemas de lazo cerrado en el control de robots móviles. Los estudios recientes coinciden en clasificar los sensores en dos grupos: propioceptivos, que miden el movimiento interno del robot, y exteroceptivos, que observan el entorno físico [64, 65].

Los sensores propioceptivos más extendidos son los encoders (codificadores de rueda) y las IMU (Unidades de Medida Inercial). Los encoders proporcionan estimaciones de desplazamiento con resolución submilimétrica, pero acumulan error sistemático por pérdidas de avance; las IMU ofrecen alta frecuencia angular-lineal y permanecen inmunes a las condiciones ambientales, si bien derivan por integración y son sensibles a vibraciones. La combinación encoder-IMU, filtrada con técnicas Kalman y fusión complementaria, demuestra reducciones en el error de posicionamiento frente a utilizar cada sensor por separado [66, 67]. Entre los sensores exteroceptivos, la visión y el LiDAR (por sus siglas en inglés, sensor de Detección y Medición por Luz) encabezan la navegación en 2D y 3D. Las cámaras monoculares, estéreo o RGB-D aportan denso contenido semántico y son económicas; el LiDAR entrega nubes de puntos métricas a gran distancia con alta precisión, aunque su precio y la carencia de color limitan ciertas aplicaciones. Las revisiones de 2025 señalan que la fusión LiDAR-cámara-IMU logra mejor robustez frente a oclusiones, interiores con pocas texturas y prolongados pasillos, superando la consistencia de la trayectoria respecto a SLAM puramente visual [67, 65].

En la actualidad la tendencia dominante es la fusión multisensor: mezclar datos cinemáticos propios con observaciones externas y anclar el sistema a referencias absolutas cuando estas estén disponibles. Modelos basados en

gráficos de factores y optimización no lineal integran en la misma ecuación encoders, IMU, cámaras, LiDAR y/o UWB (Sensor de Ultra Wide Band o Banda Ultra Ancha) [68], consiguiendo trayectorias coherentes incluso bajo pérdidas temporales de cada sensor individual y allanando el camino hacia flotas multirrobot que comparten mapas y se coordinan en tiempo real [65].

Para entender la coordinación de robots móviles tipo AGV, es necesario realizar un estudio de formación y marcha de robots móviles cinemáticos al caso de modelos dinámicos. El desarrollo de los modelos cinemáticos de robots móviles es un tópico esencial para entender y controlar a los vehículos en sus estados de posición, velocidad, fuerza, etc. cuando se encuentra en movimiento. Estos parámetros varían dependiendo del diseño del vehículo, la disposición de los neumáticos y el tipo de llanta, con los cuales se puede modelar al robot móvil, ya que cada configuración de llanta tiene un mecanismo diferentes y se traduce en un diferente modelo dinámico del robot [1]. Las configuraciones más comunes de llantas para robots móviles son: rueda fija, rueda convencional, rueda pivote y rueda omnidireccional. Cada tipo de rueda, se rige por una serie de ecuaciones cinemáticas que caracterizan sus propiedades dinámicas dependiendo de su rango de movilidad y posición, y que a su vez impacta directamente con el modelado cinemático del vehículo [1].

La formulación del problema en robótica móvil se aborda inicialmente con respecto al modelo cinemático del tipo de robot móvil y la configuración mecánica. Algunas contribuciones se han centrado en robots con tracción diferencial o tipo unicycle [8], robots omnidireccionales [9] y configuraciones tipo vehículo.

La configuración de llanta omnidireccional es de gran interés para el desarrollo de nuevos vehículos con capacidad de máxima maniobrabilidad, ya que estas permiten realizar el movimiento básico de un robot móvil en un plano cartesiano [57]. El uso y aplicación de este tipo de configuración de llanta es un modelo dinámico estudiado y se han acoplado en temas de modelado cinemático, análisis de la singularidad, y control de movimiento de robots [57]. Un tema esencial para lograr un óptimo control de los vehículos terrestres por medio de sus modelos es la caracterización dinámica del robot móvil, ya que considera la evolución de la posición, velocidad y aceleración en respuesta de los pares. El estudio de la geometría de la tracción de los robots móviles y su control en un sistema ciberfísico es un tema de estudio abierto y de carácter complejo, que se vuelve crucial para el desempeño de los vehículos conectados

y controlados por un sistema en la nube.

El estudio fundamental que precede a muchas estrategias de coordinación en sistemas multi-robot es el llamado “*problema de consenso*”, el cual constituye una de las piedras angulares del control distribuido en sistemas multi-agente. En términos generales, el consenso implica diseñar leyes de control locales para un conjunto de agentes autónomos con el fin de alcanzar un acuerdo común sobre una cierta variable de interés (posición, velocidad, orientación, etc.) a través de interacciones locales limitadas por una topología de comunicación [69]. Inicialmente formulado para agentes con dinámica de simple integrador en espacio euclidiano, el objetivo del consenso es lograr que todas las trayectorias converjan hacia un valor común a partir de información parcial y distribuida [70]. Este enfoque es particularmente relevante en la coordinación de robots móviles, ya que permite modelar y analizar comportamientos colectivos sin la necesidad de un controlador centralizado. La interacción entre agentes se modela comúnmente mediante grafos dirigidos o no dirigidos, donde los nodos representan a los agentes y las aristas la disponibilidad de información entre ellos [71]. El consenso no es un concepto exclusivo de la robótica. Ha sido ampliamente aplicado en dominios como el control de osciladores acoplados en redes eléctricas, coordinación de vehículos autónomos, regulación del tráfico vehicular, sincronización de relojes distribuidos y hasta en la propagación de opiniones en redes sociales [72]. Esta versatilidad ha llevado al desarrollo de múltiples variantes del problema clásico, incluyendo el consenso promedio, el consenso finito-tiempo y el consenso con dinámica de orden superior, cada uno adaptado a distintos tipos de agentes y contextos. Además, el análisis se ha extendido a sistemas con dinámica más compleja, como agentes con modelos de doble integrador, dinámica no lineal, sistemas híbridos o agentes gobernados por derivadas fraccionarias [73]. En estos casos, las condiciones de estabilidad y convergencia requieren herramientas avanzadas como funciones de Lyapunov, desigualdades matriciales lineales (LMI) y teoría espectral de grafos.

Para robótica móvil, el consenso se transforma en el problema de “*formación*”, donde los agentes son robots móviles en el plano o espacio, y el punto de convergencia es ahora un posicionamiento relativo deseado representado por un “*patrón de formación*” [13]. Se agrega el problema de evasión de colisiones entre robots y obstáculos en el ambiente, cuando los robots se mueven de posiciones arbitrarias iniciales hacia ese patrón de formación deseado [12]. La gráfica de comunicación debe ser conectada y en ocasiones se abordan

topologías específicas como grafos dirigidos o no dirigidos, centradas en un líder, seguimiento cíclico, tipo convoy, etc., [74, 75]. El posicionamiento relativo puede definirse con vectores estáticos, desplazamientos relativos o distancias y ángulos entre agentes [13]. Si el patrón de formación se mantiene rígido y se mueve en el plano o espacio siguiendo alguna trayectoria deseada, se denomina “*marcha*” o “*avance en formación*” [43], si el posicionamiento relativo cambia en el tiempo (se expande, contrae, rota, deforma o se pierden o añaden links de comunicación) se conoce como “*formación variante en el tiempo*” [76].

2.3.1 Temas recientes en el modelado y control de formación

La coordinación de movimiento de robots móviles y su aplicación industrial es un tópico actual de investigación interdisciplinaria [2]. Los tópicos de investigación en esta área se enfocan en múltiples agentes móviles [77], control de formación [78], coordinación basada en posiciones o distancias relativas entre agentes, el avance en formaciones tipo parvada [79] y la evasión de colisiones [80]. Las motricidades de robots móviles más estudiadas involucran a robots terrestres con ruedas tipo unicycle [81] y omnidireccionales [82] y drones multi-rotor [83]. Las estrategias de control en un grupo multi-agente de robots móviles se pueden basar en comportamientos biológicos tipo enjambres [84], redes neuronales [85] o grafos [86]. Para el caso de formaciones tipo enjambres, se ha reportado el uso de cámaras en cada agente que ayudan a conocer su entorno de forma limitada y la aplicación de leyes de control descentralizadas para mover al enjambre [87]. Otras investigaciones se han enfocado en utilizar plataformas tipo líder-seguidor [88], que se basan en el cálculo de distancias entre agentes por medio de velocidades constantes relativas medidas en su propio marco de referencia, logrando así una formación deseada con ayuda de la teoría de grafos [89]. Además de ello, se han realizado sistemas capaces de evadir obstáculos estáticos y dinámicos durante el movimiento de la formación utilizando una red neuronal [85], otros trabajos estudian pérdidas o retardos en la comunicación [84]. La implementación de los algoritmos de coordinación requiere la construcción de los robots móviles, con la implementación de sensores y sistemas de comunicación acorde a la aplicación y a la estrategia de coordinación planteada. Así, los esquemas basados en consenso posicional [90], requieren que los robots midan su propia posición respecto a su ambiente

así como la de otros robots, a través de sistemas de localización global o referenciada. Los esquemas basados en distancias y orientación son más descentralizados [91] y requieren sensores de distancia relativa, orientación (IMU), cámaras a bordo, entre otros. En el acondicionamiento y combinación de las señales de los sensores, convergen temas de procesamiento de señales, redes sensoriales [92], visión artificial [93, 94], etc. La comunicación inalámbrica entre los robots se elige de acuerdo a la cantidad y velocidad de datos compartidos y el rango de distancia de comunicación, considerando opciones como WiFi, Bluetooth, Zigbee, RF, etc. Finalmente, los requerimientos de procesamiento definen el tipo de microprocesador digital a bordo del robot móvil, que van desde micro-controladores, hasta micro computadoras como Raspberry, Jetson Nano - Nvidia o una computadora tipo industrial. Los robots móviles físicos deben ser enlazados a un sistema ciberfísico mediante el cual se realiza la etapa de control y define el proceso y acciones con el cual interactúan y se corrigen los errores. Esto lleva a los robots de una simple interacción experimental y grupal, a la solución de un problema real de ingeniería mediante la conexión a un sistema de procesamiento en la nube. Por ejemplo, si los robots realizaran una tarea en un sistema automatizado, ellos deberán comunicarse con el planeador de producción para sincronizarse con los almacenes, máquinas y estaciones de trabajo automatizados [10]. En el caso de vigilancia, el gestor deberá definir rutas y analizar en línea las imágenes provistas por los agentes móviles [95]. Se han propuesto opciones como sistemas de eventos discretos [10], lógica difusa [96] o redes neuronales [97], obedeciendo a estándares industriales [46], normas de seguridad, tipos de materiales y robustez ante pérdida de comunicación. En años recientes, los avances en la literatura se han centrado en contribuir en esquemas de control basados en el posicionamiento, que faciliten la implementación de formación y avance. El posicionamiento ha migrado gradualmente de posicionamiento absoluto a relativo. Es decir de vectores estáticos globales, que dependerían de conocer la posición absoluta de los agentes en el plano o espacio, a métricas basadas en distancias y ángulos relativos que pueden ser medidos por sensores a bordo [13]. Esto es relevante cuando los agentes no se mueven en un ambiente limitado y medido por un sensor global (GPS, cámara en el techo o sistemas de captación de movimiento) y necesitan moverse en ambientes abiertos, donde requieren detectar la ubicación de los otros robots con la fusión de cámaras a bordo, lidars y sensores inerciales, entre otros. El control de formación basado en distancia ha predominado en los últimos años [45, 12, 49]. La información de sensores de posicionamiento relativo no es precisa y se deben

considerar algunos aspectos que compensen estas deficiencias y ayuden a la estabilidad del sistema. Uno de los puntos mas importantes para lograr la comunicación entre los robots móviles es el uso de sistemas ciberfísicos, lo cual permitirá tener un sistema de sensado de posicionamiento relativo, basado en la comunicación que puedan tener entre los vehículos y su tiempo de respuesta. Uno de los problemas en los sistemas de control mediante ambientes ciberfísicos son los retardos en la comunicación [98], la pérdida instantánea de comunicación, entre otros [13]. El patrón de formación y las trayectorias a seguir en un avance en formación pueden ser generados por un planeador de rutas y de formación en la nube.

Capítulo 3

Estado del arte

Este capítulo da un panorama general de los dos enfoques utilizados en la presente tesis para abordar el modelado y control de la coordinación grupal de movimiento de robots móviles en ambientes ciberfísicos: modelado y control de formaciones ciberfísicas y sistemas lineales con retardos, como base para los siguientes capitulos.

3.1 Formaciones ciberfísicas

La formación ciberfísica surge como una evolución natural de la robótica distribuida, en lugar de concentrar a todos los robots en un único entorno, los agentes se reparten en espacios de trabajo geográficamente separados y mantienen la cohesión de grupo mediante redes de comunicación que atraviesan Internet y la nube. Esta idea aprovecha la madurez de la computación perimetral y las redes de comunicaciones para sincronizar estado y decisiones en tiempo casi real, lo que habilita aplicaciones como tele operación, patrullaje cooperativo, gemelos digitales, entre otras tareas cooperativas [99]. Los esquemas de robótica en sistemas ciberfísicos se apoyan en la infraestructura de la nube, donde se aloja el gemelo digital , agente virtual o “*avatar*”, tambien en la nube se ejecutan procesos pesados de optimización y distribución de tareas para cada grupo de robots. Este reparto jerárquico reduce la carga de cómputo local y facilita la escalabilidad cuando se añaden nuevos robots o se reconfiguran las formaciones. De acuerdo a Kivrak [100], integrar simulación, control y análisis de datos en una misma plataforma ciberfísica disminuye los

tiempos de despliegue industrial y simplifica la gestión de la seguridad. La viabilidad de las formaciones ciberfísicas se deben a dos avances recientes: a las redes inalámbricas de muy alto ancho de banda y latencia ultrabaja (5G y 6G), las cuales proporcionan canales estables para transmitir poses, estimaciones y comandos en escalas de milisegundos. Revisiones sobre 6G subrayan que el objetivo de ≤ 0.1 ms de retardo y $\geq 10^9$ bits/s está enfocado para sistemas “*machine to machine*” y flotas robóticas distribuidas [101], mientras que estudios de 6G y superficies reconfigurables perfilan coberturas resilientes entre espacios de trabajo remotos [102]. La generación 5G actual ya demuestra fiabilidad industrial y tasas de entrega con una fiabilidad casi total en sistemas de control de robots móviles dentro de fábricas piloto [103]. En segundo lugar, las arquitecturas de cómputo embebido y periféricos, como las plataformas heterogéneas con CPU + GPU + NPU procesan SLAM y control predictivo a bordo, a la vez que los dispositivos o nodos de cómputo finales (edge-computing devices) descargan inferencias de alto costo cuando la conectividad lo permite. De acuerdo a Tahir [104], menciona que colocar la lógica intensiva en nodos edge reduce la latencia respecto a la nube pura, lo cual puede alargar la vida útil de baterías móviles al disminuir picos de cómputo local o de comunicación.

3.2 Problemas actuales de formaciones ciberfísicas

La principal dificultad técnica proviene de los retardos y la pérdida eventual de paquetes en Internet. Investigaciones recientes emplean modelos de retardo variable, funcionales de Lyapunov y desigualdades lineales matriciales para delimitar la cota máxima de latencia que preserva la estabilidad de la formación. Asimismo, algoritmos predictivos basados en datos de vista previa y aprendizaje adaptativo compensan variaciones súbitas de ancho de banda, logrando que los robots mantengan la convergencia cuando la red sufre fluctuaciones en su tiempo de ida y vuelta [105]. Los esquemas de formación ciberfísica se benefician del enorme ancho de banda y la baja latencia que hoy ofrecen las redes de internet, pero siguen afrontando retos técnicos. El intercambio de información entre robots que operan en distintos espacios físicos de trabajo dependen de que los paquetes de estado y control lleguen a tiempo; experimentos recientes con dos espacios de trabajo remotos demuestran que con retardos promedio

de 80 a 120 ms, la convergencia a la formación puede mantenerse, aunque el margen de estabilidad decrece conforme la latencia varía [106, 107]. En el plano del diseño de control, las leyes clásicas de consenso deben adaptarse para funcionar sobre canales con variaciones rápidas. Modelos predictivos distribuidos y observadores de retardo, como los propuestos en [31, 32, 33], se han extendido con técnicas de vista previa que migran la carga computacional entre el borde y la nube según la calidad del enlace [108]. Los retardos variables de Internet son una amenaza dominante: si la latencia supera un umbral crítico, el lapso de muestreo efectivo se dilata y degrada la ganancia de fase, originando bifurcaciones y pérdida de cohesión de la formación, tal como se observó con barreras de control en robots diferenciales reales [109] y en ensayos de realidad mixta que evidencian patrones rotacionales inducidos por retardo [110]. Estrategias basadas en LMI confirman analíticamente los límites de retardo tolerable [34, 35]. Las pérdidas de comunicación entre espacios físicos de trabajo añaden un segundo nivel de riesgo: cuando la probabilidad de pérdida supera el 30%, aparecen colapsos parciales de la topología que pueden fragmentar el grafo de interacción; gráficos probabilísticos y control ergódico con conectividad intermitente han sido propuestos para mitigar esta situación [111], mientras que estudios clásicos sobre formaciones bajo canal con errores de bit ya alertaban de colisiones y comportamientos anómalos [36, 37]. Nuevas arquitecturas basadas en 5G con replicación y eliminación de tramas, evidencian tasas de entrega satisfactorias y reducen el impacto de la conectividad intermitente [112].

3.2.1 Control de robots en sistemas ciberfísicos

El control de formación en ambientes ciberfísicos es un tópico reciente en el contexto de robots multi-agentes. Existen temas abiertos desde el punto de vista tecnológico. Por ejemplo, el envío y recepción de datos compartidos entre robots con un servicio de la nube, el despliegue de esquemas de formación en ambientes abiertos, donde todos los sistemas de posicionamiento y localización deben estar montados en los robots. También los retos se extienden desde el punto de vista teórico, donde es necesario definir nuevas estrategias de coordinación, nuevos algoritmos con énfasis en los retardos por imprecisiones de medición, pérdidas de comunicación y mejoras en la seguridad para el humano que interactúa con los robots. El control de formación en ambientes ciberfísicos ha emergido como extensión de la robótica cooperativa tradicional al escenario en que los robots comparten datos por redes de internet y delegan

parte del cómputo en la nube o en nodos edge. Esta nueva configuración tecnológicamente viable gracias a tecnologías 5G y 6G permite sostener tasas de actualización altas [113]. Uno de los retos clave es el intercambio seguro y fiable de información entre robots y la nube, sobre todo de sincronizar información de posición, velocidad y referencias de formación, la cual se envía mediante sistemas no lineales cuando la latencia y la variabilidad del jitter superan ciertos umbrales; estrategias de precómputo y migración dinámica del control entre el sistema de control a bordo y la nube reducen el error de formación frente a esquemas fijos [105, 100]. En entornos abiertos, la totalidad de la percepción se monta en los robots, lo que exige cámaras, LiDAR o UWB de bajo consumo y fusionados con filtros que compensen oclusiones y ruido, bajo estas condiciones la robustez de estos sistemas se llega a obtener hasta 150 ms de retardo con pérdidas inferiores al 5% [114].

Desde el punto de vista teórico, los algoritmos de coordinación con retardo y pérdidas han evolucionado desde el consenso clásico a variantes predictivas [115] que incorporan observadores de red [116], barreras de seguridad [117] y gramáticas lógicas [118] para garantizar la integridad del sistema de control. Modelos con desigualdades lineales matriciales establecen cotas explícitas para la latencia y la tasa de paquetes perdidos que el sistema puede tolerar antes de perder estabilidad [119]; paralelamente, controladores con restricciones de velocidad y distancias de seguridad añaden una segunda capa de protección activa [120, 121]

La factibilidad práctica se ha demostrado en experimentos intercontinentales, como la prueba que enlazó los laboratorios de Georgia Tech (EE. UU.) y Tokyo Tech (Japón) a más de 11 000 km, donde dos equipos de robots mantuvieron una formación compartida pese a retardos de ida y vuelta de 150 ms [122, 123]. Los avances más recientes en formación ciberfísica han trasladado el concepto de coordinación de robots al dominio logístico multi-almacén. Un esquema de programación jerárquica conecta la capa de misión, la capa que decide qué almacén atiende cada robot, con la capa de despacho, la que gestiona rutas y evita la congestión dentro de cada instalación, todo ello soportado por un sistema ciberfísico industrial que monitorea stocks y flujos en tiempo real [124, 20]. La jerarquía permite que cientos de robots móviles compartan recursos globales (la nube) sin sobrecargar los enlaces locales: la computación compleja se optimiza en la nube y las trayectorias finas se refinan en dispositivos finales (edge), reduciendo los tiempos de servicio entre un 15 a 20% frente a planificadores centralizados [124]. Se han propuesto

sistemas de control ciberfísicos para robots logísticos autónomos integrando sensores de planta, PLC y middleware de automatización para habilitar toma de decisiones en tiempo real; los controladores se despliegan como micro-servicios que migran entre la SBC del robot y el clúster edge según la carga de red, manteniendo la latencia de 18 a 42 ms, incluso durante picos de tráfico [21, 32]. También se han desarrollado sistemas de planificación de trayectorias en robots mediante un sistema ciberfísico que monitorea y controla el desgaste de los motores eléctricos, donde el optimizador evita secuencias de aceleración y frenado que degraden prematuramente el bobinado y distribuye las cargas térmicas entre ejes motrices; los estudios reportan ahorros energéticos del orden del 10 al 14% y prolongación de la vida útil de los motores en un 15% a lo largo de un ciclo anual [125, 22].

3.2.2 Sistemas lineales con retardos

Los sistemas lineales con retardo describen dinámicas en las que la velocidad de cambio del estado depende, además del valor presente, de un estado pasado; esto refleja el efecto de la latencia en los lazos de comunicación de formaciones ciberfísicas. En un escenario multi-robot distribuido por Internet, los retardos son inevitables y variables; sin un modelo explícito que los contemple, las herramientas de estabilidad convencionales subestiman la ganancia de fase perdida, ya que en la práctica emergen oscilaciones o incluso colisiones de la formación. Por esa razón, es necesario considerar un análisis de sistemas lineales con retardo, ya que permite identificar los tiempos de latencia admisibles, guía la selección de ganancias y respalda analíticamente los experimentos con agentes virtuales en los diferentes espacios de trabajo que abordaremos en los siguientes capítulos.

La teoría de sistemas lineales con retardo ofrece el marco analítico para comprender cómo la latencia de comunicación afecta la estabilidad de formaciones ciberfísicas de robots móviles. Las desigualdades matriciales lineales (LMI, por sus siglas en inglés), son una herramienta convexa capaz de traducir condiciones de estabilidad, amortiguamiento y desempeño en pruebas algorítmicas resolubles con programación semidefinida [126]. De acuerdo a Mondié y Kharitonov, se plantea un modelo lineal con retardo constante y se muestra que una condición *suficiente* para la estabilidad exponencial con razón $\alpha > 0$ es la existencia de una matriz simétrica definida positiva $P \succ 0$ que satisface una desigualdad matricial lineal (LMI) dependiente de A_0 , A_1 y del retardo τ

[127].

La dinámica lineal con retardo fijo se modela como

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau), \quad \tau > 0, \quad (1)$$

con $A_0, A_1 \in \mathbb{R}^{n \times n}$.

La estabilidad exponencial con razón $\alpha > 0$ se analiza mediante el funcional de Lyapunov–Krasovski

$$V(x_t) = x^\top(t) P x(t) + \int_{t-\tau}^t e^{2\alpha(s-t)} x^\top(s) P x(s) ds, \quad P \succ 0. \quad (2)$$

Evalutando $\dot{V} + 2\alpha V$ a lo largo de (1) y aplicando el complemento de Schur, se obtiene la LMI

$$\Phi(\alpha, \tau, P) = \begin{bmatrix} A_0^\top P + P A_0 + (1 + 2\alpha)P & P A_1 \\ A_1^\top P & -(1 - e^{-2\alpha\tau})P \end{bmatrix} \prec 0, \quad (3)$$

donde $\text{He}(X) = X + X^\top$ y, en particular, el bloque $(1, 1)$ es $\text{He}(P A_0) + (1 + 2\alpha)P$.

Si existe $P \succ 0$ que satisface (3), entonces el sistema (1) es exponencialmente estable con razón α y, para toda historia inicial $x_0 \in C([- \tau, 0], \mathbb{R}^n)$,

$$\|x(t)\| \leq \kappa e^{-\alpha t} \|x_0\|_\infty, \quad \kappa = \sqrt{\frac{(1 + \tau) \lambda_{\max}(P)}{\lambda_{\min}(P)}}.$$

La comprobación numérica se reduce a un problema convexo de factibilidad resoluble con programación semidefinida, sin barridos de parámetros ni aproximaciones espectrales. [127].

Los retardos de comunicación en sistemas ciberfísicos se manifiestan como latencias constantes, variables o intermitentes en los enlaces que conectan robots con la nube o robots entre sí. Estos retardos introducen etapas adicionales que pueden degradar la ganancia efectiva de los lazos de control, originando oscilaciones, pérdida de cohesión en formaciones y, en casos extremos, inestabilidad. Experimentos recientes en robots distribuidos confirman que retardos medios de sólo 1.8-4 s son suficientes para disparar bifurcaciones en los patrones colectivos cuando no se emplean técnicas de compensación [110].

La idea central consiste en estimar el estado futuro $x(t + \tau)$ a partir del modelo dinámico y usarlo en la realimentación, de modo que la acción de control se adelante al propio retardo. Siguiendo la transformación de Artstein [128, 129], el esquema general adopta la forma

$$u(t) = K \hat{x}(t + \tau), \quad \hat{x}(t + \tau) = e^{A\tau} x(t) + \int_0^\tau e^{As} B u(t - s) ds, \quad (4)$$

y ha sido extendido a redes multi-agente mediante observadores distribuidos y algoritmos de consenso predictivo [130, 131].

Estas estrategias pueden mantener el error de formación por debajo aun con retardos variables, pero su desempeño depende de la precisión del modelo y del ancho de banda del canal de retroalimentación.

Para control con retardo se emplean funcionales de Lyapunov-Razumikhin y Lyapunov-Krasovskii. El criterio Razumikhin impone desigualdades puntuales sobre el estado retrasado, mientras que Krasovskii incorpora integrales en la historia del sistema, proporcionando resultados menos conservadores. En ambos casos se derivan condiciones de desigualdades matriciales lineales (LMI) resolubles por programación semidefinida; los trabajos de Kharitonov y Mondié ofrecen estimaciones exponenciales y métodos de síntesis robusta fundamentados en esta técnica [132, 25].

Para retardos heterogéneos y oscilatorios, se han desarrollado extensiones que emplean desigualdades de reciprocidad y de Wirtinger. Trinh presentan criterios LMI que garantizan estabilidad exponencial y, al mismo tiempo, permiten optimizar de forma simultánea las demoras dependientes del tiempo, superando las cotas obtenidas con enfoques más clásicos [133]. Otros marcos de control de retardos pueden ser el control basado en pasividad, el cual garantiza que la interconexión de subsistemas conserva energía almacenada. Notomista muestra que la combinación de funciones barrera y pasividad descentralizada mantiene la cohesión aun con retardo y pérdida de paquetes [134]. También se encuentra el control por Back-stepping y modo deslizante, el cual introduce robustez frente a incertidumbres y demoras, mediante superficies de deslizamiento o transformaciones recursivas que obligan al error a converger independientemente de variaciones moderadas de latencia; Karafyllis y otros autores reportan convergencia con derivadas acotadas incluso en sistemas no lineales con demoras internas [135].

Capítulo 4

Estrategia de control

Este capítulo presenta la contribucion teorica principal del trabajo de tesis. Se describe el conjunto de estrategias de control con y sin retardos en las comunicaciones, así como diferentes simulaciones realizadas con y sin retardos en diferentes casos de formación pero utilizando las mismas estrategias de control propuestas.

4.1 Formulación del problema

En este trabajo, el problema de formación ciberfísica se define para n robots puntuales que se desplazan en dos espacios de trabajo diferentes.

Sea $\mathcal{R} = \{R_1, \dots, R_n\}$ un conjunto de robots móviles planos que se mueven en dos espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$. Sin pérdida de generalidad, supongamos que $\mathcal{R}_1 = \{R_1, \dots, R_m\}$ y $\mathcal{R}_2 = \{R_{m+1}, \dots, R_n\}$, con $1 \leq m < n$, son los subconjuntos de robots que se desplazan en $\mathcal{W}_1$ y $\mathcal{W}_2$, respectivamente. Por lo tanto, $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ con $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Defínase $\mathbf{z}_i = [x_i, y_i]^\top \in \mathbb{R}^2$ como la posición 2D de R_i en $\mathcal{W}_1$ o $\mathcal{W}_2$.

Este trabajo se centra en la construcción de una estrategia de control de formación líder-seguidor tal que los robots en los dos espacios de trabajo converjan a una formación global como si estuvieran en el mismo espacio. El robot líder debe seguir una trayectoria deseada dada por $\mathbf{z}_d(t) \in \mathbb{R}^2$.

Un grafo de formación global deseado $\mathcal{G}$ para los n robots se define mediante una matriz de adyacencia $\mathbf{A}$, un conjunto de posiciones relativas $\mathcal{C}$ y una

matriz laplaciana $\mathbf{L}$. El desafío en una configuración ciberfísica es converger a la formación especificada por $\mathcal{G}$ en ausencia de ciertos robots en cada espacio de trabajo. Para ello, la información sobre las posiciones $\mathbf{z}_i$, $i = 1, \dots, n$, debe compartirse entre $\mathcal{W}_1$ y $\mathcal{W}_2$, como se ilustra en la Figura 4.1 utilizando Internet y la nube.

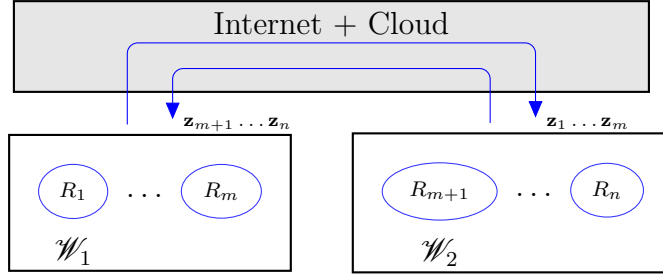


Figura 4.1: Comunicación entre los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$.

4.2 Estrategia de control

Para abordar el problema de formación ciberfísica definido en la sección anterior, este trabajo propone el uso de robots virtuales. Así, para cada $R_i \in \mathcal{R}_1$, $i = 1, \dots, m$, se define un robot virtual $V_i \in \mathcal{V}_2$, $i = 1, \dots, m$, y para cada $R_i \in \mathcal{R}_2$, $i = m + 1, \dots, n$, se define un robot virtual $V_i \in \mathcal{V}_1$, $i = m + 1, \dots, n$. De este modo, también hay n robots virtuales divididos en $\mathcal{V}_1$ y $\mathcal{V}_2$. Sea $\mathbf{v}_i \in \mathbb{R}^2$ la posición 2D de cada robot virtual V_i . Finalmente, $\mathcal{W}_i = \mathcal{R}_i \cup \mathcal{V}_i$, $i = 1, 2$ y $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$.

Para este nuevo sistema ciberfísico compuesto por $2n$ agentes generado de esta manera, se define un grafo de formación combinado $\hat{\mathcal{G}}$, derivado de $\mathcal{G}$, de la forma siguiente.

Para cada $R_j \in \mathcal{R}_i$, sean $\mathcal{N}_{ij}^R$ y $\mathcal{N}_{ij}^V$ los subconjuntos adyacentes de índices de los robots reales y virtuales en $\mathcal{W}_i$ que se comunican con R_j , según:

$$\begin{aligned} \mathcal{N}_{ij}^R &= \{k \mid \mathbf{A}_{jk} \neq 0, R_j, R_k \in \mathcal{R}_i\} \\ \mathcal{N}_{ij}^V &= \{k \mid \mathbf{A}_{jk} \neq 0, R_j \in \mathcal{R}_i, V_k \in \mathcal{V}_i\}. \end{aligned} \quad (5)$$

En adición, todos los robots virtuales $V_j \in \mathcal{W}_i$ se comunican únicamente con sus homólogos reales R_j ubicados en el otro espacio de trabajo $\mathcal{W}_{3-i}$; es decir,

sus conjuntos adyacentes correspondientes $\hat{\mathcal{N}}_{ij}^R, \hat{\mathcal{N}}_{ij}^V$ son

$$\begin{aligned}\hat{\mathcal{N}}_{ij}^R &= \{j \mid V_j \in \mathcal{W}_i, R_j \in \mathcal{W}_{3-i}\} \\ \hat{\mathcal{N}}_{ij}^V &= \emptyset.\end{aligned}\tag{6}$$

Un ejemplo de esta topología de comunicación se muestra en la Figura 4.2. Además, para cada par (j, k) con $k \in \mathcal{N}_{ij}^R \cup \mathcal{N}_{ij}^V$, sea $\mathbf{c}_{kj} \in \mathbb{R}^2$ el vector de posición relativa deseada, estático y predefinido, con $\mathbf{c}_{kj} \in \mathcal{C}$. Por diseño, es sencillo demostrar que si $\mathcal{G}$ tiene un árbol generador dirigido [136], entonces $\hat{\mathcal{G}}$ también lo tiene. Al contar con un árbol generador dirigido, es bien sabido que la matriz laplaciana de $\mathcal{G}$ y de $\hat{\mathcal{G}}$, dadas por $\mathbf{L}$ y $\hat{\mathbf{L}}$ respectivamente, tienen exactamente un valor propio nulo, siendo el resto positivos [1].

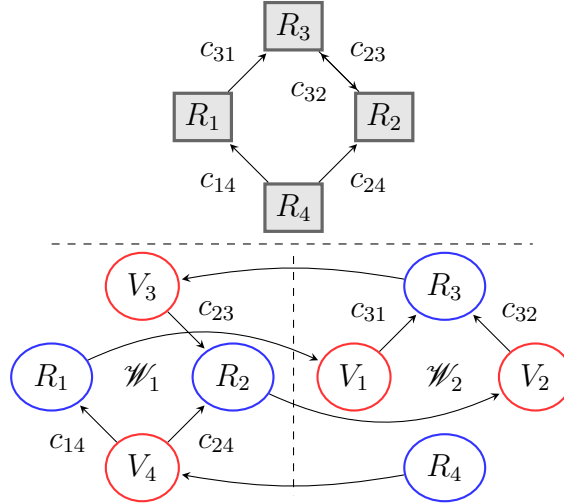


Figura 4.2: Ejemplo para $(n = 4, m = 2)$ de la relación entre el grafo de formación original $\mathcal{G}$ (arriba), los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$ y el grafo combinado $\hat{\mathcal{G}}$ (abajo).

A continuación, se definen las leyes de control para esta configuración de robots reales y virtuales. Primero, consideremos ambos como integradores simples, dados por

$$\begin{aligned}\dot{\mathbf{z}}_i &= \mathbf{u}_i, & i &= 1, \dots, n \\ \dot{\mathbf{v}}_i &= \mathbf{w}_i, & i &= 1, \dots, n\end{aligned}\tag{7}$$

donde $\mathbf{u}_i \in \mathbb{R}^2$ y $\mathbf{w}_i \in \mathbb{R}^2$ son las entradas de control para los robots reales y virtuales, respectivamente. Para la estrategia líder-seguidor introducida en este trabajo, definamos $R_n \in \mathcal{W}_2$ como el robot líder y $\mathbf{z}_d, \dot{\mathbf{z}}_d \in \mathbb{R}^2$ como la posición y velocidad deseadas de la trayectoria para el líder. Además, los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$ podrían estar a gran distancia entre sí, por lo que debe considerarse un retardo significativo en la información intercambiada.

La estrategia de control para los robots virtuales $V_j \in \mathcal{W}_i$ está diseñada para seguir a su robot real correspondiente R_j que pertenece al otro espacio de trabajo $\mathcal{W}_{3-i}$, según la ley:

$$\mathbf{w}_j(t) = k_v \left(\mathbf{z}_j(t - \tau) - \mathbf{v}_j(t) \right) + \dot{\mathbf{z}}_d(t), \quad k_v > 0, \quad j = 1, \dots, n, \quad (8)$$

donde $\tau \geq 0$ es el retardo de comunicación entre espacios y $k_v > 0$ es la ganancia de control de los robots virtuales. La ley de control para los robots reales R_j viene dada por:

$$\mathbf{u}_j(t) = k_r \sum_{k \in \mathcal{N}_{ij}^R} \left(\mathbf{z}_k(t) + \mathbf{c}_{kj} - \mathbf{z}_j(t) \right) + k_r \sum_{k \in \mathcal{N}_{ij}^V} \left(\mathbf{v}_k(t) + \mathbf{c}_{kj} - \mathbf{z}_j(t) \right) + \dot{\mathbf{z}}_d(t), \quad (9)$$

con $k_r > 0$ la ganancia de control para los robots reales y $j = 1, \dots, n-1$. La estrategia de control para el robot líder R_n es:

$$\mathbf{u}_n = k_r \sum_{k \in \mathcal{N}_{2n}^R} \left(\mathbf{z}_k + \mathbf{c}_{kn} - \mathbf{z}_n \right) + k_r \sum_{k \in \mathcal{N}_{2n}^V} \left(\mathbf{v}_k + \mathbf{c}_{kn} - \mathbf{z}_n \right) + k_c \left(\mathbf{z}_d - \mathbf{z}_n \right) + \dot{\mathbf{z}}_d, \quad (10)$$

donde la ganancia de control es $k_c > 0$. Obsérvese que la ley de control para los robots reales R_j no se ve afectada por el retardo τ entre espacios, ya que toda la información requerida está dentro de su propio espacio de trabajo.

Al definir las posiciones relativas respecto de la trayectoria deseada como $\tilde{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{z}_d$ y $\tilde{\mathbf{v}}_j = \mathbf{v}_j - \mathbf{z}_d$, las Ecuaciones (8)-(10) tienen un equilibrio de formación definido por:

$$\begin{aligned} \tilde{\mathbf{v}}_j^* &= \tilde{\mathbf{z}}_j^*, \quad \forall j, \\ 0 &= \sum_{k \in \mathcal{N}_{ij}^R} \left(\tilde{\mathbf{z}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^* \right) + \sum_{k \in \mathcal{N}_{ij}^V} \left(\tilde{\mathbf{v}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^* \right), \quad \forall j < n, \\ \tilde{\mathbf{z}}_n^* &= 0. \end{aligned} \quad (11)$$

Sea $e_j^R = \tilde{\mathbf{z}}_j - \tilde{\mathbf{z}}_j^*$ y $e_j^V = \tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_j^*$ el error relativo respecto de la formación de los robots reales y virtuales, respectivamente. Con las ecuaciones anteriores, la dinámica del error se expresa como:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{e}}^V(t) \\ \dot{\mathbf{e}}^R(t) \end{bmatrix} &= \left(\begin{bmatrix} -k_v \mathbf{I}_n & \mathbf{0}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \otimes \mathbf{I}_2 \right) \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix} \\ &+ \left(\begin{bmatrix} k_v \mathbf{I}_n \\ \mathbf{0}_n \end{bmatrix} \otimes \mathbf{I}_2 \right) \mathbf{e}^R(t - \tau), \end{aligned} \quad (12)$$

donde

$$\begin{aligned} \mathbf{e}^V(t) &= [e_1^{V^\top}(t) \dots e_n^{V^\top}(t)]^\top, \\ \mathbf{e}^R(t) &= [e_1^{R^\top}(t) \dots e_n^{R^\top}(t)]^\top, \\ \mathbf{E}_{21} + \mathbf{E}_{22} &= -\mathbf{L}, \\ \mathbf{q}_n &= [\mathbf{0}_{n-1}^\top \ 1]^\top, \end{aligned} \quad (13)$$

con $\mathbf{0}_k$ un vector nulo de dimensión k .

4.2.1 Control sin retardos en las comunicaciones

En el caso ideal sin retardos en la información intercambiada, se cumple el siguiente resultado.

Teorema 4.2.1. *Para un conjunto $\mathcal{R}$ de n robots móviles distribuidos en dos espacios físicos $\mathcal{W}_1$ y $\mathcal{W}_2$ con dinámica de primer orden dada por la Ecuación (7) y una configuración líder-seguidor definida por una topología de comunicación $\hat{\mathcal{G}}$ derivada de la topología original $\mathcal{G}$ con un árbol generador dirigido y dada por (5), (6), la estrategia de control dada por las Ecuaciones (8)-(10) sin retardo ($\tau = 0$) converge el error dinámico (14) al origen, logrando la formación definida por la Ecuación (11).*

Prueba. Cuando no hay retardo de comunicación ($\tau = 0$), la dinámica del error (12) se simplifica a

$$\begin{bmatrix} \dot{\mathbf{e}}^V(t) \\ \dot{\mathbf{e}}^R(t) \end{bmatrix} = \left(\begin{bmatrix} -k_v \mathbf{I}_n & k_v \mathbf{I}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \otimes \mathbf{I}_2 \right) \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix}. \quad (14)$$

Para demostrar la convergencia de (14), basta mostrar que todos los valores propios de la matriz $\mathbf{M}$ tienen parte real negativa. Definiendo

$$\mathbf{M} = \mathbf{K}\mathbf{F} = \begin{bmatrix} k_v \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & k_r \mathbf{I}_n \end{bmatrix} \begin{bmatrix} -\mathbf{I}_n & \mathbf{I}_n \\ \mathbf{E}_{21} & \mathbf{E}_{22} - \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \quad (15)$$

con $\hat{k}_c = k_c/k_r > 0$. Dado que $\mathbf{E}_{21} + \mathbf{E}_{22} = -\mathbf{L}$, y que todos los elementos diagonales $\{\mathbf{L}\}_{ii} = \{\mathbf{E}_{22}\}_{ii} \forall i$, al aplicar el Teorema de los círculos de Gershgorin [137] y considerando la estructura de la matriz laplaciana, es sencillo ver que la matriz $\mathbf{F}$ puede tener un valor propio nulo, mientras que el resto tienen parte real negativa. Como k_v , k_r , k_c y $\hat{k}_c$ son todos positivos, lo mismo puede afirmarse de $\mathbf{M}$. Para completar la demostración, se mostrará que $\mathbf{F}$ no posee un valor propio nulo.

Para que $\mathbf{F}$ tenga un valor propio nulo, debe existir un vector no nulo $\mathbf{p}$ tal que $\mathbf{F}\mathbf{p} = \mathbf{0}$. Usando la expresión (15) y escribiendo $\mathbf{p} = [\mathbf{p}_v^\top, \mathbf{p}_r^\top]^\top$, se obtiene

$$\mathbf{F}\mathbf{p} = \begin{bmatrix} -\mathbf{I}_n & \mathbf{I}_n \\ \mathbf{E}_{21} & \mathbf{E}_{22} - \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_v \\ \mathbf{p}_r \end{bmatrix} = \mathbf{0} \implies \begin{cases} \mathbf{p}_v = \mathbf{p}_r, \\ (\mathbf{L} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \mathbf{0}. \end{cases} \quad (16)$$

Debido a que $\mathcal{G}$ tiene un árbol generador dirigido, $\mathbf{L}$ posee un único valor propio nulo. Por tanto, su descomposición $\mathbf{L}\mathbf{U}$ presenta la estructura siguiente:

$$\mathbf{L} = \mathbf{Q}\mathbf{U} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ q_{21} & 1 & 0 & \vdots \\ \vdots & q_{ij} & \ddots & 0 \\ q_{n1} & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & \ddots & u_{ij} & \vdots \\ \vdots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \dots & \dots & 0 \end{bmatrix}, \quad (17)$$

Dado que $\mathbf{Q}$ es una matriz triangular inferior con elementos de la diagonal principal iguales a 1 y $\mathbf{U}$ es una matriz triangular superior con elementos de la diagonal principal $u_{ii} \neq 0, \forall i < n$ y $u_{nn} = 0$, obsérvese que $\mathbf{Q}$ es invertible

y que $\mathbf{Q} \mathbf{q}_n = \mathbf{Q}^{-1} \mathbf{q}_n = \mathbf{q}_n$. Al sustituir (17) en (16) se obtiene

$$(\mathbf{Q} \mathbf{U} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \mathbf{Q}(\mathbf{U} + \hat{k}_c \mathbf{Q}^{-1} \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \mathbf{0},$$

$$(\mathbf{U} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & \ddots & u_{ij} & \vdots \\ \vdots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \dots & \dots & \hat{k}_c \end{bmatrix} \mathbf{p}_r = \mathbf{0} \quad (18)$$

Como $u_{ii} \neq 0, \forall i < n$ y $\hat{k}_c > 0$, la única solución de la Ecuación (18) es $\mathbf{p}_r = \mathbf{0}$. Por tanto, la única solución de la Ecuación (16) es $\mathbf{p} = \mathbf{0}$, lo que implica que ni $\mathbf{F}$ ni $\mathbf{M}$ tienen valor propio nulo. En consecuencia, la dinámica del error (14) es estable. $\square$

4.2.2 Control con retardos en las comunicaciones

Cuando el retardo de comunicación entre espacios no puede despreciarse, debe analizarse la dinámica retardada (12). El siguiente resultado establece condiciones de estabilidad del lazo cerrado en presencia de retardos constantes en la comunicación.

Teorema 4.2.2. *Para un conjunto $\mathcal{R}$ de n robots móviles distribuidos en dos espacios físicos $\mathcal{W}_1$ y $\mathcal{W}_2$ con dinámica de primer orden dada por la Ecuación (7) y una configuración líder-seguidores definida por la topología de comunicación $\hat{\mathcal{G}}$ con un retardo fijo desconocido acotado ($0 < \tau \leq T$), y con la estrategia de control dada por las Ecuaciones (8)-(10), si existen matrices simétricas definidas positivas $\mathbf{P}, \mathbf{S}, \mathbf{Z} \in \mathbb{R}^{2n \times 2n}$ tales que*

$$\Gamma = \begin{bmatrix} A_0^\top \mathbf{P} + \mathbf{P} A_0 + \mathbf{S} + T A_0^\top \mathbf{Z} A_0 - \mathbf{Z}/T & \mathbf{P} A_1 + T A_0^\top \mathbf{Z} A_1 & \mathbf{Z}/T \\ A_1^\top \mathbf{P} + T A_1^\top \mathbf{Z} A_0 & -\mathbf{S} + T A_1^\top \mathbf{Z} A_1 & \mathbf{0} \\ \mathbf{Z}/T & 0 & -\mathbf{Z}/T \end{bmatrix} < 0, \quad (19)$$

con

$$A_0 = \begin{bmatrix} -k_v \mathbf{I}_n & \mathbf{0}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix}, \quad A_1 = \begin{bmatrix} \mathbf{0}_n & k_v \mathbf{I}_n \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix}. \quad (20)$$

la dinámica del error retardada (12) converge al origen, logrando la formación definida en la Ecuación (11).

Obsérvese que la Ecuación (19) constituye una desigualdad matricial lineal (LMI) en los coeficientes de $\mathbf{P}, \mathbf{S}, \mathbf{Z}$, parametrizable en T .

Prueba. Sea $\mathbf{x}(t)^\top = [\mathbf{e}^V(t)^\top, \mathbf{e}^R(t)^\top] \in \mathbb{R}^{2n}$ el vector de error en el instante t . Defínase también $\mathbf{x}_t(\theta)^\top = [\mathbf{e}^V(t + \theta)^\top, \mathbf{e}^R(t + \theta)^\top]$, $\theta \in [-T, 0]$, como la trayectoria de error en $[t - T, t]$. El error de la dinámica (12) se reescribe como

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}_t) = A_0 \mathbf{x}(t) + A_1 \mathbf{x}(t - \tau), \quad (21)$$

con matrices A_0, A_1 de (20). Para probar estabilidad, definimos el funcional de Lyapunov-Krasovski $V(\mathbf{x}_t)$ de la siguiente expresion:

$$\begin{aligned} V(\mathbf{x}_t) = & \mathbf{x}(t)^\top \mathbf{P} \mathbf{x}(t) + \int_{t-\tau}^t \mathbf{x}(s)^\top \mathbf{S} \mathbf{x}(s) ds \\ & + \int_{t-T}^t ds \int_s^t f(\mathbf{x}_\mu)^\top \mathbf{Z} f(\mathbf{x}_\mu) d\mu. \end{aligned} \quad (22)$$

Al escoger matrices $\mathbf{P}, \mathbf{S}$ y $\mathbf{Z}$ simétricas y definidas positivas, el funcional V resulta también positivo definido. Al derivar con respecto al tiempo la Ecuación (22) y aplicar la desigualdad de Jensen [138], se obtiene:

$$\begin{aligned} \dot{V}(\mathbf{x}_t) = & \dot{\mathbf{x}}(t)^\top \mathbf{P} \mathbf{x}(t) + \mathbf{x}(t)^\top \mathbf{P} \dot{\mathbf{x}}(t) + \mathbf{x}(t)^\top \mathbf{S} \mathbf{x}(t) - \mathbf{x}(t - \tau)^\top \mathbf{S} \mathbf{x}(t - \tau) \\ & + T f^\top(\mathbf{x}_t) \mathbf{Z} f(\mathbf{x}_t) - \int_{t-T}^t f^\top(\mathbf{x}_s) \mathbf{Z} f(\mathbf{x}_s) ds \\ \leq & \dot{\mathbf{x}}(t)^\top \mathbf{P} \mathbf{x}(t) + \mathbf{x}(t)^\top \mathbf{P} \dot{\mathbf{x}}(t) + \mathbf{x}(t)^\top \mathbf{S} \mathbf{x}(t) - \mathbf{x}(t - \tau)^\top \mathbf{S} \mathbf{x}(t - \tau) \\ & + T f^\top(\mathbf{x}_t) \mathbf{Z} f(\mathbf{x}_t) - \frac{1}{T} (\mathbf{x}(t) - \mathbf{x}(t - T))^\top \mathbf{Z} (\mathbf{x}(t) - \mathbf{x}(t - T)). \end{aligned} \quad (23)$$

Definiendo el vector $\Phi(t, \tau, T)^\top = [\mathbf{x}(t)^\top, \mathbf{x}(t - \tau)^\top, \mathbf{x}(t - T)^\top]$. Sustituyendo (21), (19) y (20) en (23) y tras reordenar términos, obtenemos:

$$\begin{aligned} \dot{V}(\mathbf{x}_t) \leq & \Phi^\top \begin{bmatrix} A_0^\top \mathbf{P} + \mathbf{P} A_0 + \mathbf{S} + T A_0^\top \mathbf{Z} A_0 - \mathbf{Z}/T & \mathbf{P} A_1 + T A_0^\top \mathbf{Z} A_1 & \mathbf{Z}/T \\ A_1^\top \mathbf{P} + T A_1^\top \mathbf{Z} A_0 & -\mathbf{S} + T A_1^\top \mathbf{Z} A_1 & \mathbf{0} \\ \mathbf{Z}/T & 0 & -\mathbf{Z}/T \end{bmatrix} \Phi \\ \dot{V} \leq & \Phi^\top(t, \tau, T) \Gamma \Phi(t, \tau, T) \end{aligned} \quad (24)$$

Si la LMI (19) se cumple, la derivada temporal del funcional de Lyapunov-Krasovski $V(\mathbf{x}_t)$, a lo largo de las trayectorias será negativa, y la dinámica del error convergerá cercana al origen, haciendo que el sistema ciberfísico alcance el patrón de formación deseado en los espacios de trabajo $\mathcal{W}_1$ y $\mathcal{W}_2$. $\square$

Observación 4.2.1. *La condición LMI aquí es conservadora, ya que el último término de $\dot{V}$ puede desarrollarse con desigualdades más ajustadas para obtener una condición LMI mas cercana.*

Observación 4.2.2. *Dado que la condición LMI depende de T pero no explícitamente de τ , si existe solución a (19), la estabilidad se garantiza para cualquier $0 \leq \tau \leq T$.*

Observación 4.2.3. *Es relativamente sencillo extender este resultado a múltiples retardos o a retardos variables con cotas en la derivada temporal de la función de retardo, obteniendo una o varias condiciones LMI análogas a (19).*

Observación 4.2.4. *Existen métodos y herramientas computacionales eficientes para resolver LMI. En este trabajo se utilizó el toolbox YALMIP de Matlab con el optimizador Gurobi [139] para verificar la condición en los experimentos presentados en las secciones siguientes.*

4.3 Simulaciones numéricas

Para la simulación numérica, se desarrolló un código en Python 3.13 utilizando las bibliotecas Matplotlib y NumPy para simular diversas trayectorias y formaciones de robots en dos espacios de trabajo distintos. Se implementaron tres formaciones:

- La primer formación consta de dos robots dispuestos en línea recta (uno por espacio de trabajo, como se muestra en la Figura 4.3)
- La segunda formación consta de cuatro robots dispuestos en línea recta (dos por espacio de trabajo, como se ilustra en la Figura 4.4)
- La tercera formación consta de cuatro robots dispuestos en configuración de rombo (dos por espacio de trabajo, como se muestra en la Figura 4.5).

En cada formación, un robot líder mantiene una trayectoria circular. Las formaciones se evaluaron empleando distintos valores de ganancias (k_c , k_r y k_v), así como diversos valores de retardo en los datos de los robots virtuales. Los errores de simulación se cuantificaron mediante tablas comparativas

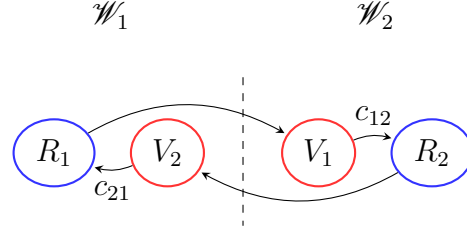


Figura 4.3: Topología de la primera formación: formación en línea horizontal de 2 robots.

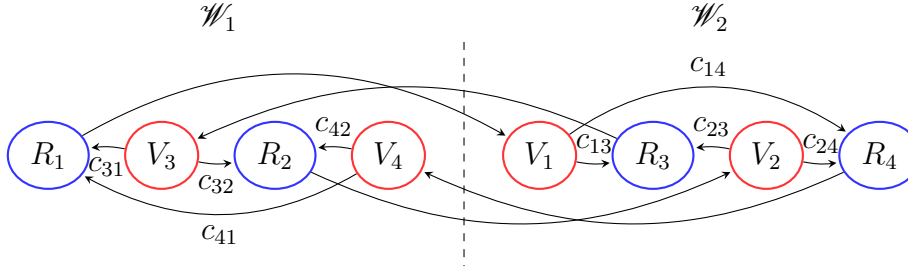


Figura 4.4: Topología de la segunda formación: formación en línea horizontal de 4 robots.

basadas en la norma L_1 (integral de la suma de errores absolutos) y la norma L_2 (integral de la suma de errores al cuadrado).

4.3.1 Control sin retardos en las comunicaciones

Formación en línea horizontal de 2 robots

La Figura 4.6 muestra la simulación numérica del enfoque de control para $n = 2$ y $m = 2$; el conjunto total de robots es $\mathcal{R} = \{R_1, R_2\}$, donde $\mathcal{R}_1 = \{R_1\} \in \mathcal{W}_1$ y $\mathcal{R}_2 = \{R_2\} \in \mathcal{W}_2$, según la topología de formación representada en la Figura 4.3. El robot líder es R_1 . Por tanto, $\mathcal{V}_1 = \{V_2\}$ y $\mathcal{V}_2 = \{V_1\}$, de modo que los miembros de cada espacio de trabajo son $\mathcal{W}_1 = \{R_1, V_2\}$ y $\mathcal{W}_2 = \{V_1, R_2\}$.

Los conjuntos adyacentes para los robots reales se definen como $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_2\}$, $\mathcal{N}_{22}^R = \emptyset$ y $\mathcal{N}_{22}^V = \{V_1\}$, tal como muestra la Figura 4.3. La formación final deseada es una línea horizontal, como si ambos robots reales estuvieran en el mismo espacio. Para lograr este patrón, los vectores relativos

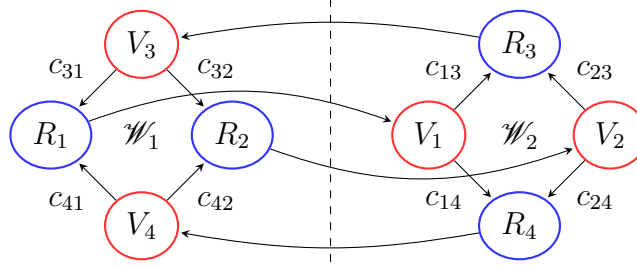


Figura 4.5: Topología de la tercera formación: formación en rombo de 4 robots.

se eligen como $c_{12} = [a, 0]^\top$ con $a = 400$ y, por construcción, $c_{12} = -c_{21}$.

La Figura 4.6a presenta las trayectorias de los robots reales y virtuales utilizando las leyes de control (8)(9)(10) con $k_c = 1$, $k_r = 1$, $k_v = 1$, y la trayectoria deseada del líder R_1 dada por

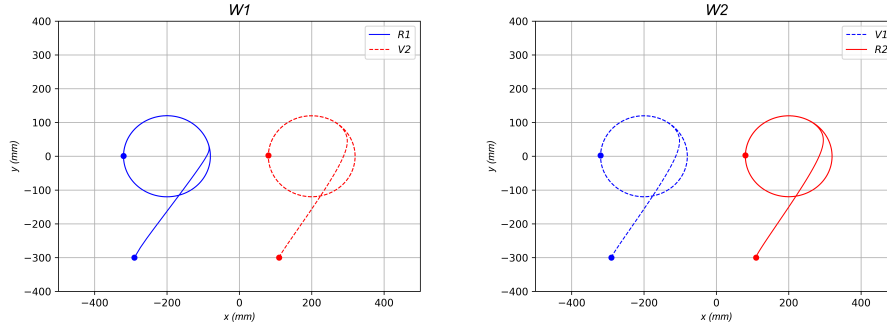
$$\mathbf{z}_d(t) = \left[-200 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top.$$

Todos los robots convergen a la línea horizontal deseada, lo que se aprecia en la convergencia de errores de la Figura 4.6b, y las entradas de control se muestran en la Figura 4.6c.

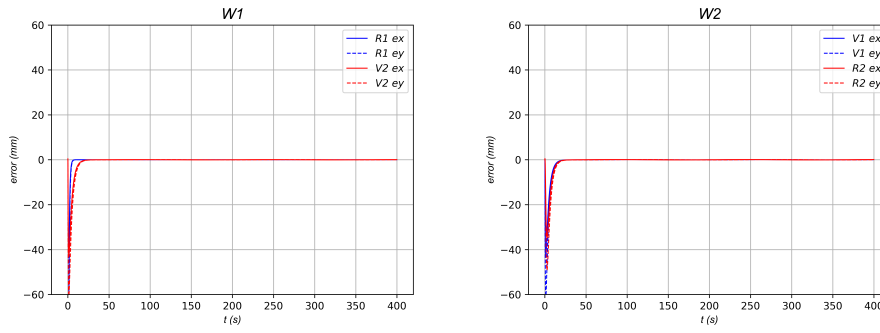
En esta primera formación (Figura 4.3), se llevaron a cabo varias simulaciones variando los valores de k_c , k_r y k_v para evaluar la estabilidad y velocidad de convergencia. Se probaron valores de k iguales a 0.1, 1 y 10 en distintas combinaciones, manteniendo siempre $k_c \geq k_r, k_v$. Para cada simulación se calcularon las normas L_1 y L_2 , sumando los errores absolutos y los errores al cuadrado, respectivamente, e incluyendo los errores de ambos espacios y de robots reales y virtuales. Los resultados se representan en los gráficos de la Figura 4.7 y se resumen en la Tabla 4.1.

Formación en línea horizontal de 4 robots

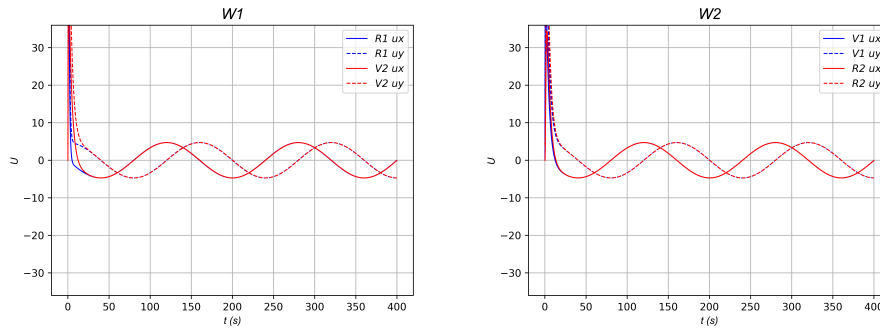
La segunda topología de formación, como se muestra en la Figura 4.8, presenta la simulación numérica del enfoque de control con $n = 4$ y $m = 2$, es decir, el conjunto total de robots es $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, donde $\mathcal{R}_1 = \{R_1, R_2\} \in \mathcal{W}_1$ y $\mathcal{R}_2 = \{R_3, R_4\} \in \mathcal{W}_2$, según la topología de la Figura 4.4. El robot líder es R_1 . Por tanto, $\mathcal{V}_1 = \{V_3, V_4\}$ y $\mathcal{V}_2 = \{V_1, V_2\}$, y los miembros de cada espacio de trabajo son $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ y $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.



(a) Trayectorias de los robots.



(b) Convergencia de errores.



(c) Entradas de control.

Figura 4.6: Simulación numérica para formación en línea horizontal de 2 robots sin retardos.

Prueba	L_1	L_2
$k_c:1, k_r:0.1, k_v:0.1$	18 092.63	8001.94
$k_c:1, k_r:0.1, k_v:1$	8874.45	5404.99
$k_c:1, k_r:1, k_v:0.1$	13 532.95	6714.88
$k_c:1, k_r:1, k_v:1$	4175.66	2313.57
$k_c:10, k_r:0.1, k_v:0.1$	18 064.60	8002.15
$k_c:10, k_r:0.1, k_v:1$	8846.89	5404.71
$k_c:10, k_r:1, k_v:0.1$	13 505.06	6714.54
$k_c:10, k_r:1, k_v:1$	4148.20	2312.31
$k_c:10, k_r:10, k_v:10$	2692.79	1992.75

Tabla 4.1: Normas L_1 y L_2 para diferentes valores de k en la simulación de formación en línea horizontal de 2 robots.

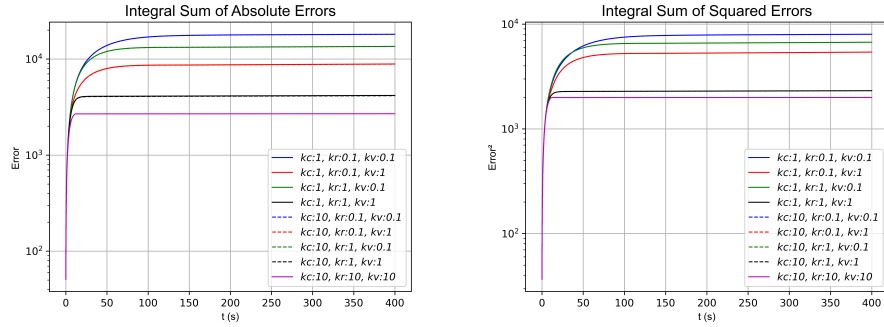


Figura 4.7: Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en línea horizontal de 2 robots.

Los conjuntos adyacentes para los robots reales se definen como $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_3, V_4\}$, $\mathcal{N}_{12}^R = \emptyset$, $\mathcal{N}_{12}^V = \{V_3, V_4\}$, $\mathcal{N}_{23}^R = \emptyset$, $\mathcal{N}_{23}^V = \{V_1, V_2\}$, $\mathcal{N}_{24}^R = \emptyset$ y $\mathcal{N}_{24}^V = \{V_1, V_2\}$, tal como muestra la Figura 4.4. La postura final deseada es una línea horizontal, como si los cuatro robots reales estuvieran en el mismo espacio. Para lograr este patrón de formación, los vectores relativos se eligen como $c_{13} = [a, 0]^\top$, $c_{14} = [3a, 0]^\top$, $c_{23} = [-a, 0]^\top$ y $c_{24} = [a, 0]^\top$ con $a = 242$, y por construcción $c_{31} = -c_{13}$, $c_{41} = -c_{14}$, $c_{32} = -c_{23}$ y $c_{42} = -c_{24}$. La Figura 4.8a muestra las trayectorias de los robots reales y virtuales empleando la ley de control (8)(9)(10) con $k_c = 1$, $k_r = 1$, $k_v = 1$ y la trayectoria deseada del líder R_1 dada por

$$\mathbf{z}_d(t) = \left[-360 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top.$$

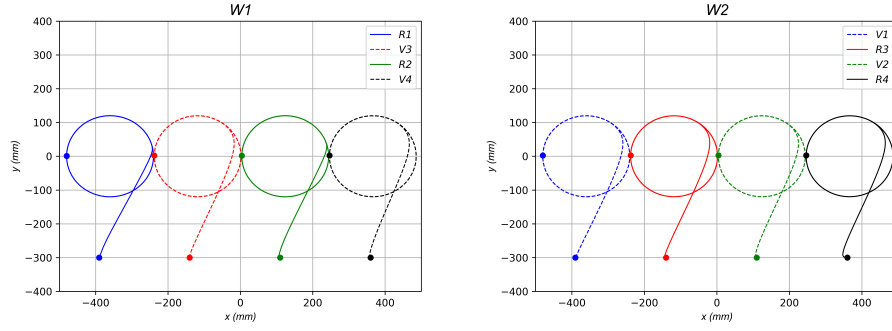
Todos los robots convergen a la formación en línea horizontal deseada, como se observa en la convergencia de errores de la Figura 4.8b, y las entradas de control se muestran en la Figura 4.8c.

En esta segunda formación (Figura 4.4), se realizaron varias simulaciones variando los valores de k_c , k_r y k_v para demostrar la estabilidad y la rapidez de convergencia de la estrategia de control. Se emplearon valores de k de 0.1, 1 y 10 en diversas combinaciones, garantizando siempre que $k_c \geq k_r, k_v$. Para cada simulación se calcularon las normas L_1 y L_2 , sumando los errores absolutos y los errores al cuadrado, respectivamente, e incluyendo los errores de ambos espacios de trabajo y de robots reales y virtuales. Los resultados se representan en los gráficos de la Figura 4.9 y se resumen en la Tabla 4.2.

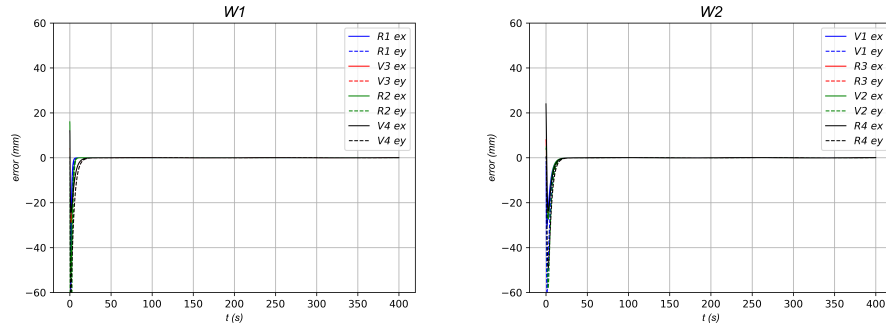
Formación en rombo de 4 robots

La tercera topología de formación para la simulación numérica, mostrada en la Figura 4.10, presenta el enfoque de control con $n = 4$ y $m = 2$; es decir, el conjunto total de robots es $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, donde $\mathcal{R}_1 = \{R_1, R_2\} \in \mathcal{W}_1$ y $\mathcal{R}_2 = \{R_3, R_4\} \in \mathcal{W}_2$, según la topología indicada en la Figura 4.5. El robot líder es R_1 . Por tanto, $\mathcal{V}_1 = \{V_3, V_4\}$ y $\mathcal{V}_2 = \{V_1, V_2\}$, y los miembros de cada espacio de trabajo son $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ y $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.

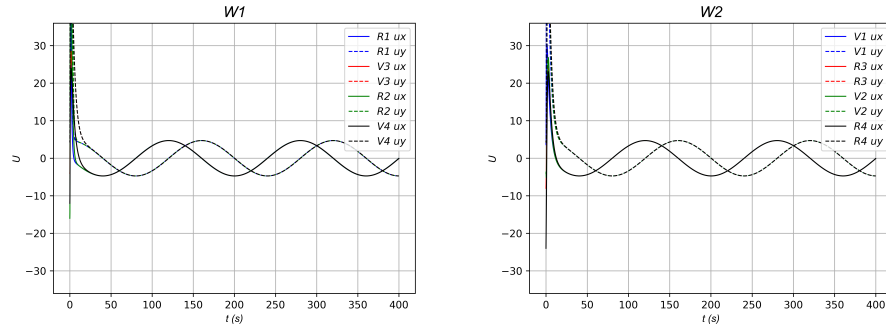
Los conjuntos adyacentes para los robots reales se definen como $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_3, V_4\}$, $\mathcal{N}_{12}^R = \emptyset$, $\mathcal{N}_{12}^V = \{V_3, V_4\}$, $\mathcal{N}_{23}^R = \emptyset$, $\mathcal{N}_{23}^V = \{V_1, V_2\}$, $\mathcal{N}_{24}^R = \emptyset$ y $\mathcal{N}_{24}^V = \{V_1, V_2\}$, tal como ilustra la Figura 4.5. La configuración final deseada es una formación en rombo, como si los cuatro robots reales



(a) Trayectorias.



(b) Errores.



(c) Entradas de control.

Figura 4.8: Simulación numérica de formación en línea horizontal de 4 robots sin retardos.

Prueba	L_1	L_2
$k_c:1, k_r:0.1, k_v:0.1$	33 554.19	10 729.37
$k_c:1, k_r:0.1, k_v:1$	17 981.90	7408.51
$k_c:1, k_r:1, k_v:0.1$	21 512.64	8508.63
$k_c:1, k_r:1, k_v:1$	5518.18	2377.26
$k_c:10, k_r:0.1, k_v:0.1$	33 525.20	10 732.97
$k_c:10, k_r:0.1, k_v:1$	17 953.18	7411.35
$k_c:10, k_r:1, k_v:0.1$	21 484.32	8511.37
$k_c:10, k_r:1, k_v:1$	5490.62	2378.52
$k_c:10, k_r:10, k_v:10$	2523.83	1877.37

Tabla 4.2: Normas L_1 y L_2 para diferentes valores de k en la simulación de formación en línea horizontal de 4 robots.

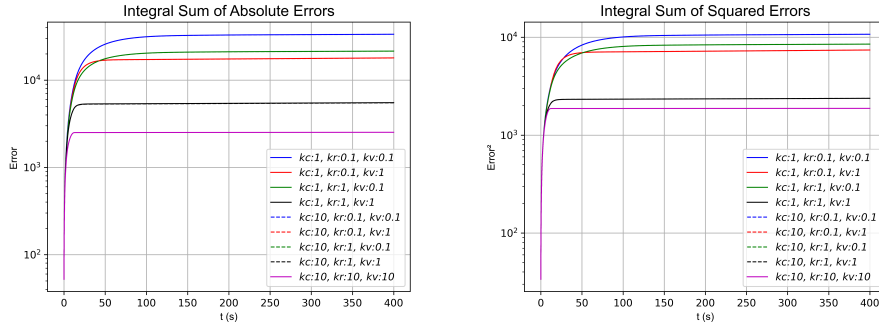


Figura 4.9: Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en línea horizontal de 4 robots.

operasen en un mismo espacio. Para conseguirla, los vectores relativos se escogen como

$$c_{13} = [a, b]^\top, \quad c_{14} = [a, -b]^\top, \quad c_{23} = [-a, b]^\top, \quad c_{24} = [-a, -b]^\top,$$

con $a = 300$, $b = 150$, y por construcción $c_{31} = -c_{13}$, $c_{41} = -c_{14}$, $c_{32} = -c_{23}$ y $c_{42} = -c_{24}$.

La Figura 4.10 muestra las trayectorias de los robots reales y virtuales aplicando la ley de control (8)(9)(10) con ganancias unitarias $k_c = 1$, $k_r = 1$ y $k_v = 1$, y la trayectoria deseada del líder R_1 dada por

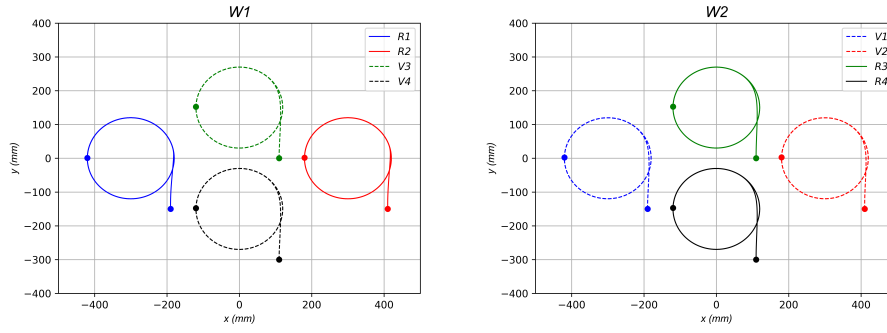
$$\mathbf{z}_d(t) = \left[-300 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top.$$

Se observa que los robots convergen a la formación deseada, como evidencia la convergencia de errores en la Figura 4.10b, y las entradas de control aparecen en la Figura 4.10c.

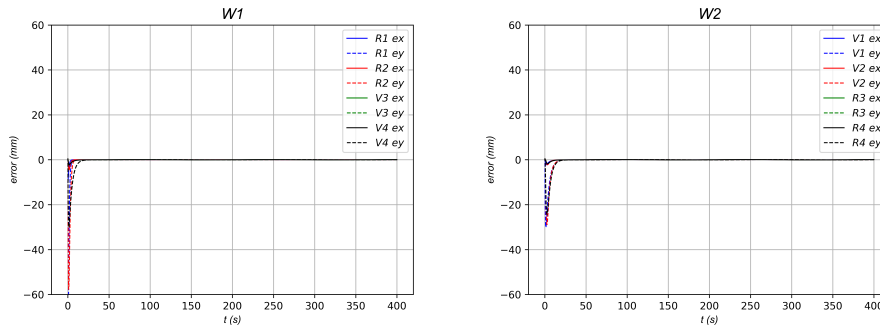
En esta tercera formación (Figura 4.5), se efectuaron múltiples simulaciones variando los parámetros k_c , k_r y k_v para evaluar la estabilidad y velocidad de convergencia. Se probaron valores de k de 0.1, 1 y 10 en diversas combinaciones, asegurando siempre $k_c \geq k_r, k_v$. Para cada simulación se calcularon las normas L_1 y L_2 , sumando los errores absolutos y al cuadrado, respectivamente, incluyendo los errores de ambos espacios y de robots reales y virtuales. Los resultados se representan en los gráficos de la Figura 4.11 y se resumen en la Tabla 4.3.

Comparación

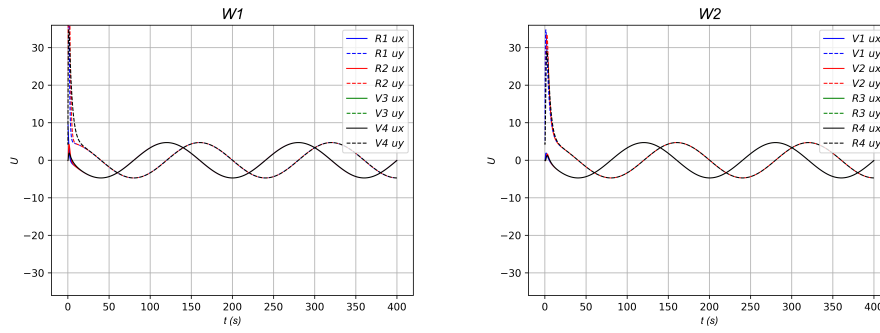
Como se muestra en las simulaciones numéricas y en los resultados presentados en las Tablas 4.1–4.3 para las distintas topologías y formaciones, así como en las simulaciones realizadas con diversos valores de la ganancia k , incrementos en k producen una convergencia más rápida y marcada. Esto se evidencia en las figuras que ilustran las normas L_1 y L_2 para tres configuraciones: dos robots en línea horizontal (Figura 4.7), cuatro robots en línea horizontal (Figura 4.9) y cuatro robots en una formación en rombo (Figura 4.11). Las ganancias más influyentes en el modelo de control son k_r y k_v , ya que valores idénticos de estos parámetros producen tasas de convergencia similares y valores similares de las normas L_1 y L_2 aun cuando k_c varía. Por ejemplo, al analizar la curva negra (línea continua o discontinua) correspondiente a



(a) Trayectorias.



(b) Errores.



(c) Entradas de control.

Figura 4.10: Simulación numérica de formación en rombo de 4 robots sin retardos.

Prueba	L_1	L_2
$k_c:1, k_r:0.1, k_v:0.1$	12 438.28	4772.66
$k_c:1, k_r:0.1, k_v:1$	6533.89	3306.62
$k_c:1, k_r:1, k_v:0.1$	8118.66	3595.80
$k_c:1, k_r:1, k_v:1$	1790.64	799.26
$k_c:10, k_r:0.1, k_v:0.1$	12 416.72	4773.17
$k_c:10, k_r:0.1, k_v:1$	6511.51	3306.86
$k_c:10, k_r:1, k_v:0.1$	8096.59	3595.59
$k_c:10, k_r:1, k_v:1$	1768.38	799.40
$k_c:10, k_r:10, k_v:10$	589.10	467.78

Tabla 4.3: Normas L_1 y L_2 para diferentes valores de k en la simulación de formación en rombo de 4 robots.

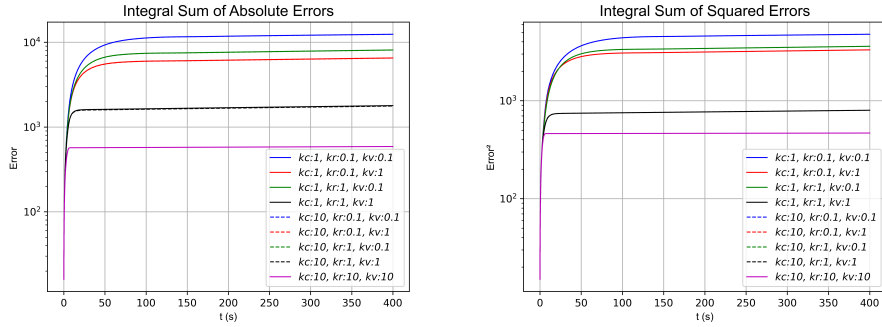


Figura 4.11: Normas L_1 y L_2 con diferentes valores de k en la simulación de formación en rombo de 4 robots.

$k_c = 1$ o 10 con k_r y k_v ambos fijos en 0.1 , ambas simulaciones muestran velocidades de convergencia y comportamiento semejantes; la variación en k_c también conduce a valores similares de las normas L_1 y L_2 .

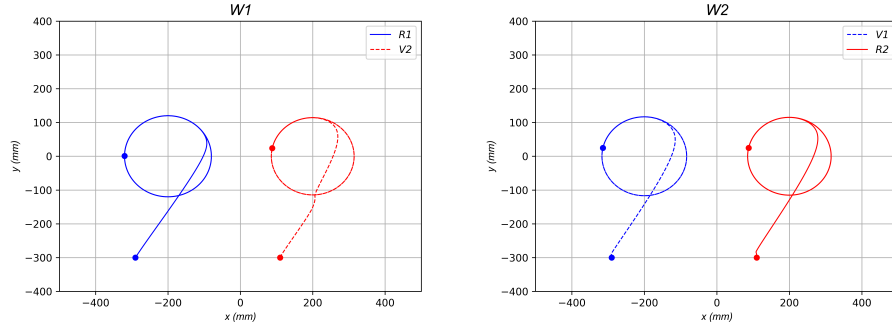
4.3.2 Control con retardos en las comunicaciones

Se llevaron a cabo simulaciones numéricas empleando la estrategia de control descrita anteriormente con valores fijos de ganancias ($k_c = 1$, $k_r = 1$, $k_v = 1$). Se utilizaron las mismas tres topologías y configuraciones de espacios de trabajo presentadas previamente, añadiendo un retardo constante controlado en la información de posición de los robots virtuales, los cuales reciben datos directamente de su robot físico correspondiente en el espacio de trabajo opuesto. Este retardo, denotado por τ , se probó con valores de 0.1 , 0.2 , 0.5 , 1 , 2 , 5 y 10 segundos para evaluar la velocidad de convergencia y la estabilidad de la estrategia de control en presencia de retardos. El valor máximo de retardo simulado se seleccionó con base en el trabajo experimental discutido en las Secciones 5.2 y 5.3. Este estudio se centra en el uso de la plataforma de comunicación MQTT, que exhibe un retardo medio de $0.2s$ y puede alcanzar retardos máximos de hasta $2s$. Además, se midió un retardo de $0.1s$ en el sistema de visión por computadora para la detección de la posición de los robots. En consecuencia, se evaluaron en las simulaciones numéricas varios valores intermedios de retardo dentro de este rango.

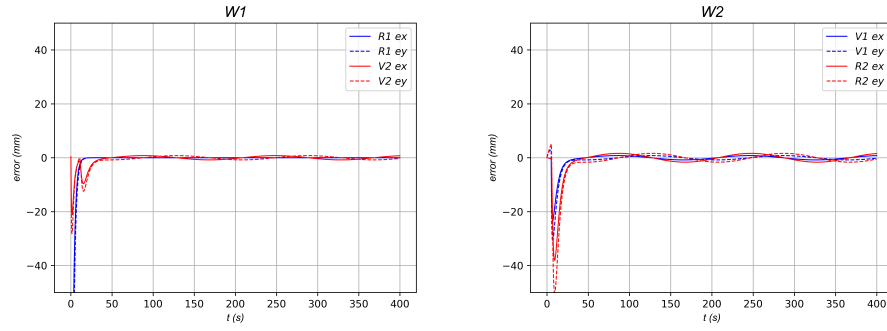
Formación en línea horizontal de 2 robots

La Figura 4.12 muestra un ejemplo de simulación numérica basado en la topología de la Figura 4.3, ahora desarrollado con un retardo de $\tau = 5s$ en los datos de los robots virtuales. Obsérvese que, en comparación con la simulación de la Figura 4.6, se produce un incremento en los valores de error y un desfase en la posición relativa de los robots reales y virtuales.

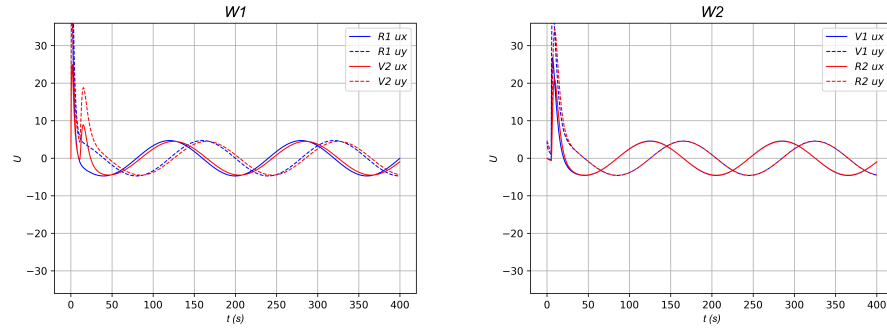
En la primera formación (Figura 4.3), se realizaron varias simulaciones variando el retardo τ para evaluar la estabilidad y la velocidad de convergencia de la estrategia de control. Para cada simulación, se calcularon las normas L_1 y L_2 sumando, respectivamente, los errores absolutos y los errores al cuadrado, considerando los errores tanto de los robots reales como de los virtuales en ambos espacios de trabajo. Los resultados se muestran en los gráficos de la Figura 4.13 y se resumen en la Tabla 4.4.



(a) Trayectorias con retardos.



(b) Convergencia de errores con retardos.



(c) Entradas de control con retardos.

Figura 4.12: Simulación numérica de formación en línea horizontal de 2 robots con retardos.

Prueba	L_1	L_2
$\tau:0.1$ s	4417.46	2402.97
$\tau:0.2$ s	4446.70	2421.57
$\tau:0.5$ s	4534.75	2477.42
$\tau:1$ s	4682.41	2570.56
$\tau:2$ s	4980.26	2756.62
$\tau:5$ s	5881.10	3307.36
$\tau:10$ s	7368.16	4160.87

Tabla 4.4: Normas L_1 y L_2 para diferentes valores de τ en la simulación de formación en línea horizontal de 2 robots.

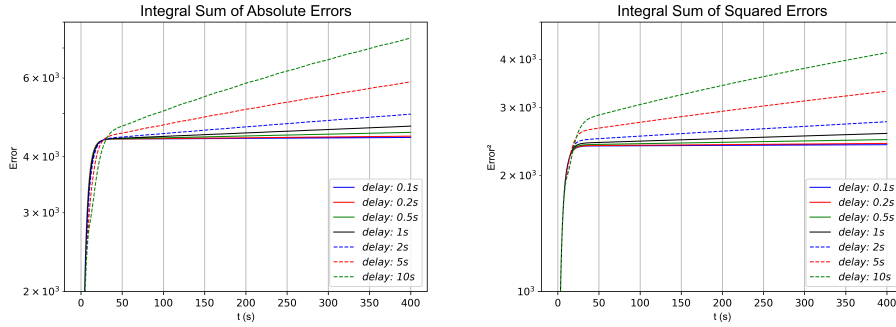


Figura 4.13: Normas L_1 y L_2 con diferentes retardos en la simulación de formación en línea horizontal de 2 robots.

La Figura 4.14 muestra la misma simulación numérica, nuevamente con un retardo de $\tau = 5$ s en los datos de los robots virtuales, basada en la topología de la Figura 4.4. En comparación con la simulación de la Figura 4.8, se aprecia un aumento en los valores de error y un desfase en la posición relativa entre robots reales y virtuales.

Formación en línea horizontal de 4 robots

En la segunda formación (Figura 4.4), se repitieron estas simulaciones variando el retardo τ para evaluar la estabilidad y la velocidad de convergencia de la estrategia de control. Para cada simulación se calcularon las normas L_1 y L_2 ;

Prueba	L_1	L_2
$\tau:0.1$ s	6110.49	2515.99
$\tau:0.2$ s	6179.19	2544.67
$\tau:0.5$ s	6385.92	2630.63
$\tau:1$ s	6731.94	2773.52
$\tau:2$ s	7423.70	3057.82
$\tau:5$ s	9482.80	3899.20
$\tau:10$ s	12 845.24	5230.16

Tabla 4.5: Normas L_1 y L_2 para diferentes valores de τ en la simulación de formación en línea horizontal de 4 robots.

los resultados se presentan en los gráficos de la Figura 4.15 y se resumen en la Tabla 4.5.

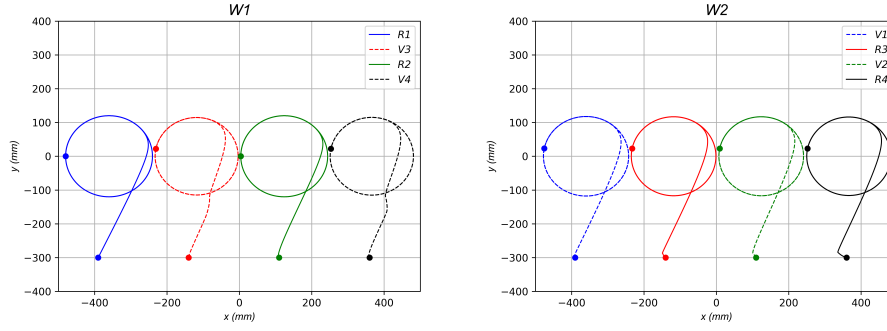
La Figura 4.16 presenta un ejemplo de la misma simulación numérica desarrollada con un retardo de $\tau = 5$ s en los datos de los robots virtuales, basada en la topología de la Figura 4.5. En comparación con la simulación de la Figura 4.10, se observa un aumento en los valores de error y un desfase en la posición relativa entre robots reales y virtuales.

Formación en rombo de 4 robots

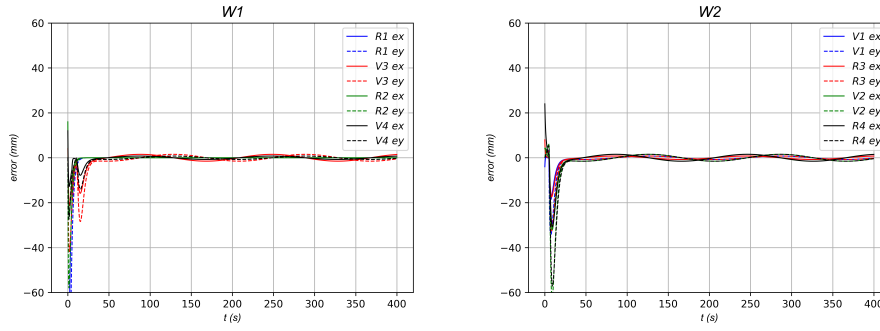
En la tercera formación (Figura 4.5), se llevaron a cabo las mismas simulaciones variando el retardo τ para evaluar la estabilidad y la velocidad de convergencia de la estrategia de control. Para cada simulación se calcularon las normas L_1 y L_2 ; los resultados se representan en los gráficos de la Figura 4.17 y se resumen en la Tabla 4.6.

Comparación

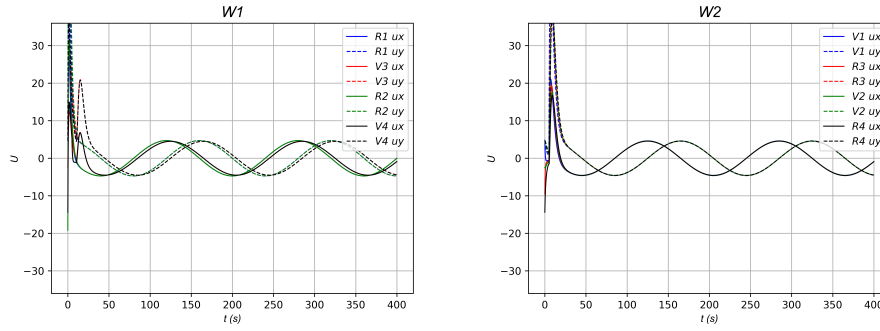
Como se muestra en las simulaciones numéricas y en los resultados presentados en las Tablas 4.4–4.6 para las distintas topologías y formaciones, así como en las simulaciones realizadas con diversos valores de retardo τ , un valor de retardo menor, idealmente cercano a cero, produce una convergencia más rápida y pronunciada. Esto se aprecia en las figuras que ilustran las normas L_1 y L_2 para las tres configuraciones en las Figuras 4.13, 4.15 y 4.17. Todos los ensayos convergen independientemente del valor de retardo τ seleccionado.



(a) Trayectorias con retardos.



(b) Convergencia de errores con retardos.



(c) Entradas de control con retardos.

Figura 4.14: Simulación numérica de formación en línea horizontal de 4 robots con retardos.

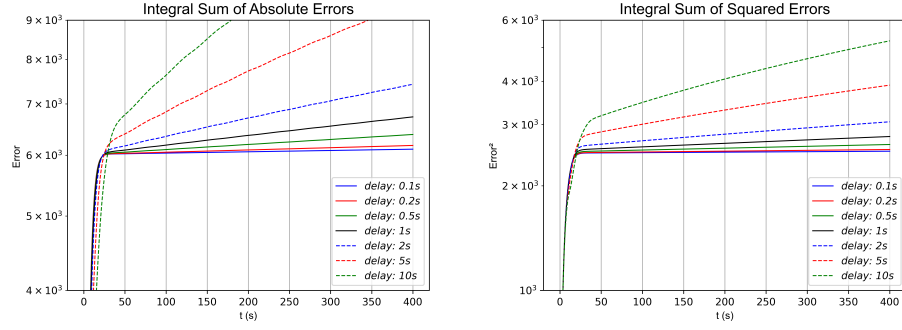
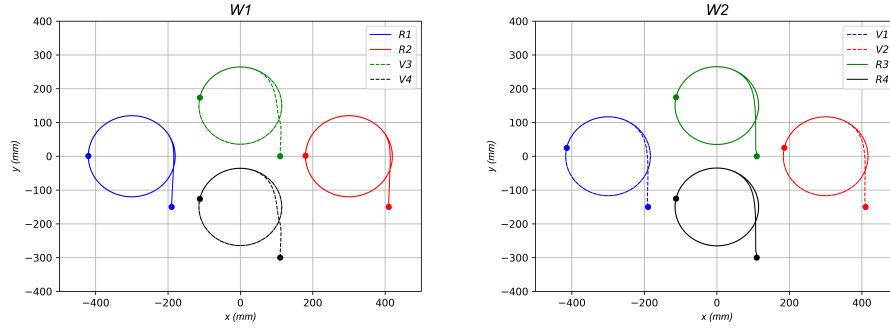


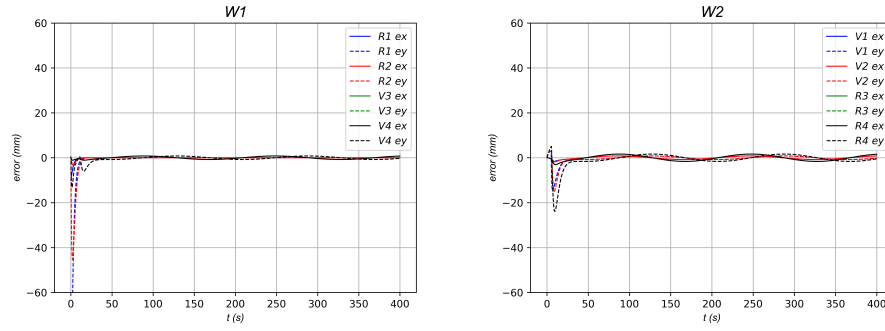
Figura 4.15: Normas L_1 y L_2 con diferentes retardos en la simulación de formación en línea horizontal de 4 robots.

Prueba	L_1	L_2
$\tau:0.1$ s	1964.33	875.06
$\tau:0.2$ s	2022.92	897.39
$\tau:0.5$ s	2199.31	964.38
$\tau:1$ s	2495.02	1075.87
$\tau:2$ s	3091.26	1297.98
$\tau:5$ s	4894.16	1953.13
$\tau:10$ s	7886.94	2991.31

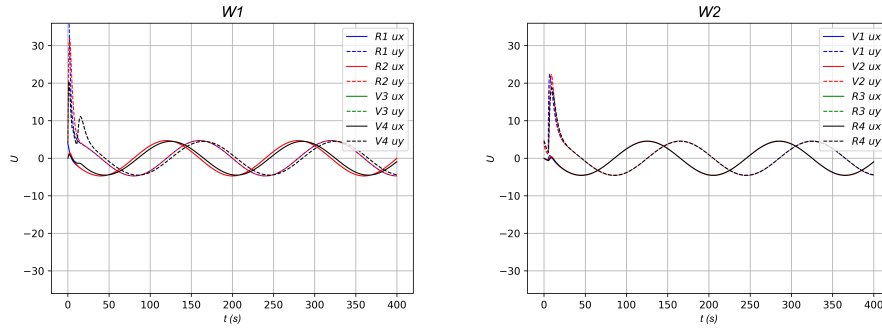
Tabla 4.6: Normas L_1 y L_2 para diferentes valores de τ en la simulación de formación en rombo de 4 robots.



(a) Trayectorias de los robots con retardo.



(b) Convergencia de los errores con retardo.



(c) Entradas de control con retardo.

Figura 4.16: Simulación numérica de la formación en rombo de cuatro robots con un retardo de comunicación de $\tau = 5$ s.

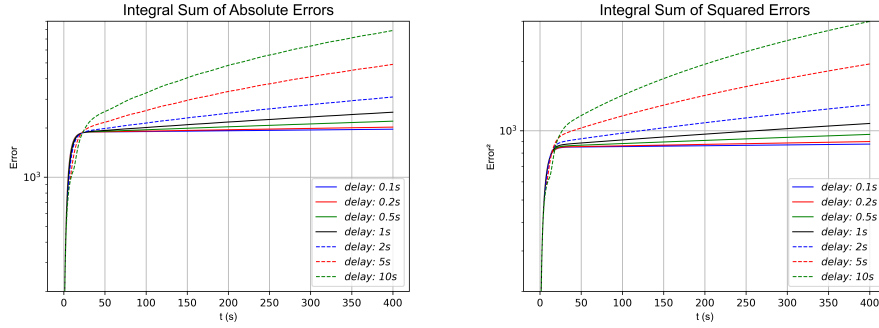


Figura 4.17: Normas L_1 y L_2 con diferentes retardos en la simulación de formación en rombo de 4 robots.

En particular, como se mencionó anteriormente, nuestro interés principal radica en retardos próximos a 0.2 s, el retardo medio de las comunicaciones MQTT y en retardos esporádicos de hasta 2 s. En todas las pruebas realizadas, incluso con valores de retardo superiores a los esperados en experimentos futuros, las simulaciones numéricas de las tres formaciones bajo esta estrategia de control convergieron con éxito. Aunque retardos mayores provocan una tasa de convergencia más lenta y un aumento del error acumulado, todos los experimentos mantuvieron resultados estables, presentando únicamente oscilaciones menores atribuibles al retardo de los datos.

Capítulo 5

Trabajo experimental

La implementación del control para todos los experimentos ciberfísicos se muestra en la Figura 5.1. Los n robots se distribuyen en dos espacios de trabajo diferentes. Cada espacio cuenta con una cámara web (Modelo C920, Logitech) de resolución 1280×720 píxeles, instalada a 400 mm de altura, que cubre un área aproximada de 750×400 mm. Cada cámara está conectada a un PC (Modelo Core i7, Intel). El PC lee las marcas ArUco (módulo OpenCV-contrib, OpenCV.org) a una tasa de muestreo de 0.1s, detectando el ID de cada robot y calculando su posición y orientación mediante OpenCV 4.8.0 (OpenCV.org). Estas posiciones y orientaciones se procesan en el PC correspondiente y se envían por Internet mediante el protocolo IoT MQTT (v3.1.1 y 1.6.9, Eclipse Foundation) usando distintos brokers. La comunicación es bidireccional, conectando ambos espacios de trabajo a través de la red.

La ley de control se implementó en cada PC con Python 3.13, enviando las señales de control a los robots vía comunicación inalámbrica Bluetooth. Este montaje es de bajo costo y escalable; además, la plataforma experimental es modular y emplea componentes comerciales, lo que permite ampliar fácilmente el número de robots y espacios de trabajo. La Figura 5.2 muestra un ejemplo de la implementación del control en la plataforma experimental para la formación en rombo de cuatro robots basada en la topología de la Figura 4.5.

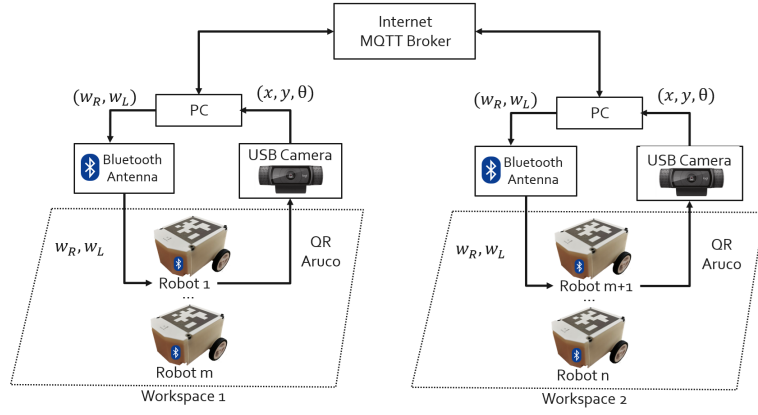
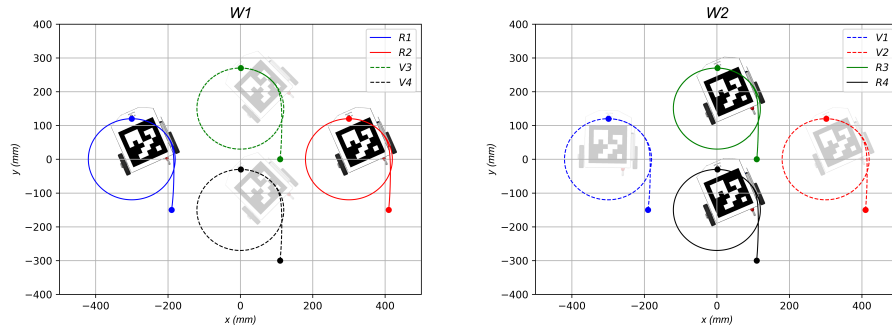
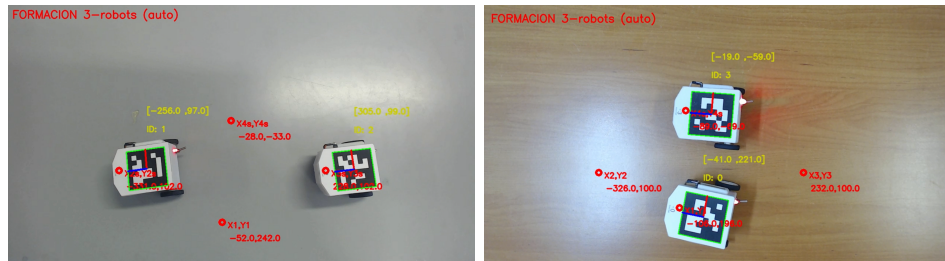


Figura 5.1: Diagrama de los espacios de trabajo para los experimentos ciberfísicos.



(a) Distribución de robots en los espacios.



(b) Robots reales en los espacios de trabajo.

Figura 5.2: Ejemplo de implementación del control en la plataforma experimental para la formación en rombo de cuatro robots.

5.1 Robots tipo unicycle

La estrategia de control se probó en una plataforma de robots de bajo costo compuesta por cuatro robots móviles (Figura 5.3). Cada robot tipo unicycle integra dos motores DC (Micro metal gear motors HP 6 V 1000:1, Pololu) con ruedas de plástico, controlados por un microcontrolador ATmega32U4 (Arduino Pro Micro, Arduino) con comunicación inalámbrica Bluetooth. El chasis se imprimió en 3D y lleva una marca ArUco en la parte superior. El tamaño de cada robot es 100×100 mm, este tamaño se seleccionó durante el desarrollo para proporcionar una plataforma versátil y de fácil implementación, adecuada tanto para entornos académicos como de investigación.

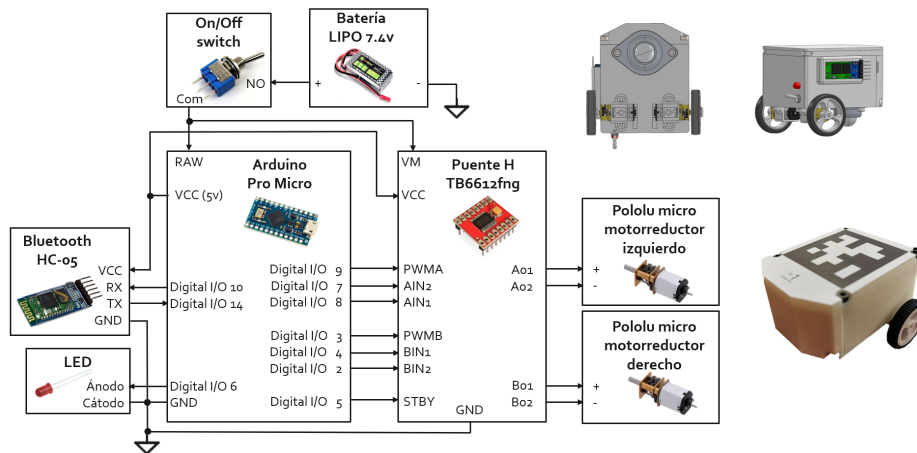


Figura 5.3: Diagrama del sistema electrónico de la plataforma experimental de robots.

A continuación se presenta un desglose detallado de los componentes principales y sus precios aproximados en la Tabla 5.1. El costo total por unidad se estima entre 40 y 70 USD, lo cual es significativamente inferior al de plataformas similares descritas en la literatura (normalmente entre 80 y 150 USD).

Este diseño económico reduce el costo de experimentación para múltiples robots o espacios de trabajo y facilita la replicación y escalabilidad en entornos académicos y de investigación.

Componente	Modelo / Fabricante	Costo (USD)
Microcontrolador	ATmega32U4 (Arduino SA)	5-7
Módulo Bluetooth	HC-05 (Wavesen)	3-5
Driver de motores	TB6612FNG (Toshiba)	3-4
Motores DC (x2)	Pololu 6 V 1000:1 (Pololu)	5-10 c/u
Batería LiPo	7.4 V 2S (Line Power)	10-14
Switch y LED	Genérico (Steren)	1-2
Chasis y piezas 3D	PLA impreso	3-5
Ruedas (x2)	Pololu 32x7 mm (Pololu)	2-6
Rueda esférica	Pololu Ball Caster 1/2" (Pololu)	3-6
Total estimado		40-70

Tabla 5.1: Componentes y costos del robot tipo unicycle.

El modelo cinemático de los robots tipo unicycle mostrado en la Figura 5.3, según la Figura 5.4, viene dado por

$$\dot{x}_i = v_i \cos \theta_i \quad (25)$$

$$\dot{y}_i = v_i \sin \theta_i \quad (26)$$

$$\dot{\theta}_i = w_i \quad (27)$$

con $i = 1, \dots, n$, donde $[x_i, y_i]^\top$ es la coordenada bidimensional del robot y θ_i su ángulo de orientación respecto al eje horizontal. Las variables v_i y w_i son las velocidades de avance y rotación en el punto medio entre las ruedas. Estas velocidades se convierten en las velocidades angulares de las ruedas mediante

$$w_{R_i} = \frac{v_i}{r} + \frac{w_i L}{2r}, \quad w_{L_i} = \frac{v_i}{r} - \frac{w_i L}{2r} \quad (28)$$

donde w_{R_i} y w_{L_i} son las velocidades angulares de la rueda derecha e izquierda, respectivamente, r el radio de las ruedas y L la distancia entre ellas.

El punto frontal se elige como salida de control, tal como muestra la Figura 5.4, con

$$\alpha_i = [x_i + \ell \cos \theta_i, \quad y_i + \ell \sin \theta_i]^\top \quad (29)$$

donde $\ell \neq 0$. Su dinámica viene dada por

$$\dot{\alpha}_i = A_i(\theta_i) \begin{bmatrix} v_i \\ w_i \end{bmatrix} \quad (30)$$

siendo

$$A_i(\theta_i) = \begin{bmatrix} \cos \theta_i & -\ell \sin \theta_i \\ \sin \theta_i & \ell \cos \theta_i \end{bmatrix}$$

Puesto que $\det A_i(\theta_i) = \ell \neq 0$, es posible linealizar la dinámica de α_i mediante el control

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = A_i(\theta_i)^{-1} \mathbf{u}_i \quad (31)$$

con $\mathbf{u}_i \in \mathbb{R}^2$ como control auxiliar. Así, en lazo cerrado (30)-(31) la dinámica se reduce a $\dot{\alpha}_i = \mathbf{u}_i$, eliminando las no linealidades y obteniendo un integrador simple (7). Por tanto, el enfoque de control se extiende a los robots tipo unicycle empleando (31), donde $\mathbf{u}_i$ se define por las Ecuaciones (8)-(10) pero en función de las coordenadas α_i .

Para el caso experimental de la Sección 5.4, los parámetros de los robots son $r = 17 \text{ mm}$, $\ell = 90 \text{ mm}$ y $L = 70 \text{ mm}$, donde r es el radio de las ruedas, ℓ la distancia al punto frontal auxiliar y L la separación entre centros de rueda. Además, con motores Pololu[®] 6V 1000:1 (33RPM máximo) y rueda de $r = 17 \text{ mm}$, el programa limita la velocidad angular de rueda a $W_{\max} = 2.5 \text{ rad/s}$ y $W_{\min} = -2.5 \text{ rad/s}$, aproximadamente el 75% de la velocidad máxima del motor.

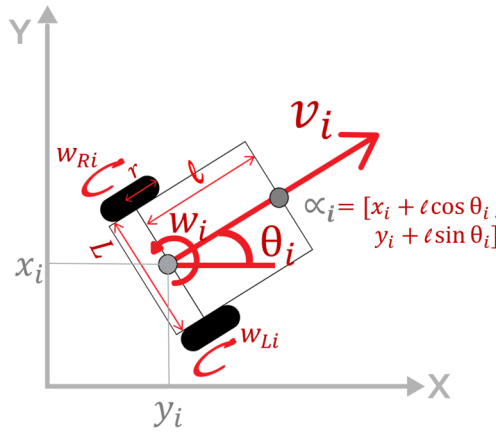


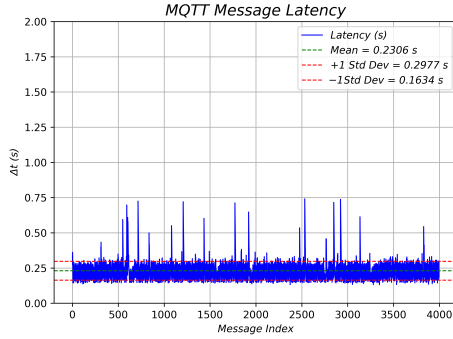
Figura 5.4: Modelo cinemático de un robot tipo unicycle.

5.2 Broker MQTT

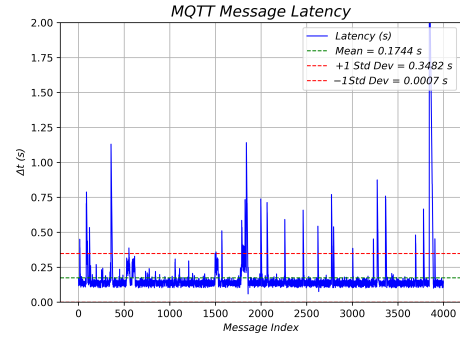
Para la conexión IoT mediante MQTT se emplearon Python 3.13 y la biblioteca `paho.mqtt` para establecer la comunicación entre el ordenador y el servidor MQTT (broker). Un broker MQTT actúa como intermediario de mensajes, facilitando la comunicación entre dispositivos y aplicaciones mediante el protocolo MQTT; gestiona la publicación y suscripción a mensajes y temas. En este experimento se utilizaron dos brokers distintos: un broker comercial público, `test.mosquitto.org`, y un broker personalizado alojado en Google Cloud. El broker `test.mosquitto.org` es un servicio gratuito accesible por el puerto 1883 que no requiere cifrado ni autenticación, por lo que se emplea habitualmente para desarrollo y pruebas MQTT. Adicionalmente, se desarrolló un broker MQTT personalizado en la plataforma Google Cloud. Se creó una instancia de máquina virtual tipo e2-medium con dos CPU virtuales, 4 GB de RAM y Ubuntu como sistema operativo. Se habilitó el puerto 1883 e instalaron los paquetes `mosquitto` y `mosquitto-client` (versiones 1.6.9 y 3.1.1). Tras la configuración, se obtuvo la IP pública del servidor para direccionar los mensajes al IP y puerto correspondientes. Para ambos brokers se realizaron pruebas de latencia y comunicación enviando 4000 paquetes de datos a una tasa de muestreo de 0.05 segundos, midiendo el tiempo de transmisión de datos entre ordenadores. En la Figura 5.5 se muestran los tiempos de latencia del broker `test.mosquitto.org`, con un promedio de 0.23 segundos y desviación estándar de 0.0671 segundos, lo que indica un broker muy estable con latencias máximas ocasionales de 0.75 segundos. En contraste, el broker de Google Cloud presenta una latencia media de 0.1744 segundos y desviación estándar de 0.1738 segundos, siendo menos estable que el broker comercial y con picos esporádicos de hasta 2 segundos.

5.3 Detección de posición con ArUco y visión por computadora

Para la detección de posición de los robots se empleó Python 3.13 junto con la biblioteca OpenCV para detectar códigos QR de tipo ArUco. El algoritmo de detección proporciona las coordenadas x y y , así como el ángulo de orientación θ de cada robot. Se realizaron pruebas de latencia para evaluar la velocidad de procesamiento y posibles demoras al detectar 1, 2 o 4 códigos QR, como se muestra en la Tabla 5.2, observándose que la latencia se mantiene



(a) test.mosquitto.org



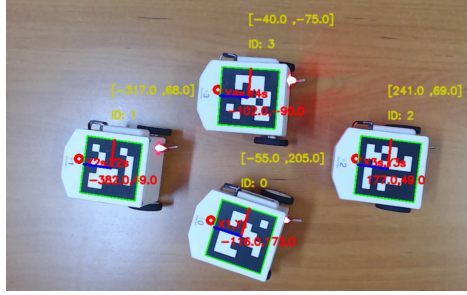
(b) Google Cloud

Figura 5.5: Latencia de los brokers MQTT (media y desviación estándar).

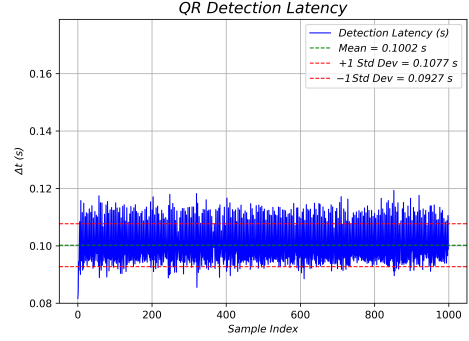
Códigos	Latencia (s)	(s)
1	0.1003	0.0073
2	0.1002	0.0075
4	0.1002	0.0075

Tabla 5.2: Latencia media y desviación estándar en detección de códigos ArUco.

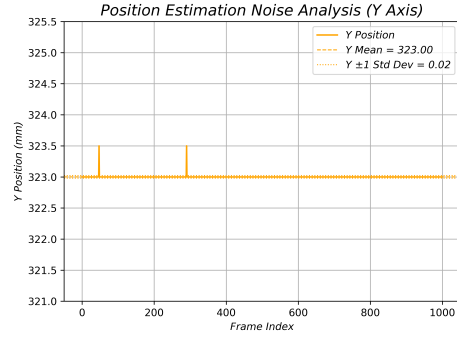
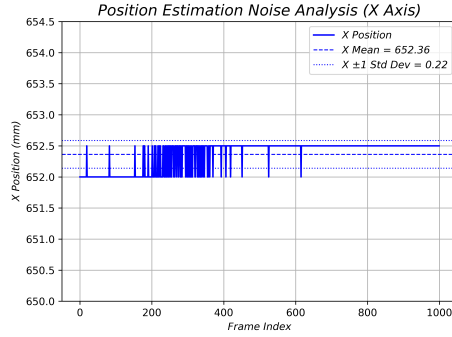
constante en aproximadamente 0.1 segundos con una desviación estándar mínima de 0.0075 segundos. En la Figura 5.6b, los resultados de una prueba de 1000 cuadros capturando cuatro códigos QR confirman que esta latencia se mantiene durante todo el experimento. Adicionalmente, el sistema de visión emplea una cámara Logitech C920[®] con resolución 1280x720px. La cámara se sitúa a 400mm sobre el área de formación de los robots, cubriendo un campo de visión de 750x400 mm, lo que equivale a 0.36 mm² por píxel. Se realizó una prueba de estabilidad midiendo las posiciones x e y de un código QR fijo sin mover ni el código ni la cámara. Como muestra la Figura 5.6c, a partir de 1000 cuadros obtenidos de un QR estacionario, la desviación estándar fue de 0.22 mm, indicando una resolución inferior a 1 mm.



(a) Detección de 4 robots con códigos ArU-co.



(b) Latencia en detección de 4 códigos (1000 frames).



(c) Estabilidad en posición x, y (1000 frames).

Figura 5.6: Latencia y estabilidad del sistema de visión computacional.

5.4 Resultados experimentales

Utilizando la plataforma experimental, se llevaron a cabo varios experimentos con las configuraciones de formación presentadas en la sección de simulación 4.3 (Figuras 4.3-4.5). Sobre la base de las simulaciones numéricas, se seleccionaron ganancias unitarias ($k_c = 1$, $k_r = 1$ y $k_v = 1$) para los experimentos físicos.

Cada experimento se realizó con un tiempo de muestreo distinto (dt), que determina tanto la frecuencia de cálculo como la tasa de envío de datos al broker MQTT. Se probaron $dt = 0.5, 0.2, 0.1, 0.05$ y 0.01 s. Cabe señalar que varios de estos tiempos de muestreo superan la tasa de transmisión media del sistema MQTT (0.2s) y la tasa de muestreo de la cámara (0.1s), lo que

introduce errores y retardos esperados, tal como se discutió en la sección de simulación numérica sobre la robustez del modelo de control.

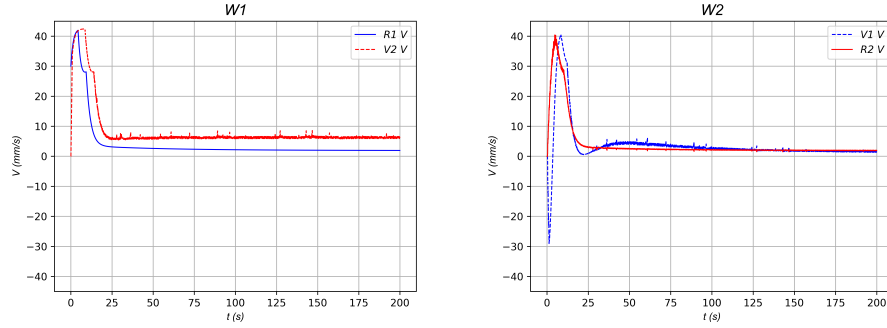
Además de variar dt , los experimentos se llevaron a cabo con dos brokers MQTT diferentes: el broker comercial `test.mosquitto.org` y un broker personalizado en Google Cloud. Los errores de todos los experimentos se cuantificaron mediante tablas comparativas basadas en las normas L_1 (suma de errores absolutos) y L_2 (suma de errores al cuadrado).

Formación de 2 robots en línea horizontal

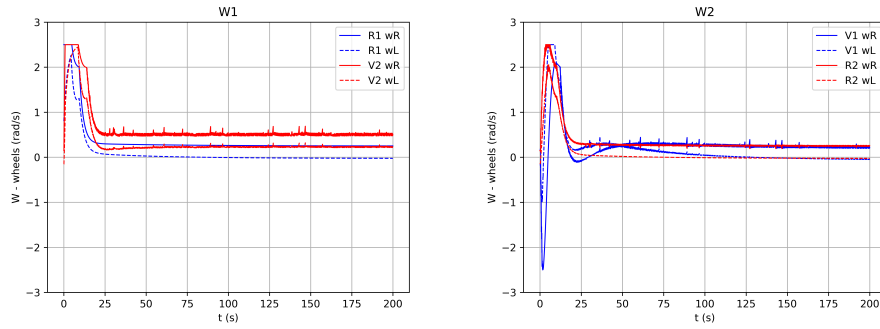
La Figura 5.8 muestra los resultados experimentales con $dt = 0.05$. La trayectoria deseada del robot líder R_1 está dada por

$$\mathbf{z}_d(t) = \left[-200 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top$$

en milímetros, y el broker MQTT seleccionado para este experimento fue el comercial `test.mosquitto.org`. Los robots convergen al patrón de formación deseado, como se aprecia en las trayectorias de la Figura 5.8a y en la convergencia de errores de la Figura 5.8b. Obsérvese que los errores de los robots reales son menores que los de los virtuales en los datos experimentales. Las entradas de control se presentan en la Figura 5.8c, donde se ve que la señal de control está saturada en un valor máximo $U_{\max} = 30$ y mínimo $-U_{\max}$. El desempeño del control se ve afectado por la resolución de la cámara, el tiempo de muestreo y las dinámicas no modeladas (rozamiento del suelo y otros efectos reales).



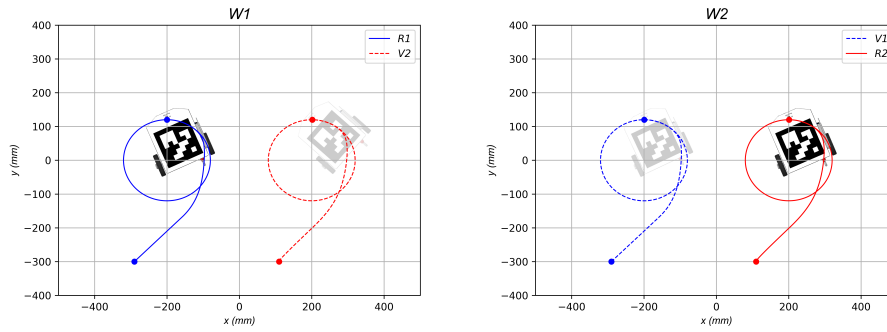
(a) Señal de control V .



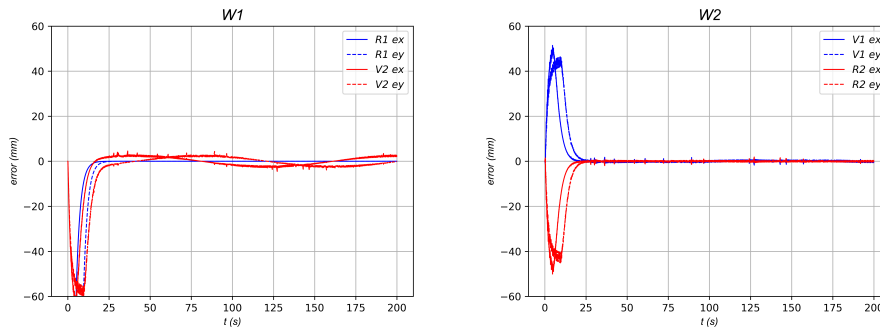
(b) Velocidades angulares W_R y W_L .

Figura 5.7: V , W_R y W_L en el experimento de formación de 2 robots en línea horizontal.

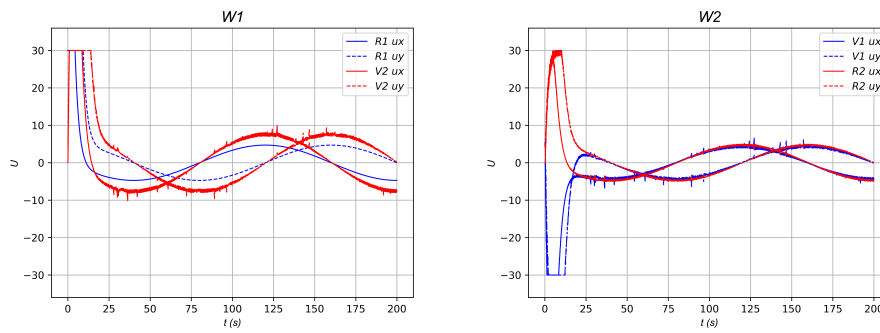
La Figura 5.7 muestra los valores de velocidad longitudinal V con respecto al punto medio entre los ejes de las ruedas (ver Figura 5.7a). Además, la Figura 5.7b presenta las velocidades angulares W_R y W_L de las ruedas derecha e izquierda de cada robot, respectivamente. Obsérvese que dichas velocidades angulares están saturadas en un valor máximo de $W_{\max} = 2.5 \text{ rad/s}$ y un mínimo de $-W_{\max}$, debido a la velocidad máxima alcanzable por los motores físicos de la plataforma experimental.



(a) Trayectorias de los robots.



(b) Convergencia de los errores.



(c) Entradas de control.

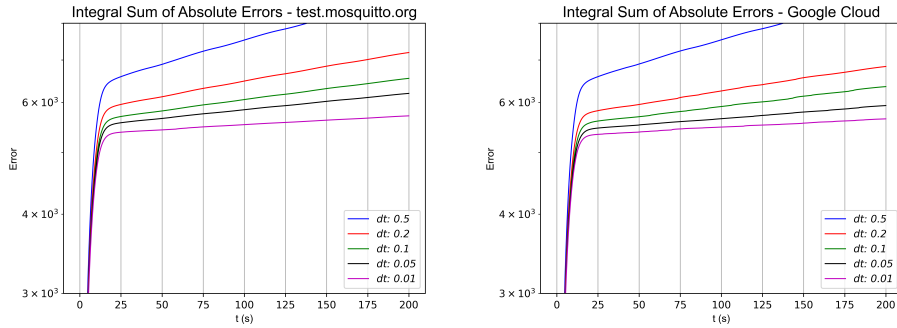
Figura 5.8: Resultados experimentales para la formación de 2 robots en línea horizontal.

En esta primera topología de formación (Figura 4.3), se llevaron a cabo varios experimentos variando el tiempo de muestreo dt para evaluar la estabilidad y

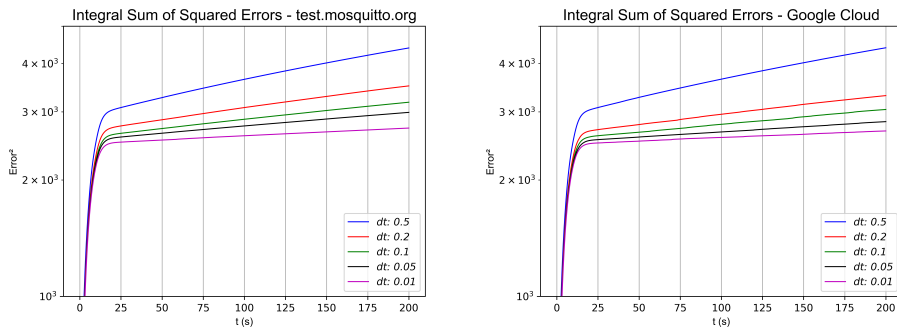
la rapidez de convergencia de la estrategia de control. Se probaron valores de dt de 0.5, 0.2, 0.1, 0.05 y 0.01 segundos, y cada uno se evaluó con dos brokers MQTT diferentes: `test.mosquitto.org` y un broker personalizado en Google Cloud. Para cada simulación se calcularon las normas L_1 y L_2 , sumando los errores absolutos y los errores al cuadrado, respectivamente, para ambos espacios de trabajo y para robots reales y virtuales. Los resultados de cada broker se muestran en los gráficos de la Figura 5.9 y se resumen en la Tabla 5.3. Además, se generaron gráficos del error cuadrático instantáneo (sumando los errores de cada robot real o virtual en ambos espacios) para comparar qué configuraciones de dt y de broker MQTT ofrecen mayor estabilidad y convergencia más rápida, a pesar de los retardos del sistema y otros desafíos inherentes a los experimentos con robots reales. Estos resultados se ilustran en la Figura 5.10.

(a)		
Test	L_1	L_2
$dt = 0.5$	8781.32	4390.35
$dt = 0.2$	7189.94	3502.40
$dt = 0.1$	6546.99	3178.08
$dt = 0.05$	6199.24	2991.69
$dt = 0.01$	5714.09	2724.57
(b)		
Test	L_1	L_2
$dt = 0.5$	8788.56	4395.22
$dt = 0.2$	6835.13	3305.84
$dt = 0.1$	6353.93	3043.51
$dt = 0.05$	5931.30	2831.08
$dt = 0.01$	5651.87	2678.36

Tabla 5.3: Normas L_1 y L_2 en el experimento de 2 robots en línea horizontal. (a) Broker `test.mosquitto.org`; (b) Broker Google Cloud.

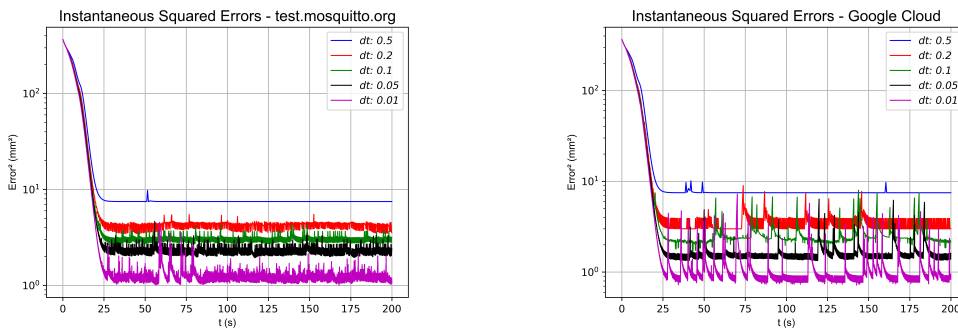


(a) test.mosquitto.org vs. Google Cloud: L_1 .



(b) test.mosquitto.org vs. Google Cloud: L_2 .

Figura 5.9: Normas L_1 y L_2 en el experimento de 2 robots en línea horizontal.



(a) Broker test.mosquitto.org

(b) Broker Google Cloud

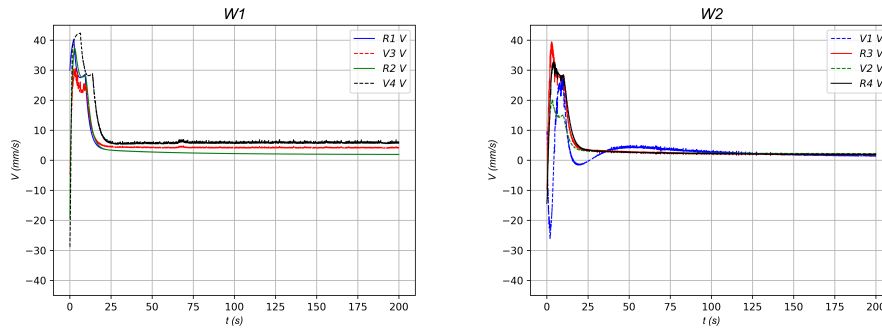
Figura 5.10: Error cuadrático instantáneo en el experimento de 2 robots en línea horizontal.

Formación de 4 robots en línea horizontal

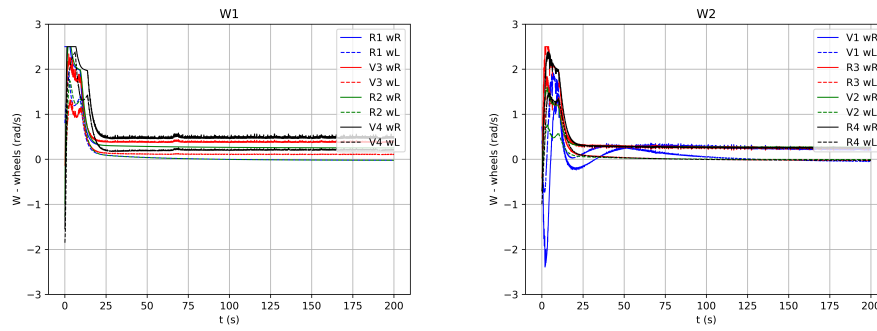
La Figura 5.12 muestra el segundo experimento con $dt = 0.05$; la trayectoria deseada del robot líder R_1 viene dada por

$$\mathbf{z}_d(t) = \left[-360 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top$$

en milímetros, empleando el mismo broker MQTT `test.mosquitto.org`. Los robots convergen al patrón de formación deseado, como se observa en las trayectorias de la Figura 5.12a y en la convergencia de errores de la Figura 5.12b. Nótese que los errores de los robots reales son menores que los de los virtuales en los datos experimentales. Las entradas de control aparecen en la Figura 5.12c, donde la señal de control está saturada en un valor máximo $U_{\max} = 30$ y mínimo $-U_{\max}$. El desempeño del control se ve afectado por la resolución de la cámara, el tiempo de muestreo y las dinámicas no modeladas (rozamiento con el suelo y otros efectos reales).

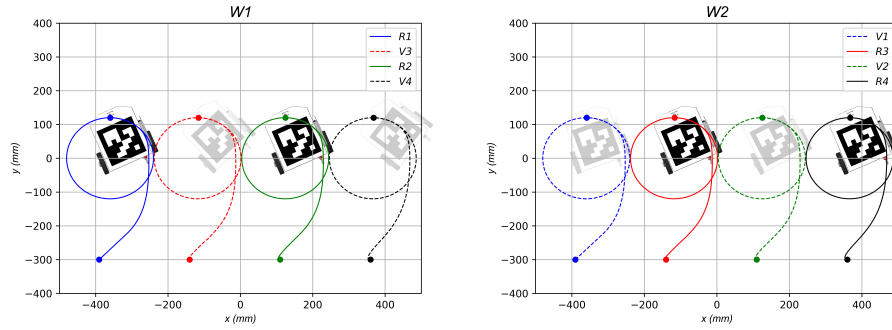


(a) Señal de control V .

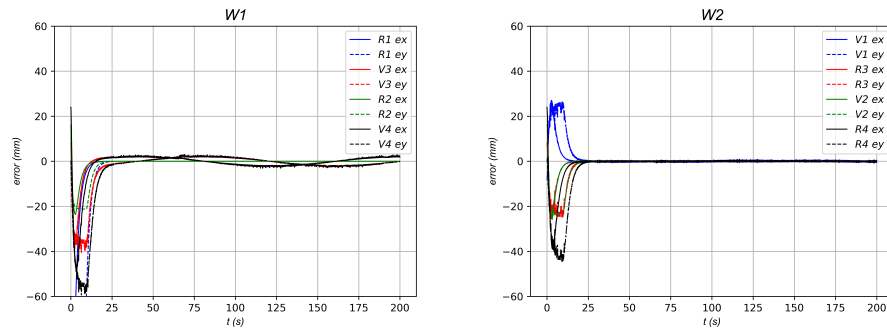


(b) Velocidades angulares W_R y W_L .

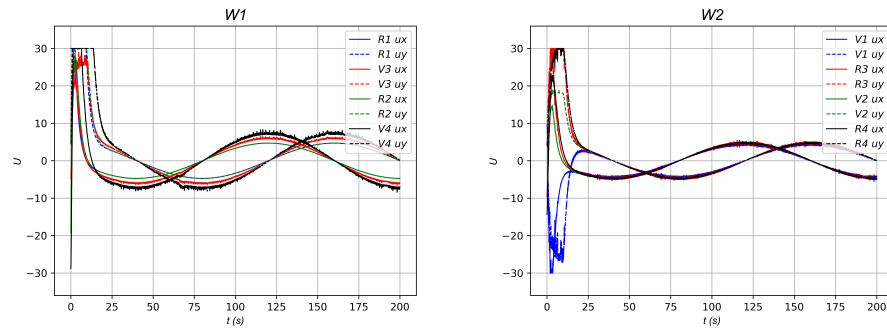
Figura 5.11: V , W_R y W_L en el experimento de formación de 4 robots en línea horizontal.



(a) Trayectorias de los robots.



(b) Convergencia de los errores.



(c) Entradas de control.

Figura 5.12: Resultados experimentales de la formación de 4 robots en línea horizontal.

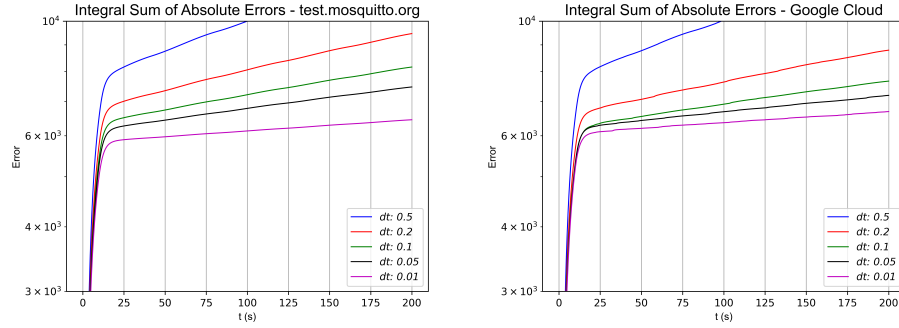
La Figura 5.11 muestra los valores de la velocidad longitudinal V referidos al punto medio entre los ejes de las ruedas (ver Figura 5.11a). Asimismo,

la Figura 5.11b presenta las velocidades angulares W_R y W_L de las ruedas derecha e izquierda de cada robot, respectivamente. Obsérvese que estas velocidades angulares están saturadas en un valor máximo $W_{\max} = 2.5 \text{ rad/s}$ y mínimo $-W_{\max}$, debido a la velocidad máxima que pueden alcanzar los motores físicos de la plataforma experimental.

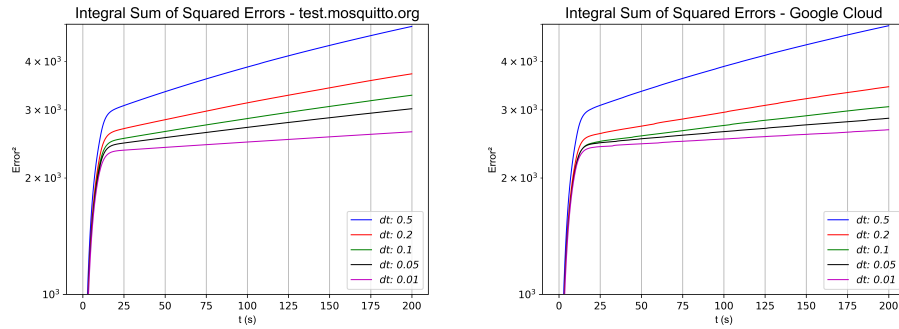
En la segunda topología de formación (Figura 4.4), se realizaron múltiples experimentos variando el tiempo de muestreo dt para evaluar la estabilidad y la rapidez de convergencia de la estrategia de control. En cada uno de estos ensayos también se compararon distintos brokers MQTT (`test.mosquitto.org` y un broker personalizado en Google Cloud). Para cada simulación se calcularon las normas L_1 y L_2 sumando los errores absolutos y los errores al cuadrado, respectivamente, considerando los fallos de robots reales y virtuales en ambos espacios de trabajo. Los resultados de estas pruebas se muestran en los gráficos de la Figura 5.13 y se resumen en la Tabla 5.4.

(a)		
Test	L_1	L_2
$dt = 0.5$	12 502.65	4 932.40
$dt = 0.2$	9 459.63	3 719.16
$dt = 0.1$	8 150.74	3 274.36
$dt = 0.05$	7 458.17	3 021.21
$dt = 0.01$	6 444.53	2 633.53
(b)		
Test	L_1	L_2
$dt = 0.5$	12 558.79	4 952.44
$dt = 0.2$	8 786.86	3 444.10
$dt = 0.1$	7 654.73	3 057.90
$dt = 0.05$	7 184.19	2 854.59
$dt = 0.01$	6 683.78	2 665.83

Tabla 5.4: Normas L_1 y L_2 en el experimento de formación de 4 robots en línea horizontal. (a) Broker `test.mosquitto.org`. (b) Broker Google Cloud.



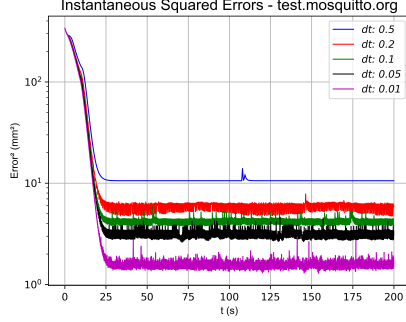
(a) `test.mosquitto.org` vs. Google Cloud: L_1 .



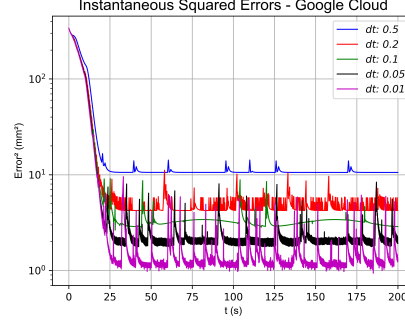
(b) `test.mosquitto.org` vs. Google Cloud: L_2 .

Figura 5.13: Normas L_1 y L_2 en el experimento de formación de 4 robots en línea horizontal.

Adicionalmente, se generaron gráficos del error cuadrático instantáneo, sumando en cada instante los errores de todos los robots reales y virtuales, para comparar qué configuraciones de dt y qué broker MQTT ofrecen mayor estabilidad y convergencia más rápida, pese a los retardos de comunicación y otros desafíos de los experimentos físicos. Estos resultados aparecen en la Figura 5.14.



(a) Broker `test.mosquitto.org`



(b) Broker Google Cloud

Figura 5.14: Error cuadrático instantáneo en el experimento de formación de 4 robots en línea horizontal.

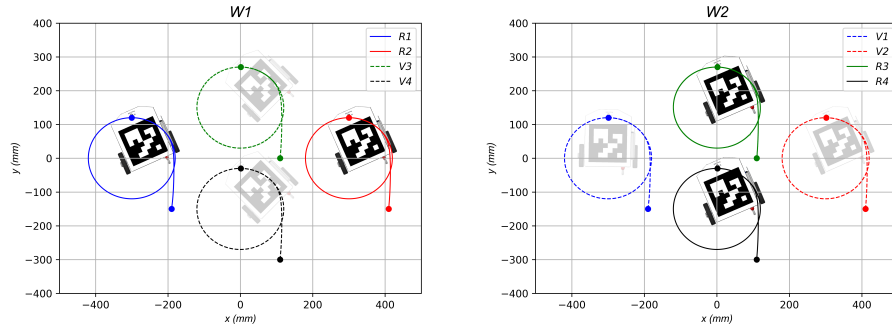
Formación en rombo de 4 robots

La Figura 5.15 presenta los resultados experimentales de la tercer topología con $dt = 0.05$. La trayectoria deseada del robot líder R_1 viene dada por

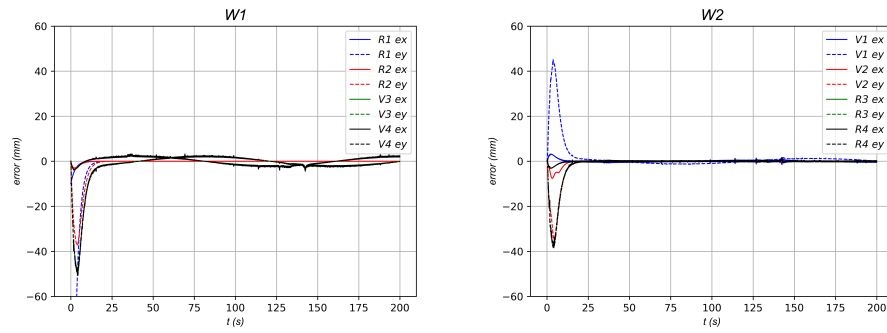
$$\mathbf{z}_d(t) = \left[-300 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top$$

(en mm), utilizando el broker MQTT `test.mosquitto.org`. Se observa que los robots convergen al patrón de formación deseado: en la subfigura 5.15a aparecen las trayectorias y en la subfigura 5.15b la convergencia de los errores. Cabe destacar que los errores de los robots reales son menores que los de los virtuales. Las entradas de control, saturadas en $\pm U_{\max} = 30$, se muestran en la subfigura 5.15c. El desempeño del control se ve influenciado por la resolución de la cámara, el tiempo de muestreo y las dinámicas no modeladas (rozamiento del suelo, etc.).

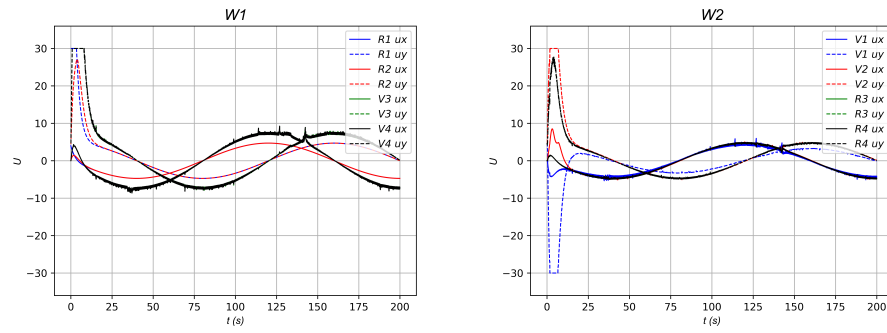
La Figura 5.16 muestra (a) la velocidad longitudinal V respecto al punto medio entre los ejes de las ruedas y (b) las velocidades angulares W_R y W_L de las ruedas derecha e izquierda de cada robot. Ambas señales están saturadas en $\pm W_{\max} = 2.5 \text{ rad/s}$, acorde con la velocidad máxima de los motores físicos.



(a) Trayectorias de los robots.

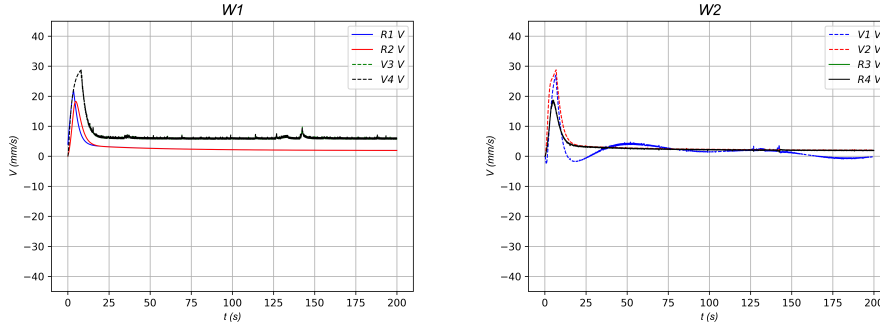


(b) Convergencia de los errores.

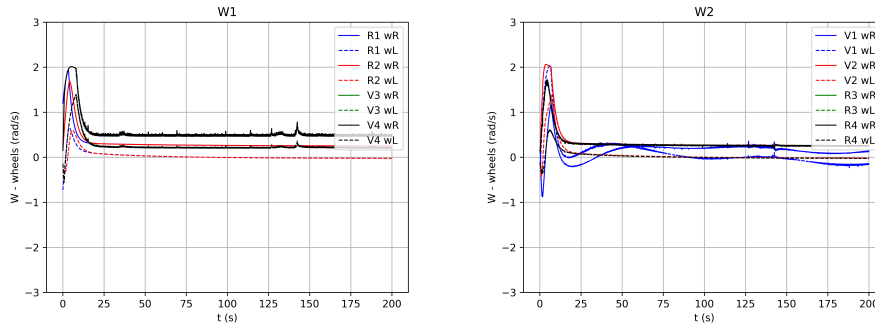


(c) Entradas de control.

Figura 5.15: Resultados experimentales para la formación en rombo de 4 robots en dos espacios.



(a) Señal de control V .



(b) Velocidades angulares W_R y W_L .

Figura 5.16: V , W_R y W_L en el experimento de formación en rombo de 4 robots.

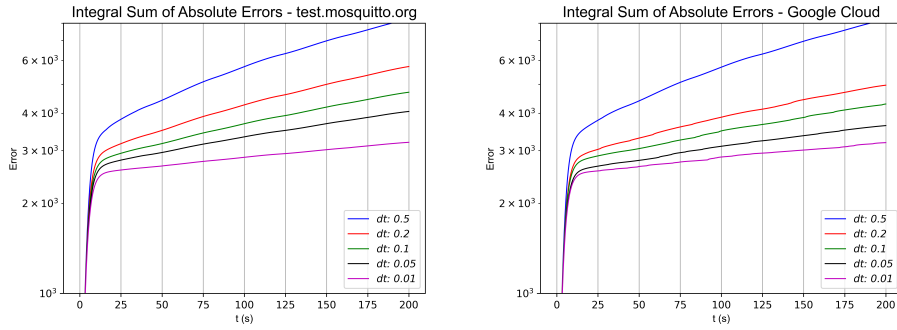
En la tercera topología de formación (Figura 4.5), se llevaron a cabo múltiples experimentos variando el tiempo de muestreo dt para evaluar la estabilidad y la velocidad de convergencia de la estrategia de control. Asimismo, se compararon diferentes brokers MQTT (test.mosquitto.org y uno alojado en Google Cloud) en cada conjunto de pruebas. Para cada experimento se calcularon las normas L_1 y L_2 , sumando los errores absolutos y los errores al cuadrado de todos los robots (reales y virtuales) en ambos espacios de trabajo. Los resultados de estas comparativas se presentan en la Figura 5.17 y se resumen en la Tabla 5.5.

(a)		
Test	L_1	L_2
$dt = 0.5$ s	8 238.39	3 393.61
$dt = 0.2$ s	5 728.74	2 285.73
$dt = 0.1$ s	4 700.82	1 892.68
$dt = 0.05$ s	4 054.56	1 623.58
$dt = 0.01$ s	3 197.88	1 260.00

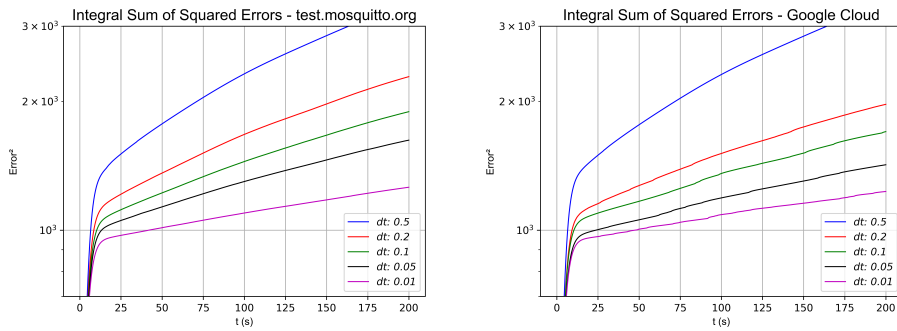
(b)		
Test	L_1	L_2
$dt = 0.5$ s	8 216.36	3 385.04
$dt = 0.2$ s	4 962.72	1 969.13
$dt = 0.1$ s	4 294.15	1 700.06
$dt = 0.05$ s	3 636.71	1 422.18
$dt = 0.01$ s	3 189.84	1 231.52

Tabla 5.5: Normas L_1 y L_2 en el experimento en rombo de 4 robots. (a) Broker `test.mosquitto.org`. (b) Broker Google Cloud.

Además, se generaron gráficos del error cuadrático instantáneo, la suma en cada instante de los cuadrados de los errores de todos los agentes, para comparar qué configuraciones de dt y de broker MQTT ofrecen mayor estabilidad y convergencia más rápida, pese a los retardos y las incertidumbres del experimento real. Estos resultados aparecen en la Figura 5.18.

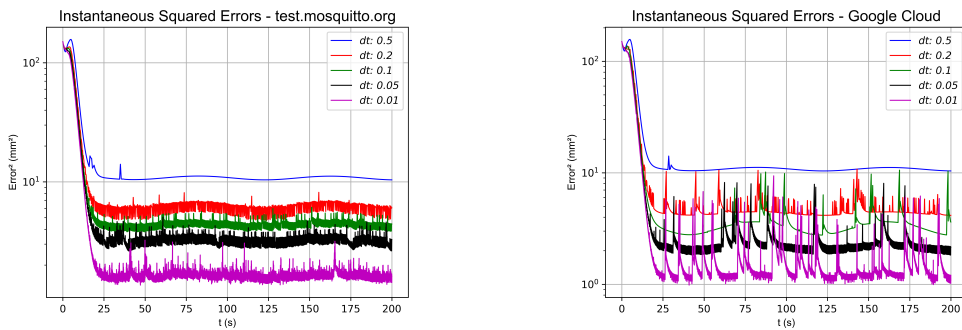


(a) test.mosquitto.org vs. Google Cloud: L_1 .



(b) test.mosquitto.org vs. Google Cloud: L_2 .

Figura 5.17: Normas L_1 y L_2 en el experimento en rombo de 4 robots.



(a) test.mosquitto.org

(b) Google Cloud

Figura 5.18: Error cuadrático instantáneo en el experimento en rombo de 4 robots.

Comparación

Como se muestra en las simulaciones numéricas y en los resultados presentados en las Tablas 5.3 - 5.5 para las distintas topologías y formaciones, así como en los experimentos realizados con varios tiempos de muestreo dt y diferentes brokers MQTT, un dt menor, idealmente próximo a cero, aun con más errores de comunicación, produce una convergencia más rápida y pronunciada. Esto se evidencia en las figuras que ilustran las normas L_1 y L_2 para las tres configuraciones en las Figuras 5.9, 5.13 y 5.17.

Además, el análisis de las tablas de normas L_1 y L_2 de los tres experimentos no revela diferencias significativas entre el broker comercial `test.mosquitto.org` y el broker de Google Cloud. En todos los casos, el broker de Google Cloud mostró valores de L_1 y L_2 ligeramente inferiores, lo cual se relaciona directamente con las latencias medidas en la Sección 5.2: `test.mosquitto.org` tuvo una latencia media de 0.23 s, mientras que el broker de Google Cloud presentó 0.17 s (Figura 5.5).

Es importante señalar que, aunque el broker de Google Cloud exhibe mayor desviación típica y ocasionalmente retrasos de 0.5, 1 y 2 s, la estrategia de control sigue siendo lo bastante robusta para beneficiarse de su menor latencia media y sus tiempos de muestreo más rápidos. Las Figuras 5.10, 5.14 y 5.18 muestran el error cuadrático instantáneo para ambos brokers en las tres topologías: con `test.mosquitto.org` el error es relativamente constante con picos puntuales, mientras que con Google Cloud el error medio es menor pero más errático. Sin embargo, al integrar los errores para calcular las normas L_1 y L_2 , los valores resultan menores con Google Cloud.

Respecto a la velocidad de convergencia, los resultados de simulación (Tablas 4.1 - 4.3) muestran que ganancias de control mayores (k_c, k_r, k_v) aceleran la convergencia. Otro factor crítico es la latencia de la comunicación MQTT y los retardos del sistema de visión por computadora: simulaciones con distintos valores de τ (Tablas 4.4 - 4.6) confirman que un τ menor mejora notablemente la velocidad de convergencia. Por último, la velocidad máxima de los robots limitada a $W_{\max} = 2.5$ rad/s según la Sección 5.1 y las posiciones iniciales (cuanto más alejadas, mayor tiempo de convergencia) también condicionan el desempeño.

Un detalle adicional es que los errores medidos en los robots reales son menores que en los virtuales, debido a que estos últimos incorporan la latencia de transmisión de datos, mientras que los reales solo sufren la demora del sistema

de visión. Además, en el espacio $\mathcal{W}_1$ los robots virtuales acumulan doble retardo: primero al recibir datos de $\mathcal{W}_2$ y luego al retransmitir la posición relativa de los reales de $\mathcal{W}_2$ de vuelta a $\mathcal{W}_1$, lo que genera dos retardos consecutivos en el ciclo experimental.

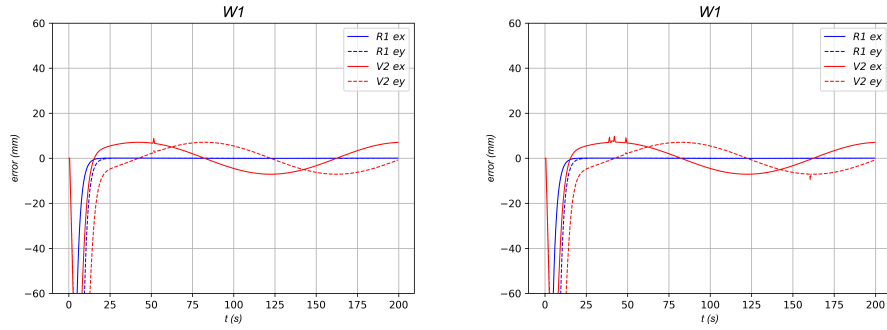
Resultados experimentos completos

Los resultados completos de los experimentos realizados pueden observarse en las siguientes gráficas, donde podemos observar una comparativa de los diferentes brokers MQTT: `test.mosquitto.org` y un broker personalizado en Google Cloud. Se presenta la comparativa con diferentes tasas de muestreo (dt), que determina tanto la frecuencia de cálculo como la tasa de envío de datos al broker MQTT: $dt = 0.5, 0.2, 0.1, 0.05$ y 0.01 segundos. Las ganancias de estos experimentos se encuentran con ganancias unitarias ($k_c = 1, k_r = 1$ y $k_v = 1$).

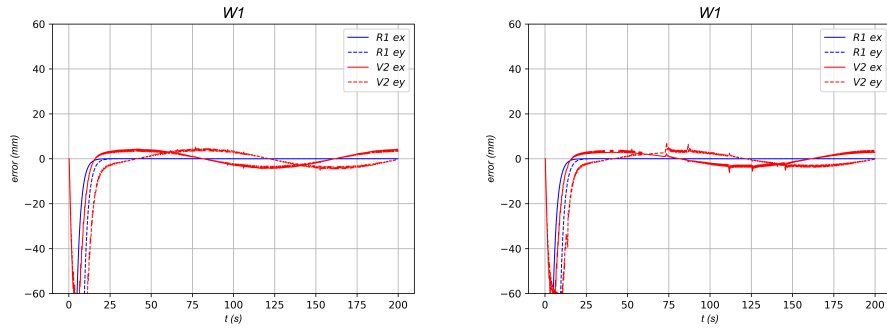
De la primera formación de 2 robots en línea horizontal podemos observar el error en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.19 y 5.20, el error en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.21 y 5.22, ganancia de control en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.23 y 5.24, y ganancia de control en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.25 y 5.26.

La segunda formación de 4 robots en línea horizontal podemos observar el error en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.27 y 5.28, el error en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.27 y 5.28, ganancia de control en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.31 y 5.32, y ganancia de control en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.33 y 5.34.

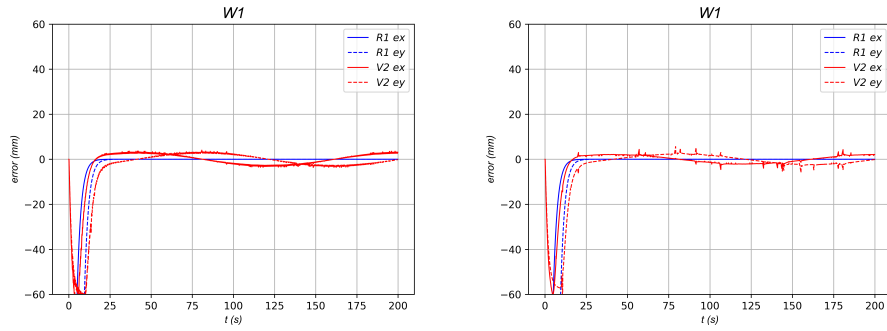
Finalmente la tercera formación de 4 robots en rombo, podemos observar el error en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.35 y 5.36, el error en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.35 y 5.36, ganancia de control en espacio de trabajo $\mathcal{W}_1$ en la Figura 5.39 y 5.40 y ganancia de control en espacio de trabajo $\mathcal{W}_2$ en la Figura 5.41 y 5.42.



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

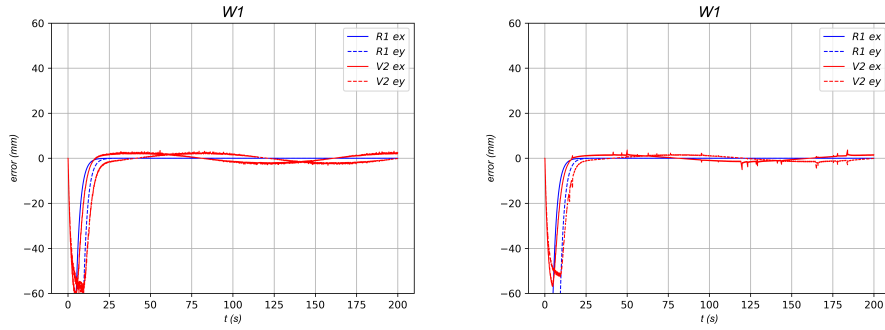


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

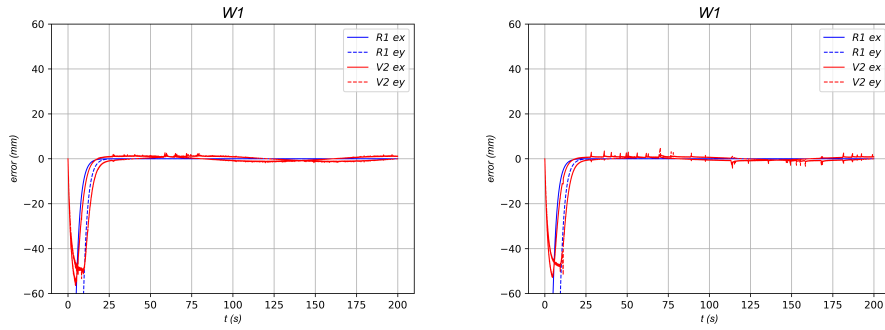


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.19: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).

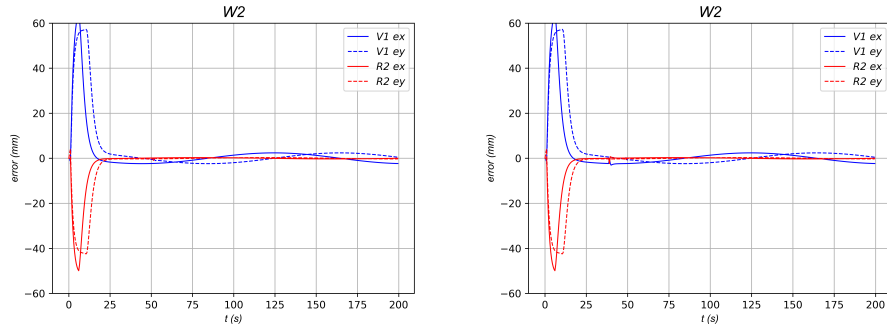


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

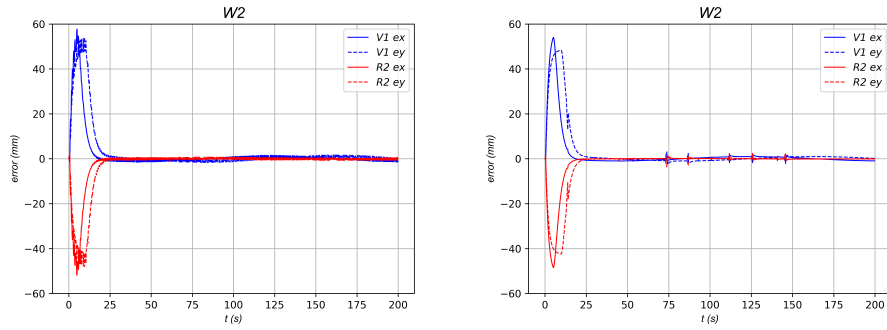


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

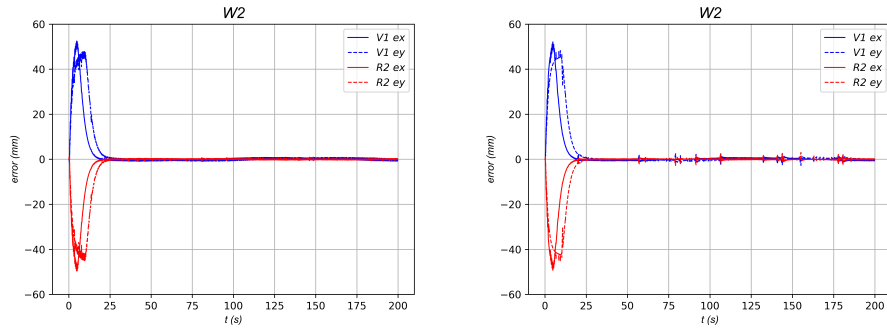
Figura 5.20: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

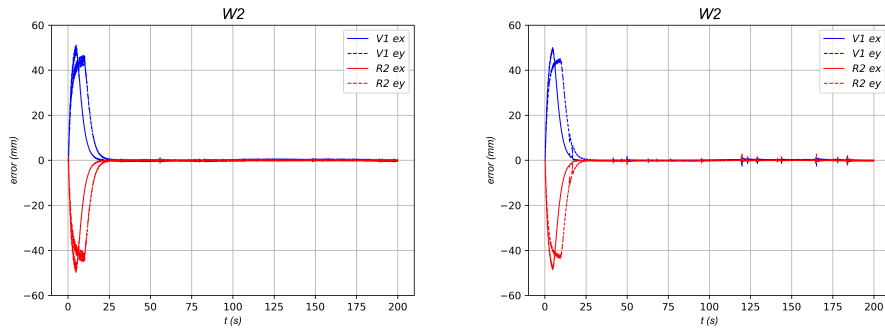


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

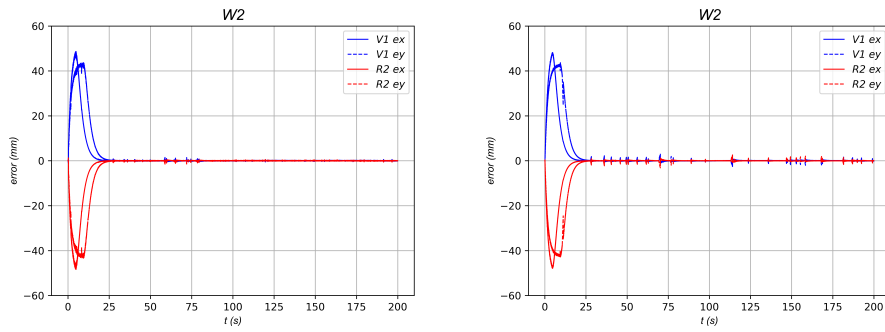


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.21: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).

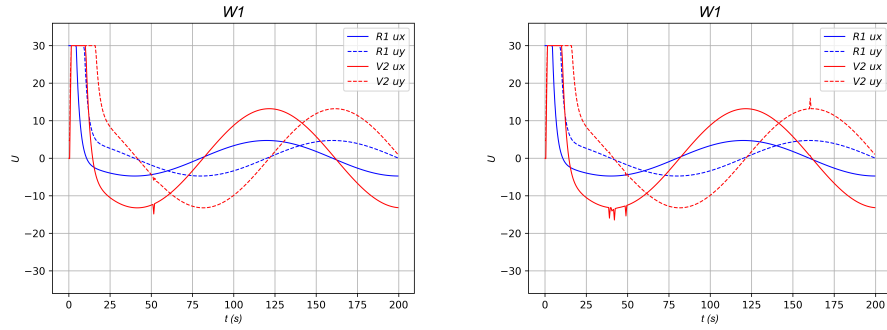


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

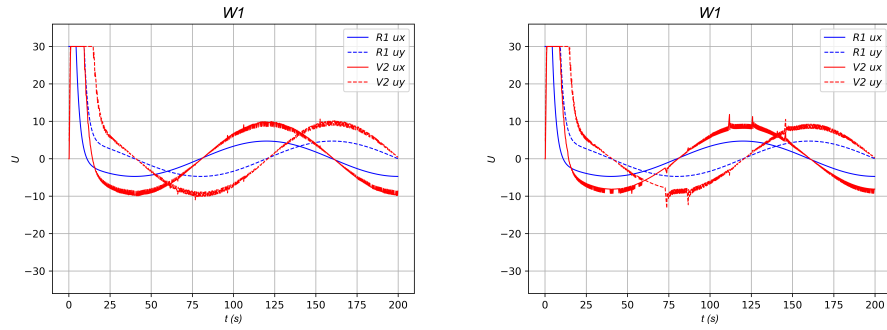


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

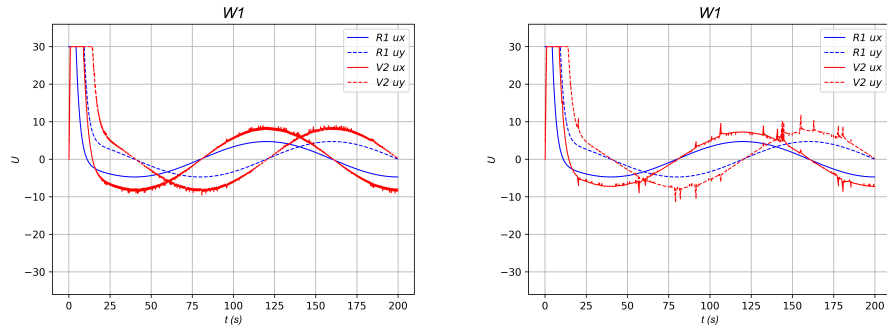
Figura 5.22: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

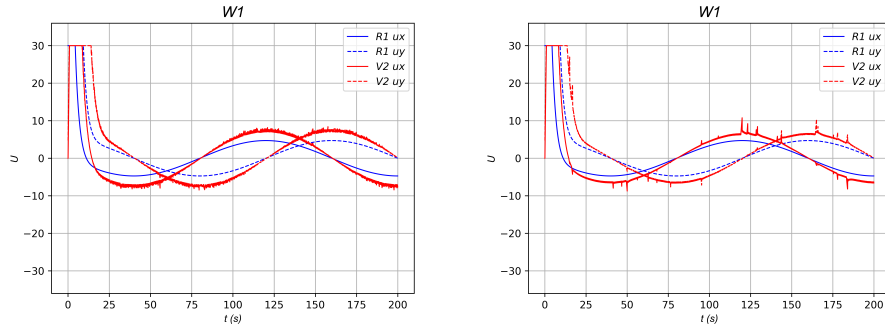


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

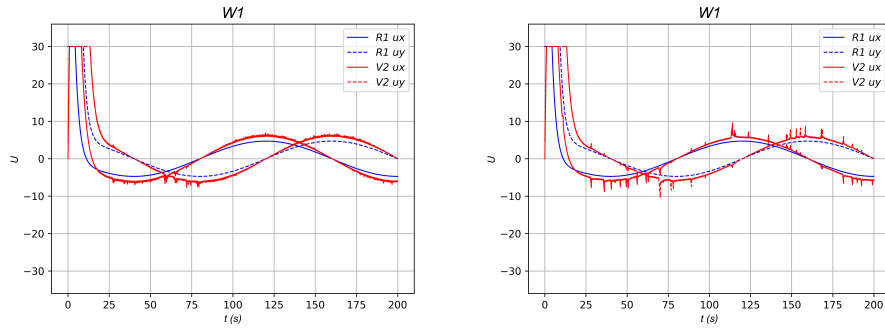


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.23: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).

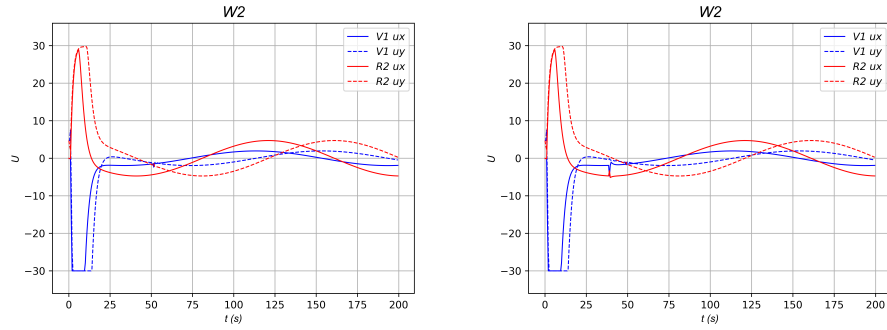


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

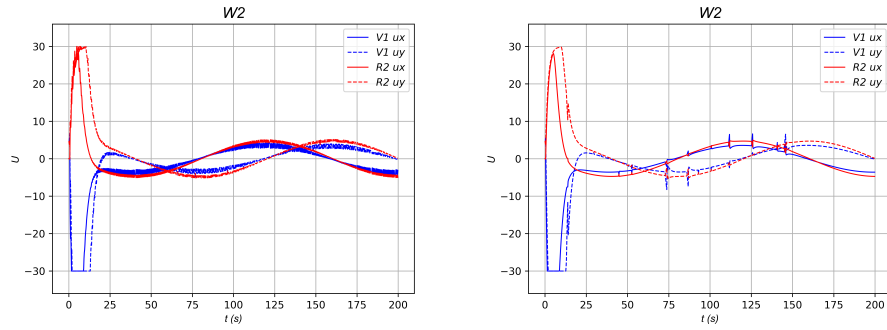


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

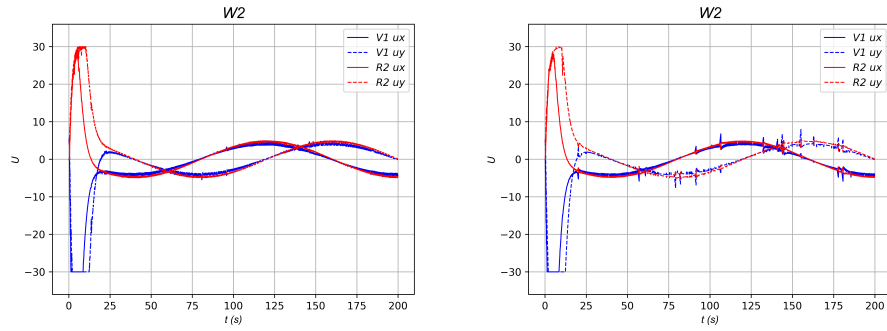
Figura 5.24: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

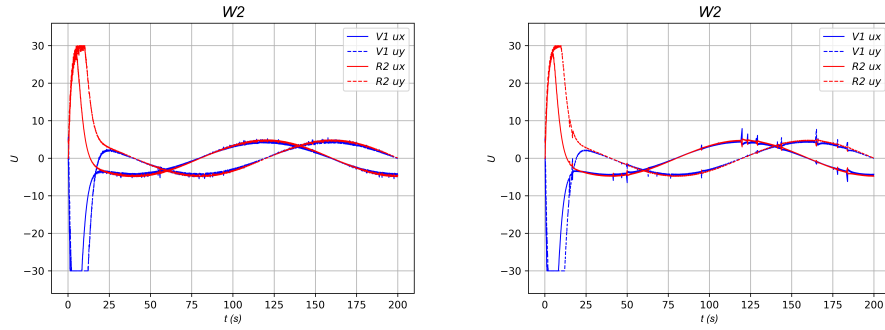


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

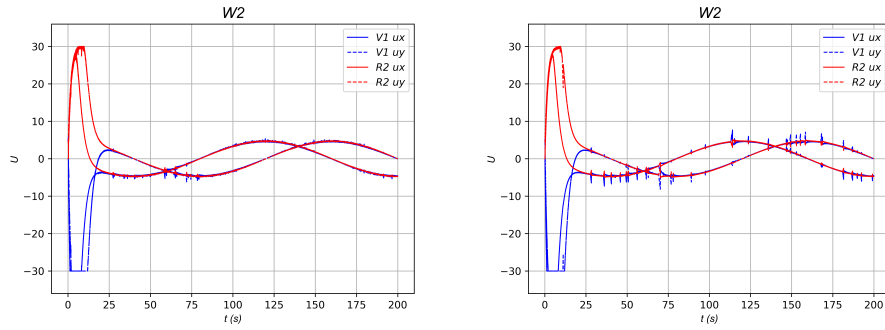


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.25: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (1/2).

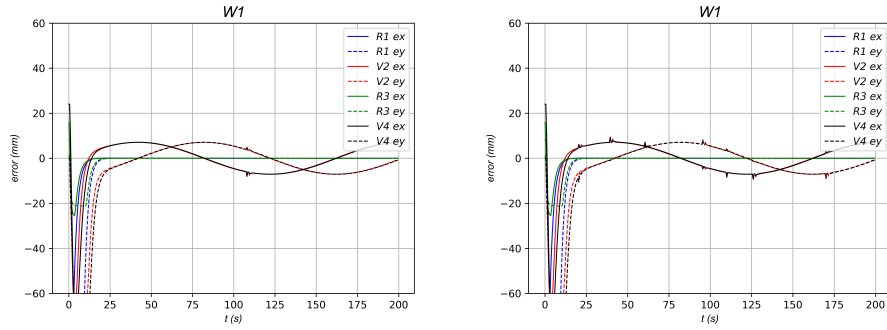


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

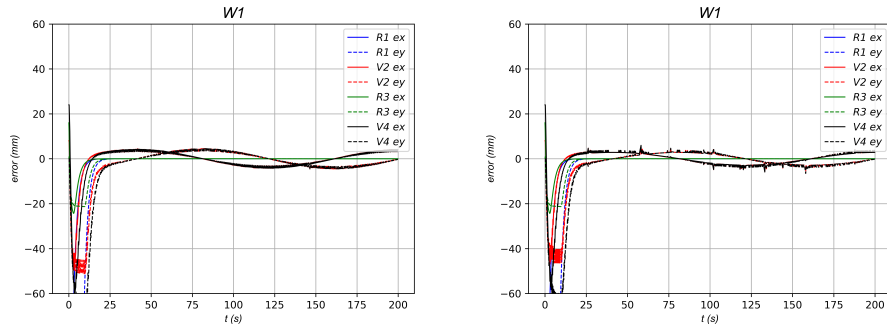


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

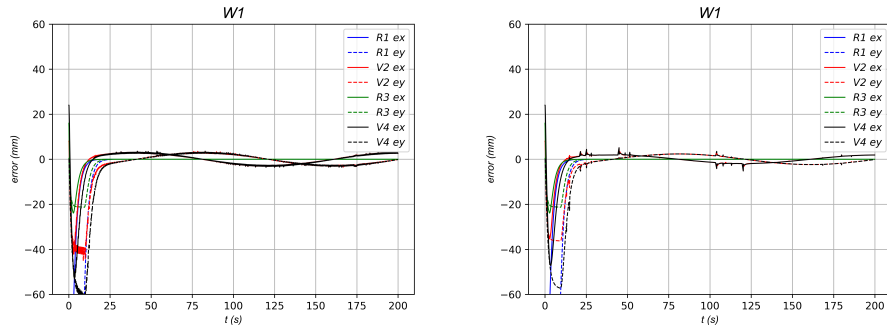
Figura 5.26: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 2 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

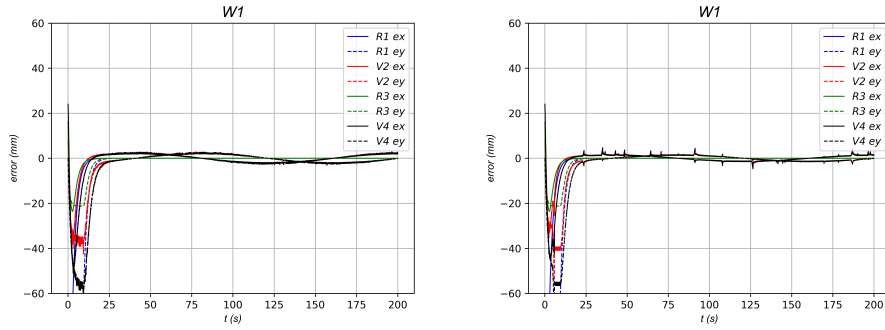


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

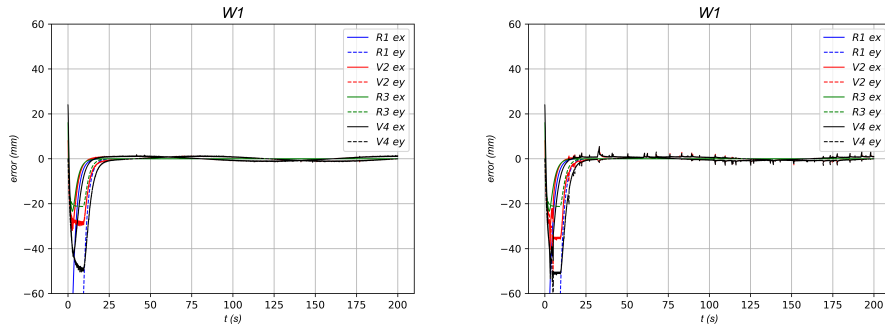


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.27: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).

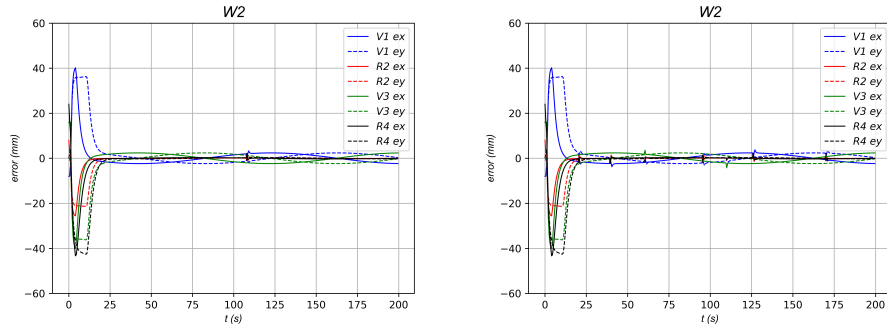


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

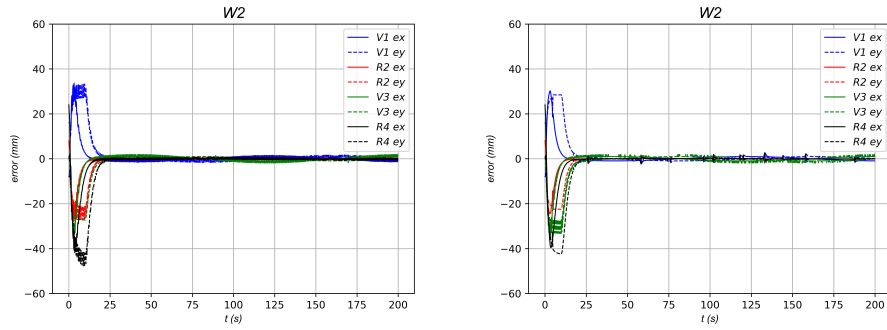


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

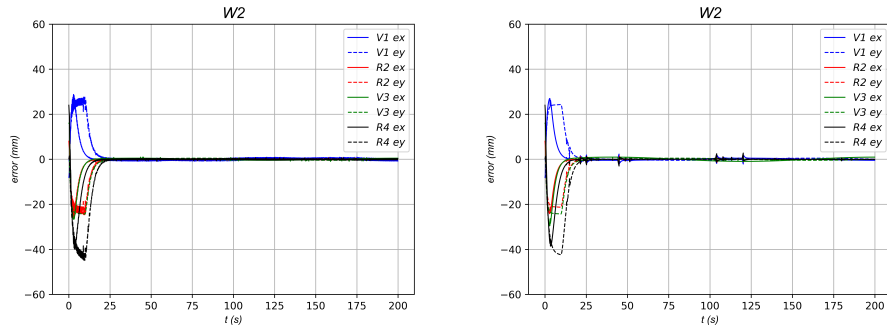
Figura 5.28: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

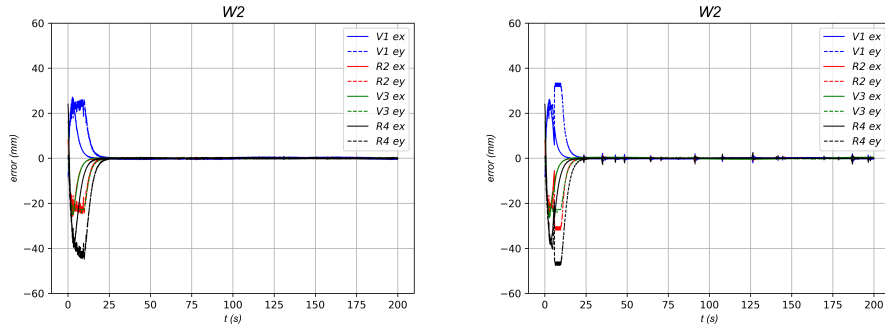


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

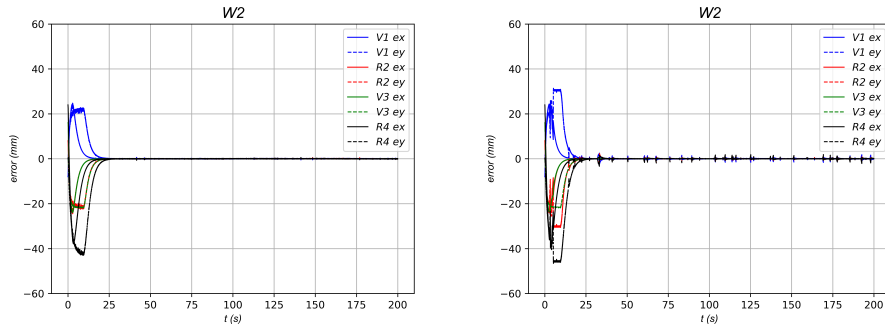


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.29: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).

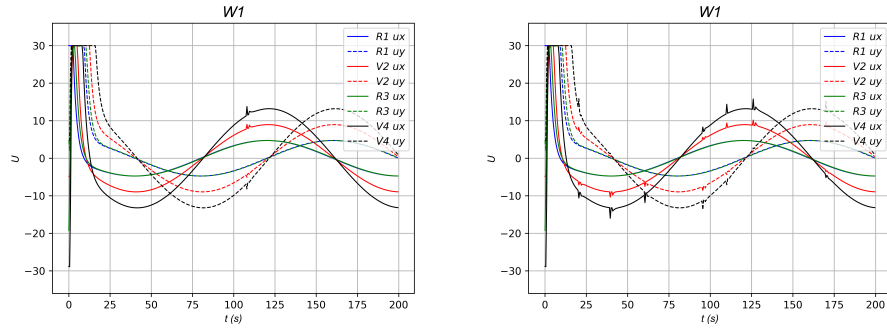


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

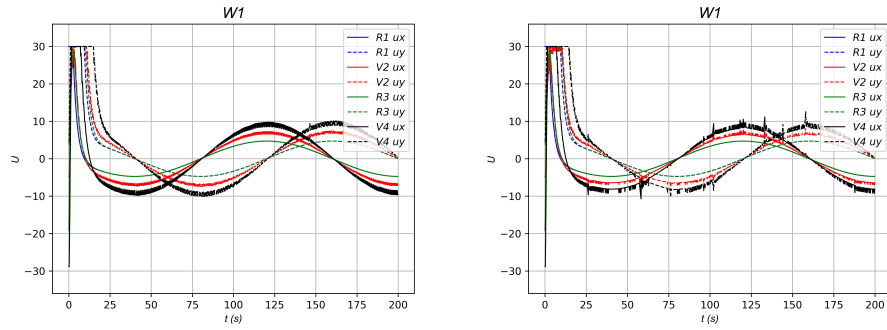


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

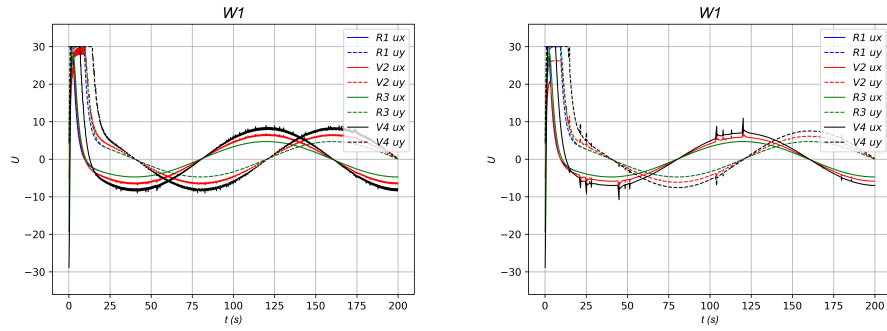
Figura 5.30: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

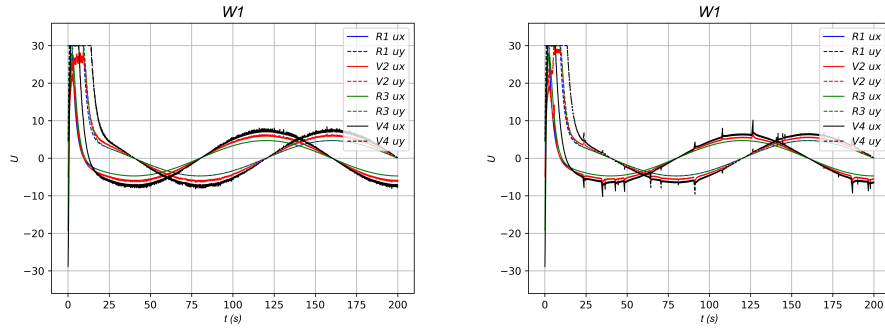


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

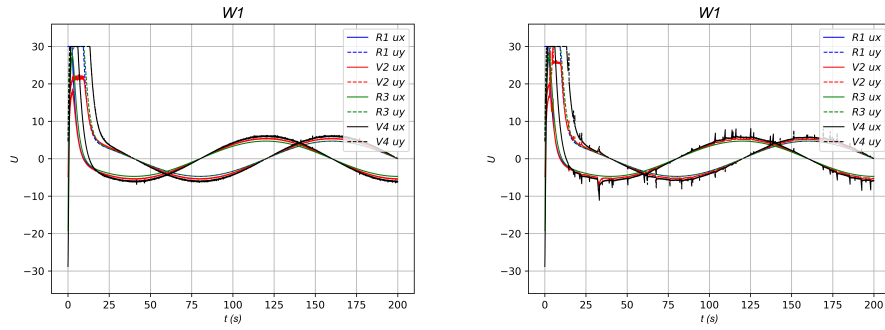


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.31: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).

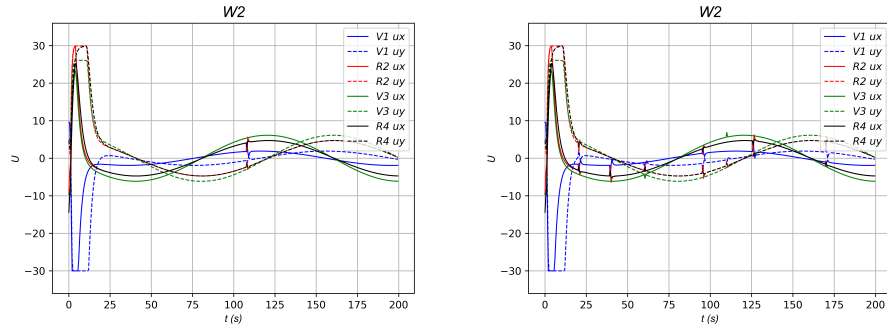


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

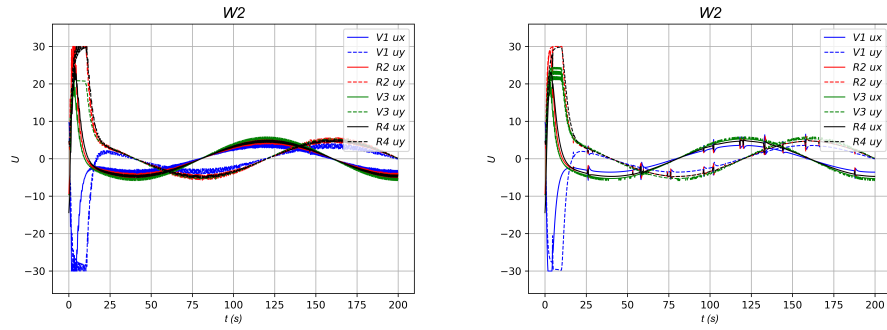


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

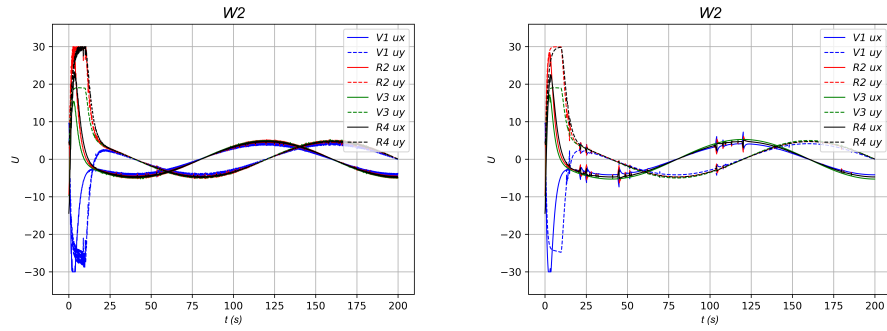
Figura 5.32: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

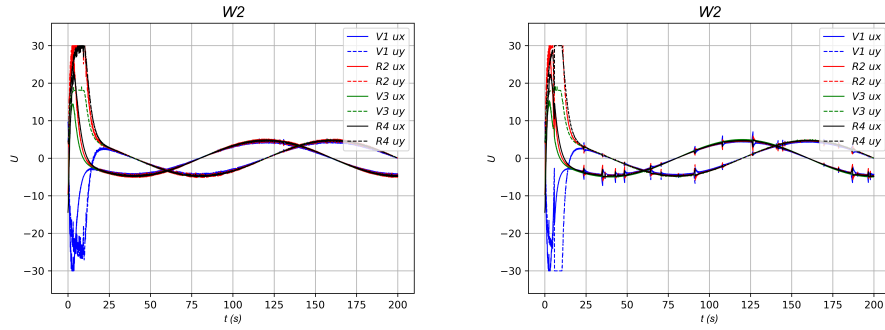


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

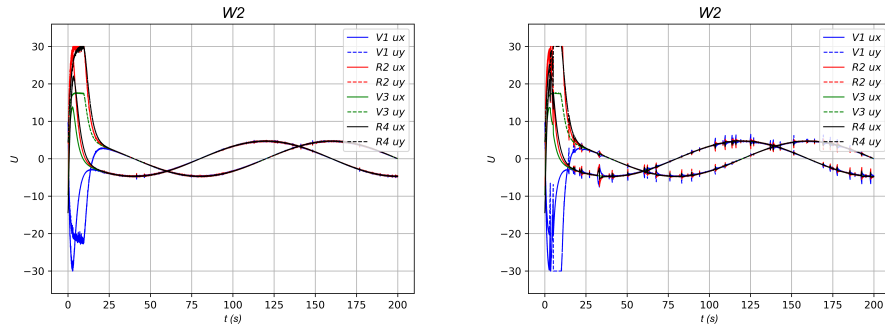


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.33: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (1/2).

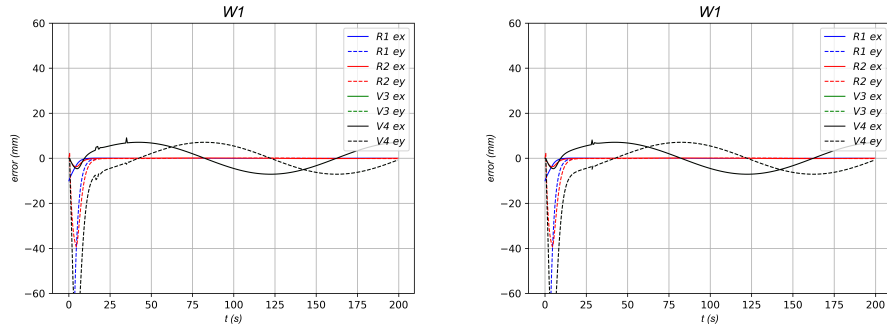


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

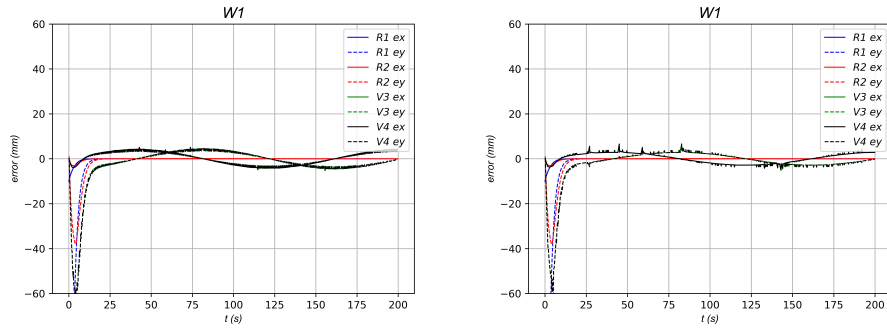


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

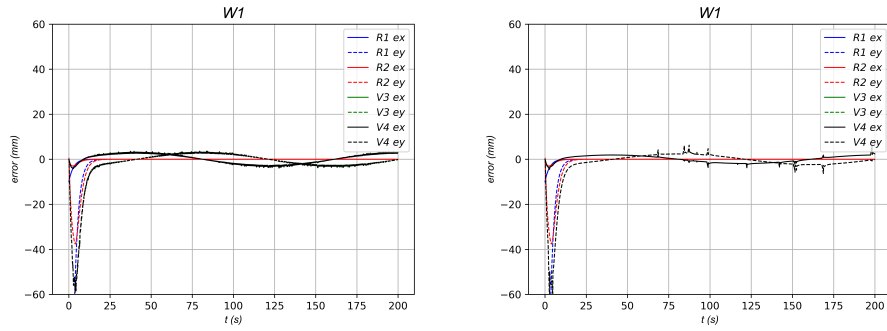
Figura 5.34: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en línea horizontal (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

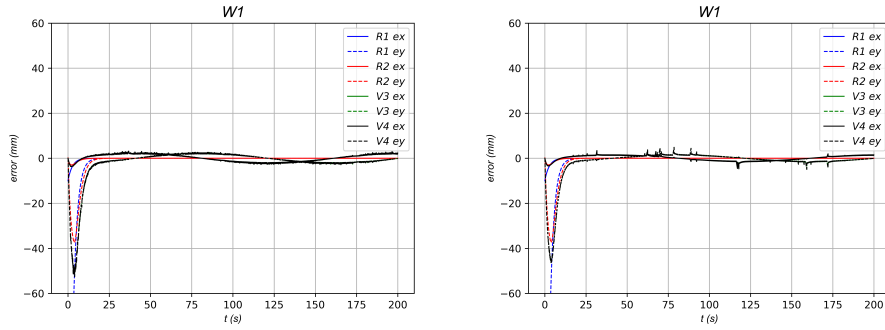


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

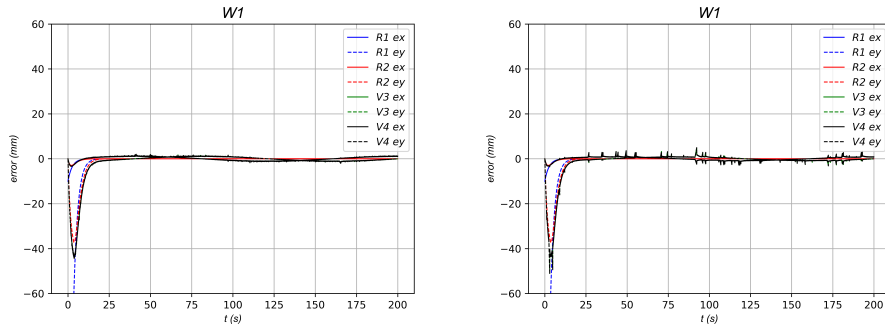


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.35: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo ($1/2$).

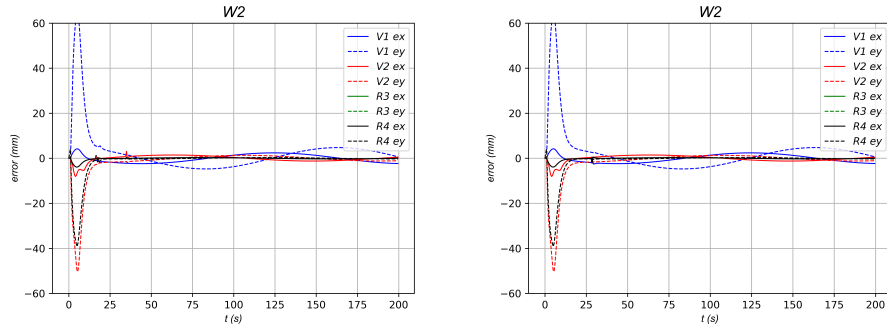


(a) $dt = 0.05$, `test.mosquitto.org`(izquierda) vs. Google Cloud(derecha).

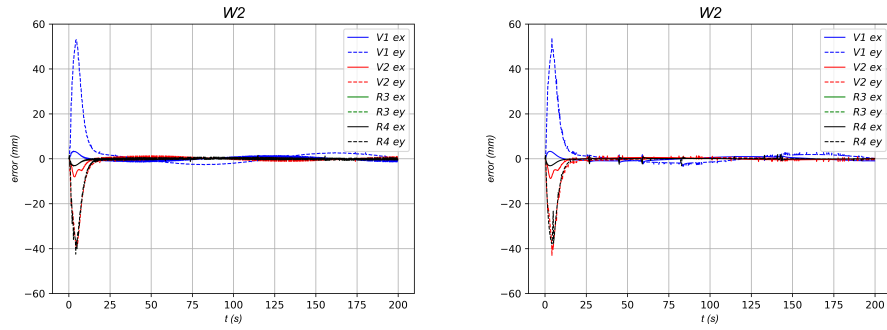


(b) $dt = 0.01$, `test.mosquitto.org`(izquierda) vs. Google Cloud(derecha).

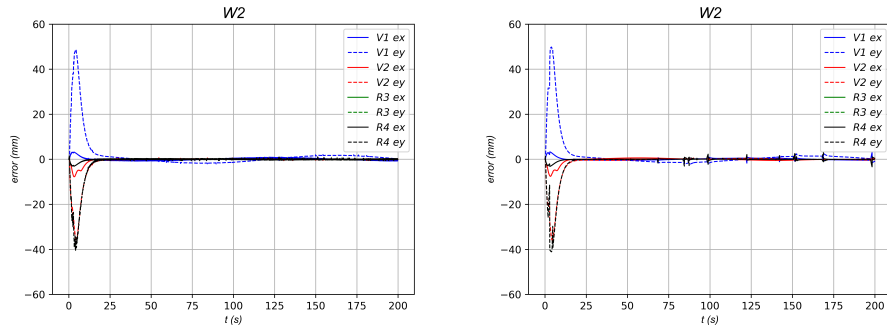
Figura 5.36: Error del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

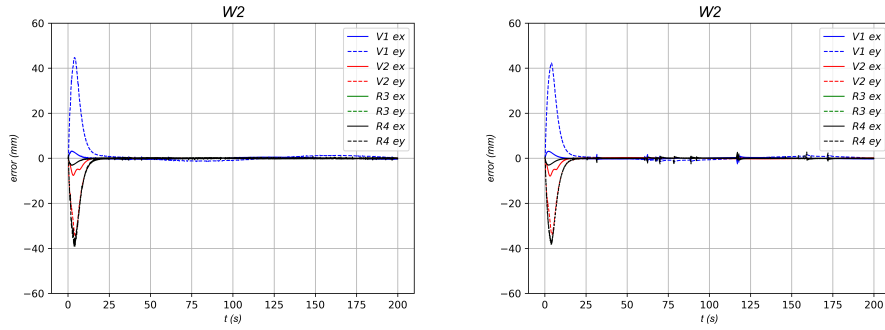


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

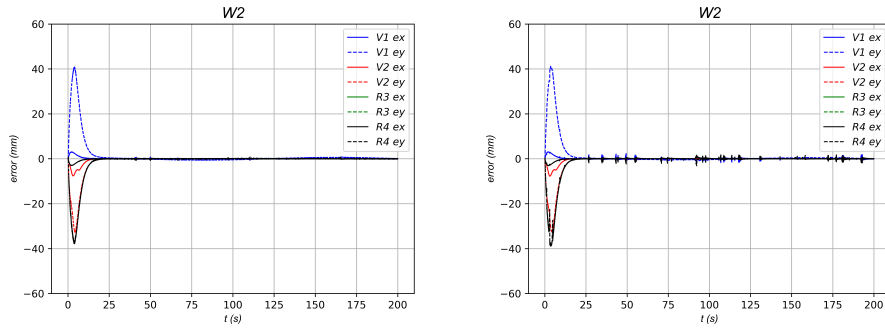


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.37: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (1/2).

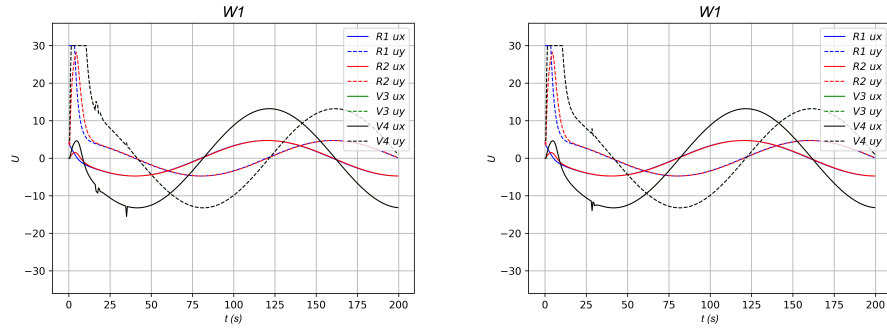


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

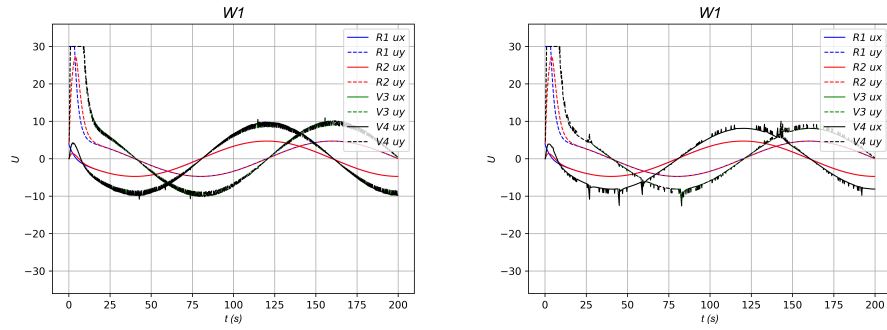


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

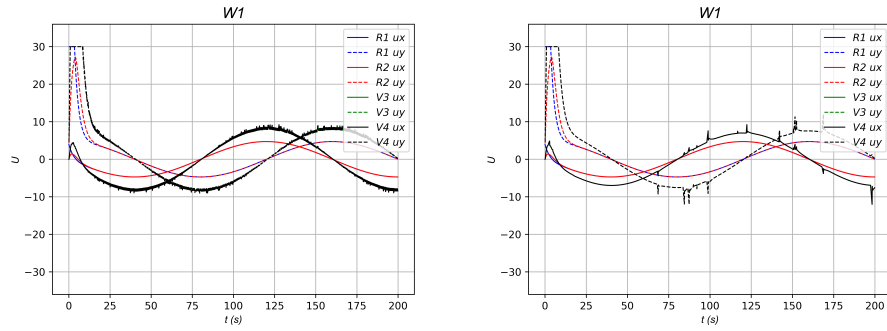
Figura 5.38: Error del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

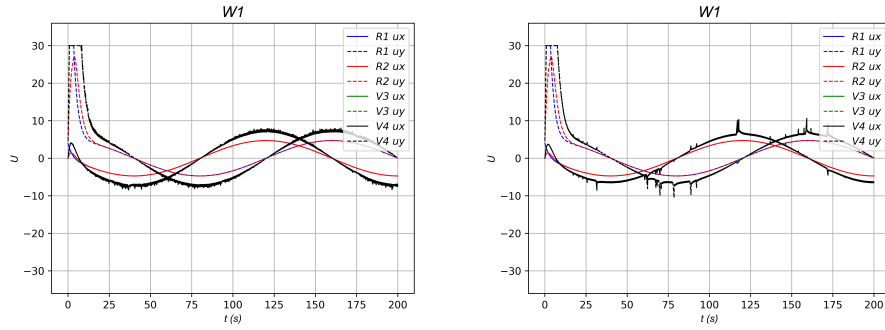


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

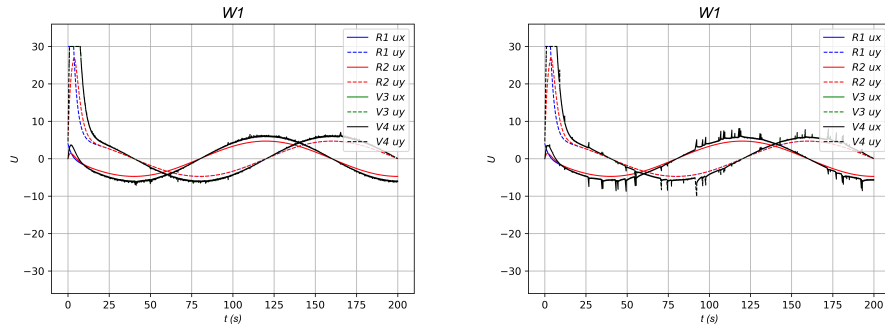


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.39: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo ($1/2$).

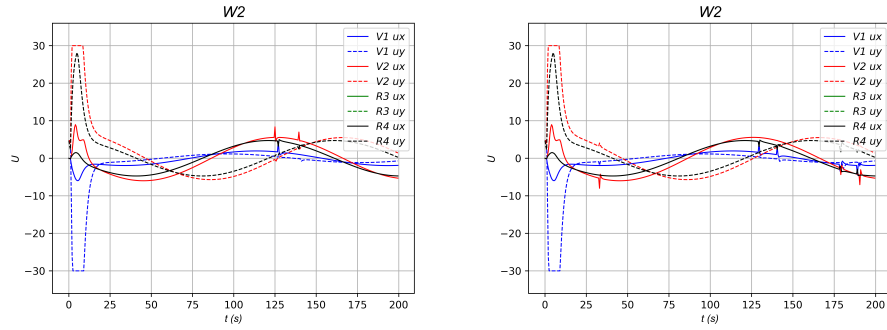


(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

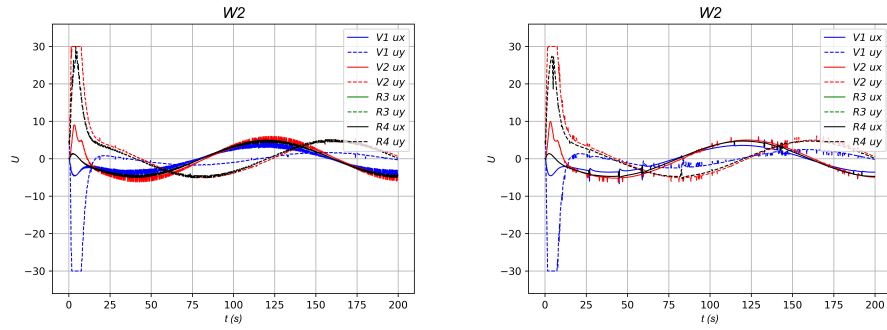


(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

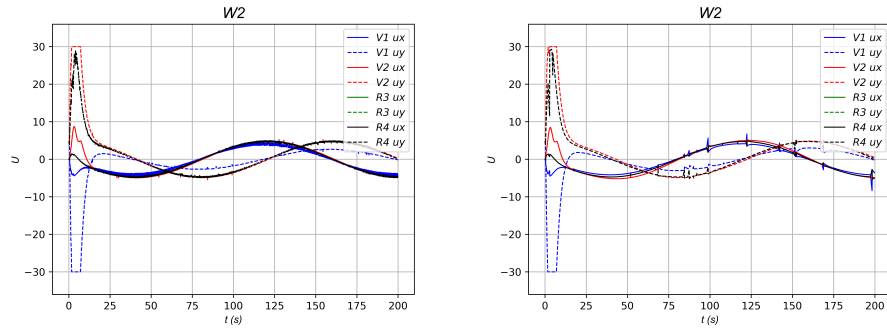
Figura 5.40: Ganancia del control U del espacio de trabajo W_1 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).



(a) $dt = 0.5$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

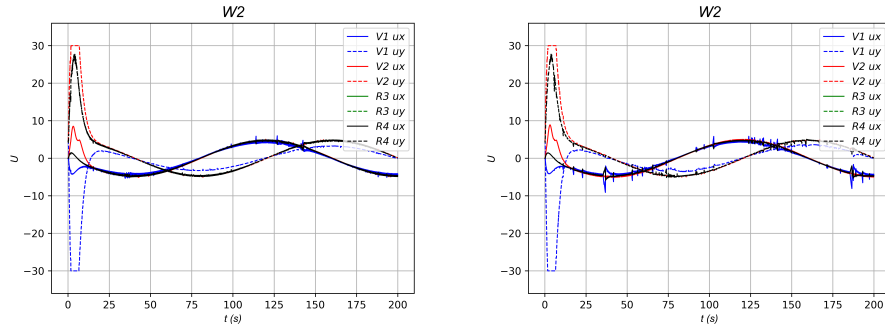


(b) $dt = 0.2$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

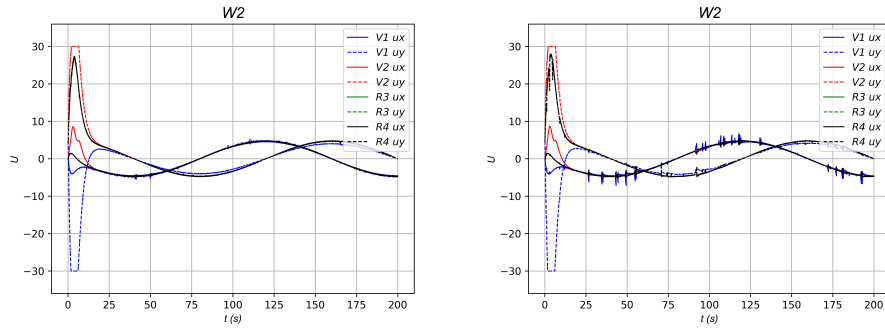


(c) $dt = 0.1$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.41: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo ($1/2$).



(a) $dt = 0.05$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).



(b) $dt = 0.01$, test.mosquitto.org(izquierda) vs. Google Cloud(derecha).

Figura 5.42: Ganancia del control U del espacio de trabajo W_2 con diferentes tasas de muestreo (dt) en el experimento de 4 robots en formación de rombo (2/2).

Capítulo 6

Conclusiones

6.1 Contribuciones

Las principales contribuciones de este trabajo son las siguientes:

- El diseño e implementación de una estrategia de control para formaciones ciberfísicas de sistemas de robots móviles que se desplazan en dos espacios de trabajo. El enfoque de control se basa en el uso de agentes virtuales que convergen junto con los robots físicos que se mueven en el espacio de trabajo opuesto, asegurando que los robots se comporten como si operaran en un único espacio. La convergencia se analiza para el caso general de n robots. Bajo este esquema de control, los robots se comportan como si se desplazaran en un único espacio de trabajo.
- Se formuló el problema para robots de integrador simple en 2D, con alcance general al demostrar que diferentes plataformas de robots móviles no holonómicos pueden reducirse a esta representación simple mediante una transformación de linealización entrada-salida adecuada.
- Se formuló y evaluó una estrategia de control para robots integradores simples en 2D los cuales se comportan como si estuvieran en el mismo espacio, utilizando agentes virtuales en un sistema con y sin retardos en las comunicaciones, evaluando la estabilidad y convergencia del error de la estrategia de control ante retardos en las comunicaciones de los agentes virtuales.
- El esquema de control, inicialmente desarrollado para robots de integra-

dor simple, se extendió al caso de robots tipo unicycle con dos ruedas y se validó mediante simulaciones numéricas.

- Se construyó una plataforma experimental de bajo costo empleando electrónica comercial y piezas impresas en 3D. Se realizaron diferentes experimentos con máximo cuatro robots distribuidos en dos espacios físicos distintos, conectados mediante protocolo MQTT y Bluetooth, evaluando distintas topologías de formación, tasas de muestreo y brokers MQTT.

6.2 Conclusión

En esta tesis se evaluó este nuevo modelo en más de 48 simulaciones y treinta experimentos de formaciones ciberfísicas, proporcionando una visión clara de su diseño, control y experimentación. La aproximación considera robots móviles modelados como integradores simples en un plano; dichos robots se desplazan en dos espacios de trabajo diferentes, compartiendo sus posturas a través de Internet. La ley de control está diseñada para garantizar la convergencia de los robots a un patrón de formación como si se movieran en un único espacio y se formuló para el caso general de n robots.

La estrategia de control emplea agentes virtuales que convergen hacia los robots físicos que se desplazan en el espacio de trabajo opuesto. Se estudiaron dos escenarios: primero, la convergencia de la formación sin retrasos en la comunicación de las posturas, y segundo, el efecto de los retrasos de comunicación cuando las posturas de los robots se transmiten por Internet. Este último caso resulta especialmente relevante en aplicaciones reales con espacios de trabajo distantes, donde la transmisión de información puede sufrir demoras significativas u otros efectos prácticos. La convergencia de la ley de control se analiza y valida mediante simulaciones numéricas.

Además, como prueba de concepto, la ley de control se extiende al caso de robots tipo unicycle. Se diseñó y montó una plataforma experimental de bajo costo empleando electrónica comercial y técnicas de fabricación con impresora 3D. Los robots se ubicaron en dos espacios de trabajo físicos distintos, comunicados por Bluetooth y protocolo MQTT. Esta configuración permitió ensayar experimentalmente ambos escenarios de control (con y sin retrasos), incluyendo la medición del tiempo de entrega de paquetes del servidor MQTT. Los resultados demuestran el desempeño de la ley de control

y su aplicabilidad a problemas prácticos.

Cabe destacar que el enfoque de agentes virtuales puede extenderse a escenarios ciberfísicos con más de dos espacios de trabajo, incluso a ambientes 3D, y es aplicable a diversas tareas multi-robot donde la coordinación remota aporta ventajas, por ejemplo, emular efectos espejo entre robots al aire libre y una estación de control remota, o formaciones en plantas industriales conectadas por redes y routers distintos. Asimismo, es posible alternar estratégicamente entre agentes reales y virtuales para mantener la formación ante fallos de robots reales. Una contribución clave de este trabajo es la demostración de una configuración mínima basada en un único modelo de robot como fundamento para futuras extensiones y aplicaciones.

6.3 Trabajo a futuro

En trabajos futuros se abordarán las siguientes líneas de investigación:

- Modelación dinámica de los robots y extensión a formaciones heterogéneas que incluyan modelos tipo unicycle, omnidireccionales, ackermann y otros vehículos.
- Modelación y extensión del control de robot móviles en otras topologías de formación basadas en distancias, áreas y ángulos.
- Desarrollo de mecanismos robustos para evitar colisiones y garantizar la seguridad durante las fases transitorias de la formación multi-robot.
- Profundización en los desafíos de comunicación en entornos ciberfísicos, como intermitencia de enlaces, retardos variables y otras irregularidades típicas de redes de Internet.
- Explorar escenarios ciberfísicos con más de dos espacios de trabajo remotos, interconectados mediante un sistema ciberfísico.
- Evaluar estrategias de control en un robot tipo AGV a escala real.
- Probar esquemas de control con agentes virtuales implementados como avatares en entornos de realidad virtual.
- Analizar fenómenos de la comunicación como la intermitencia y la pérdida de datos, entre otros.
- Iterar la plataforma robótica de experimentación para fortalecer sus

capacidades de investigación y docencia en sistemas ciberfísicos.

6.4 Publicaciones y distinciones

Las publicaciones en congresos y revistas que formaron parte del desarrollo de esta tesis se enlistan a continuación:

Giron-Nieto, Huber; Ochoa-Garcia, Oliver; Hernandez-Martinez, Eduardo Gamaliel; Ramirez-Neria, Mario; Fernández Anaya, Guillermo; Ferreira, Enrique D.; Flores-Godoy, Jose-Job. Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments. Memorias del Congreso Nacional de Control Automático, pp. 507-512, 2023. <https://doi.org/10.58571/CNCA.AMCA.2023.105> Apendice B. [140]

Giron-Nieto, H., Hernandez-Martinez, E. G., Fernandez-Anaya, G., Ferreira-Vazquez, E. D., Flores-Godoy, J.-J., Ramírez-Neria, M., & Molano-Jimenez, A. (2025). Cyber-Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach. Electronics, 14(9), 1869. <https://doi.org/10.3390/electronics14091869> Apendice C. [141]

Giron-Nieto, Huber. Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments. Ponente de divulgación en el 3er Simposio Anual del Grupo de Investigación en Ciencias Básicas, [142]

Bibliografia

- [1] Carlos Canudas de Wit, Bruno Siciliano, and Georges Bastin, editors. *Theory of Robot Control*. Springer London, 1996.
- [2] Tiago Valentim, Rita Cunha, Paulo Oliveira, David Cabecinhas, and Carlos Silvestre. Multi-vehicle cooperative control for load transportation. *IFAC-PapersOnLine*, 52(12):358–363, 2019. 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019.
- [3] Ryosuke Matsuo, Shinya Yasuda, Taichi Kumagai, Natsuhiko Sato, Hiroshi Yoshida, and Takehisa Yairi. Residual reinforcement learning for logistics cart transportation. *Advanced Robotics*, 36(8):404–421, 2022.
- [4] Sara L. Lopes, Aristides I. Ferreira, Rui Prada, and Ralf Schwarzer. Social robots as health promoting agents: An application of the health action process approach to human-robot interaction at the workplace. *International Journal of Human-Computer Studies*, 180:103124, 2023.
- [5] Mateus V. Gasparino, Vitor A.H. Higuti, Arun N. Sivakumar, Andres E.B. Velasquez, Marcelo Becker, and Girish Chowdhary. Cropnav: a framework for autonomous navigation in real farms. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11824–11830, 2023.
- [6] Shweta Vichare, Snehal Katare, and Amruta Pawar. Wireless military defense robot. *International Research Journal of Engineering and Technology (IRJET)*, 7(05), 2020.
- [7] R Benazir Begam, V Yogalakshmi, M Sanjai Nagaraaja, A Sanjay Kumar, and A Santhosh Priyan. Iot based multipurpose agricultural robot. In *2024 3rd International Conference on Applied Artificial Intelligence*

- and Computing (ICAAIC)*, pages 1755–1761, 2024.
- [8] Khac Duc Do. Formation tracking control of unicycle-type mobile robots. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 2391–2396. IEEE, 2007.
 - [9] Yang Fei, Peng Shi, Chee Peng Lim, and Xin Yuan. Finite-time observer-based formation tracking with application to omnidirectional robots. *IEEE Transactions on Industrial Electronics*, 70(10):10598–10606, 2022.
 - [10] Eduardo G Hernandez-Martinez, Sergio A Foyo-Valdes, Erika S Puga-Velazquez, and Jesús A Meda-Campaña. Hybrid architecture for coordination of agvs in fms. *Int. Journal of Advanced Robotic Systems*, 11(3):41, 2014.
 - [11] Jean Luc Farrugia and Simon G. Fabri. Swarm robotics for object transportation. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 353–358, 2018.
 - [12] Enrique D Ferreira-Vazquez, Eduardo Gamaliel Hernández-Martínez, José-Job Flores-Godoy, Guillermo Fernandez-Anaya, and P Paniagua-Contro. Distance-based formation control using angular information between robots. *Journal of Intelligent & Robotic Systems*, 83(3):543–560, 2016.
 - [13] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.
 - [14] A Lopez-Gonzalez, ED Ferreira, Eduardo Gamaliel Hernández-Martínez, José-Job Flores-Godoy, Guillermo Fernandez-Anaya, and P Paniagua-Contro. Multi-robot formation control using distance and orientation. *Advanced Robotics*, 30(14):901–913, 2016.
 - [15] K. L. Keung, C. K. M. Lee, P. Ji, and Kam K. H. Ng. Cloud-based cyber-physical robotic mobile fulfillment systems: A case study of collision avoidance. *IEEE Access*, 8:89318–89336, 2020.
 - [16] Thai-Viet Dang and Ngoc-Tam Bui. Obstacle avoidance strategy for mobile robot based on monocular camera. *Electronics*, 12(8):1932, 2023.
 - [17] Luis Escobar, Cristian Moyano, Gabriel Aguirre, Gabriel Guerra, Luis Allauca, and David Loza. Multi-robot platform with features of cyber-physical systems for education applications. In *2020 IEEE ANDESCON*, pages 1–6, 2020.

- [18] Farzad Samie, Lars Bauer, and Jörg Henkel. Iot technologies for embedded computing: A survey. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 1–10, 2016.
- [19] Bhushankumar Nemade, Neelam Phadnis, Aaditya Desai, and Kiran Krushnakant Mungekar. Enhancing connectivity and intelligence through embedded internet of things devices. *ICTACT Journal on Microelectronics*, 9(4):1670–1674, 2024.
- [20] Yindong Lian, Wei Xie, Qifan Yang, Langwen Zhang, Danqi Lin, and Yajing Zhou. A novel multi-warehouse mobile robot hierarchical scheduling strategy based on industrial cyber-physical system. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 263–269, 2021.
- [21] Heiko Pikner, Raivo Sell, Kristo Karjust, Ehsan Malayjerdi, and Tarmo Velsker. Cyber-physical control system for autonomous logistic robot. In *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pages 699–704, 2021.
- [22] T. Kruglova, I. Schmelev, I. Sushkov, and R. Filatov. Cyber-physical system of the mobile robots optimal trajectory planning with taking into account electric motors deterioration. In *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–5, 2019.
- [23] Yang Deng, Vincent Léchappé, Emmanuel Moulay, Zhang Chen, Bin Liang, Franck Plestan, and Qing-Long Han. Predictor-based control of time-delay systems: a survey. *International Journal of Systems Science*, 53(12):2496–2534, 03 2022.
- [24] Vladimir Kharitonov. *Time-delay systems: Lyapunov functionals and matrices*. Springer Science & Business Media, 2012.
- [25] S. Mondie and V.L. Kharitonov. Exponential estimates for retarded time-delay systems: an lmi approach. *IEEE Transactions on Automatic Control*, 50(2):268–273, 2005.
- [26] Hieu Trinh et al. Exponential stability of time-delay systems via new weighted integral inequalities. *Applied Mathematics and Computation*, 275:335–344, 2016.

- [27] Gennaro Notomista, Xiaoyi Cai, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. *CoRR*, abs/1904.04801, 2019.
- [28] Iasson Karafyllis, Michael Malisoff, Frederic Mazenc, and Eds. Pierdomenico Pepe. Recent advances in nonlinear delay control systems. In *Advances in Delays and Dynamics*. Springer, 2016.
- [29] Jiansheng Guan, Wende Zhou, Shaobo Kang, Yuan Sun, and Zibo Liu. Robot formation control based on internet of things technology platform. *IEEE Access*, 8:96767–96776, 2020.
- [30] Héber Renato Fadel de Moraes, João Paulo Lima Silva de Almeida, and Lúcia Valéria Ramos de Arruda. Running cooperative tasks in a multi-robot system with cyber-physical features. In *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pages 217–222. IEEE, 2022.
- [31] Weilin Li, Jing Wu, and Chengnian Long. Formation control for unmanned surface vehicles based on integrative apf and mpc. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 201–206. IEEE, 2022.
- [32] Hasan Kivrak, Muhammed Zahid Karakusak, Simon Watson, and Barry Lennox. Cyber-physical system architecture of autonomous robot ecosystem for industrial asset monitoring. *Computer Communications*, 218:72–84, 2024.
- [33] Jiacheng Xie, Shuguang Liu, and Xuewen Wang. Framework for a closed-loop cooperative human cyber-physical system for the mining industry driven by vr and ar: Mhcps. *Computers & Industrial Engineering*, 168:108050, 2022.
- [34] Kyung Jin Kim, Il Hong Suh, Sung Hoon Kim, and Sang Rok Oh. A novel real-time control architecture for internet-based thin-client robot; simulacrum-based approach. In *2008 IEEE International Conference on Robotics and Automation*, pages 4080–4085, 2008.
- [35] Luca Nanu, Luigi Colangelo, Carlo Novara, and Carlos Perez Montenegro. Embedded model control of networked control systems: An experimental robotic application. *Mechatronics*, 99:103160, 2024.

- [36] Wang-Tai Lo, Yun-Hui Liu, I. Elhajj, Ning Xi, Yinghai Shi, and Yuechao Wang. Co-operative control of internet based multi-robot systems with force reflection. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 4414–4419, 2003.
- [37] Takahiro Ishikawa, Michiya Masuda, Shunsuke Uda, Koichiro Yokoyama, Kenniti Hamamoto, and Kiminao Kogiso. Constrained remote control of construction machine with time-varying delay and packet loss. *Advanced Robotics*, 37(1-2):46–60, 2023.
- [38] Aamir Shahzad and Hubert Roth. Bilateral telecontrol of automerlin mobile robot with fix communication delay. In *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pages 1–6, 2016.
- [39] Addie Irawan, Mohammad Fadhil Abas, and Nurulfadzilah Hasan. Robot local network using tqc protocol for land-to-underwater communications. *Journal of Telecommunications and Information Technology*, 75(1):23–30, 2019.
- [40] Hongxia Wang, Zixing Li, Tao Liu, and Xiao Liang. Infinite-horizon optimal control for networked control systems with markovian packet losses. *Optimal Control Applications and Methods*, 45(1):29–44, 2024.
- [41] Eduardo G Hernandez-Martinez and Eduardo Aranda-Bricaire. Decentralized formation control of multi-agent robot systems based on formation graphs. *Studies in Informatics and Control*, 21(1):8, 2012.
- [42] Eduardo Gamaliel Hernandez-Martinez, José Job Flores-Godoy, and Guillermo Fernandez-Anaya. Decentralized discrete-time formation control for multirobot systems. *Discrete Dynamics in Nature and Society*, 2013:746713, 2013.
- [43] Eduardo Gamaliel Hernández-Martínez, José-Job Flores-Godoy, and Guillermo Fernandez-Anaya. Marching control based on the leader-followers scheme and formation graphs. *Advanced Robotics*, 27(14):1123–1135, 2013.
- [44] Jaime González-Sierra, Eduardo Aranda-Bricaire, and Eduardo Gamaliel Hernandez-Martinez. Formation tracking with orientation convergence for groups of unicycles. *International Journal of Advanced Robotic Systems*, 10(3):180, 2013.

- [45] Juan Carlos Pérez Sampieri, Eduardo Aranda Bricaire, Eduardo Gamaliel Hernández Martínez, et al. Supervisory control of agvs for flexible manufacturing cells. In *Proceedings of the Congreso Nacional de Control Automático 2013*, Ensenada, Baja California, México, Oct. 16–18 2013.
- [46] Eduardo Gamaliel Hernández Martínez, Erika Selene Puga Velázquez, Sergio A. Foyo Valdés, Jesús Alberto Meda Campaña, et al. Modeling framework for automated manufacturing systems based on petri nets and isa standards. *Studies in Informatics and Control*, 22(2):163–174, June 2013.
- [47] E. G. Hernández-Martínez, Sergio A. Foyo-Valdés, Erika S. Puga-Velázquez, and J. A. Meda Campaña. Motion coordination of agv's in fms using petri nets. *IFAC-PapersOnLine*, 48(3):187–192, 2015.
- [48] Enrique D Ferreira Vázquez, Eduardo Gamaliel Hernández Martínez, and José Job Flores Godoy. Formation control of multiple robots avoiding local minima. In *Proceedings of the XVI Latin American Congress of Automatic Control*, pages 1–6. International Federation of Automatic Control, 2014.
- [49] Eduardo Gamaliel Hernandez-Martinez, Guillermo Fernandez-Anaya, ED Ferreira, José Job Flores-Godoy, and Alexandro Lopez-Gonzalez. Trajectory tracking of a quadcopter uav with optimal translational control. *IFAC-PapersOnLine*, 48(19):226–231, 2015.
- [50] Tommaso Gallo, Chiara Cagnetti, Cecilia Silvestri, and Alessandro Ruggieri. Industry 4.0 tools in lean production: A systematic literature review. *Procedia Computer Science*, 180:394–403, 2021. Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020).
- [51] Francisco Javier Folgado, David Calderón, Isaías González, and Antonio José Calderón. Review of industry 4.0 from the perspective of automation and supervision systems: Definitions, architectures and recent trends. *Electronics*, 13(4), 2024.
- [52] Izabela Rojek, Piotr Kotlarz, Janusz Doroyski, and Dariusz Mikoajewski. Sixth-generation (6g) networks for improved machine-to-machine (m2m) communication in industry 4.0. *Electronics*, 13(10), 2024.
- [53] Marek Nagy, Marcel Figura, Katarina Valaskova, and George Lzroiou. Predictive maintenance algorithms, artificial intelligence digital twin

technologies, and internet of robotic things in big data-driven industry 4.0 manufacturing systems. *Mathematics*, 13(6), 2025.

- [54] Elias Dritsas and Maria Trigka. A survey on the applications of cloud computing in the industrial internet of things. *Big Data and Cognitive Computing*, 9(2), 2025.
- [55] Luca Pietrantoni, Marco Favilla, Federico Fraboni, Elvis Mazzoni, Sofia Morandini, Martina Benvenuti, and Marco De Angelis. Integrating collaborative robots in manufacturing, logistics, and agriculture: Expert perspectives on technical, safety, and human factors. *Frontiers in Robotics and AI*, Volume 11 - 2024, 2024.
- [56] Eneko Villar, Imanol Martín Toral, Isidro Calvo, Oscar Barambones, and Pablo Fernández-Bustamante. Architectures for industrial aiot applications. *Sensors*, 24(15), 2024.
- [57] M. Fugerth and Z. Vámosy. Autonomus navigation system for mobile robot. In *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*, pages 165–170, 2013.
- [58] Hyunhak Cho, Eun Kyeong Kim, and Sungshin Kim. Indoor slam application using geometric and icp matching methods based on line features. *Robotics and Autonomous Systems*, 100:206–224, 2018.
- [59] Abdurrahman Yilmaz and Hakan Temeltas. Self-adaptive monte carlo method for indoor localization of smart agvs using lidar data. *Robotics and Autonomous Systems*, 122:103285, 2019.
- [60] André Araújo, David Portugal, Micael S. Couceiro, Jorge Sales, and Rui P. Rocha. Desarrollo de un robot móvil compacto integrado en el middleware ros. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 11(3):315–326, 2014.
- [61] Research in China. Global and china automated guided vehicle (agv) industry report. Technical report, Research in China, 2019. Industry report.
- [62] Mordor Intelligence. Automated guided vehicle (agv) market – growth, trends, covid-19 impact, and forecasts (2022–2027). Technical report, Mordor Intelligence, 2022. Market research report.
- [63] Hui Liu. Chapter 4 - automatic guided vehicles (agvs) in the rail transit intelligent manufacturing environment. In Hui Liu, editor, *Robot*

- Systems for Rail Transit Applications*, pages 143–188. Elsevier, 2020.
- [64] Yu Liu, Shuting Wang, Yuanlong Xie, Tifan Xiong, and Mingyuan Wu. A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 24(4), 2024.
 - [65] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in Python*, volume 146 of *Springer Tracts in Advanced Robotics*. Springer, Cham, 3 edition, 2023.
 - [66] Heonmoo Kim and Yosoon Choi. 3d location estimation and tunnel mapping of autonomous driving robots through 3d point cloud registration on underground mine rampways. *Underground Space*, 22:1–20, 2025.
 - [67] Zheng Fan, Lele Zhang, Xueyi Wang, Yilan Shen, and Fang Deng. Lidar, imu, and camera fusion for simultaneous localization and mapping: a systematic review. *Artificial Intelligence Review*, 58:174, 2025.
 - [68] I. Jarraya, A. Al-Batati, M. B. Kadri, et al. Gnss-denied unmanned aerial vehicle navigation: analyzing computational complexity, sensor fusion, and localization methodologies. *Satellite Navigation*, 6:9, 2025.
 - [69] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
 - [70] Wei Ren, Randal W Beard, and Ella M Atkins. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.
 - [71] Magdi S. Mahmoud and Yuanqing Xia. Chapter 3 - consensus of systems over graphs. In Magdi S. Mahmoud and Yuanqing Xia, editors, *Discrete Networked Dynamic Systems*, pages 91–136. Academic Press, 2021.
 - [72] Luca Scardovi and Rodolphe Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11):2557–2562, 2009.
 - [73] G. Nava-Antonio, Guillermo Fernández-Anaya, Eduardo Gamaliel Hernández-Martínez, José-Job Flores-Godoy, and Enrique D. Ferreira-Vázquez. Consensus of multiagent systems described by various noninteger derivatives. *Complexity*, 2019:1–14, February 2019.
 - [74] Mohamed A. Kamel, Xiang Yu, and Youmin Zhang. Formation control and coordination of multiple unmanned ground vehicles in normal and

- faulty situations: A review. *Annual Reviews in Control*, 49:128–144, 2020.
- [75] Pablo Paniagua-Contro, Eduardo Gamaliel Hernández-Martínez, Oscar González-Medina, Jaime González-Sierra, José Job Flores-Godoy, Enrique D. Ferreira-Vázquez, and Guillermo Fernández-Anaya. Extension of leader-follower behaviours for wheeled mobile robots in multirobot coordination. *Mathematical Problems in Engineering*, 2019:1–16, April 2019.
 - [76] J. Santiaguillo-Salinas and E. Aranda-Bricaire. Time-varying formation tracking with collision avoidance for multi-agent systems**this work was partially supported by conacyt, méxico, through scholarship holder no. 243226. *IFAC-PapersOnLine*, 50(1):309–314, 2017. 20th IFAC World Congress.
 - [77] Dongyu Li, Shuzhi Sam Ge, Wei He, Guangfu Ma, and Lihua Xie. Multi-layer formation control of multi-agent systems. *Automatica*, 109:108558, 2019.
 - [78] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.
 - [79] Ertug Olcay, Katharina Gabrich, and Boris Lohmann. Optimal control of a swarming multi-agent system through guidance of a leader-agent. *IFAC-PapersOnLine*, 52(20):1–6, 2019. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.
 - [80] Yutong Liu, Hongjun Yu, Peng Shi, and Cheng-Chew Lim. Formation control and collision avoidance for a class of multi-agent systems. *Journal of the Franklin Institute*, 356(10):5395–5420, 2019.
 - [81] Christopher Heintz and Jesse B. Hoagg. Formation control for agents modeled with extended unicycle dynamics that includes orientation kinematics on $so(m)$ and speed constraints. *Systems & Control Letters*, 146:104784, 2020.
 - [82] Flavio Roberti, Juan Marcos Toibero, Raquel Frizera Vassallo, and Ricardo Carelli. Control estable de formación basado en visión omnidireccional para robots móviles no holonómicos. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(1):29–37, 2011.
 - [83] Hao Liu, Yu Tian, Frank L. Lewis, Yan Wan, and Kimon P. Valava-

- nis. Robust formation tracking control for multiple quadrotors under aggressive maneuvers. *Automatica*, 105:179–185, 2019.
- [84] Nadia Nedjah, Luigi Maciel Ribeiro, and Luiza de Macedo Mourelle. Communication optimization for efficient dynamic task allocation in swarm robotics. *Applied Soft Computing*, 105:107297, 2021.
 - [85] Shun Yang, Weiwei Bai, Tieshan Li, Quan Shi, Yue Yang, Yue Wu, and C. L. Philip Chen. Neural-network-based formation control with collision, obstacle avoidance and connectivity maintenance for a class of second-order nonlinear multi-agent systems. *Neurocomputing*, 439:243–255, 2021.
 - [86] Hui Liu, Zhiyun Lin, Ming Cao, Xiaoping Wang, and Jinhu Lü. Coordinate-free formation control of multi-agent systems using rooted graphs. *Systems & Control Letters*, 119:8–15, 2018.
 - [87] Yueqian Liang, Qi Dong, and Yanjie Zhao. Adaptive leaderfollower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances. *Chinese Journal of Aeronautics*, 33(11):2972–2988, 2020. SI: Emerging Technologies of Unmanned Aerial Vehicles.
 - [88] Ning Wang and He Li. Leader-follower formation control of surface vehicles: A fixed-time control approach. *ISA Transactions*, 2020.
 - [89] Qi Li, Jinyuan Wei, Qiuxiong Gou, and Zhiqi Niu. Distributed adaptive fixed-time formation control for second-order multi-agent systems with collision avoidance. *Information Sciences*, 564:27–44, 2021.
 - [90] Eduardo G. Hernández-Martínez and Eduardo Aranda-Bricaire. Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach. In *Multi-agent SystemsModelling, Control, Programming, Simulations and Applications*, pages 105–126. InTech, Austria, 2011.
 - [91] Yan-Wu Wang, Meng Liu, Zhi-Wei Liu, and Jing-Wen Yi. Formation tracking of the second-order multi-agent systems using position-only information via impulsive control with input delays. *Applied Mathematics and Computation*, 246:572–585, 2014.
 - [92] Ellips Masehian, Marjan Jannati, and Taher Hekmatfar. Cooperative mapping of unknown environments by multiple heterogeneous mobi-

- le robots with limited sensing. *Robotics and Autonomous Systems*, 87:188–218, 2017.
- [93] Héctor Ríos, Jaime González-Sierra, and Alejandro Dzul. Robust tracking output-control for a quad-rotor: A continuous sliding-mode approach. *Journal of the Franklin Institute*, 354(15):6672–6691, 2017.
 - [94] Nikolaos Stefas, Haluk Bayram, and Volkan Isler. Vision-based uav navigation in orchards. *IFAC-PapersOnLine*, 49(16):10–15, 2016. 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016.
 - [95] Bharadwaj R.K. Mantha, Min Kyu Jung, Borja García de Soto, Carol C. Menassa, and Vineet R. Kamat. Generalized task allocation and route planning for robots with multiple depots in indoor building environments. *Automation in Construction*, 119:103359, 2020.
 - [96] Sami El Ferik, Mohammad Tariq Nasir, and Uthman Baroudi. A behavioral adaptive fuzzy controller of multi robots in a cluster space. *Applied Soft Computing*, 44:117–127, 2016.
 - [97] EG Hernandez-Martinez and JM Flores Albino. Hybrid architecture of multi-robot systems based on formation control and som neural networks. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 941–946. IEEE, 2011.
 - [98] Liang Han, Yuxin Xie, Xiaoduo Li, Xiwang Dong, Qingdong Li, and Zhang Ren. Time-varying group formation tracking control for second-order multi-agent systems with communication delays and multiple leaders. *Journal of the Franklin Institute*, 357(14):9761–9780, 2020.
 - [99] Yong Xu, Malathy Batumalay, Choon Kit Chan, Walton Wider, Kui Fu, Lingmin Yang, and Jiansheng Peng. Hot topics and frontier evolution of formation control research in multiple robots. *Journal of Robotics*, 2025(1):1–18, 2025.
 - [100] Hasan Kivrak, Muhammed Zahid Karakusak, Simon Watson, and Barry Lennox. Cyberphysical system architecture of autonomous robot ecosystem for industrial asset monitoring. *Computer Communications*, 218:72–84, 2024.
 - [101] Pengfei Li, Jiaxin Fan, and Jianhong Wu. Exploring the key technologies and applications of 6g wireless communication network. *iScience*,

28(5):112281, 2025.

- [102] Wagdy M. Othman, Abdelhamied A. Ateya, Mohamed E. Nasr, Ammar Muthanna, Mohammed ElAffendi, Andrey Koucheryavy, and Azhar A. Hamdi. Key enabling technologies for 6g: The role of uavs, terahertz communication, and intelligent reconfigurable surfaces in shaping the future of wireless networks. *Journal of Sensor and Actuator Networks*, 14(2), 2025.
- [103] Pierre E. Kehl, Junaid Ansari, Mikael Lovrin, Praveen Mohanram, Chi-Chuan (Eric) Liu, Jun-Lin (Larry) Yeh, and Robert H. Schmitt. 5g-tsn integrated prototype for reliable industrial communication using frame replication and elimination for reliability. *Electronics*, 14(4), 2025.
- [104] Nazish Tahir and Ramviyas Parasuraman. Edge computing and its application in robotics: A survey. *Journal of Sensor and Actuator Networks*, 14(4), 2025.
- [105] Chenzhuolei Chao, Haoran Tan, Xueming Zhang, Gang Wang, You Wu, and Yaonan Wang. Data-driven adaptive formation control based on preview mechanism for networked multi-robot systems with communication delays. *Neurocomputing*, 620:129151, 2025.
- [106] Will Ferenc, Hannah Kastein, Lauren Lieu, Ryan Wilson, Yuan Rick Huang, Jérôme Gilles, Andrea L. Bertozzi, Balaji R. Sharma, Baisra-van HomChaudhuri, Subramanian Ramakrishnan, and Manish Kumar. Robot swarming over the internet. *arXiv preprint arXiv:2411.03614*, 2024.
- [107] Jakub Krejčí, Marek Babiuch, Jiří Suder, Václav Krys, and Zdenko Bobovský. Latency-sensitive wireless communication in dynamically moving robots for urban mobility applications. *Smart Cities*, 8(4), 2025.
- [108] Lei Cao, Da-Wei Zhang, Clara Mihaela Ionescu, and Guo-Ping Liu. A distributed predictive formation control strategy for cyber-physical multi-agent systems under communication constraints. *Information Sciences*, 641:119092, 2023.
- [109] Beatrice Capelli, Hassan Fouad, Giovanni Beltrame, and Lorenzo Sabattini. Decentralized connectivity maintenance with time delays using control barrier functions. *arXiv preprint arXiv:2103.12614*, 2021.
- [110] Victoria Edwards, Philip deZonia, M. Ani Hsieh, Jason Hindes, Ioana

- Triandaf, and Ira B. Schwartz. Delayinduced swarm pattern bifurcations in mixed reality experiments. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(7):1–14, 2020.
- [111] Yongce Liu and Zhongqiang Ren. A probabilistic measure of multirobot connectivity and ergodic optimal control. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2025. Paper 100.
 - [112] Tianyu Zhang, Gang Wang, Chuanyu Xue, Jiachen Wang, Mark S. Nixon, and Song Han. Time sensitive networking (tsn) for industrial automation: Current advances and future directions. *ACM Computing Surveys*, 57(2):Article 30 (38 pp.), 2025. NASA NTRS Document ID 20250002365.
 - [113] Nazish Tahir and Ramvijas Parasuraman. Edge computing and its application in robotics: A survey. *arXiv preprint arXiv:2507.00523*, 2025.
 - [114] Zhihui Du, Xiaobo Qu, Jingping Shi, and Jingchao Lu. Formation control of fixed-wing uavs with communication delay. *ISA Transactions*, 146:154–164, 2024.
 - [115] Zhenfeng Jiang, Hongchun Qu, Guanghui Liu, and Zhihao Cao. Distributed model predictive control for multi-agent systems with packet loss and actuator saturation. *Authorea Preprints*, 2023.
 - [116] Weijie Weng and Fanglai Zhu. Observer-based event-triggered consensus control for multi-agent systems with nonlinearity and unknown inputs. *International Journal of Robust and Nonlinear Control*, 34(2):e7102, 2024.
 - [117] Faheem Gulzar, Noor Muhammad Khan, Yasir Awais Butt, and Aamer Iqbal Bhatti. Constraint-oriented formation control of multi-robot system in leaderless consensus under confined conditions. *Systems Science & Control Engineering*, 12(1):2436666, 2024.
 - [118] Giuseppe Silano, Alvaro Caballero, Davide Liuzza, Luigi Iannelli, Stjepan Bogdan, and Martin Saska. A signal temporal logic approach for task-based coordination of multi-aerial systems: a wind turbine inspection case study. *Robotics and Autonomous Systems*, 186:104905, 2025.
 - [119] Hyeon-Woo Na and PooGyeon Park. Lmi approach of h_∞ consensus

- for multi-agent systems under markov switching topology by dynamic output-feedback controller. *ISA Transactions*, 157:1–10, 2025.
- [120] Ayush Rai and Shaoshuai Mou. Safe region multi-agent formation control with velocity tracking. *Systems & Control Letters*, 186:105776, 2024.
 - [121] Gang Wang, Zongyu Zuo, and Chaoli Wang. Robust consensus control of second-order uncertain multi-agent systems with velocity and input constraints. *Automatica*, 157:111226, 2023.
 - [122] R. Funada, X. Cai, G. Notomista, M. W. S. Atman, J. Yamauchi, M. Fujita, and M. Egerstedt. Coordination of robot teams over long distances: From georgia tech to tokyo tech and back-an 11,000-km multirobot experiment. *IEEE Control Systems Magazine*, 40(4):53–79, 2020.
 - [123] Riku Funada, Xiaoyi Cai, Gennaro Notomista, Made Widhi Surya Atman, Junya Yamauchi, Masayuki Fujita, and Magnus Egerstedt. Coordination of robot teams over long distances: From georgia tech to tokyo tech and back-an 11,000-km multirobot experiment. *IEEE Control Systems Magazine*, 40(4):53–79, 2020.
 - [124] Yindong Lian, Qifan Yang, Yanru Liu, and Wei Xie. A spatio-temporal constrained hierarchical scheduling strategy for multiple warehouse mobile robots under industrial cyberphysical system. *Advanced Engineering Informatics*, 52:101572, 2022.
 - [125] Giovanni Carabin and Lorenzo Scalera. On the trajectory planning for energy efficiency in industrial robotic systems. *Robotics*, 9(4), 2020.
 - [126] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.
 - [127] Sabine Mondi and Vladimir L. Kharitonov. Exponential estimates for retarded time-delay systems: An lmi approach. *IEEE Transactions on Automatic Control*, 50(2):268–273, 2005.
 - [128] Zvi Artstein. Linear systems with delayed controls: A reduction. *IEEE Transactions on Automatic Control*, 27(4):869–879, August 1982.
 - [129] Wook Hyun Kwon and PooGyeon Park. *Stabilizing and Optimizing*

Control for Time-Delay Systems: Including Model Predictive Controls. Communications and Control Engineering. Springer, Cham, 2019.

- [130] Kento Fujita and Daisuke Tsubakino. A predictor-based approach to communication delay compensation for multi-agent systems. *IEEE Access*, 12:196148–196158, 2024.
- [131] Yang Deng, Vincent, Emmanuel Moulay, Zhang Chen, Bin Liang, Franck Plestan, and Qing Long Han. Predictor based control of time delay systems: A survey. *International Journal of Systems Science*, 53(12):2496–2534, 2022.
- [132] Bin Zhou and Alexey V. Egorov. Razumikhin and krasovskii stability theorems for time-varying time-delay systems. *Automatica*, 71:281–291, 2016.
- [133] Wei Yin Leong, Hieu Trinh, and Le Van Hien. An lmi-based functional estimation scheme of large-scale time-delay systems with strong interconnections. *Journal of the Franklin Institute*, 353(11):2482–2510, 2016.
- [134] Gennaro Notomista, Xiaoyi Cai, Junya Yamauchi, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. *arXiv preprint arXiv:1904.04801*, 2019.
- [135] Adrian E. Onyeka, Xing-Gang Yan, and Jianqiu Mu. Sliding mode control of time-delay systems with delayed nonlinear uncertainties. *IFAC-PapersOnLine*, 50(1):2696–2701, 2017. 20th IFAC World Congress.
- [136] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [137] Richard S. Varga. *Gershgorin and His Circles*, volume 36 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2004.
- [138] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- [139] Johan Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan,

2004.

- [140] Huber Giron-Nieto, Oliver Ochoa-Garcia, Eduardo Gamaliel Hernandez-Martinez, Mario Ramirez-Neria, Guillermo Fernández Anaya, Enrique D. Ferreira, and Jose-Job Flores-Godoy. Cyber-physical multi-robot formation: Virtual agents approach and low-cost experiments. In *Proc. of Congreso Nacional de Control Automático*, pages 507–512, 2023.
- [141] Huber Giron-Nieto, Eduardo Gamaliel Hernandez-Martinez, Guillermo Fernandez-Anaya, Enrique D. Ferreira-Vazquez, José-Job Flores-Godoy, Mario Ramírez-Neria, and Andres Molano-Jimenez. Cyberphysical multi-robot formation with a communication delays and a virtual agent approach. *Electronics*, 14(9), 2025.
- [142] Huber Giron Nieto. Cyberphysical multirobot formation: Virtual agents approach and lowcost experiments. 3er Simposio Anual del Grupo de Investigación en Ciencias Básicas, nov 2023. Conference presentation, 17 November 2023.

Apéndice A

Listado de empresas fabricantes de robots AGV

Proveedor	País	Descripción de la empresa
AGILOX	Austria	https://agilox.net/en/agilox-igv
DS AUTOMOTION	Austria	https://www.ds-automotion.com/
EGEMIN AU- TOMATION DEMATIC	Bélgica	https://www.dematic.com/agv
MOTUM	Bélgica	http://motum.be/
CASUN	China	http://www.casun.cn/en/index.html
COTEK	China	http://www.cotek-robotics.com/products/products1/productS3/
Zhejiang Guozi Robotics	China	http://www.smartlogisticsolutions.com/about/index.html
GEEK+	China	https://www.geekplusrobotics.com/
HANGZHOU HIKROBOT TECHNOLOGY CO.,LTD	China	http://en.hikrobotics.com/robot/robotlist.htm
IKV ROBOT	China	https://www.ikvrobot.com/
JIADA	China	http://www.jduagv.com/
KSEC	China	https://www.cstckm.cn/

Shenzhen Mircolomay Technology Co. Ltd.	China	https://www.agvnetwork.com/agv-producers/mircolomay
SIASUN	China	http://www.siasun.com/
SUZHOU AGV	China	http://www.agvsz.com/
LOGIC ROBOTICS	China	https://logic-os.com/
WELLWIT ROBOTICS	China	https://www.wellwit.com.cn/
ROCLA	Finlandia	https://www.rocla.com/
SOLVING	Finlandia	https://www.solving.com/
AXTER AUTOMATION	Francia	https://www.axter.net/
BALYO	Francia	https://www.balyo.com/
BA SYSTEMES	Francia	https://www.basystemes.com/
OPTIMUS ROBOTICS	Francia	https://www.optimus-robotics.com/
SCALLOG	Francia	https://www.scallog.com/
CREFORM	Alemania	http://www.creform.de/home/
EK AUTOMATION	Alemania	https://ek-automation.com/
GRENZEBACH	Alemania	https://www.grenzebach.com/
JUNGHEINRICH	Alemania	https://www.jungheinrich.com/
KOKEISL	Alemania	http://www.kokeisl.eu/
LOGISTICJET	Alemania	https://www.logistic-jet.com/
LINDE	Alemania	https://www.linde-mh.com/
MLR SYSTEM	Alemania	https://www.mlr.de/
SAFELOG	Alemania	https://www.safelog.de/
STILL	Alemania	https://www.still.de/
SWISSLOG	Suiza	https://www.swisslog.com/
GreyOrange	India	https://www.greyorange.com/
GRIDBOTS	India	https://www.gridbots.com/
SIMPLEX ROBOTICS	India	https://www.simplexrobotics.com/
The Hi-tech Robotic Systemz Ltd.	India	https://www.hitechroboticsystemz.com/
CASSIOLI	Italia	https://www.cassioli.com/
COMAU	Italia	https://www.comau.com/

DIMASIMMA	Italia	https://www.dimasimma.com/
ELETTRIC80	Italia	https://www.elettric80.com/
INDEVA	Italia	https://www.indevagroup.com/
MAD AUTOMATION	Italia	https://www.madautomation.com/
MOVINCAR	Italia	https://www.movincar.com/
OCME	Italia	https://www.ocme.com/
OPPENT	Italia	https://www.oppent.com/
PROXAUT	Italia	https://www.proxaut.com/
TECNOFERRARI	Italia	https://www.tecnoferrari.com/
AICHIKIKAI	Japón	https://www.aichikikai.co.jp/
DAIFUKU	Japón	https://www.daifuku.com/
KOLEC	Japón	https://www.kolec.co.jp/
MEIDEN	Japón	https://www.meidensha.com/
EUROTEC	Países Bajos	https://www.eurotec.com/
MOVIGO	Países Bajos	https://www.movigo.nl/
OCEANEERING	Países Bajos	https://www.oceaneering.com/
WEWO EUROPE	Países Bajos	https://www.wewo-europe.com/
AIUT Sp. z o.o.	Polonia	https://aiut.com/
LEAN-TECH	Polonia	https://lean-tech.pl/
RB POLAND	Polonia	https://rbpoland.pl/
ACTIVE SPACE AUTOMATION	Portugal	https://www.activespaceautomation.com/
PINTO BRASIL	Portugal	https://pintobrasil.com/
CEIT AGV	Eslovaquia	https://www.asseco-ceit.com/
AMS Robotics	España	https://www.ams-robotics.com/
ASTI MOBILE ROBOTICS	España	https://www.astimobilerobotics.com/
DTA	España	https://dta.es/
KIVNON	España	https://www.kivnon.com/
ROBOTNIK	España	https://robotnik.eu/
SYNERSIGHT	España	https://synersight.es/

AGVE	Suecia	https://www.agvegroup.com/
ATAB & MAX AGV	Suecia	https://www.atab.se/
MAX AGV	Suecia	https://www.maxagv.com/
ESATROLL	Suiza	https://www.esatroll.com/
iROB	Turquía	https://www.irobagv.com/
MODOYA	Turquía	https://www.modoya.com/
OPPENT TUR- KEY	Turquía	https://www.oppent.com/
ROBOS AGV	Turquía	https://robos.com.tr/
TRANSOLT	Turquía	https://www.transolt.de/
AGVE SOLU- TIONS	Turquía	https://www.agvegroup.com/
AMERDEN AGVS	EE. UU.	https://www.amerden.com/
America In Motion (AIM)	EE. UU.	https://www.aim-agv.com/
AUTOGUIDE	EE. UU.	https://agmobilerobots.com/
DAIFUKU (USA)	EE. UU.	https://www.daifukuamerica.com/
DEMATIC	EE. UU.	https://www.dematic.com/
ECKHART	EE. UU.	https://www.eckhartusa.com/
FETCH ROBO- TICS	EE. UU.	https://fetchrobotics.com/
FORI AUTOMA- TION	EE. UU.	https://www.foriauto.com/
HYSTER-YALE	EE. UU.	https://www.hyster.com/
JBT	EE. UU.	https://www.jbtc.com/
LOCUS ROBO- TICS	EE. UU.	https://locusrobotics.com/
OTTO MOTORS	EE. UU.	https://ottomotors.com/
SAVANT	EE. UU.	https://www.savantautomation.com/
SEEGRID	EE. UU.	https://seegrid.com/
TRANSBOTICS SCOTT AUTOMA- TION	EE. UU.	https://www.scottautomation.com/
VECNA ROBO- TICS	EE. UU.	https://www.vecnarobotics.com/
WARD SYSTEMS	EE. UU.	https://wardventures.com/

Apéndice B

Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments

Giron-Nieto, Huber; Ochoa-Garcia, Oliver; Hernandez-Martinez, Eduardo Gamaliel; Ramirez-Neria, Mario; Fernández Anaya, Guillermo; Ferreira, Enrique D.; Flores-Godoy, Jose-Job. Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments. Memorias del Congreso Nacional de Control Automático, pp. 507-512, 2023. <https://doi.org/10.58571/CNCA.AMCA.2023.105> [140]

Cyber-physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments ^{*}

Huber Giron-Nieto ^{*} Oliver Ochoa-García ^{*}
Eduardo Gamaliel Hernandez-Martinez ^{**}
Mario Ramírez-Neria ^{**} Guillermo Fernandez-Anaya ^{***}
Enrique D. Ferreira-Vazquez ^{****} Jose Job Flores-Godoy [†]

^{*} *Department of Science and engineering, Universidad Iberoamericana Puebla, 072820 Puebla, México. (e-mail: huber.giron2@iberopuebla.mx, oliver.ochoa2@iberopuebla.mx).*

^{**} *Institute of Applied Research and Technology, Universidad Iberoamericana Ciudad de México, 01219, Mexico City, Mexico. (e-mail: eduardo.gamaliel@ibero.mx, mario.ramirez@ibero.mx)*

^{***} *Physics and Math Department, Universidad Iberoamericana Ciudad de México, 01219, Mexico City, Mexico. (e-mail: guillermo.fernandez@ibero.mx)*

^{****} *Engineering Department, Universidad Católica del Uruguay, Montevideo 11600, Uruguay. (enrique.ferreira@ucu.edu.uy)*

[†] *Exact and Natural Sciences Department, Universidad Católica del Uruguay, Montevideo 11600, Uruguay. (jose.flores@ucu.edu.uy)*

Abstract: A cyber-physical formation includes all the strategies to coordinate mobile robots moving in different physical workspaces sharing information through internet and the cloud. This work addresses a formation scheme of robots moving in two different workspaces. The control strategy is based on virtual agents used like "avatars" which must converge to the position of the robots in the opposite workspace. The control approach is designed for robots modeled as single integrators and extended to the case of unicycle-type robots. Results of numerical simulations and real experiments are shown using a low-cost cyber-physical micro-robot platform based on a camera as position and orientation sensor.

Keywords: Cyber-physical, formation control, mobile robots, unicycles, multi-agent systems.

1. INTRODUCTION

The motion coordination of multiple mobile robots is an important theoretical and technological research area in the industrial and service fields (Canudas de Wit et al., 1996; Valentim et al., 2019). Groups of robots have been widely used for logistics (Hernandez-Martinez et al., 2014), the transportation of large objects (Farugia and Fabri, 2018) surveillance and perimeter vigilance (Mantha et al., 2020), among others. The main contributions have been addressed for differential-drive or unicycle-type wheeled robots (González-Sierra et al., 2013), omnidirectional robots (Hernandez-Martinez et al., 2013) and car-like configurations. The basic group coordination problems are: formation control (Ferreira-Vazquez

et al., 2016), the formation tracking or marching control (Oh et al., 2015), and the inter-robot collision avoidance (Lopez-Gonzalez et al., 2016).

The fourth industrial revolution, known as Industry 4.0, incorporates new enabling technologies for a better communicated world, as in Gallo et al. (2021). Wireless communication devices with high-bandwidth, the use of the cloud for data storage and analysis and the embedded computing architectures promote new opportunities for an enhanced distributed motion coordination of mobile robots. Thus, the robots can achieve coordinated motion across different and remote workspaces. Besides, the formation control laws could be executed onboard the robots themselves or in the cloud. Therefore, the group coordination becomes a cyber-physical setup with decentralized real hardware working online with the internet and the cloud, Escobar et al. (2020).

^{*} The authors acknow the financial support from Universidad Iberoamericana Puebla through the Institute Design and Technological Innovation (IDIT), and from Universidad Iberoamericana Ciudad de México through the project fund DINVP-0051.

The cyber-physical formation schemes are a recent focus for the multi-robot control community. Recent advances are shown in Lian et al. (2021), Pikner et al. (2021), Kruglova et al. (2019). The formation problem across different workspaces, sharing information through the internet, generates new challenges for the control schemes. Kim et al. (2008) have studied the effect of delays in the internet communication, Lo et al. (2003) the problem of communication losses between workspaces and Shahzad and Roth (2016) for the intermittence in the internet links.

Recently, some robotics platforms have been built for the experimentation of cyber-physical formation approaches. For example in Escobar et al. (2020), a multi-robot platform is designed for educational purposes with cyber-physical features using omnidirectional mobile robots and MQTT protocol. Thus, the study of cyber-physical behaviors in multi-robot systems requires a flexible platform that enables monitoring and control of the multi-workspace setup.

This work is a first stage for the modeling, control and experimentation of cyber-physical formations of mobile robots. The main contributions are highlighted by the next points:

- The design and implementation of a control strategy for a cyber-physical formation of mobile robot systems moving in two workspaces.
- The control approach is based on the use of virtual agents which converge to the physical robots moving in the opposite workspace. The convergence is analyzed for the general case of n robots. Under this control scheme the robots behave as if they move in a unique workspace.
- The control approach is developed for 2D single integrator robots, but it is extended to unicycle-type robots with two wheels and it is validated by numerical simulations.
- Finally, a novel low cost experimental platform is built. An experiment is shown using four robots in two different physical spaces communicated through an internet connection with a MQTT protocol.

The paper is organized as follows. The problem definition is given in Section 2. The control strategy is presented in the Section 3. Numerical simulations are shown in Section 4. The experimental platform and results are shown in Section 5. Finally, some concluding remarks are proposed in Section 6.

2. PROBLEM FORMULATION

In this section, the cyber-physical formation problem is defined for n point-robots moving in two different workspaces.

Let $\mathcal{R} = \{R_1, \dots, R_n\}$ be a set of mobile robots. Consider that these n robots move in two different workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$. Without loss of generality, consider that

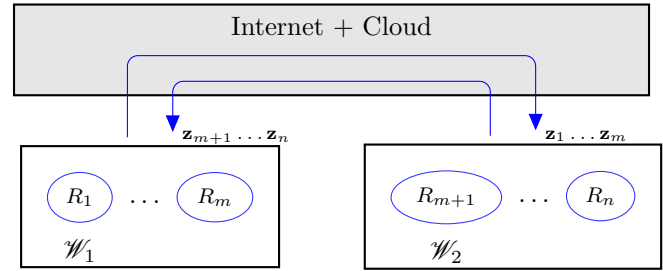


Fig. 1. Communication between workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$.

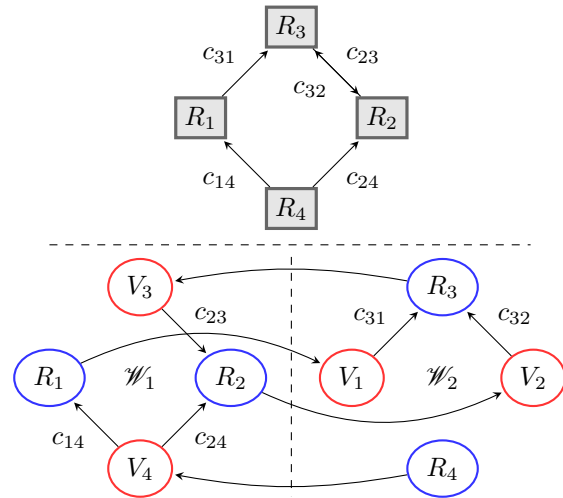


Fig. 2. Example of the relationship between the original formation graph $\mathcal{G}$ (above) and workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$ and combined graph $\hat{\mathcal{G}}$ (below).

$\mathcal{R}_1 = \{R_1, \dots, R_m\}$, with $1 < m < n$ is the subset of robots moving in $\mathcal{W}_1$ and $\mathcal{R}_2 = \{R_{m+1}, \dots, R_n\}$ is the subset of robots moving in $\mathcal{W}_2$. Therefore $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ with $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Define $\mathbf{z}_i = [x_i, y_i]^\top \in \mathbb{R}^2$ as the 2D position of R_i in $\mathcal{W}_1$ or $\mathcal{W}_2$.

The main goal of this work is to construct a formation control strategy such that the robots in the two workspaces converge to a global formation as if they were in the same space. To do this, the information about the positions $\mathbf{z}_i$, $i = 1, \dots, n$ must be shared between $\mathcal{W}_1$ and $\mathcal{W}_2$ as illustrated in Fig. 1 using internet and the cloud. Let us assume a desired global formation graph $\mathcal{G}$ for n robots is defined by an adjacency matrix $\mathbf{A}$, a set of relative positions $\mathcal{C}$ and a Laplacian matrix $\mathbf{L}$. The challenge in a cyber-physical configuration is to converge to the formation $\mathcal{G}$ with the absence of certain robots in each workspace. This paper proposes the use of virtual robots. Thus, for each $R_i \in \mathcal{R}_1$, $i = 1, \dots, m$ define a virtual robot $V_i \in \mathcal{V}_2$, $i = 1, \dots, m$ and for each $R_i \in \mathcal{R}_2$, $i = m+1, \dots, n$ define a virtual robot $V_i \in \mathcal{V}_1$, $i = m+1, \dots, n$. Thus, there are also n virtual robots divided in $\mathcal{V}_1$ and $\mathcal{V}_2$. Let $\mathbf{v}_i \in \mathbb{R}^2$ be the 2D position of each virtual robot V_i . Finally, $\mathcal{W}_i = \mathcal{R}_i \cup \mathcal{V}_i$, $i = 1, 2$.

Consider that both, the real and virtual robots are modeled as single integrators as follows:

$$\begin{aligned}\dot{\mathbf{z}}_i &= \mathbf{u}_i, & i &= 1, \dots, n \\ \dot{\mathbf{v}}_i &= \mathbf{w}_i, & i &= 1, \dots, n\end{aligned}\quad (1)$$

where $\mathbf{u}_i \in \mathbb{R}^2$ and $\mathbf{w}_i \in \mathbb{R}^2$ are the control inputs for the real and virtual robots, respectively.

For all robots in any workspace $\mathcal{W}_i$ a new formation graph $\mathcal{G}$ is derived from $\mathcal{G}$ as follows. For each $R_j \in \mathcal{R}_i$ let $\mathcal{N}_{ij}^R$ and $\mathcal{N}_{ij}^V$ be the adjacent subsets of indexes of the real and virtual robots in $\mathcal{W}_i$ who are communicated to R_j according to the following:

$$\begin{aligned}\mathcal{N}_{ij}^R &= \{k | \mathbf{A}_{jk} \neq 0, R_j, R_k \in \mathcal{R}_i\} \\ \mathcal{N}_{ij}^V &= \{k | \mathbf{A}_{jk} \neq 0, R_j \in \mathcal{R}_i, V_k \in \mathcal{V}_i\}\end{aligned}\quad (2)$$

Besides, all virtual robots V_j are communicated only to their real counterpart R_j located in the other workspace $\mathcal{W}_{3-i}$. An example of the communication topology is shown in Fig. 2. On the other side, for each $(j, k) | k \in \mathcal{N}_{ij}^R \cup \mathcal{N}_{ij}^V$ let $\mathbf{c}_{kj}$ define the static and predefined desired relative position vector $\mathbf{c}_{kj} \in \mathbb{R}^2$, $\mathbf{c}_{kj} \in \mathcal{C}$. By design, it is straightforward to show that if $\mathcal{G}$ has a directed spanning tree then $\mathcal{G}$ also has one. Having a directed spanning tree, it is well-known that the laplacian matrix of $\mathcal{G}$ and $\mathcal{G}$ given by $\mathbf{L}$ and $\hat{\mathbf{L}}$ respectively, have the eigenvalue set $E = \{\lambda_1, \dots, \lambda_n\}$ with $\lambda_1 = 0$ and $\lambda_i > 0$, $i = 2, \dots, n$ (Olfati-Saber, 2007).

3. CONTROL STRATEGY

The control strategy needs to have at least a leader robot. Let us define $R_n \in \mathcal{W}_2$ as the leader robot, and $\mathbf{z}_d, \dot{\mathbf{z}}_d \in \mathbb{R}^2$ and as the desired position and velocity of the trajectory for the leader. The control strategy for the virtual robots $V_j \in \mathcal{W}_i$ is designed to track its corresponding real robot R_j that belongs to the other workspace $\mathcal{W}_{3-i}$, according to the following law:

$$\mathbf{w}_j = k_v(\mathbf{z}_j - \mathbf{v}_j) + \dot{\mathbf{z}}_d, \quad k_v > 0, j = 1, \dots, n. \quad (3)$$

with $k_v > 0$ the control gain for the virtual robots. The control law for the real robots R_j is given by the following:

$$\begin{aligned}\mathbf{u}_j &= k_r \sum_{k \in \mathcal{N}_{ij}^R} (\mathbf{z}_k + \mathbf{c}_{kj} - \mathbf{z}_j) + \dots \\ &+ k_r \sum_{k \in \mathcal{N}_{ij}^V} (\mathbf{v}_k + \mathbf{c}_{kj} - \mathbf{z}_j) + \dot{\mathbf{z}}_d,\end{aligned}\quad (4)$$

with $k_r > 0$ the control gain for the real robots and $j = 1, \dots, n-1$. The control strategy for the leader robot R_n is given by

$$\begin{aligned}\mathbf{u}_n &= k_r \sum_{k \in \mathcal{N}_{2n}^R} (\mathbf{z}_k + \mathbf{c}_{kn} - \mathbf{z}_n) + \dots \\ &+ k_r \sum_{k \in \mathcal{N}_{2n}^V} (\mathbf{v}_k + \mathbf{c}_{kn} - \mathbf{z}_n) + k_c(\mathbf{z}_d - \mathbf{z}_n) + \dot{\mathbf{z}}_d.\end{aligned}\quad (5)$$

Defining the relative positions of the robots with respect to the desired trajectory $\tilde{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{z}_d$ and $\tilde{\mathbf{v}}_j = \mathbf{v}_j - \mathbf{z}_d$,

equations (3),(4) and (5) have a relative consensus defined by

$$\begin{aligned}\tilde{\mathbf{v}}_j &= \tilde{\mathbf{z}}_j^*, \quad \forall j \\ 0 &= \sum_{k \in \mathcal{N}_{ij}^R} (\tilde{\mathbf{z}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^*) + \dots \\ &+ \sum_{k \in \mathcal{N}_{ij}^V} (\tilde{\mathbf{v}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^*) \quad \forall j < n \\ \tilde{\mathbf{z}}_n^* &= 0.\end{aligned}\quad (6)$$

Let $e_j^R = \tilde{\mathbf{z}}_j - \tilde{\mathbf{z}}_j^*$ and $e_j^V = \tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_j^*$ be the relative error with respect to the consensus trajectory of the real and virtual robots respectively. Using the above equations, the error dynamics can be expressed as

$$\begin{bmatrix} \dot{\mathbf{e}}^V \\ \dot{\mathbf{e}}^R \end{bmatrix} = \left(\begin{bmatrix} -k_v \mathbf{I}_n & k_v \mathbf{I}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} + k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \otimes \mathbf{I}_2 \right) \begin{bmatrix} \mathbf{e}^V \\ \mathbf{e}^R \end{bmatrix} \quad (7)$$

where

$$\mathbf{e}^V = [e_1^V \dots e_n^V]^\top \quad (8)$$

$$\mathbf{e}^R = [e_1^R \dots e_n^R]^\top \quad (9)$$

$$\mathbf{E}_{21} + \mathbf{E}_{22} = -\mathbf{L} \quad (10)$$

$$\mathbf{q}_n = [\mathbf{0}_{n-1}^\top \ 1]^\top. \quad (11)$$

with $\mathbf{0}_{n-1}$ a zero vector with dimension $n-1$.

Using the fact that the communication topology is given by $\mathcal{G}$, it can be shown that the system dynamics (7) converges to the origin, achieving relative consensus given by equations (6). A Gershgorin disk-based proof is omitted here for reasons of space.

4. NUMERICAL SIMULATION

The numerical simulation was programmed in Matlab-Simulink©R2022b with a variable-step continuous implicit solver.

Fig. 3 shows the numerical simulation of the control approach with $n = 4$, and $m = 2$, i.e. the total set of robots is $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, where $\mathcal{R}_1 = \{R_1, R_2\} \in \mathcal{W}_1$ and $\mathcal{R}_2 = \{R_3, R_4\} \in \mathcal{W}_2$ as shown in Fig. 3a. The leader robot is R_4 . Therefore $\mathcal{V}_1 = \{V_3, V_4\}$ and $\mathcal{V}_2 = \{V_1, V_2\}$, and the total members of each workspace are $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ and $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.

The adjacent subsets for the real robots are defined as $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_3, V_4\}$, $\mathcal{N}_{12}^R = \emptyset$, $\mathcal{N}_{12}^V = \{V_3, V_4\}$, $\mathcal{N}_{23}^R = \emptyset$, $\mathcal{N}_{23}^V = \{V_1, V_2\}$, $\mathcal{N}_{24}^R = \emptyset$, $\mathcal{N}_{24}^V = \{V_1, V_2\}$ as shown in Fig. 3a. The desired final posture of the robots is a diamond-shaped formation, as if all 4 real robots would be in the same workspace. To achieve this desired formation pattern, the relative vectors are given by $c_{13} = [a, b]^\top$, $c_{14} = [a, -b]^\top$, $c_{23} = [-a, b]^\top$, $c_{24} = [-a, -b]^\top$, with $a = 280$, $b = 140$ and by construction $c_{31} = -c_{13}$, $c_{41} = -c_{14}$, $c_{32} = -c_{23}$ and $c_{42} = -c_{24}$.

Fig. 3b shows the trajectories of the real and virtual robots using the control law (3)-(4)-(5) with $k_v = 100$,

$k_r = 100$, $k_c = 100$ and the desired trajectory of the leader robot R_4 given by $\mathbf{z}_d = [72 \cos t, 84 + 72 \sin t]^\top$. Note that the robots converge to the desired formation pattern which is observed by the errors' convergence shown in the Fig. 3c. Finally, the control inputs are depicted in the Fig. 3d.

5. EXPERIMENTAL WORK

5.1 Experimental platform

The control approach was proved in a novel low-cost experimental robotics platform composed by the four mobile robots shown in Fig. 4. Each unicycle-type robots are integrated by two DC motors manufactured by Pololu© with plastic wheels, controlled by an Arduino pro-micro with Bluetooth wireless communicated to a PC. The case of the robots is made with 3D printer with an Aruco-Tag on the top.

The control implementation is based on the scheme presented in Fig. 5. The four robots are placed in two different workspaces. Every workspace has a Logitech© web camera model C920 with a resolution of 1280×720 pixels, installed to a height of 40 mm, that generates a viewed area of around $75 \text{ mm} \times 40 \text{ mm}$. Therefore, the resolution of every square pixel is about 0.36 mm^2 . Every camera is connected to an Intel© core i7 PC. The PC reads the Aruco tags at a sample rate of 0.1 s detecting the *ID* of the robots and calculating the position and orientation of every robot using the OpenCV visual processing toolbox. The positions and orientations of the robots are sent to the internet through a MQTT IoT protocol using a HIVEMQ© broker. The communication is bidirectional and the two workspaces are connected online. The control law is programmed in every PC using Phyton version 3.10 at the same sample rate of 0.1s, sending the control inputs to every robot using the Bluetooth wireless communication. Note that the control setup is low-cost and scalable. Also, the experimental setup is modular using commercial components allowing its expansion to more robots and workspaces. The use of the HIVEMQ© broker is free. Thus, this experimental setup becomes the base of a ad-hoc platform for proving future and more complex cyber-physical formation approaches.

5.2 Extension of the control to unicycle-type robots

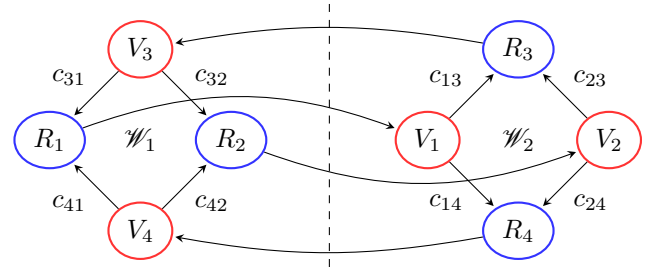
The kinematic model of the unicycle-type robots shown in Fig. 4 is given, according to the Fig. 6, by

$$\dot{x}_i = v_i \cos \theta_i \quad (12)$$

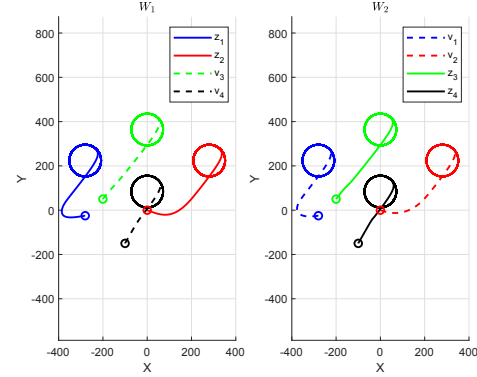
$$\dot{y}_i = v_i \sin \theta_i \quad (13)$$

$$\dot{\theta}_i = w_i \quad (14)$$

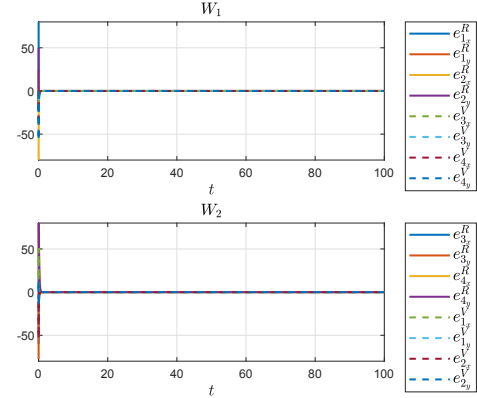
with $i = 1, \dots, n$, where $[x_i, y_i]^\top$ is the 2D coordinate of the robot and θ_i is its orientation angle with respect to the horizontal axis. The v_i and w_i are the longitudinal and rotational control inputs respect to the middle point



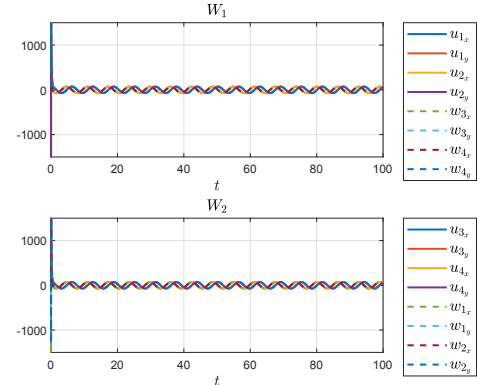
(a) Experimental workspaces and formation topology.



(b) Trajectories of robots.



(c) Convergence of the errors.



(d) Control inputs.

Fig. 3. Numerical simulation with $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ and $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.

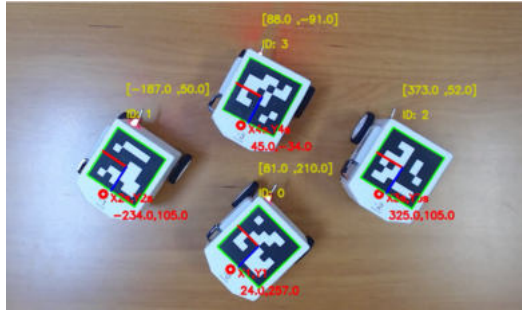


Fig. 4. Photo of the experimental robots.

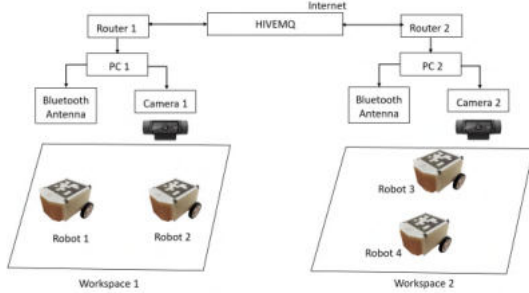


Fig. 5. Control implementation.

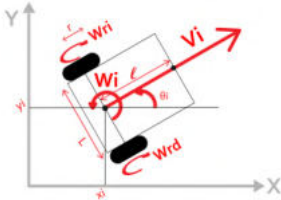


Fig. 6. Kinematic model of a unicycle-type robot.

of the axis' wheels. These two body-level velocities can be converted to the velocities of the wheels through

$$w_{R_i} = \frac{v_i}{r} + \frac{w_i L}{2r}, \quad w_{L_i} = \frac{v_i}{r} - \frac{w_i L}{2r} \quad (15)$$

where w_{R_i} and w_{L_i} are the angular velocities of the right and left wheels, respectively, r is the radius of the wheels and L is the length between the wheels.

Choosing the frontal point as control output, shown in Fig. 6, with

$$\alpha_i = [x_i + \ell \cos \theta_i, y_i + \ell \sin \theta_i]^T \quad (16)$$

where $\ell \neq 0$, its dynamics is given by

$$\dot{\alpha}_i = A_i(\theta_i)[v_i, w_i]^T \quad (17)$$

where $A_i(\theta_i) = \begin{bmatrix} \cos \theta_i & -\ell \sin \theta_i \\ \sin \theta_i & \ell \cos \theta_i \end{bmatrix}$.

Since $\det A_i(\theta_i) = \ell \neq 0$, it is possible to linearize the dynamics of α_i using the control

$$[v_i, w_i]^T = A_i^{-1}(\theta_i)\mathbf{u}_i \quad (18)$$

with $\mathbf{u}_i \in \mathbb{R}^2$ as an auxiliary control. Note that in the closed-loop control (17)-(18), the dynamics is reduced $\dot{\alpha}_i = \mathbf{u}_i$ and the nonlinearities are eliminated resulting

in the single integrator given by (1). Therefore, the control approach can be extended to the unicycle-type robots using the control law (18) where $\mathbf{u}_i$ is defined by the equations (3), (4) and (5) but dependent on the coordinates α_i of the robots.

5.3 Experimental results

Fig. 7 shows the experimental results of the simulation case developed in section 4. The parameters of the robots are $r = 15$ mm, $\ell = 10$ mm and $L = 45$ mm. The control gains are given by $k_v = 100$, $k_r = 100$, $k_c = 100$ and the desired trajectory of the leader robot R_4 is given by $\mathbf{z}_d = [72 \cos t, 84 + 72 \sin t]^T$ in millimeters. The robots converge to the desired formation pattern, which is observed by the robots' trajectories in Fig. 7a and the errors' convergence shown in Fig. 7b. Note that the errors of the real robots are smaller than the errors of the virtual robots in the experimental data. The control inputs are depicted in Fig. 7c. Note that the control performance is affected by the resolution of the camera, the sample time and the non-modeled dynamics presented by the floor friction and other real effects.

6. CONCLUSIONS

This paper addresses the modeling, control and experimentation of cyber-physical formations for mobile single integrator 2D robots over two workspaces. The control strategy employs virtual agents that converge to the physical robots moving in the opposite workspace. Convergence is guaranteed for a general case of n robots and the robots behave as if they moved in a single workspace. The proposed approach can be used for a family of non-holonomic mobile robots that can be reduced to single 2D integrators by an appropriate feedback linearization. The control strategy was validated through numerical simulations. In addition, an innovative low-cost experimental platform was built, where an experiment was run using four unicycle robots in two different physical workspaces with internet communication. Further work will address other robot dynamics and delays and intermittence effects due to the internet communication.

REFERENCES

- Canudas de Wit, C., Siciliano, B., and Bastin, G. (eds.) (1996). *Theory of Robot Control*. Springer London.
- Escobar, L., Moyano, C., Aguirre, G., Guerra, G., Al-lauca, L., and Loza, D. (2020). Multi-robot platform with features of cyber-physical systems for education applications. In *2020 IEEE ANDESCON*, 1–6.
- Farrugia, J.L. and Fabri, S.G. (2018). Swarm robotics for object transportation. In *2018 UKACC 12th International Conference on Control (CONTROL)*, 353–358.
- Ferreira-Vazquez, E.D., Hernández-Martínez, E.G., Flores-Godoy, J.J., Fernández-Anaya, G., and Paniagua-Contro, P. (2016). Distance-based formation control using angular information between robots.

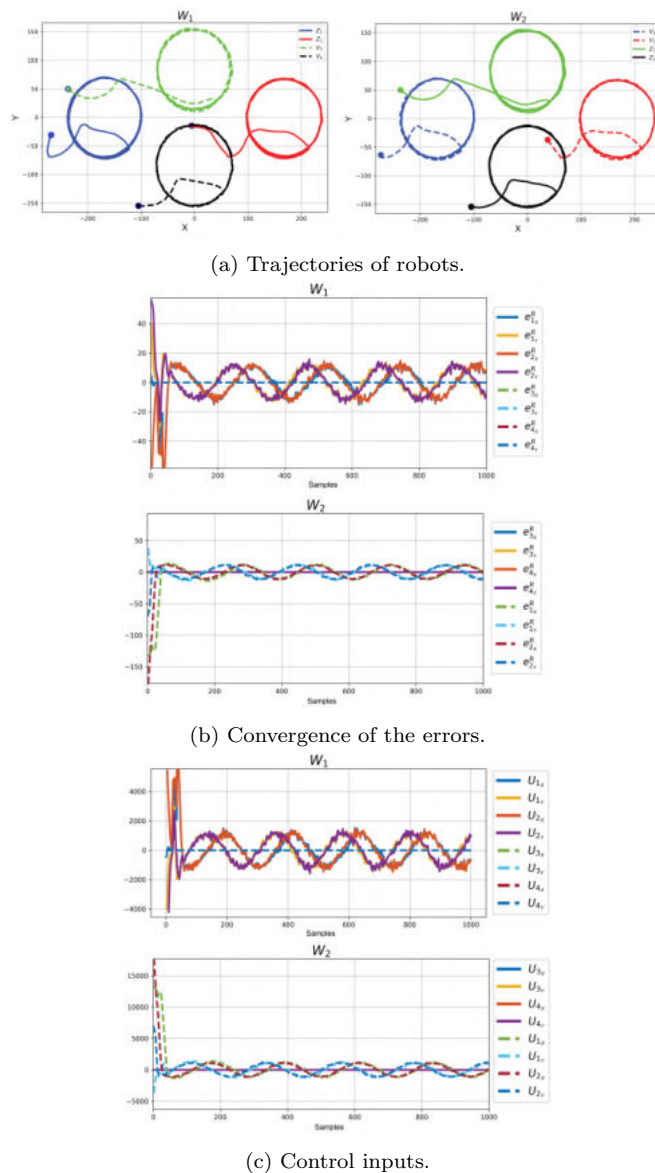


Fig. 7. Experimental work.

Journal of Intelligent & Robotic Systems, 83(3), 543–560.

Gallo, T., Cagnetti, C., Silvestri, C., and Ruggieri, A. (2021). Industry 4.0 tools in lean production: A systematic literature review. *Procedia Computer Science*, 180, 394–403. Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020).

González-Sierra, J., Aranda-Bricaire, E., and Hernandez-Martinez, E.G. (2013). Formation tracking with orientation convergence for groups of unicycles. *International Journal of Advanced Robotic Systems*, 10(3), 180.

Hernandez-Martinez, E.G., Foyo-Valdes, S.A., Puga-Velazquez, E.S., and Meda-Campaña, J.A. (2014). Hybrid architecture for coordination of agvs in fms. *Int. Journal of Advanced Robotic Systems*, 11(3), 41.

Hernandez-Martinez, E.G., Flores-Godoy, J.J., and Fernandez-Anaya, G. (2013). Decentralized discrete-time formation control for multirobot systems. *Discrete Dynamics in Nature and Society*, 2013, 746713.

Kim, K.J., Suh, I.H., Kim, S.H., and Oh, S.R. (2008). A novel real-time control architecture for internet-based thin-client robot; simulacrum-based approach. In *2008 IEEE International Conference on Robotics and Automation*, 4080–4085.

Kruglova, T., Schmelev, I., Sushkov, I., and Filatov, R. (2019). Cyber-physical system of the mobile robot's optimal trajectory planning with taking into account electric motors deterioration. In *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 1–5.

Lian, Y., Xie, W., Yang, Q., Zhang, L., Lin, D., and Zhou, Y. (2021). A novel multi-warehouse mobile robot hierarchical scheduling strategy based on industrial cyber-physical system. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 263–269.

Lo, W.T., Liu, Y.H., Elhajj, I., Xi, N., Shi, Y., and Wang, Y. (2003). Co-operative control of internet based multi-robot systems with force reflection. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, 4414–4419.

Lopez-Gonzalez, A., Ferreira, E., Hernández-Martínez, E.G., Flores-Godoy, J.J., Fernandez-Anaya, G., and Paniagua-Contro, P. (2016). Multi-robot formation control using distance and orientation. *Advanced Robotics*, 30(14), 901–913.

Mantha, B.R., Jung, M.K., García de Soto, B., Menassa, C.C., and Kamat, V.R. (2020). Generalized task allocation and route planning for robots with multiple depots in indoor building environments. *Automation in Construction*, 119, 103359.

Oh, K.K., Park, M.C., and Ahn, H.S. (2015). A survey of multi-agent formation control. *Automatica*, 53, 424–440.

Olfati-Saber, R. (2007). Design of behavior of swarms: From flocking to data fusion using microfilter networks. In *Cooperative Control of Distributed Multi-Agent Systems*, 19–41. John Wiley & Sons, Ltd.

Pikner, H., Sell, R., Karjust, K., Malayjerdi, E., and Velsker, T. (2021). Cyber-physical control system for autonomous logistic robot. In *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, 699–704.

Shahzad, A. and Roth, H. (2016). Bilateral telecontrol of automerlin mobile robot with fix communication delay. In *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 1–6.

Valentim, T., Cunha, R., Oliveira, P., Cabecinhas, D., and Silvestre, C. (2019). Multi-vehicle cooperative control for load transportation. *IFAC-PapersOnLine*, 52(12), 358–363. 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019.

Apéndice C

Cyber-Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach

Giron-Nieto, H., Hernandez-Martinez, E. G., Fernandez-Anaya, G., Ferreira-Vazquez, E. D., Flores-Godoy, J.-J., Ramírez-Neria, M., & Molano-Jimenez, A. (2025). Cyber-Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach. *Electronics*, 14(9), 1869. <https://doi.org/10.3390/electronics14091869> [141]

Article

Cyber–Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach

Huber Giron-Nieto ¹, Eduardo Gamaliel Hernandez-Martinez ^{2,*}, Guillermo Fernandez-Anaya ³,
Enrique D. Ferreira-Vazquez ⁴, José-Job Flores-Godoy ⁵, Mario Ramírez-Neria ²,
and Andres Molano-Jimenez ⁶

- ¹ Departamento de Ciencias e Ingenierías, Universidad Iberoamericana Puebla, Puebla 72820, Mexico; huber.giron2@iberopuebla.mx
- ² Instituto de Investigación Aplicada y Tecnología, Universidad Iberoamericana Ciudad de México, Mexico City 01219, Mexico; mario.ramirez@ibero.mx
- ³ Departamento de Física y Matemáticas, Universidad Iberoamericana Ciudad de México, Mexico City 01219, Mexico; guillermo.fernandez@ibero.mx
- ⁴ Engineering Department, Universidad Católica del Uruguay, Montevideo 11600, Uruguay; enferrei@ucu.edu.uy
- ⁵ Departamento de Ciencias Exactas y Naturales, Universidad Católica del Uruguay, Montevideo 11600, Uruguay; jose.flores@ucu.edu.uy
- ⁶ Departamento de Estudios en Ingeniería para la Innovación, Universidad Iberoamericana Ciudad de México, Mexico City 01219, Mexico; andres.molano@ibero.mx
- * Correspondence: eduardo.gamaliel@ibero.mx; Tel.: +52-55-5950-4014

Abstract: A cyber–physical multi-robot system integrates robotic agents that share data over communication networks in real time to achieve common objectives by making decisions collectively based on the knowledge of their surroundings. This work introduces a formation control strategy for two groups of mobile robots placed in two separate workspaces connected by a communication network. The control technique generates two similar formations on each workspace using virtual agents that mirror the behavior of the corresponding physical robot in the opposite workspace. Control laws are derived for a single integrator and unicycle-type real and virtual robots that converge to the desired formation, even in the presence of communication delays. The numerical simulations performed show the convergence of the control strategy. A low-cost cyber–physical micro-robot platform was developed to run experiments with real robots. The setup uses a camera as a position and orientation sensor and the MQTT protocol for server communication and data exchange. Results obtained on this platform show the feasibility of the approach in an actual physical setting.

Keywords: cyber–physical; formation control; mobile robots; unicycles; multi-agent systems; communication delays



Academic Editors: Cecilio Angulo and Siqi Chen

Received: 23 February 2025

Revised: 10 April 2025

Accepted: 30 April 2025

Published: 3 May 2025

Citation: Giron-Nieto, H.; Hernandez-Martinez, E.G.; Fernandez-Anaya, G.; Ferreira-Vazquez, E.D.; Flores-Godoy, J.-J.; Ramírez-Neria, M.; Molano-Jimenez, A. Cyber–Physical Multi-Robot Formation with a Communication Delays and a Virtual Agent Approach. *Electronics* **2025**, *14*, 1869. <https://doi.org/10.3390/electronics14091869>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The control and coordination of multiple mobile robots [1] is an important research area for current technological advances in the use of industrial and service robots [2]. In recent years, the use of mobile agents in logistics [3], health [4], field [5], and military defense tasks robots [6] has increased [7].

The problem formulation in mobile robotics is initially addressed with respect to the kinematic model of the mobile robot type and mechanical configuration. Some contributions have been addressed for differential-drive or unicycle-type wheeled robots [8], omnidirectional robots [9] and car-like configurations. In addition to the individual control

of mobile robots, it is possible to coordinate a group of robots for logistics and transportation tasks [10,11]. The fundamental problems in the coordination of mobile robots are formation control [12], formation tracking or marching control [13], and inter-robot collision avoidance [14–16].

The cyber–physical formation schemes are a recent focus for the multi-robot control community. These schemes include the strategies to coordinate mobile robots moving in different physical workspaces, sharing information through the internet and the cloud [17]. This new formation scheme is possible thanks to wireless communication devices with high bandwidth [18] and embedded robust computing architectures [19], which allow formation control laws that could be executed onboard the robots themselves or in the cloud. In these cyber–physical systems, the robots can achieve coordinated motion across remote workspaces.

Communication delays have an effect on cyber–physical systems workspaces, which can significantly affect their performance if they are not addressed properly. Predictor-based methods have been extensively used to estimate future states and compensate for delays [20]. Stability analysis has employed Lyapunov–Razumikin and Lyapunov–Krasovski functionals, leading to linear matrix inequality conditions that can be efficiently solved [21,22], which can handle multiple and variable delays and reach tighter conditions [23]. Other approaches include passivity-based [24], back-stepping, and sliding-mode control [25], among others.

Recent advances in cyber–physical formation schemes are shown [26] that propose a hierarchical scheduling strategy for mobile robots in multi-warehouse environments, integrating industrial cyber–physical systems to optimize logistics operations. Ref. [27] presents a cyber–physical control system designed for autonomous logistic robots, focusing on real-time decision-making and integration with industrial automation platforms. Ref. [28], introduces a cyber–physical system for optimal trajectory planning of mobile robots, explicitly considering the deterioration of electric motors to improve long-term performance and energy efficiency.

The main issues in this scheme across different workspaces are sharing information through the internet [29,30], new challenges for control schemes [31–33], delay effects in internet communications [34,35], the problem of communication losses between workspaces [36,37], and the impact of intermittent connectivity on internet links [38–40].

This work includes the development of a low-cost platform of a multi-robot system with a cyber–physical connection. Some multi-robot platforms for educational purposes are based on the development of omnidirectional robots [41,42], cooperative task systems [30], and cyber–physical [17].

This article presents an extension of our previous work [43], including detailed proofs handling communication delays and adding new experiments to show the robustness of the theoretical results presented. The main contributions are as follows.

- The design and implementation of a control strategy for a cyber–physical formation of mobile robot systems moving in two workspaces. The control approach is based on the use of virtual agents that converge with the physical robots moving in the opposite workspace. The convergence is analyzed for the general case of n robots. Under this control scheme, the robots behave as if they move in a unique workspace.
- The problem formulation is established for single integrator robots in $2D$. However, it has a broader scope since it is known that different nonholonomic mobile robots can be reduced to this simple representation using an appropriate input–output linearization transformation.
- The control approach is developed for $2D$ single integrator robots, but it is extended to unicycle-type robots with two wheels, and it is validated by numerical simulations.
- Finally, a low-cost experimental platform is built. An experiment is shown using four robots in two different physical spaces communicated through an internet connection

with an MQTT protocol, different formation topologies, sample rates, and MQTT brokers are tested.

The paper is organized as follows. The problem definition is given in Section 2. The control strategy is presented in the Section 3. Numerical simulations are shown in Section 4. The experimental platform is introduced and its results are reported in Section 5. Finally, concluding remarks and future research directions are drawn in Section 6.

2. Problem Formulation

In this work, the cyber–physical formation problem is defined for n point-robots moving in two different workspaces.

Let $\mathcal{R} = \{R_1, \dots, R_n\}$ be a set of mobile planar robots moving in two different workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$. Without loss of generality, assume that $\mathcal{R}_1 = \{R_1, \dots, R_m\}$ and $\mathcal{R}_2 = \{R_{m+1}, \dots, R_n\}$, with $1 \leq m < n$, are the subsets of robots moving in $\mathcal{W}_1$ and $\mathcal{W}_2$, respectively. Therefore, $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ with $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Define $\mathbf{z}_i = [x_i, y_i]^\top \in \mathbb{R}^2$ as the 2D position of R_i in $\mathcal{W}_1$ or $\mathcal{W}_2$.

This work is focused on the construction of a leader–follower formation control strategy such that the robots in the two workspaces converge to a global formation as if they were in the same space. The leader robot should follow a desired trajectory given by $\mathbf{z}_d(t) \in \mathbb{R}^2$.

A desired global formation graph $\mathcal{G}$ for the n robots is given by an adjacency matrix $\mathbf{A}$, a set of relative positions $\mathcal{C}$, and a Laplacian matrix $\mathbf{L}$. The challenge in a cyber–physical configuration is to converge to the formation specified by $\mathcal{G}$ in the absence of certain robots in each workspace. To carry this out, the information about the positions $\mathbf{z}_i, i = 1, \dots, n$ must be shared between $\mathcal{W}_1$ and $\mathcal{W}_2$, as illustrated in Figure 1 using the internet and the cloud.

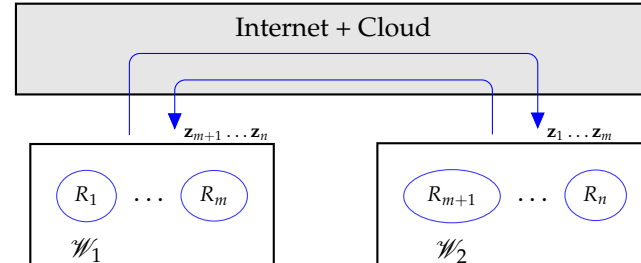


Figure 1. Communication between workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$.

3. Control Strategy

To tackle the cyber–physical formation problem defined in the previous section, this work proposes the use of virtual robots. Thus, for each $R_i \in \mathcal{R}_1, i = 1, \dots, m$, define a virtual robot $V_i \in \mathcal{V}_2, i = 1, \dots, m$, and for each $R_i \in \mathcal{R}_2, i = m + 1, \dots, n$, define a virtual robot $V_i \in \mathcal{V}_1, i = m + 1, \dots, n$. Thus, there are also n virtual robots divided into $\mathcal{V}_1$ and $\mathcal{V}_2$. Let $\mathbf{v}_i \in \mathbb{R}^2$ be the 2D position of each virtual robot V_i . Finally, $\mathcal{W}_i = \mathcal{R}_i \cup \mathcal{V}_i, i = 1, 2$ and $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$.

For this new cyber–physical system composed of $2n$ agents generated in this way, a combined formation graph $\mathcal{G}$, derived from $\mathcal{G}$, is defined as follows.

For each $R_j \in \mathcal{R}_i$, let $\mathcal{N}_{ij}^R$ and $\mathcal{N}_{ij}^V$ be the adjacent subsets of the indexes of the real and virtual robots in $\mathcal{W}_i$ that are communicated to R_j according to the following:

$$\begin{aligned} \mathcal{N}_{ij}^R &= \{k \mid \mathbf{A}_{jk} \neq 0, R_j, R_k \in \mathcal{R}_i\} \\ \mathcal{N}_{ij}^V &= \{k \mid \mathbf{A}_{jk} \neq 0, R_j \in \mathcal{R}_i, V_k \in \mathcal{V}_i\}. \end{aligned} \quad (1)$$

In addition, all virtual robots $V_j \in \mathcal{W}_i$ are communicated only to their real counterparts R_j located in the other workspace $\mathcal{W}_{3-i}$; i.e., their corresponding adjacent subsets $\mathcal{N}_{ij}^R, \mathcal{N}_{ij}^V$ are

$$\begin{aligned}\mathcal{N}_{ij}^R &= \{j \mid V_j \in \mathcal{W}_i, R_j \in \mathcal{W}_{3-i}\} \\ \mathcal{N}_{ij}^V &= \emptyset.\end{aligned}\quad (2)$$

An example of such a communication topology is shown in Figure 2. Furthermore, for each (j, k) with $k \in \mathcal{N}_{ij}^R \cup \mathcal{N}_{ij}^V$, let the desired static and predefined relative position vector be $\mathbf{c}_{kj} \in \mathbb{R}^2$ with $\mathbf{c}_{kj} \in \mathcal{C}$. By design, it is straightforward to show that if $\mathcal{G}$ has a directed spanning tree, then $\hat{\mathcal{G}}$ also has one. Having a directed spanning tree, it is well known that the Laplacian matrix of $\mathcal{G}$ and $\hat{\mathcal{G}}$, given by $\mathbf{L}$ and $\hat{\mathbf{L}}$ respectively, have exactly one zero eigenvalue, the rest being positive [1].

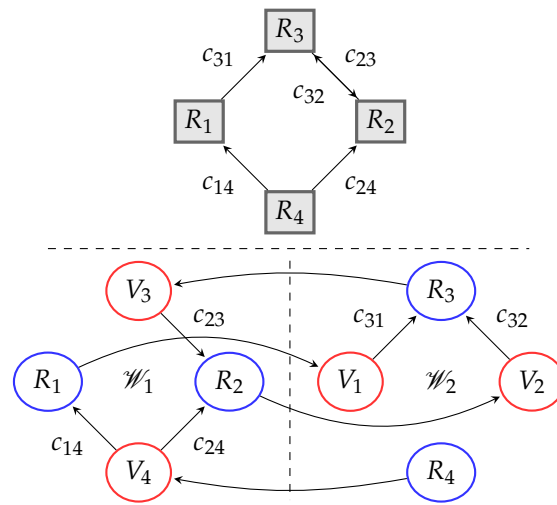


Figure 2. An example for $(n = 4, m = 2)$ of the relationship between the original formation graph $\mathcal{G}$ (above) and workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$ and combined graph $\hat{\mathcal{G}}$ (below).

In the following, the control laws for this configuration of real and virtual robots are defined. First, consider both real and virtual robots as single integrators, given by

$$\begin{aligned}\dot{\mathbf{z}}_i &= \mathbf{u}_i, & i &= 1, \dots, n \\ \dot{\mathbf{v}}_i &= \mathbf{w}_i, & i &= 1, \dots, n\end{aligned}\quad (3)$$

where $\mathbf{u}_i \in \mathbb{R}^2$ and $\mathbf{w}_i \in \mathbb{R}^2$ are the control inputs for the real and virtual robots, respectively. For the leader–follower control strategy introduced in this work, let us define $R_n \in \mathcal{W}_2$ as the leader robot and $\mathbf{z}_d, \dot{\mathbf{z}}_d \in \mathbb{R}^2$ as the desired position and velocity of the trajectory for the leader. Furthermore, workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$ could be a long distance away from each other, and therefore, a sizable delay should have to be taken into account on the information passed between them.

The control strategy for the virtual robots $V_j \in \mathcal{W}_i$ is designed to track its corresponding real robot R_j that belongs to the other workspace $\mathcal{W}_{3-i}$, according to the following law:

$$\mathbf{w}_j(t) = k_v(\mathbf{z}_j(t - \tau) - \mathbf{v}_j(t)) + \dot{\mathbf{z}}_d(t), \quad k_v > 0, \quad j = 1, \dots, n. \quad (4)$$

where $\tau \geq 0$ is the communication delay between workspaces, and $k_v > 0$ is the control gain for the virtual robots. The control law for the real robots R_j is given by the following:

$$\mathbf{u}_j(t) = k_r \sum_{k \in \mathcal{N}_{ij}^R} (\mathbf{z}_k(t) + \mathbf{c}_{kj} - \mathbf{z}_j(t)) + k_r \sum_{k \in \mathcal{N}_{ij}^V} (\mathbf{v}_k(t) + \mathbf{c}_{kj} - \mathbf{z}_j(t)) + \dot{\mathbf{z}}_d(t), \quad (5)$$

with $k_r > 0$ being the control gain for the real robots and $j = 1, \dots, n-1$. The control strategy for the leader robot R_n is given by

$$\mathbf{u}_n = k_r \sum_{k \in \mathcal{N}_{2n}^R} (\mathbf{z}_k + \mathbf{c}_{kn} - \mathbf{z}_n) + k_r \sum_{k \in \mathcal{N}_{2n}^V} (\mathbf{v}_k + \mathbf{c}_{kn} - \mathbf{z}_n) + k_c(\mathbf{z}_d - \mathbf{z}_n) + \dot{\mathbf{z}}_d, \quad (6)$$

where the control gain $k_c > 0$. Note that the control law for the real robots R_j is not affected by the communication delay τ between the workspaces, since all required information is within its own workspace. Defining the relative positions of the robots with respect to the desired trajectory $\tilde{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{z}_d$ and $\tilde{\mathbf{v}}_j = \mathbf{v}_j - \mathbf{z}_d$, Equations (4)–(6) have a relative equilibrium formation defined by

$$\begin{aligned} \tilde{\mathbf{v}}_j^* &= \tilde{\mathbf{z}}_j^*, \quad \forall j \\ 0 &= \sum_{k \in \bigcup_i \mathcal{N}_{ij}^R} (\tilde{\mathbf{z}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^*) + \sum_{k \in \bigcup_i \mathcal{N}_{ij}^V} (\tilde{\mathbf{v}}_k^* + \mathbf{c}_{kj} - \tilde{\mathbf{z}}_j^*), \quad \forall j < n \\ \tilde{\mathbf{z}}_n^* &= 0. \end{aligned} \quad (7)$$

Let $e_j^R = \tilde{\mathbf{z}}_j - \tilde{\mathbf{z}}_j^*$ and $e_j^V = \tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_j^*$ be the relative error with respect to the formation trajectory of the real and virtual robots, respectively. Using the above equations, the error dynamics can be expressed as

$$\begin{bmatrix} \dot{\mathbf{e}}^V(t) \\ \dot{\mathbf{e}}^R(t) \end{bmatrix} = \left(\begin{bmatrix} -k_v \mathbf{I}_n & \mathbf{0}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \otimes \mathbf{I}_2 \right) \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix} + \left(\begin{bmatrix} k_v \mathbf{I}_n \\ \mathbf{0}_n \end{bmatrix} \otimes \mathbf{I}_2 \right) \mathbf{e}^R(t - \tau) \quad (8)$$

where

$$\begin{aligned} \mathbf{e}^V(t) &= [e_1^{V\top}(t) \dots e_n^{V\top}(t)]^\top \\ \mathbf{e}^R(t) &= [e_1^{R\top}(t) \dots e_n^{R\top}(t)]^\top \\ \mathbf{E}_{21} + \mathbf{E}_{22} &= -\mathbf{L} \\ \mathbf{q}_n &= [\mathbf{0}_{n-1}^\top \quad 1]^\top, \end{aligned} \quad (9)$$

with $\mathbf{0}_k$ being a zero vector of dimension k .

3.1. Without Communication Delays

In an ideal case in which there are no delays in the information carried out between workspaces, we can state the following result.

Theorem 1. For a set $\mathcal{R}$ of n mobile robots distributed over two physical workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$ with first-order dynamics given by Equation (3) and a leader–follower formation setup given by a communication topology $\mathcal{G}$ derived from the original topology $\mathcal{G}$ with a directed spanning tree and defined by (1), (2), the control strategy given by Equations (4)–(6) without delay ($\tau = 0$) reaches convergence of the error dynamics (10) to the origin, achieving the relative formation defined by Equation (7).

Proof. When there is no communication delay, i.e., $\tau = 0$, the error dynamics (8) can be simplified to

$$\begin{bmatrix} \dot{\mathbf{e}}^V(t) \\ \dot{\mathbf{e}}^R(t) \end{bmatrix} = \left(\begin{bmatrix} -k_v \mathbf{I}_n & k_v \mathbf{I}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \otimes \mathbf{I}_2 \right) \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{e}^V(t) \\ \mathbf{e}^R(t) \end{bmatrix}. \quad (10)$$

In order to prove the convergence of dynamics (10), it is sufficient to show that all the eigenvalues of the matrix $\mathbf{M}$ have a negative real part. Note that

$$\mathbf{M} = \mathbf{K}\mathbf{F} = \begin{bmatrix} k_v \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & k_r \mathbf{I}_n \end{bmatrix} \begin{bmatrix} -\mathbf{I}_n & \mathbf{I}_n \\ \mathbf{E}_{21} & \mathbf{E}_{22} - \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \quad (11)$$

with $\hat{k}_c = k_c/k_r > 0$. Since $\mathbf{E}_{21} + \mathbf{E}_{22} = -\mathbf{L}$, where all the diagonal elements $\{\mathbf{L}\}_{ii} = \{\mathbf{E}_{22}\}_{ii} \forall i$, by applying the Gershgorin Circle's Theorem and taking into account the structure of the Laplacian matrix, it is straightforward to see that the matrix $\mathbf{F}$ may have one zero eigenvalue while the rest have a negative real part. Since k_v , k_r , k_c , and $\hat{k}_c$ are all positive, the same can be said of $\mathbf{M}$. To complete the proof, it will be shown that $\mathbf{F}$ does not have a zero eigenvalue.

For $\mathbf{F}$ to have a zero eigenvalue, there must exist a non-null vector $\mathbf{p}$ such that $\mathbf{F}\mathbf{p} = \mathbf{0}$. Using expression (11) and writing $\mathbf{p} = [\mathbf{p}_v^\top, \mathbf{p}_r^\top]^\top$

$$\mathbf{F}\mathbf{p} = \begin{bmatrix} -\mathbf{I}_n & \mathbf{I}_n \\ \mathbf{E}_{21} & \mathbf{E}_{22} - \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_v \\ \mathbf{p}_r \end{bmatrix} = \mathbf{0} \implies \begin{cases} \mathbf{p}_v = \mathbf{p}_r, \\ (\mathbf{L} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \mathbf{0} \end{cases} \quad (12)$$

Because of the assumption that $\mathcal{G}$ has a directed spanning tree, $\mathbf{L}$ has a single zero eigenvalue. Thus, its LU decomposition has the following structure:

$$\mathbf{L} = \mathbf{Q}\mathbf{U} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ q_{21} & 1 & 0 & \vdots \\ \vdots & q_{ij} & \ddots & 0 \\ q_{n1} & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & \ddots & u_{ij} & \vdots \\ \vdots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \dots & \dots & 0 \end{bmatrix}, \quad (13)$$

where $\mathbf{Q}$ is a lower triangular matrix with its main diagonal elements equal to 1 and matrix $\mathbf{U}$ is an upper triangular matrix with its main diagonal elements $u_{ii} \neq 0, \forall i < n$, and $u_{nn} = 0$. Observe that $\mathbf{Q}$ is invertible and also $\mathbf{Q}\mathbf{q}_n = \mathbf{Q}^{-1}\mathbf{q}_n = \mathbf{q}_n$. Substituting (13) into (12) gives

$$\begin{aligned} (\mathbf{Q}\mathbf{U} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r &= \mathbf{Q}(\mathbf{U} + \hat{k}_c \mathbf{Q}^{-1} \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r = \mathbf{0} \\ (\mathbf{U} + \hat{k}_c \mathbf{q}_n \mathbf{q}_n^\top) \mathbf{p}_r &= \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & \ddots & u_{ij} & \vdots \\ \vdots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \dots & \dots & \hat{k}_c \end{bmatrix} \mathbf{p}_r = \mathbf{0} \end{aligned} \quad (14)$$

Since $u_{ii} \neq 0, \forall i < n$, and $\hat{k}_c > 0$, the only solution of Equation (14) is $\mathbf{p}_r = \mathbf{0}$. Therefore, the only solution to Equation (12) is also $\mathbf{p} = \mathbf{0}$, which implies that $\mathbf{F}$ and $\mathbf{M}$ do not have a zero eigenvalue. Consequently, the error dynamics (10) are stable. $\square$

3.2. With Communication Delays

When a delay in communication between workspaces cannot be neglected, delayed dynamics (8) must be analyzed. The next result looks at conditions to have stability of the closed loop in the event of having communication delays.

Theorem 2. For a set $\mathcal{R}$ of n mobile robots distributed over two physical workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$ with first-order dynamics given by Equation (3) and a leader–followers formation setup given by a communication topology $\mathcal{G}$ with unknown fixed bounded delay ($0 < \tau \leq T$) and a control strategy given by Equations (4)–(6), if there exist symmetric positive definite matrices $\mathbf{P}, \mathbf{S}, \mathbf{Z} \in \mathbb{R}^{2n \times 2n}$ such that

$$\Gamma = \begin{bmatrix} A_0^\top \mathbf{P} + \mathbf{P}A_0 + \mathbf{S} + TA_0^\top \mathbf{Z}A_0 - \mathbf{Z}/T & \mathbf{P}A_1 + TA_0^\top \mathbf{Z}A_1 & \mathbf{Z}/T \\ A_1^\top \mathbf{P} + TA_1^\top \mathbf{Z}A_0 & -\mathbf{S} + TA_1^\top \mathbf{Z}A_1 & \mathbf{0} \\ \mathbf{Z}/T & 0 & -\mathbf{Z}/T \end{bmatrix} < 0, \quad (15)$$

with

$$A_0 = \begin{bmatrix} -k_v \mathbf{I}_n & \mathbf{0}_n \\ k_r \mathbf{E}_{21} & k_r \mathbf{E}_{22} - k_c \mathbf{q}_n \mathbf{q}_n^\top \end{bmatrix}, \quad A_1 = \begin{bmatrix} \mathbf{0}_n & k_v \mathbf{I}_n \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix}. \quad (16)$$

the closed loop error dynamics (8) reaches convergence to the origin, achieving relative formation defined by Equation (7).

It should be noted that Equation (15) constitutes a Linear Matrix Inequality (LMI) in the coefficients of the matrices $\mathbf{P}, \mathbf{S}$ and $\mathbf{Z}$ parameterizable in T .

Proof. Let $\mathbf{x}(t)^\top = [\mathbf{e}^V(t)^\top, \mathbf{e}^R(t)^\top] \in \mathbb{R}^{2n}$ describe the error vector at time t . Let us also define $\mathbf{x}_t(\theta)^\top = [\mathbf{e}^V(t+\theta)^\top, \mathbf{e}^R(t+\theta)^\top]$, $\theta \in [-T, 0]$ as the error trajectory in the delay interval $[t-T, t]$. The error dynamics (8) can be rewritten as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}_t) = A_0 \mathbf{x}(t) + A_1 \mathbf{x}(t-\tau), \quad (17)$$

with matrices A_0 and A_1 given by (16). In order to prove the stability of (17), we define a Lyapunov–Krasovski functional $V(\mathbf{x}_t)$ by the following expression:

$$V(\mathbf{x}_t) = \mathbf{x}(t)^\top \mathbf{P} \mathbf{x}(t) + \int_{t-\tau}^t \mathbf{x}(s)^\top \mathbf{S} \mathbf{x}(s) ds + \int_{t-T}^t ds \int_s^t f^\top(\mathbf{x}_\mu) \mathbf{Z} f(\mathbf{x}_\mu) d\mu. \quad (18)$$

Choosing matrices $\mathbf{P}, \mathbf{S}$ and $\mathbf{Z}$ symmetric and positive definite makes the functional V positive definite as well. Taking the time derivative of (18) and applying Jensen’s inequality, we obtain

$$\begin{aligned} \dot{V}(\mathbf{x}_t) &= \dot{\mathbf{x}}(t)^\top \mathbf{P} \mathbf{x}(t) + \mathbf{x}(t)^\top \mathbf{P} \dot{\mathbf{x}}(t) + \mathbf{x}(t)^\top \mathbf{S} \mathbf{x}(t) - \mathbf{x}(t-\tau)^\top \mathbf{S} \mathbf{x}(t-\tau) \\ &\quad + T f^\top(\mathbf{x}_t) \mathbf{Z} f(\mathbf{x}_t) - \int_{t-T}^t f^\top(\mathbf{x}_s) \mathbf{Z} f(\mathbf{x}_s) ds \\ &\leq \dot{\mathbf{x}}(t)^\top \mathbf{P} \mathbf{x}(t) + \mathbf{x}(t)^\top \mathbf{P} \dot{\mathbf{x}}(t) + \mathbf{x}(t)^\top \mathbf{S} \mathbf{x}(t) - \mathbf{x}(t-\tau)^\top \mathbf{S} \mathbf{x}(t-\tau) \\ &\quad + T f^\top(\mathbf{x}_t) \mathbf{Z} f(\mathbf{x}_t) - \frac{1}{T} (\mathbf{x}(t) - \mathbf{x}(t-T))^\top \mathbf{Z} (\mathbf{x}(t) - \mathbf{x}(t-T)). \end{aligned} \quad (19)$$

Let us define the vector $\Phi(t, \tau, T)^\top = [\mathbf{x}(t)^\top, \mathbf{x}(t-\tau)^\top, \mathbf{x}(t-T)^\top]$. Substituting Equations (17), (15) and (16) into (19) and rearranging terms, we obtain

$$\begin{aligned} \dot{V}(\mathbf{x}_t) &\leq \Phi^\top \begin{bmatrix} A_0^\top \mathbf{P} + \mathbf{P}A_0 + \mathbf{S} + TA_0^\top \mathbf{Z}A_0 - \mathbf{Z}/T & \mathbf{P}A_1 + TA_0^\top \mathbf{Z}A_1 & \mathbf{Z}/T \\ A_1^\top \mathbf{P} + TA_1^\top \mathbf{Z}A_0 & -\mathbf{S} + TA_1^\top \mathbf{Z}A_1 & \mathbf{0} \\ \mathbf{Z}/T & 0 & -\mathbf{Z}/T \end{bmatrix} \Phi \\ \dot{V} &\leq \Phi^\top(t, \tau, T) \Gamma \Phi(t, \tau, T) \end{aligned} \quad (20)$$

If the LMI given by (15) is satisfied, the time derivative of the Lyapunov–Krasovski functional $V(\mathbf{x}_t)$ along the trajectories will be negative, and the error dynamics will converge to the origin, making the cyberphysical system converge to the desired formation pattern in workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$. $\square$

Remark 1. The LMI condition here is conservative, since the last term of $\dot{V}$ can be further developed using tighter inequalities to obtain a closer LMI condition.

Remark 2. Since the LMI condition depends on T but not explicitly on τ , if there is a solution to (15), the stability can be ensured for any $0 \leq \tau \leq T$.

Remark 3. It is fairly straightforward to extend this result to multiple delays and time varying delays with bounds on the time derivative of the delay function, arriving at one or several LMI conditions like (15).

Remark 4. There are efficient computational methods and tools to solve LMIs. In this work, the YALMIP toolbox for Matlab with the Gurobi optimizer has been used to check the condition in the experiments shown in the following sections.

4. Numerical Simulations

For the numerical simulation, a Python 3.13 code was developed using Matplotlib and NumPy libraries to simulate various trajectories and robot formations within two distinct workspaces. Three formations were implemented: the first consists of two robots arranged in a straight line (one per workspace, as shown in Figure 3); the second consists of four robots arranged in a straight line (two per workspace, as depicted in Figure 4); and the third consists of four robots arranged in a diamond configuration (two per workspace, as illustrated in Figure 5). In each formation, a leader robot maintains a circular trajectory. The formations were evaluated using different gain parameters (k_c , k_r , and k_v), as well as various delay values in the virtual robots' data. Simulation errors were quantified through comparative tables based on the L_1 norm (integral of the sum of absolute errors) and the L_2 norm (integral of the sum of squared errors).

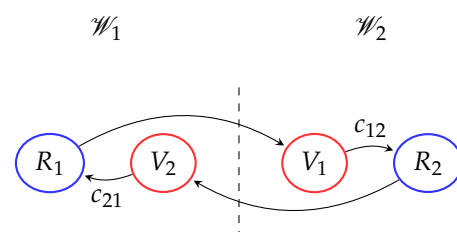


Figure 3. First formation topology: 2-robot horizontal line shape formation.

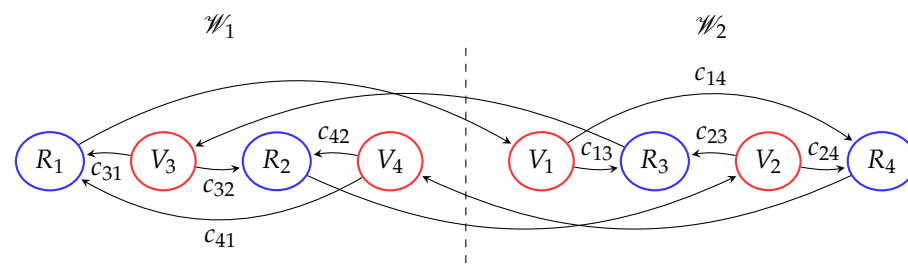


Figure 4. Second formation topology: 4-robot horizontal line shape formation.

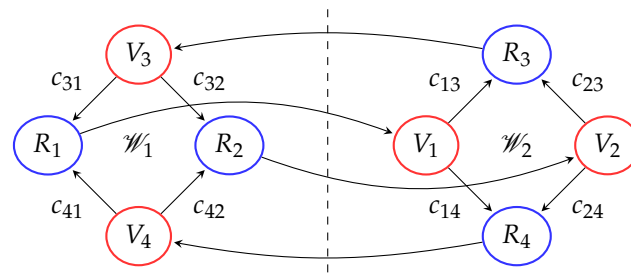


Figure 5. Third formation topology: 4 Diamond shape robots.

4.1. Without Communication Delays

Figure 6 shows the numerical simulation of the control approach with $n = 2$, and $m = 2$; the total number of robots is $\mathcal{R} = \{R_1, R_2\}$, where $\mathcal{R}_1 = \{R_1\} \in \mathcal{W}_1$ and $\mathcal{R}_2 = \{R_2\} \in \mathcal{W}_2$, based on the workspace and topology formation, as shown in Figure 3. The leader robot is R_1 . Therefore, $\mathcal{V}_1 = \{V_2\}$ and $\mathcal{V}_2 = \{V_1\}$, and the total members of each workspace are $\mathcal{W}_1 = \{R_1, V_2\}$ and $\mathcal{W}_2 = \{V_1, R_2\}$.

The adjacent subsets for the real robots are defined as $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_2\}$, $\mathcal{N}_{22}^R = \emptyset$, $\mathcal{N}_{22}^V = \{V_1\}$, as shown in Figure 3. The desired final formation of the robots is a horizontal line, as if both real robots were in the same workspace. To achieve this desired formation pattern, the relative vectors are given by $c_{12} = [a, 0]^\top$, with $a = 400$ and by construction $c_{12} = -c_{21}$.

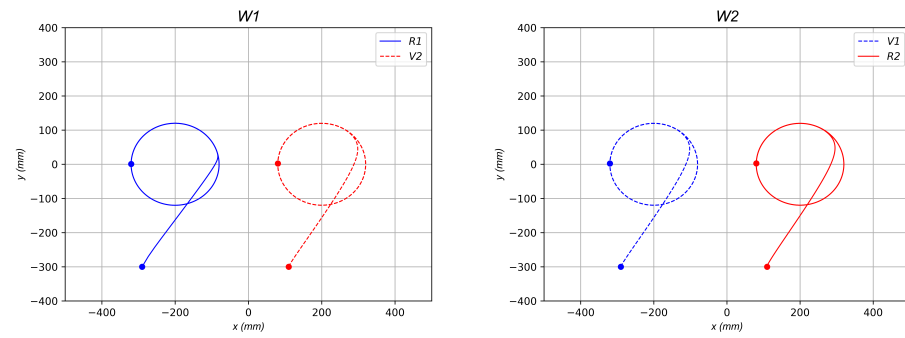
Figure 6a shows the trajectories of the real and virtual robots using the control law (4)–(5)–(6) with $k_c = 1$, $k_r = 1$, $k_v = 1$ and the desired trajectory of the leader robot R_1 given by $\mathbf{z}_d(t) = [-200 + 120 \cos(\frac{\pi}{80}t), 120 \sin(\frac{\pi}{80}t)]^\top$. All robots converge to the desired horizontal line formation pattern, which is observed by the convergence of the errors as shown in Figure 6b, and the control inputs are observed in Figure 6c.

In the first formation, shown in Figure 3, several simulations were conducted by varying the values of k_c , k_r , and k_v to demonstrate the stability and convergence speed of the control strategy. Simulations were performed using k values of 0.1, 1, and 10, as well as their various combinations, while ensuring that k_c remained greater than or equal to k_r and k_v . For each simulation, the L_1 and L_2 norms were computed by summing the absolute errors and the squared errors, respectively, taking into account errors from both workspaces, real and virtual robots. The results are depicted in the graphs in Figure 7 and summarized in Table 1.

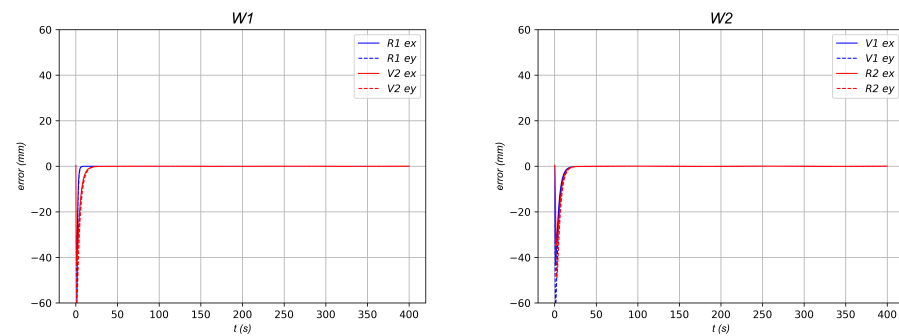
The second formation topology, as shown in the Figure 8, shows the numerical simulation of the control approach with $n = 4$, and $m = 2$, i.e., the total set of robots is $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, where $\mathcal{R}_1 = \{R_1, R_2\} \in \mathcal{W}_1$ and $\mathcal{R}_2 = \{R_3, R_4\} \in \mathcal{W}_2$, based on the workspace and topology formation, as shown in Figure 4. The leader robot is R_1 . Therefore, $\mathcal{V}_1 = \{V_3, V_4\}$ and $\mathcal{V}_2 = \{V_2, V_3\}$ and the total members of each workspace are $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ and $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.

Table 1. L_1 and L_2 norms for different k values in the simulation of 2-robot horizontal line shape formation.

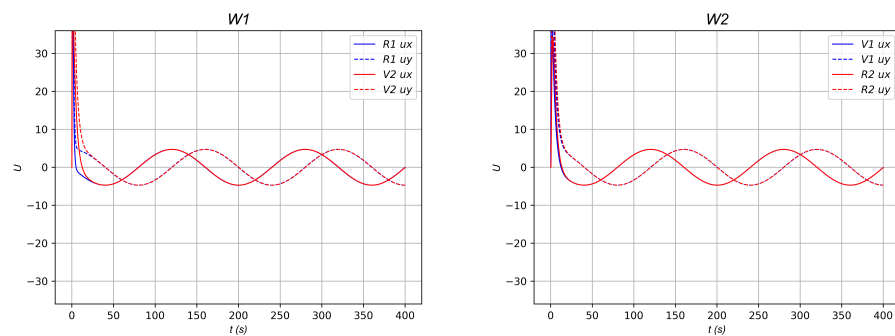
Test	L_1	L_2
$k_c: 1, k_r: 0.1, k_v: 0.1$	18,092.63	8001.94
$k_c: 1, k_r: 0.1, k_v: 1$	8874.45	5404.99
$k_c: 1, k_r: 1, k_v: 0.1$	13,532.95	6714.88
$k_c: 1, k_r: 1, k_v: 1$	4175.66	2313.57
$k_c: 10, k_r: 0.1, k_v: 0.1$	18,064.60	8002.15
$k_c: 10, k_r: 0.1, k_v: 1$	8846.89	5404.71
$k_c: 10, k_r: 1, k_v: 0.1$	13,505.06	6714.54
$k_c: 10, k_r: 1, k_v: 1$	4148.20	2312.31
$k_c: 10, k_r: 10, k_v: 10$	2692.79	1992.75



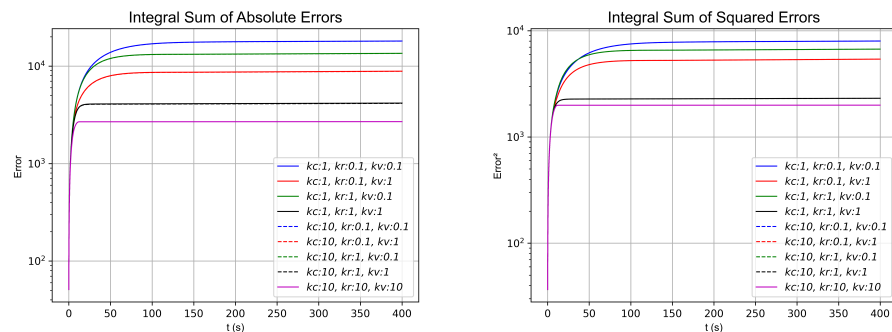
(a) Robots Trajectories.



(b) Convergence of the errors.



(c) Control Inputs.

Figure 6. Numerical simulation for a 2-Robots horizontal line shape formation.**Figure 7.** L_1 and L_2 norms with different k values in the simulation of 2-robot horizontal line shape formation.

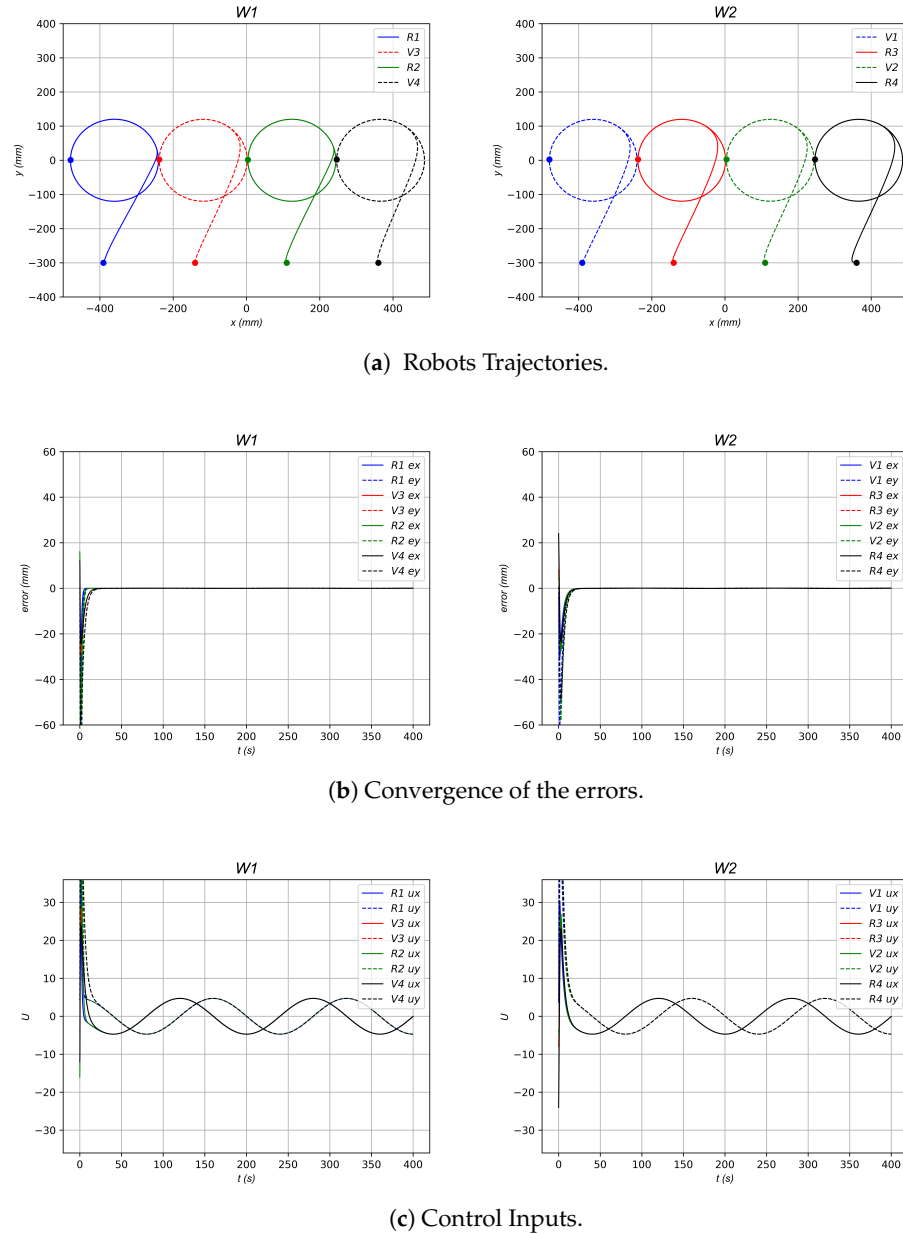


Figure 8. Numerical simulation of 4-robot horizontal line shape formation.

The adjacent subsets for the real robots are defined as $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_3, V_4\}$, $\mathcal{N}_{12}^R = \emptyset$, $\mathcal{N}_{12}^V = \{V_3, V_4\}$, $\mathcal{N}_{23}^R = \emptyset$, $\mathcal{N}_{23}^V = \{V_1, V_2\}$, $\mathcal{N}_{24}^R = \emptyset$, $\mathcal{N}_{24}^V = \{V_1, V_2\}$, as shown in Figure 4. The desired final posture of the robots is a horizontal line formation, as if all four real robots would be in the same workspace. To achieve this desired formation pattern, the relative vectors are given by $c_{13} = [a, 0]^\top$, $c_{14} = [3a, 0]^\top$, $c_{23} = [-a, 0]^\top$, $c_{24} = [a, 0]^\top$, with $a = 242$ and by construction $c_{31} = -c_{13}$, $c_{41} = -c_{14}$, $c_{32} = -c_{23}$ and $c_{42} = -c_{24}$.

Figure 8a shows the trajectories of the real and virtual robots using the control law (4)–(5)–(6) and again with $k_c = 1$, $k_r = 1$, $k_v = 1$ and the desired trajectory of the leader robot R_1 given by $\mathbf{z}_d(t) = \left[-360 + 120 \cos\left(\frac{\pi}{80}t\right), 120 \sin\left(\frac{\pi}{80}t\right) \right]^\top$. All robots converge to the desired horizontal line formation pattern, which is observed by the error convergence shown in Figure 8b, and the control inputs are observed in the Figure 8c.

In the second formation, shown in Figure 4, several simulations were conducted by varying the values of k_c , k_r , and k_v to demonstrate the stability and convergence speed

of the control strategy. Simulations were performed using k values of 0.1, 1, and 10, as well as their various combinations, while ensuring that k_c remained greater than or equal to k_r and k_v . For each simulation, the L_1 and L_2 norms were computed by summing the absolute errors and the squared errors, respectively, taking into account errors from both workspaces, real and virtual robots. The results are depicted in the graphs in Figure 9 and summarized in Table 2.

Table 2. L_1 and L_2 norms for different k values in the simulation of 4-robot horizontal line shape formation.

Test	L_1	L_2
$k_c: 1, k_r: 0.1, k_v: 0.1$	33,554.19	10,729.37
$k_c: 1, k_r: 0.1, k_v: 1$	17,981.90	7408.51
$k_c: 1, k_r: 1, k_v: 0.1$	21,512.64	8508.63
$k_c: 1, k_r: 1, k_v: 1$	5518.18	2377.26
$k_c: 10, k_r: 0.1, k_v: 0.1$	33,525.20	10,732.97
$k_c: 10, k_r: 0.1, k_v: 1$	17,953.18	7411.35
$k_c: 10, k_r: 1, k_v: 0.1$	21,484.32	8511.37
$k_c: 10, k_r: 1, k_v: 1$	5490.62	2378.52
$k_c: 10, k_r: 10, k_v: 10$	2523.83	1877.37

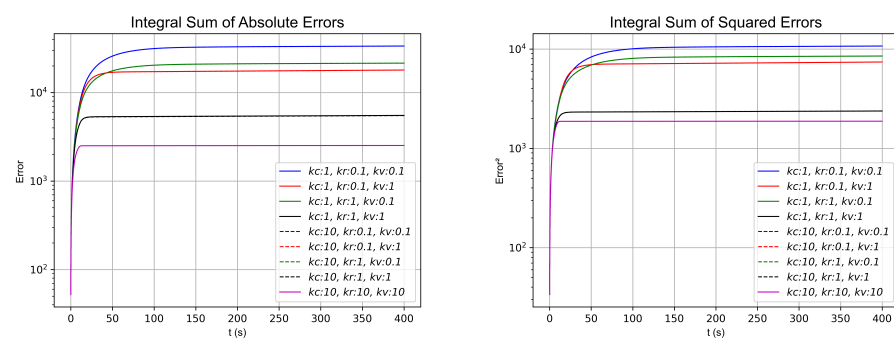


Figure 9. L_1 and L_2 norms with different k values in simulation of 4-robot horizontal line shape formation.

The third formation topology for numerical simulation, as shown in Figure 10, presents the control approach with $n = 4$ and $m = 2$; i.e., the total set of robots is $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$, where $\mathcal{R}_1 = \{R_1, R_2\} \in \mathcal{W}_1$ and $\mathcal{R}_2 = \{R_3, R_4\} \in \mathcal{W}_2$, based on the workspace and topology formation, as shown in Figure 5. The leader robot is R_1 . Therefore, $\mathcal{V}_1 = \{V_3, V_4\}$ and $\mathcal{V}_2 = \{V_1, V_2\}$, and the total members of each workspace are $\mathcal{W}_1 = \{R_1, R_2, V_3, V_4\}$ and $\mathcal{W}_2 = \{V_1, V_2, R_3, R_4\}$.

The adjacent subsets for the real robots are defined as $\mathcal{N}_{11}^R = \emptyset$, $\mathcal{N}_{11}^V = \{V_3, V_4\}$, $\mathcal{N}_{12}^R = \emptyset$, $\mathcal{N}_{12}^V = \{V_3, V_4\}$, $\mathcal{N}_{23}^R = \emptyset$, $\mathcal{N}_{23}^V = \{V_1, V_2\}$, $\mathcal{N}_{24}^R = \emptyset$, $\mathcal{N}_{24}^V = \{V_1, V_2\}$, as shown in Figure 5. The target final pattern configuration for the robots is a diamond-shaped formation, as if all four real robots were operating within the same workspace. To achieve this arrangement, the corresponding relative vectors are defined by $c_{13} = [a, b]^\top$, $c_{14} = [a, -b]^\top$, $c_{23} = [-a, b]^\top$, $c_{24} = [-a, -b]^\top$, with $a = 300$, $b = 150$, and by construction $c_{31} = -c_{13}$, $c_{41} = -c_{14}$, $c_{32} = -c_{23}$, and $c_{42} = -c_{24}$.

Figure 10 shows the trajectories of the real and virtual robots using the control law (4)–(5)–(6) with unitary values of gains $k_c = 1$, $k_r = 1$, and $k_v = 1$ and the desired trajectory of the leader robot R_1 given by $\mathbf{z}_d(t) = [-300 + 120 \cos(\frac{\pi}{80}t), 120 \sin(\frac{\pi}{80}t)]^\top$. It can be observed that the robots converge to the intended formation, as evidenced by the error convergence presented in Figure 10b. Additionally, Figure 10c illustrates the control inputs.

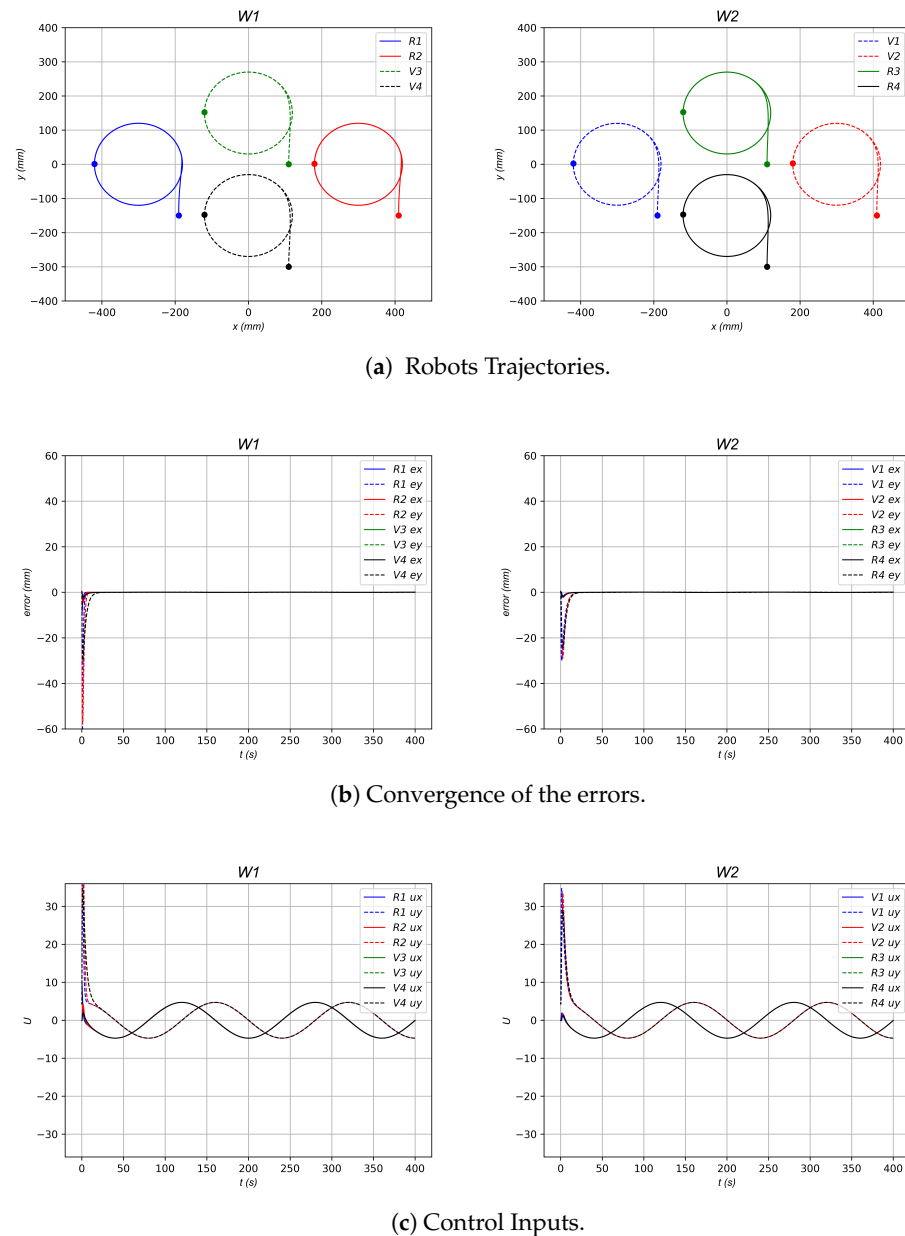


Figure 10. Numerical simulation with the 4 Robot diamond-shaped formation topology.

In the third formation, as shown in Figure 5, several simulations were performed by varying the parameters k_c , k_r , and k_v to assess the stability and convergence rate of the control strategy. Simulations were conducted using k values of 0.1, 1, and 10, as well as their different combinations, while ensuring that k_c remained greater than or equal to both k_r and k_v . For each simulation, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and from both real and virtual robots. The results are depicted in the graphs in Figure 11 and summarized in Table 3.

As shown in the numerical simulations and the results presented in Tables 1–3 for the different topologies and formations, as well as in the simulations conducted with various k gain values, higher k gains yield a faster and more pronounced convergence. This is evident in the figures illustrating the L_1 and L_2 norms for the three configurations: two robots arranged in a horizontal line (Figure 7), four robots arranged in a horizontal line (Figure 9), and four robots in a diamond-shaped formation (Figure 11). The most influential gains in the control model are k_r and k_v , since identical values for these parameters result in similar convergence rates

and comparable L_1 and L_2 norm values even when k_c varies. For example, when examining the black curve (a continuous or dashed line) corresponding to k_c set to 1 or 10 with k_r and k_v both fixed to 0.1, both simulations exhibit similar convergence speeds and behavior, despite the variation in k_c also having a similar value to L_1 and L_2 norm values.

Table 3. L_1 and L_2 norms for different k values in the 4 Robot diamond-shaped simulation.

Test	L_1	L_2
$k_c: 1, k_r: 0.1, k_v: 0.1$	12,438.28	4772.66
$k_c: 1, k_r: 0.1, k_v: 1$	6533.89	3306.62
$k_c: 1, k_r: 1, k_v: 0.1$	8118.66	3595.80
$k_c: 1, k_r: 1, k_v: 1$	1790.64	799.26
$k_c: 10, k_r: 0.1, k_v: 0.1$	12,416.72	4773.17
$k_c: 10, k_r: 0.1, k_v: 1$	6511.51	3306.86
$k_c: 10, k_r: 1, k_v: 0.1$	8096.59	3595.59
$k_c: 10, k_r: 1, k_v: 1$	1768.38	799.40
$k_c: 10, k_r: 10, k_v: 10$	589.10	467.78

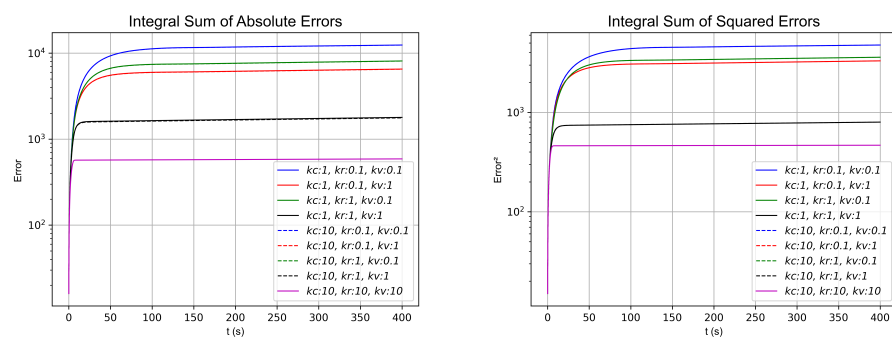


Figure 11. L_1 and L_2 norms with different k values in the diamond-shaped Robot simulation.

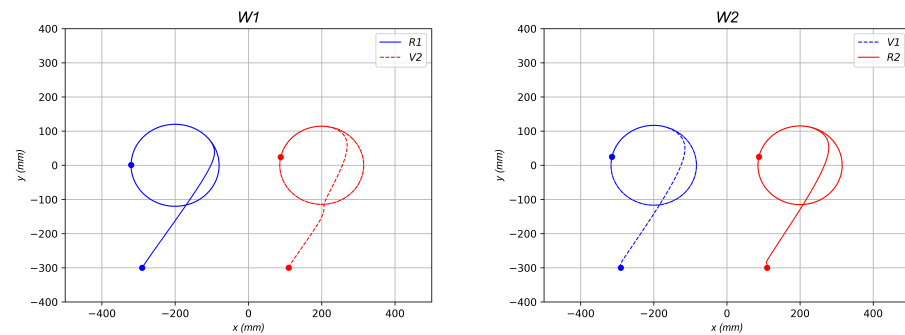
4.2. With Communication Delays

Numerical simulations were performed using the previously described control strategy with fixed gain values ($k_c = 1$, $k_r = 1$, $k_v = 1$). The same three topologies and workspace configurations presented earlier were employed, with the added feature of introducing a controlled constant delay in the position information of the virtual robots, which receive data directly from their corresponding real robot in the other workspace. This delay, denoted as τ , was tested at various values (0.1, 0.2, 0.5, 1, 2, 5, and 10 s) to evaluate the convergence speed and stability of the control strategy in the presence of delays. The maximum delay value simulated for these tests was selected based on the experimental work discussed in Sections 5.2 and 5.3. This study focuses on using the MQTT communication platform, which exhibits an average delay of 0.2 s and can reach maximum delays of up to 2 s. Also, a delay of 0.1 s in the computer vision system was measured for robot position detection. Consequently, intermediate delay values within this range were evaluated in various numerical simulation tests in the current section.

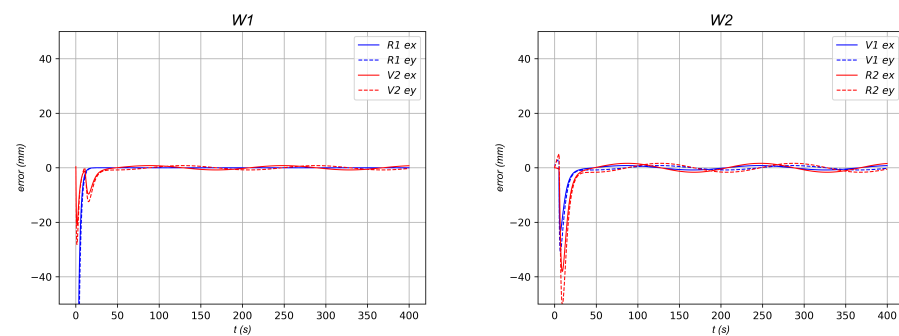
Figure 12 shows an example of a numerical simulation based on the workspace and topology formation shown in Figure 3, now developed in this example with a delay $\tau = 5$ s with respect to the data of the virtual robots. Note that an increase in the error values and a delay in the relative position of virtual and real robots was generated compared to the simulation shown in Figure 6.

In the first formation, as shown in Figure 3, several simulations were performed by varying the delay τ to assess the stability and convergence rate of the control strategy. For each simulation, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and

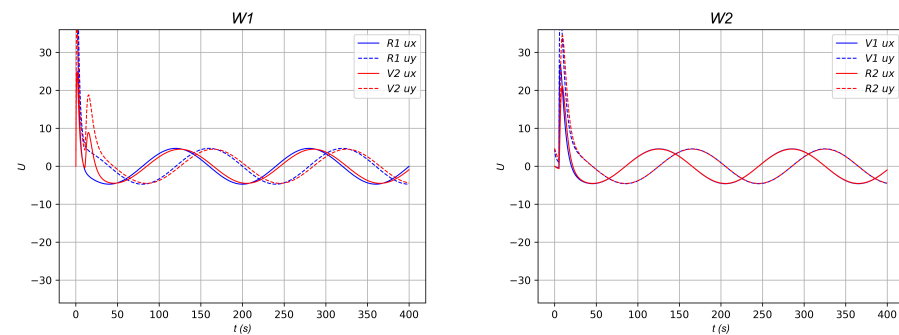
from both real and virtual robots. The results are depicted in the graphs in Figure 13 and summarized in Table 4.



(a) Robot trajectories with delays.



(b) Convergence of the errors with delays.



(c) Control inputs with delays.

Figure 12. Numerical simulation for 2-robot horizontal line shape formation with delays.

Table 4. L_1 and L_2 norms for different τ values for the simulation of 2-robot horizontal line shape formation.

Test	L_1	L_2
τ : 0.1 s	4417.46	2402.97
τ : 0.2 s	4446.70	2421.57
τ : 0.5 s	4534.75	2477.42
τ : 1 s	4682.41	2570.56
τ : 2 s	4980.26	2756.62
τ : 5 s	5881.10	3307.36
τ : 10 s	7368.16	4160.87

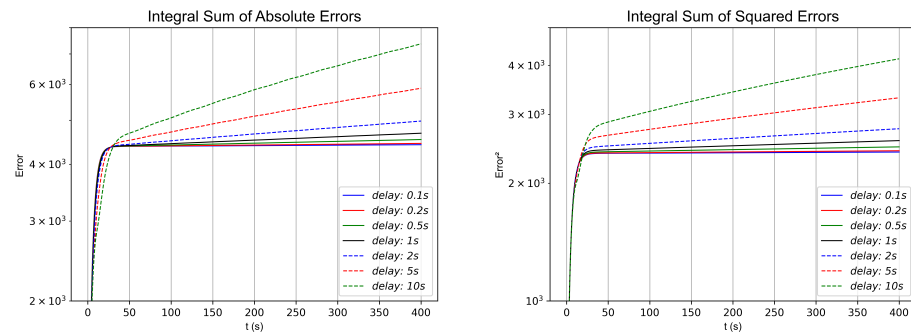


Figure 13. L_1 and L_2 norms with different delays for the simulation of 2-robot horizontal line shape formation.

In Figure 14, the same numerical simulation is also developed with a delay $\tau = 5$ s with respect to the data of the virtual robots, now based on the workspace and topology formation as shown in Figure 4. An increase in error values can be observed, along with a delay in the relative positioning between virtual and real robots, compared to the simulation presented in Figure 8.

In the second formation, as shown in Figure 4, the same simulations were performed, varying the delay τ to assess the stability and convergence rate of the control strategy. For each simulation, the L_1 and L_2 norms were calculated, and the results are depicted in the graphs in Figure 15 and summarized in Table 5.

Figure 16 is an example of the same numerical simulation developed with a delay $\tau = 5$ s with respect to the data of the virtual robots, now based on the workspace and topology formation shown in Figure 5. An increase in error values can be observed, along with a delay in the relative positioning between virtual and real robots, compared to the simulation presented in Figure 10.

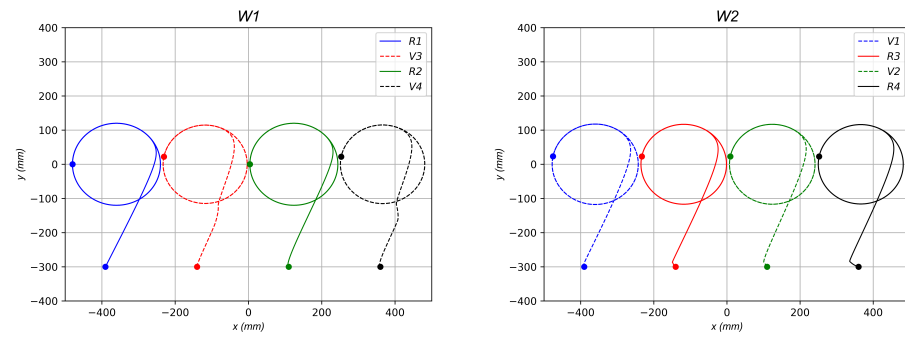
In the third formation, as shown in Figure 5, the same simulations were performed, varying the delay τ to assess the stability and convergence rate of the control strategy. For each simulation, the L_1 and L_2 norms were calculated; the results are depicted in the graphs in Figure 17 and summarized in Table 6.

Table 5. L_1 and L_2 norms for different τ values in simulation of 4-robot horizontal line shape formation.

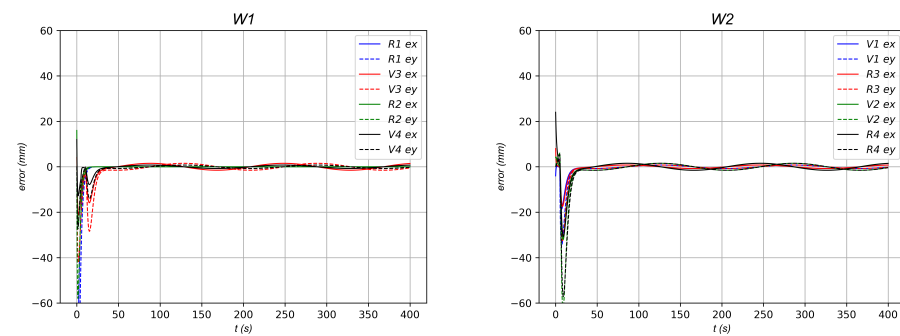
Test	L_1	L_2
τ : 0.1 s	6110.49	2515.99
τ : 0.2 s	6179.19	2544.67
τ : 0.5 s	6385.92	2630.63
τ : 1 s	6731.94	2773.52
τ : 2 s	7423.70	3057.82
τ : 5 s	9482.80	3899.20
τ : 10 s	12,845.24	5230.16

Table 6. L_1 and L_2 norms for different τ values in the 4 Robot diamond-shaped simulation.

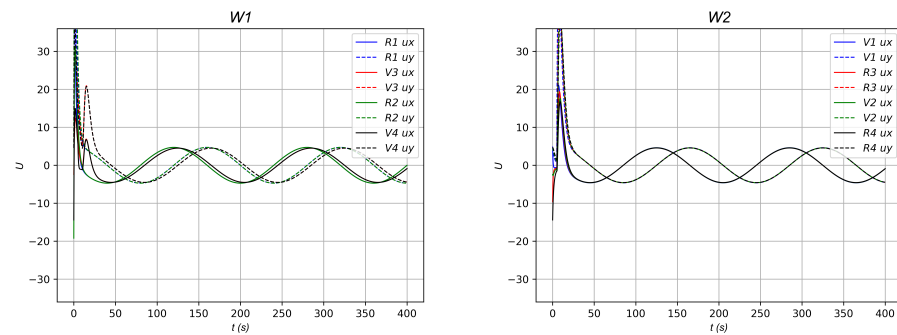
Test	L_1	L_2
τ : 0.1 s	1964.33	875.06
τ : 0.2 s	2022.92	897.39
τ : 0.5 s	2199.31	964.38
τ : 1 s	2495.02	1075.87
τ : 2 s	3091.26	1297.98
τ : 5 s	4894.16	1953.13
τ : 10 s	7886.94	2991.31



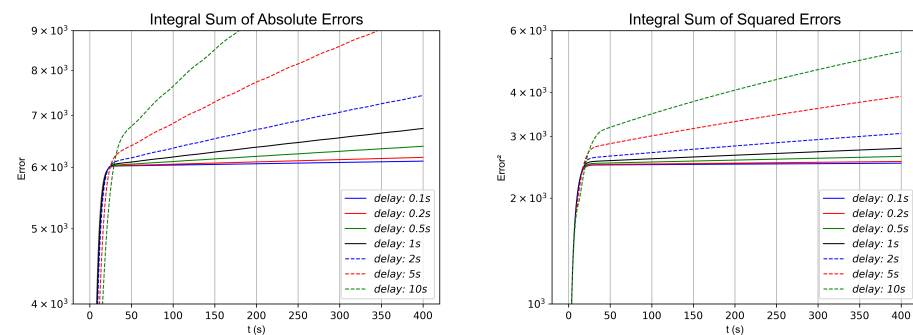
(a) Robot trajectories with delays.

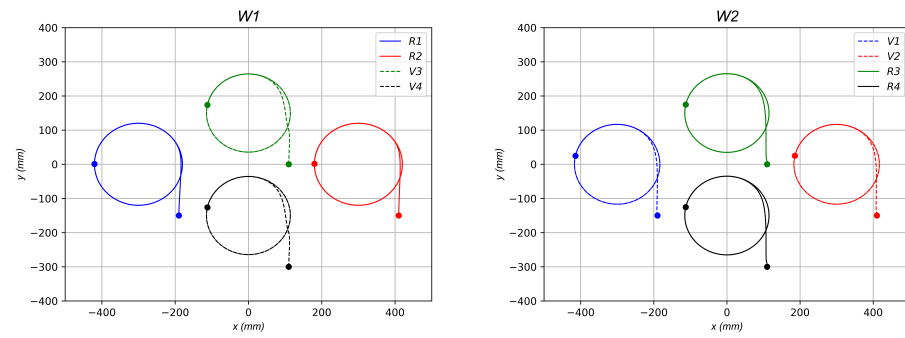


(b) Convergence of the errors with delays.

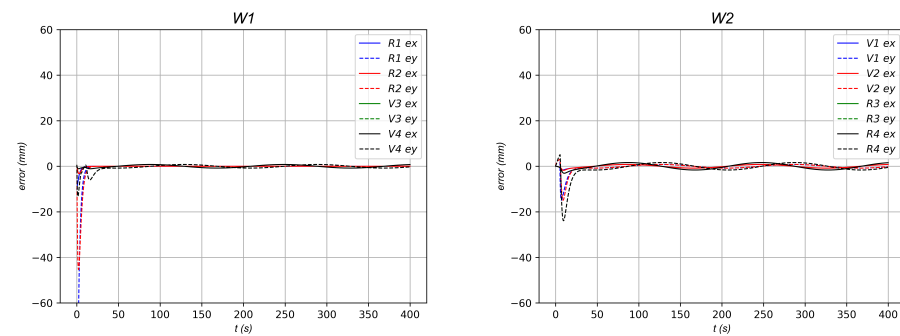


(c) Control inputs with delays.

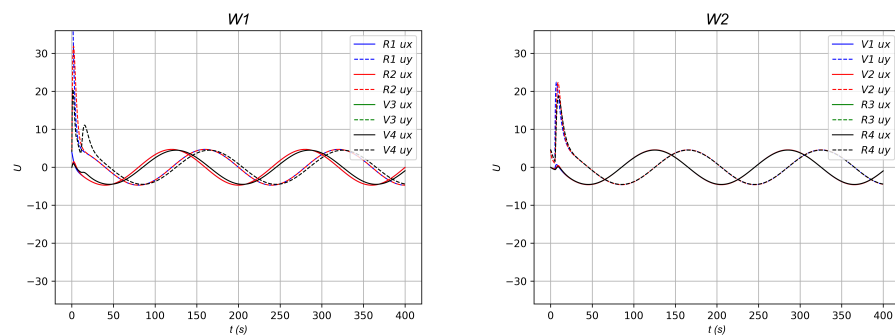
Figure 14. Numerical simulation of 4-robot horizontal line shape formation with delays.**Figure 15.** L_1 and L_2 norms with different delays in simulation of 4-robot horizontal line shape formation.



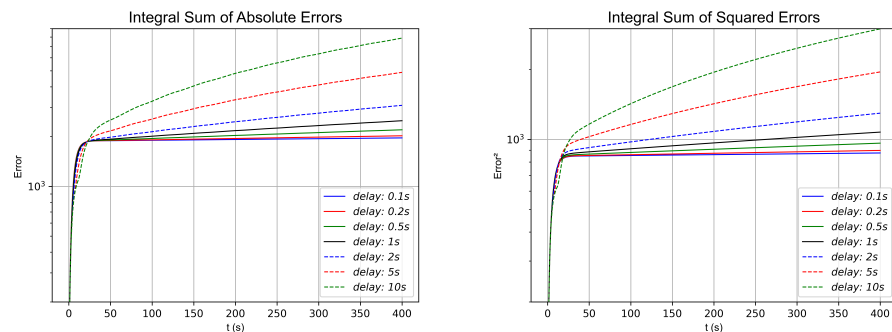
(a) Robot trajectories with delays.



(b) Convergence of the errors with delays.



(c) Control inputs with delays.

Figure 16. Numerical simulation with the 4 Robot diamond-shaped formation topology with delays.**Figure 17.** L_1 and L_2 norms different delay in the 4 Robot diamond-shaped simulation.

As shown in the numerical simulations and the results presented in Tables 4–6 for the different topologies and formations, as well as in the simulations conducted with various delay values at τ , a lower delay value, ideally approaching zero, yields a faster and more

pronounced convergence. This is evident in the figures illustrating the norms L_1 and L_2 for the three configurations in Figures 13, 15, and 17. All tests converge regardless of the selected τ delay value. In particular, as mentioned earlier, our primary interest lies in delays close to 0.2 s, the average delay of MQTT communications, and sporadic delays of up to 2 s. In all tests conducted, even with delay values exceeding those expected in future experiments, the numerical simulations of the three formations under this control strategy successfully converged. Although higher delays result in a slower convergence rate and an increase in accumulated error, all experiments maintained stable results, exhibiting only minor oscillations attributable to the data delay.

5. Experimental Work

The control implementation for all cyber–physical experiments is presented in Figure 18. The n robots are placed in two different workspaces. Every workspace has a web camera (Model C920, Logitech, Lausanne, Switzerland) with a resolution of 1280×720 pixels, installed at a height of 400 mm, that generates a viewing area of around $750 \text{ mm} \times 400 \text{ mm}$. Every camera is connected to a PC computer (Model Core i7, Intel Corp., Santa Clara, CA, USA). The PC reads the Aruco tags (OpenCV-contrib module, OpenCV.org, Palo Alto, CA, USA) at a sample rate of 0.1 s, detecting the *ID* of the robots and calculating the position and orientation of every robot using a visual processing library (OpenCV 4.8.0, OpenCV.org, Palo Alto, CA, USA). The positions and orientations are processed by the computer in the workspace for the real robots, and this same information is sent to the internet through an IoT protocol (MQTT v3.1.1 and 1.6.9, Eclipse Foundation, Ottawa, ON, Canada). using different brokers. The communication is bidirectional, and the two workspaces are connected via the internet. The control law was programmed in each PC using Python (Version 3.13, Python Software Foundation, Wilmington, DE, USA), sending the control inputs to every robot using Bluetooth wireless communication. Note that the control setup is low-cost and scalable. Also, the experimental setup is modular using commercial components, allowing its expansion to more robots and workspaces. Figure 19 is an example of the control implementation using the experimental platform of the topology and the formation structure of the four-robot diamond shape based on the topology shown in Figure 5.

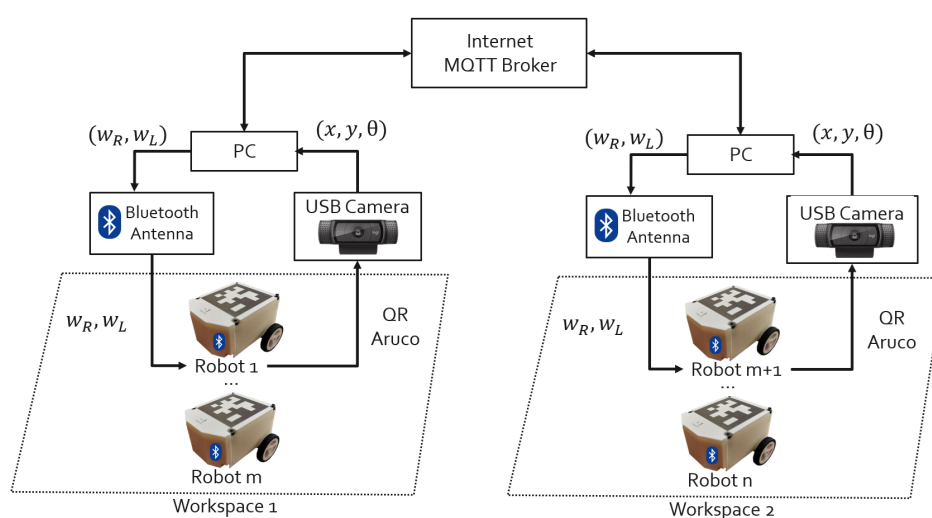
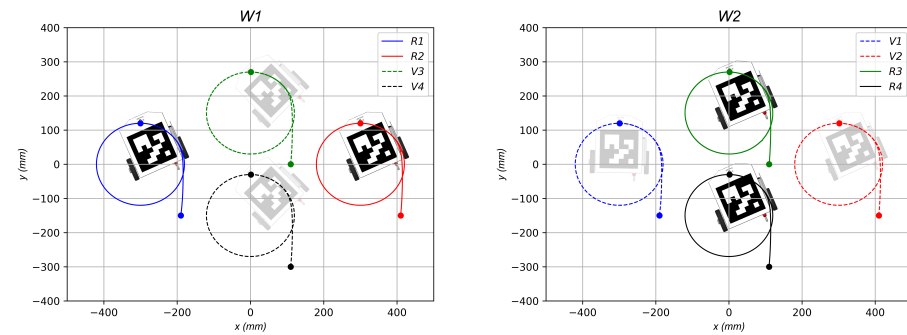
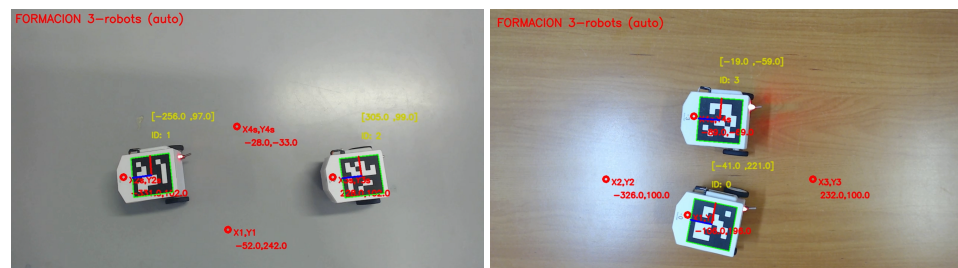


Figure 18. Workspace diagram for the cyber–physical experiments.



(a) Robot distribution workspaces.

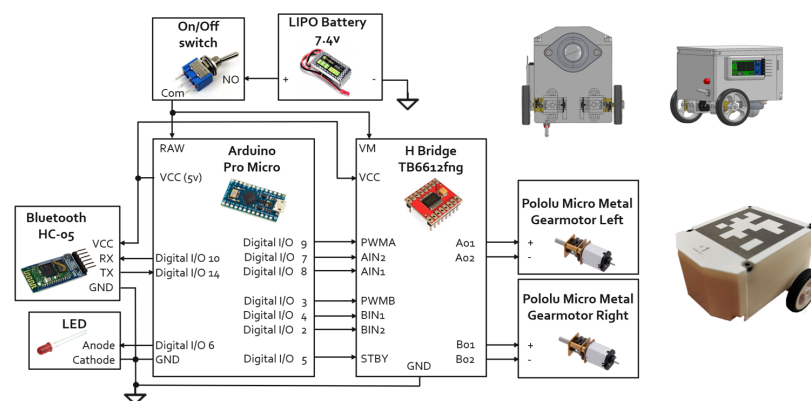


(b) Real robot workspaces.

Figure 19. Control implementation workspace example.

5.1. Unicycle-Type Robots

The control approach was tested in a low-cost experimental robotic platform composed of the four mobile robots shown in Figure 20. Each unicycle-type robot was integrated with two DC motors (Micro metal gear motors HP 6v 1000:1, Pololu Corporation, Las Vegas, NV, USA) with plastic wheels, controlled by a microcontroller (ATmega32U4-based microcontroller, Arduino SA, Somerville, MA, USA). with Bluetooth wireless communication to a PC. The robot case was made with a 3D printer with an Aruco tag on top. The unicycle robots measure 100 mm × 100 mm; this size was selected during development to provide a versatile, easily maneuverable platform that is well suited for both academic and research environments.

**Figure 20.** Diagram of the electronics system for the robots' experimental setup.

A detailed breakdown of the main hardware components and their estimated prices is given in Table 7. The total cost per unit is estimated to be in the range of USD 40 to 70, which is significantly lower than similar platforms described in the literature (typically USD 80 to 150).

Table 7. Unicycle type robot components and costs.

Component	Model	Cost (USD)
Arduino Pro Micro	ATmega32U4-based microcontroller Arduino SA, Somerville, MA, USA	5–7
Bluetooth Module	HC-05 Wavesen, Shenzhen, China	3–5
Motor Driver	TB6612FNG dual motor driver Toshiba Corp., Tokyo, Japan	3–4
DC Motors (2 units)	Small Pololu motors 6v 1000:1 Pololu Corp., Las Vegas, NV, USA	5–10 each
LiPo Battery	7.4 V, 2-cell (2S) battery Line Power, Shenzhen, China	10–14
On/Off Switch + LED	Generic switch with indicator Stereon, Mexico City, Mexico	1–2
Chassis and 3D Printed Parts	PLA material components	3–5
Wheel (2 units)	Pololu Wheel 32 × 7 mm Pololu Corp., Las Vegas, NV, USA	2–6
Ball caster	Pololu Ball Caster 1/2" Metal Ball Pololu Corp., Las Vegas, NV, USA	3–6
Total		40–70

This cost-effective design reduces the experimentation cost for multiple robots or multiple workspaces and also facilitates easy replication and scalability in academic and research settings.

The kinematic model of the unicycle-type robots shown in Figure 20 is given, according to Figure 21, by

$$\dot{x}_i = v_i \cos \theta_i \quad (21)$$

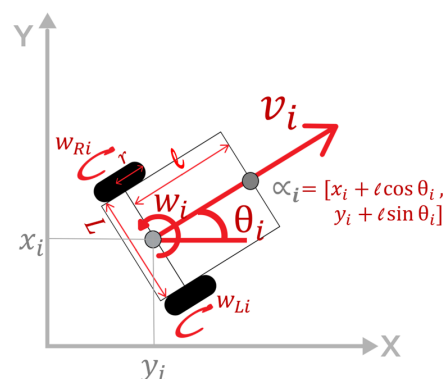
$$\dot{y}_i = v_i \sin \theta_i \quad (22)$$

$$\dot{\theta}_i = w_i \quad (23)$$

with $i = 1, \dots, n$, where $[x_i, y_i]^\top$ is the 2D coordinate of the robot and θ_i is its orientation angle with respect to the horizontal axis. The v_i and w_i are the longitudinal and rotational control inputs with respect to the middle point of the axis wheels. These two body-level velocities can be converted to the velocities of the wheels through

$$w_{R_i} = \frac{v_i}{r} + \frac{w_i L}{2r}, \quad w_{L_i} = \frac{v_i}{r} - \frac{w_i L}{2r} \quad (24)$$

where w_{R_i} and w_{L_i} are the angular velocities of the right and left wheels, respectively, r is the radius of the wheels, and L is the distance between the wheels.

**Figure 21.** Kinematic model of a unicycle-type robot.

The frontal point is chosen as control output, shown in Figure 21 with

$$\alpha_i = \begin{bmatrix} x_i + \ell \cos \theta_i & y_i + \ell \sin \theta_i \end{bmatrix}^\top \quad (25)$$

where $\ell \neq 0$. Its dynamics are given by

$$\dot{\alpha}_i = A_i(\theta_i)[v_i, w_i]^\top \quad (26)$$

where $A_i(\theta_i) = \begin{bmatrix} \cos \theta_i & -\ell \sin \theta_i \\ \sin \theta_i & \ell \cos \theta_i \end{bmatrix}$. Since $\det(A_i(\theta_i)) = \ell \neq 0$, it is possible to linearize the dynamics of α_i using the control

$$[v_i, w_i]^\top = A_i^{-1}(\theta_i)\mathbf{u}_i \quad (27)$$

with $\mathbf{u}_i \in \mathbb{R}^2$ as an auxiliary control. Note that in the closed-loop control (26) and (27), the dynamics is reduced to $\dot{\alpha}_i = \mathbf{u}_i$ and the nonlinearities are eliminated, resulting in the single integrator given by (3). Therefore, the control approach can be extended to the unicycle-type robots using the control law (27), where $\mathbf{u}_i$ is defined by the Equations (4)–(6) but dependent on the coordinates α_i of the robots. For the experimental work case developed in Section 5.4, the parameters of the robots constructed for the experimental platform in this study are $r = 17$ mm, $\ell = 90$ mm, and $L = 70$ mm. Here, r is defined as the radius of the robot's wheel, ℓ is the distance to an auxiliary frontal point, and L is the distance between the centers of the two wheels. These values are determined by the size of the robot and the commercial components used. Additionally, due to the selected motor type Pololu® 6V with a 1000:1 gear ratio and a maximum speed of 33 RPM and given the wheel radius of $r = 17$ mm, the robot and control programming impose a maximum wheel angular velocity of $W_{max} = 2.5$ rad/s and a minimum value of $-W_{max} = -2.5$ rad/s, which is approximately 75% of the motors' maximum speed.

5.2. MQTT Broker

For the MQTT IoT connection, Python 3.13 and the paho.mqtt library were used to establish communication between the computer and the MQTT server (broker). An MQTT broker is a message intermediary that facilitates communication between devices and applications via the MQTT protocol; it manages the publication and subscription of messages and topics. In this experiment, two different brokers were employed: a popular commercial broker, `test.mosquitto.org`, and a custom-developed broker hosted on Google Cloud.

The broker `test.mosquitto.org`, is a free commercial broker accessible on port 1883 that does not require encryption or authentication, making it commonly used for MQTT development and testing. Also, a custom MQTT broker was developed using the Google Cloud platform. A virtual machine instance of type “e2-medium” with two virtual CPUs, 4 GB of RAM, and Ubuntu as the operating system was created. Port 1883 was enabled, and the `mosquitto` and `mosquitto-client` packages (versions 1.6.9 and 3.1.1) were installed. Once configured, the public IP of the server was obtained for directing messages to the appropriate IP and port.

For both brokers, latency and communication tests were performed by sending 4000 data packets at a sampling rate of 0.05 s, measuring the time required for data transmission between computers. In Figure 22, the latency times for the `test.mosquitto.org` broker are shown, with an average of 0.23 s and a standard deviation of 0.0671 s, indicating a highly stable broker with occasional maximum latencies of 0.75 s. In contrast, the Google

Cloud broker exhibits an average latency of 0.1744 s and a standard deviation of 0.1738, making it less stable than the commercial broker, with sporadic latencies reaching 2 s.

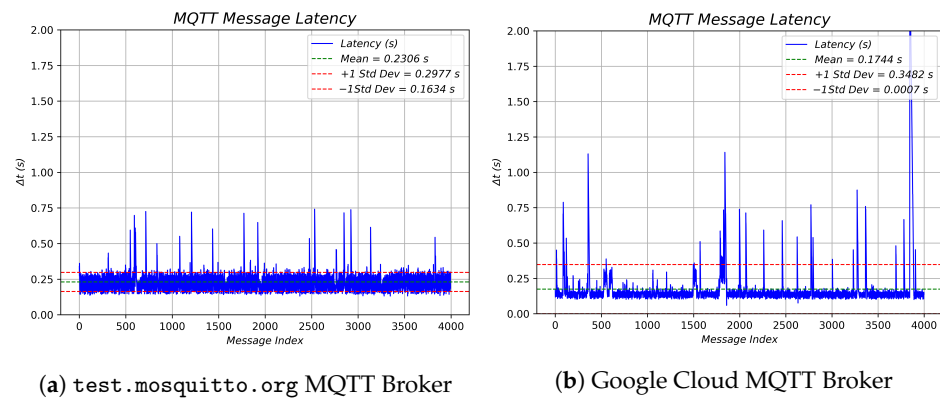


Figure 22. MQTT brokers latency, mean, and standard deviation.

5.3. Position Detection with Aruco QR Code and Computer Vision

For robot position detection, Python 3.13 was utilized along with the OpenCV library to detect ArUco-type QR codes. The detection algorithm provides the x and y coordinates, as well as the orientation angle θ for each robot. Latency tests were carried out to evaluate the processing speed and potential delays when detecting 1, 2, or 4 QR codes, as shown in Table 8, with the latency remaining constant at approximately 0.1 s and a minimal standard deviation of 0.0075 s. In Figure 23b, the results from a test involving 1000 frames capturing four QR codes on four robots confirm that the latency is maintained throughout the experiment.

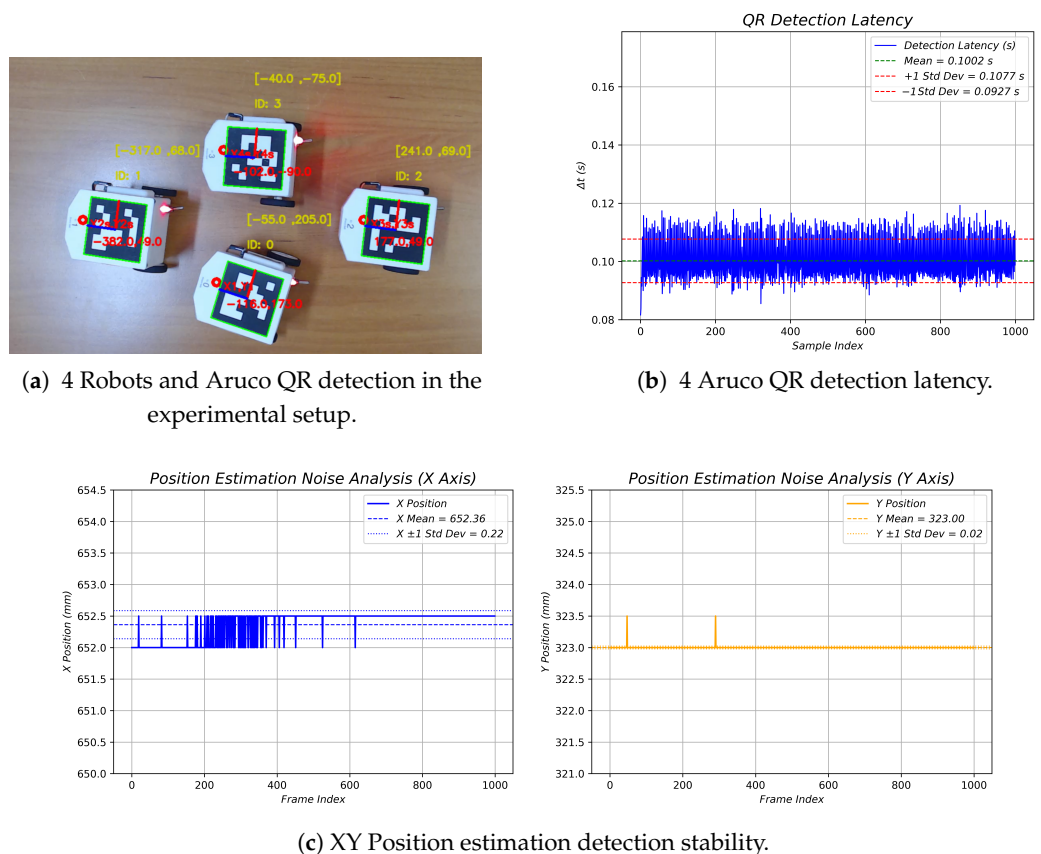


Figure 23. QR computer vision latency and stability.

Table 8. Mean and standard deviation of the Aruco QR code detection latency.

Test	Latency	Std Dev
1 Aruco QR	0.1003	0.0073
2 Aruco QR	0.1002	0.0075
4 Aruco QR	0.1002	0.0075

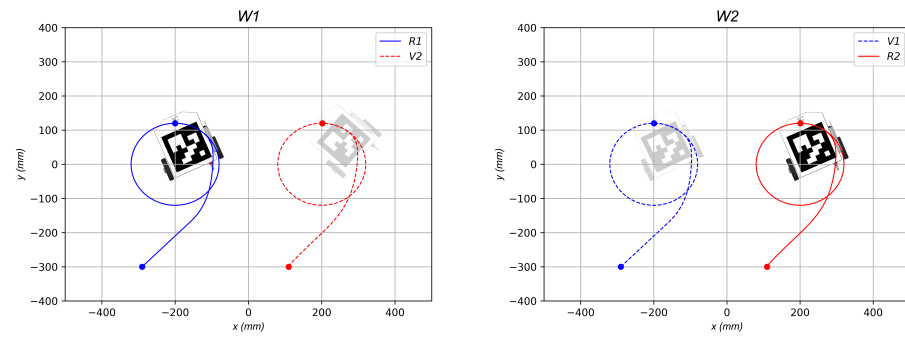
Additionally, the vision system employs a Logitech C920© camera with a resolution of 1280×720 pixels. The camera is positioned 400 mm above the robot formation area, and its field of view covers an area of 750 mm by 400 mm, resulting in a per-pixel area of 0.36 mm^2 . A stability test was performed by measuring the x and y positions of a fixed QR code without moving either the QR code or the camera, thereby assessing the system's stability. As depicted in Figure 23c, based on 1000 frames captured of a stationary QR code, the measurements yielded a standard deviation of 0.22 mm, indicating a resolution of less than 1 mm.

5.4. Experimental Results

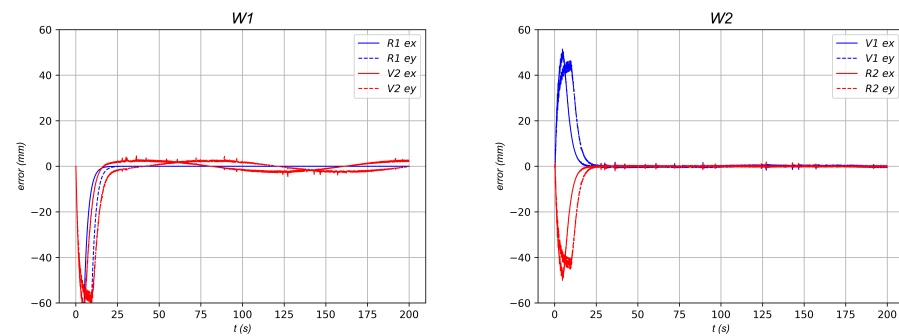
Using the experimental platform, several experiments were conducted with the formation configurations presented in the simulation Section 4 (Figures 3–5). Based on the numerical simulations, a unit gain ($K_c = 1$, $k_r = 1$ and $k_v = 1$) was selected for the physical experiments. Each experiment was carried out with a different sampling time (dt), which determines both the calculation frequency and the rate at which data are transmitted to the MQTT broker. Sampling times of 0.5, 0.2, 0.1, 0.05 and 0.01 s were tested. It is important to note that several of these sampling rates exceed the previously studied data transmission rate of the MQTT system (an average of 0.2 s) and the camera sampling rate (0.1 s). This introduces expected errors and delays in the data, as discussed in the numerical simulation section regarding the robustness of the control model. In addition to testing different dt parameters, the experiments were carried out using two distinct MQTT brokers: a popular commercial broker, `test.mosquitto.org`, and a custom-developed broker hosted in Google Cloud. The errors of all experiments were quantified using comparative tables based on the integral of the L_1 norm (sum of absolute errors) and the L_2 norm (sum of squared errors).

Figure 24 shows the experimental results of the simulation with $dt = 0.05$. The desired trajectory of the leader robot R_1 is given by $\mathbf{z}_d(t) = [-200 + 120 \cos(\frac{\pi}{80}t), 120 \sin(\frac{\pi}{80}t)]^T$ in millimeters, and the MQTT broker selected for this specific experiment was the commercial MQTT broker `test.mosquitto.org`. The robots converge to the desired formation pattern, which is observed by the robots' trajectories in Figure 24a and the error convergence shown in Figure 24b. Note that the errors of the real robots are smaller than the errors of the virtual robots in the experimental data. The control inputs are depicted in Figure 24c, and note that the control input is saturated at a maximum value $U_{max} = 30$ and a minimum value of $-U_{max}$. Note that the control performance is affected by the resolution of the camera, the sample time, and the non-modeled dynamics presented by the floor friction and other real effects.

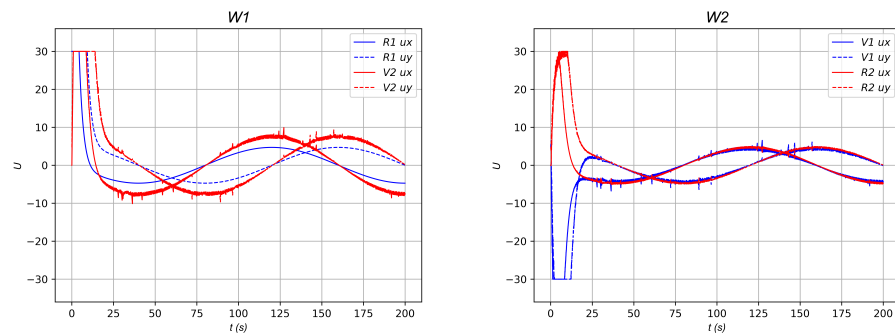
Figure 25 illustrates the longitudinal velocity values V relative to the midpoint between the wheel axes; see Figure 25a. Furthermore, Figure 25b presents the angular velocities, W_R and W_L , corresponding to the right and left wheels of each robot, respectively. Note that the angular wheel velocities are saturated at a maximum value of $W_{max} = 2.5 \text{ rad/s}$ and a minimum value of $-W_{max}$, due to the maximum speed that the physical motors on the experimental platform can achieve.



(a) Robots Trajectories.



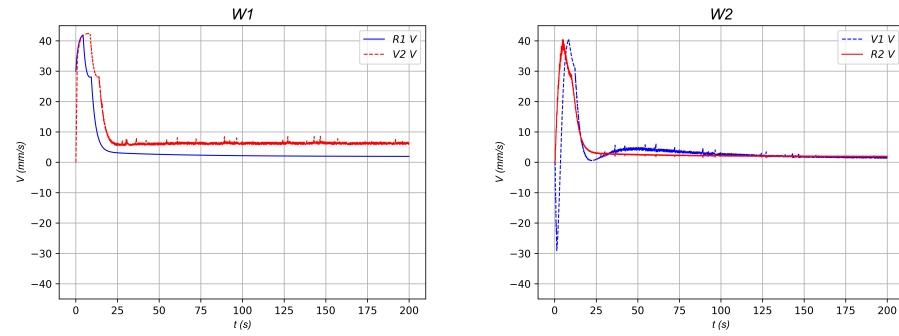
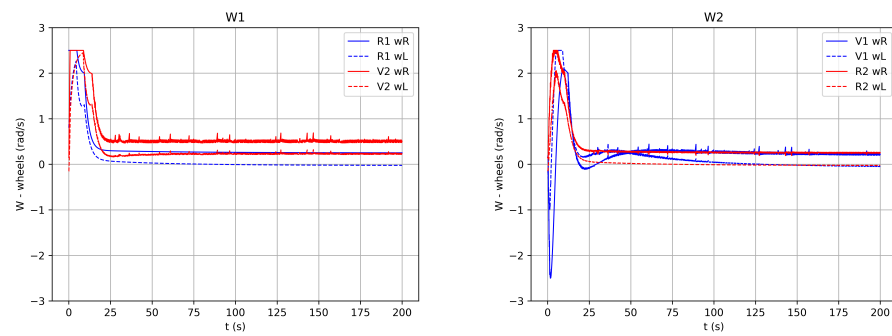
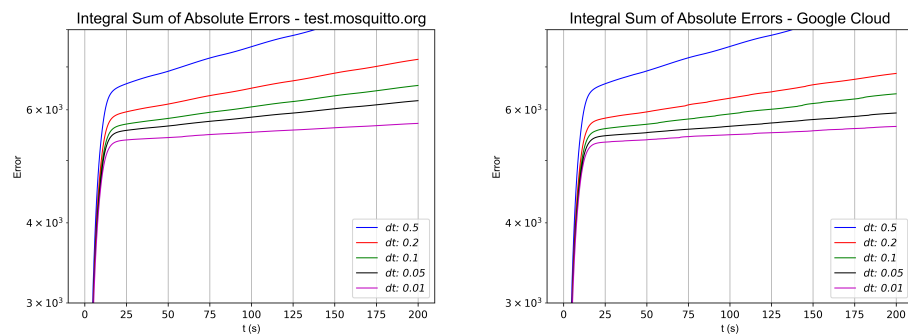
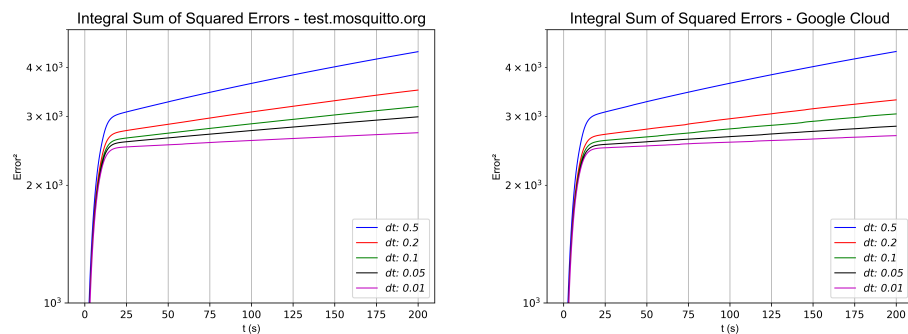
(b) Convergence of the errors.



(c) Control Inputs.

Figure 24. Experimental results for a 2-robot horizontal line shape formation.

In this first formation topology, as shown in Figure 3, several experiments were performed by varying the sample rate dt to assess the stability and convergence rate of the control strategy. The simulations were carried out using different dt values of 0.5, 0.2, 0.1, 0.05, and 0.01, and all the different dt values were evaluated with different MQTT brokers: `test.mosquitto.org` and a custom-developed broker hosted on Google Cloud. For each simulation, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and from both real and virtual robots. The results of the different broker test are shown in the graphs in Figure 26 and summarized in Table 9. Additionally, graphs were generated that depict the instantaneous sum of squared errors (computed by aggregating the errors of each real or virtual robot across both workspaces) to compare which sampling times and MQTT broker configurations offer the greatest stability and the fastest convergence, despite system delays and other challenges inherent in physical robot experiments. These results are illustrated in Figure 27.

(a) Robots V input control value.(b) Robots' W_R and W_L wheel values.**Figure 25.** V , W_R , and W_L values in the experiment of 2-robot horizontal line shape formation.(a) test.mosquitto.org vs. Google cloud MQTT broker L_1 norm values.(b) test.mosquitto.org vs. Google cloud MQTT broker L_2 norm values.**Figure 26.** L_1 and L_2 norms in the experiment of 2-robot horizontal line shape formation.

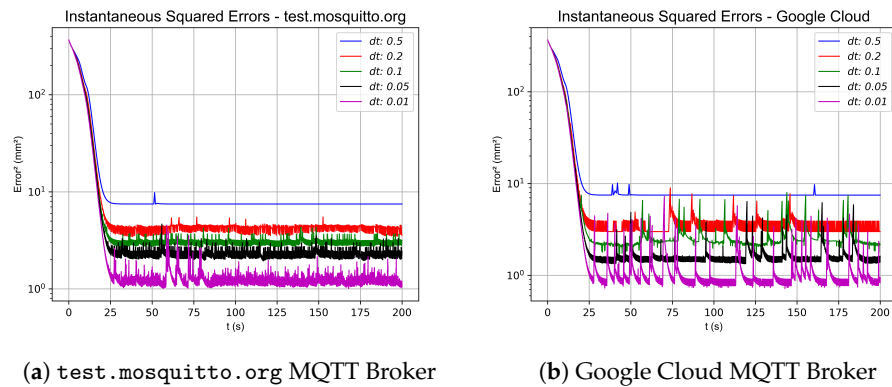


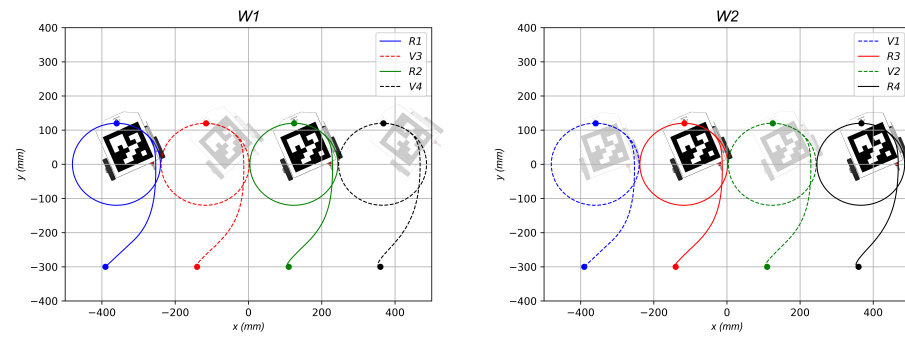
Figure 27. Instantaneous squared error in the experiment of 2-robot horizontal line shape formation.

Table 9. L_1 and L_2 norms in the experiment of 2-robot horizontal line shape formation. (a) test.mosquitto.org MQTT Broker; (b) Google Cloud MQTT Broker.

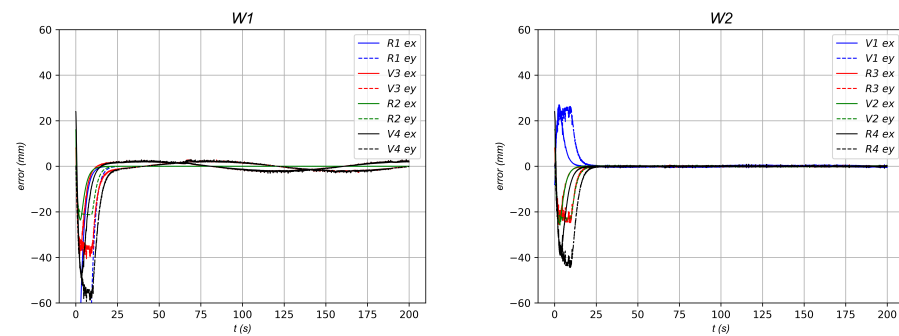
(a)		
Test	L_1	L_2
$dt: 0.5$	8781.32	4390.35
$dt: 0.2$	7189.94	3502.40
$dt: 0.1$	6546.99	3178.08
$dt: 0.05$	6199.24	2991.69
$dt: 0.01$	5714.09	2724.57
(b)		
Test	L_1	L_2
$dt: 0.5$	8788.56	4395.22
$dt: 0.2$	6835.13	3305.84
$dt: 0.1$	6353.93	3043.51
$dt: 0.05$	5931.30	2831.08
$dt: 0.01$	5651.87	2678.36

Figure 28 shows the second experiment with a $dt = 0.05$; the desired trajectory of the leader robot R_1 is given by $\mathbf{z}_d(t) = [-360 + 120 \cos(\frac{\pi}{80}t), 120 \sin(\frac{\pi}{80}t)]^\top$ in millimeters and the same MQTT broker, namely test.mosquitto.org. The robots converge to the desired formation pattern, which is observed by the robots' trajectories in Figure 28a and the error convergence shown in Figure 28b. Note that the errors of the real robots are smaller than the errors of the virtual robots in the experimental data. The control inputs are depicted in Figure 28c, and note that the control input is saturated at a maximum value $U_{max} = 30$ and a minimum value of $-U_{max}$. Note that the control performance is affected by the resolution of the camera, the sample time, and the non-modeled dynamics presented by the floor friction and other real effects.

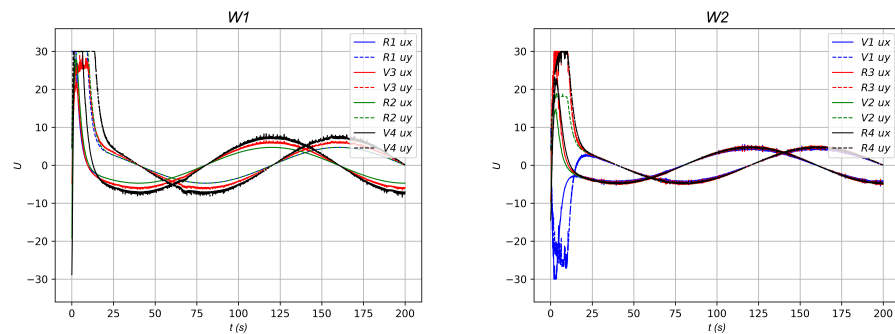
Figure 29 illustrates the longitudinal velocity values V relative to the midpoint between the wheel axes; see Figure 29a. Furthermore, Figure 29b presents the angular velocities, W_R and W_L , corresponding to the right and left wheels of each robot, respectively. Note that the angular wheel velocities are saturated at a maximum value of $W_{max} = 2.5$ rad/s and a minimum value of $-W_{max}$ due to the maximum speed that the physical motors on the experimental platform can achieve.



(a) Robots Trajectories.



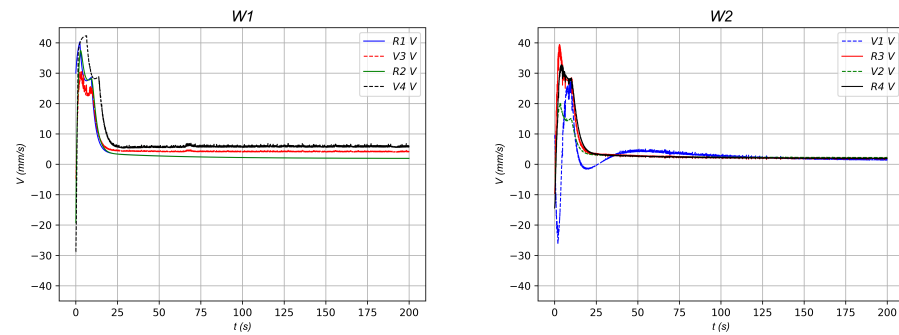
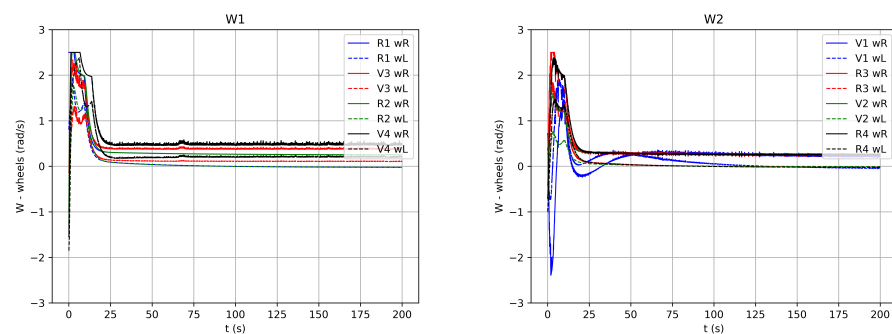
(b) Convergence of the errors.



(c) Control Inputs.

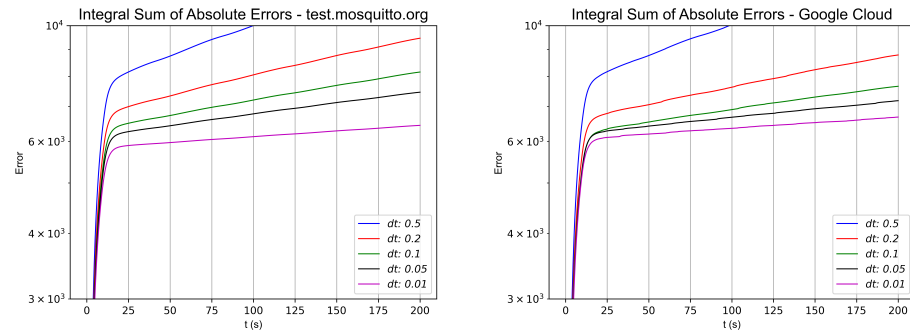
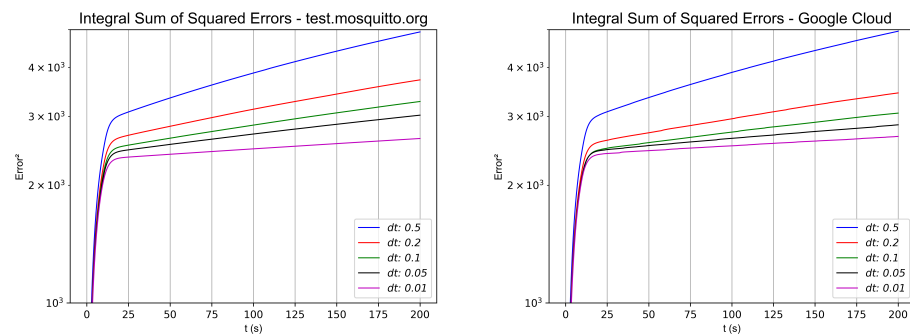
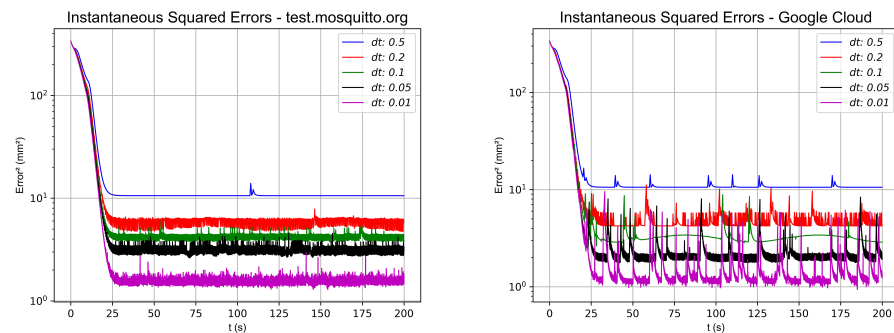
Figure 28. Experimental results of 4-robot horizontal line shape formation.

In the second formation topology, as shown in Figure 4, several experiments were performed by varying the sample rate dt to assess the stability and convergence rate of the control strategy. In this formation, new experiments were also carried out using different dt values and with different MQTT brokers. For each simulation, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and from both real and virtual robots. The results of the different broker test are shown in the graphs in Figure 30 and summarized in Table 10. Additionally, graphs were generated that depict the instantaneous sum of squared errors (computed by aggregating the errors of each real or virtual robot across both workspaces) to compare which sampling times and MQTT broker configurations offer the greatest stability and the fastest convergence, despite system delays and other challenges inherent in physical robot experiments. These results are illustrated in Figure 31.

(a) Robots V input control value.(b) Robots' W_R and W_L wheel values.**Figure 29.** V , W_R , and W_L values in the experiment of 4-robot horizontal line shape formation.**Table 10.** L_1 and L_2 norms in the experiment of 4-robot horizontal line shape formation. (a) test.mosquitto.org MQTT broker. (b) Google Cloud MQTT broker.

(a)		
Test	L_1	L_2
$dt: 0.5$	12,502.65	4932.40
$dt: 0.2$	9459.63	3719.16
$dt: 0.1$	8150.74	3274.36
$dt: 0.05$	7458.17	3021.21
$dt: 0.01$	6444.53	2633.53
(b)		
Test	L_1	L_2
$dt: 0.5$	12,558.79	4952.44
$dt: 0.2$	8786.86	3444.10
$dt: 0.1$	7654.73	3057.90
$dt: 0.05$	7184.19	2854.59
$dt: 0.01$	6683.78	2665.83

Figure 24 shows the experimental results with a $dt = 0.05$; the desired trajectory of the leader robot R_1 is given by $\mathbf{z}_d(t) = [-300 + 120 \cos(\frac{\pi}{80}t), 120 \sin(\frac{\pi}{80}t)]^\top$ in millimeters and the MQTT broker test.mosquitto.org. The robots converge to the desired formation pattern. The robots' trajectories are observed in Figure 32a, and error convergence is shown in Figure 32b. Note that the errors of the real robots are smaller than the errors of the virtual robots in the experimental data. The control inputs are depicted in Figure 32c, and note that the control input is saturated at a maximum value $U_{max} = 30$ and a minimum value of $-U_{max}$. Note that the control performance is affected by the resolution of the camera, the sample time, and the non-modeled dynamics effects.

(a) test.mosquitto.org vs. Google cloud MQTT broker L_1 norm values.(b) test.mosquitto.org vs. Google cloud MQTT broker L_2 norm values.**Figure 30.** L_1 and L_2 norms in the experiment of 4-robot horizontal line shape formation.

(a) test.mosquitto.org MQTT broker

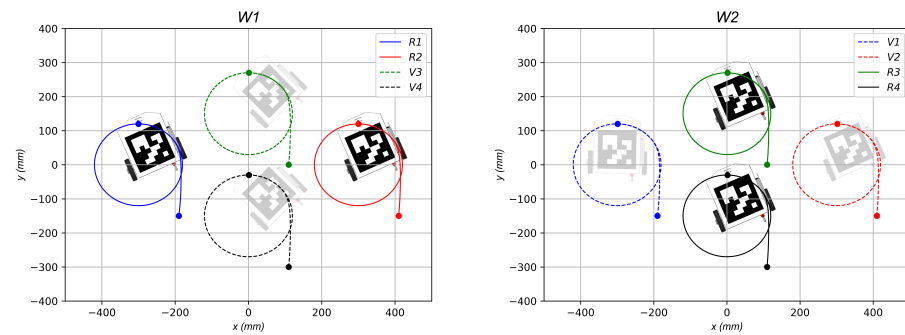
(b) Google Cloud MQTT Broker

Figure 31. Instantaneous Squared error in the experiment of 4-robot horizontal line shape formation.

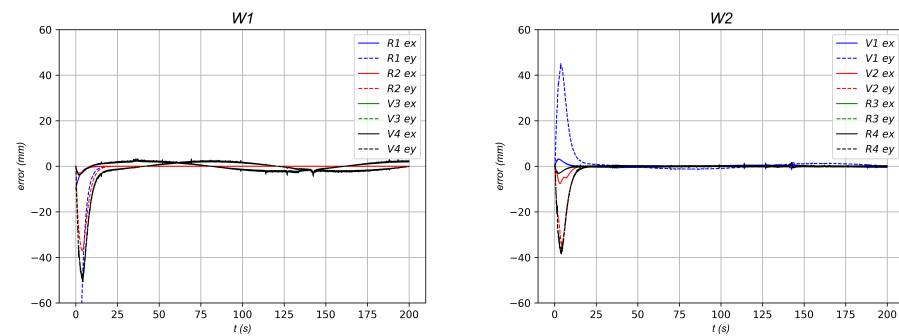
Figure 33 illustrates the longitudinal velocity values V relative to the midpoint between the wheel axes (see Figure 33a). Furthermore, Figure 33b presents the angular velocities, W_R and W_L , corresponding to the right and left wheels of each robot. Both are saturated at a maximum value of $W_{max} = 2.5$ and a minimum value of $-W_{max}$, due to the maximum speed that the physical motors of the real robots can achieve.

In the third formation topology, as shown in Figure 5, several experiments were performed by varying the sample rate dt to assess the stability and convergence rate of the control strategy. In this formation, new experiments were also carried out using different dt values and with different MQTT brokers. For each experiment, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and from both real and virtual robots. For each simulation, the L_1 and L_2 norms were calculated by summing the absolute errors and squared errors, respectively, taking into account errors from both workspaces and from both real and virtual robots. The results of the different broker test are shown in the graphs

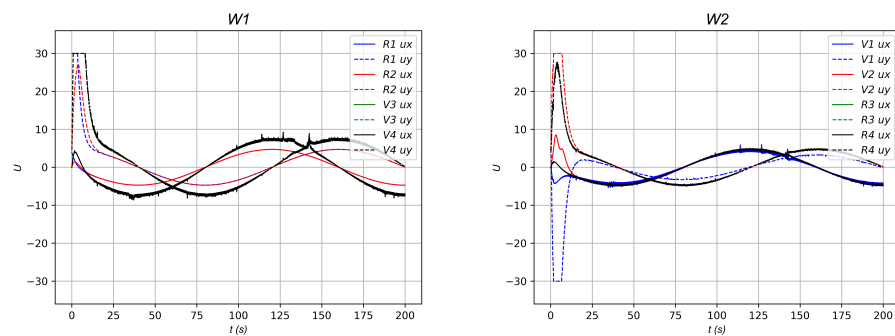
in Figure 34 and summarized in Table 11. Furthermore, graphs were generated that depict the instantaneous sum of squared errors (computed by aggregating the errors of each real or virtual robot across both workspaces) to compare which sampling times and MQTT broker configurations offer the greatest stability and the fastest convergence, despite system delays and other challenges inherent in physical robot experiments. These results are illustrated in Figure 35.



(a) Robots Trajectories.

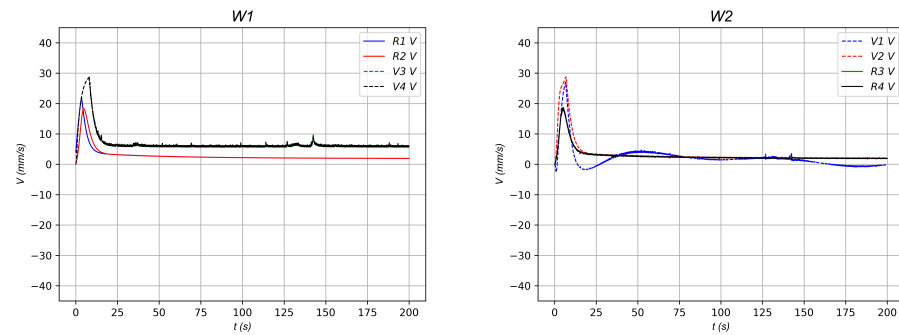
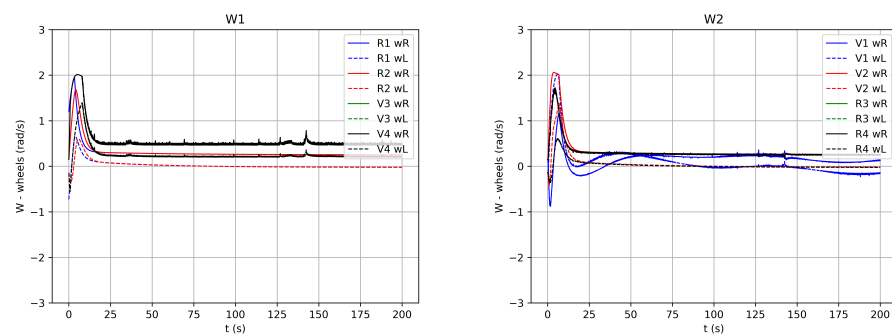


(b) Convergence of the errors.



(c) Control inputs.

Figure 32. Experimental work with the 4 Robot diamond-shaped workspace and formation topology.

(a) Robots V input control value.(b) Robots' W_R and W_L wheel values.**Figure 33.** V , W_R and W_L values in the 4 Robot diamond-shaped formation topology experiment.**Table 11.** L_1 and L_2 norms with in the 4 diamond-shaped Robot experiment. (a) `test.mosquitto.org` MQTT broker. (b) Google Cloud MQTT broker.

(a)		
Test	L_1	L_2
$dt: 0.5$	8238.39	3393.61
$dt: 0.2$	5728.74	2285.73
$dt: 0.1$	4700.82	1892.68
$dt: 0.05$	4054.56	1623.58
$dt: 0.01$	3197.88	1260.00
(b)		
Test	L_1	L_2
$dt: 0.5$	8216.36	3385.04
$dt: 0.2$	4962.72	1969.13
$dt: 0.1$	4294.15	1700.06
$dt: 0.05$	3636.71	1422.18
$dt: 0.01$	3189.84	1231.52

As shown in the numerical simulations and the results presented in Tables 9–11 for the different topologies and formations, as well as in the simulations conducted with various sampling times dt , and also using different MQTT brokers, a lower sampling time dt , ideally approaching zero, even with more communication errors, yields a faster and more pronounced convergence. This is evident in the figures illustrating the norms L_1 and L_2 for the three configurations in Figures 26, 30, and 34.

Furthermore, an analysis of the L_1 and L_2 norm tables from the three experiments shows no significant differences between the commercial broker `test.mosquitto.org` and the Google Cloud broker. In particular, in all experiments, the Google Cloud broker yielded

slightly lower L_1 and L_2 norm values, indicating reduced errors with this MQTT service. This result is directly related to the latency and transmission speeds measured in Section 5.2, where `test.mosquitto.org` exhibited an average latency of 0.23 s, while the Google Cloud broker showed a latency of 0.17 s, as illustrated in Figure 22.

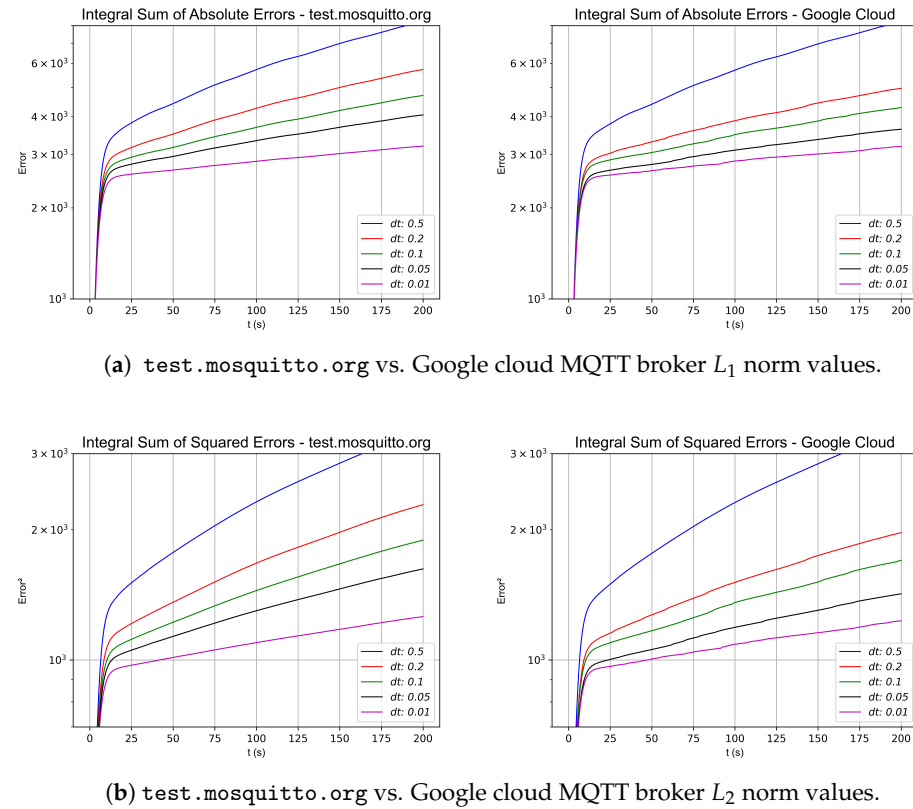


Figure 34. L_1 and L_2 norms in the 4 diamond-shaped Robot experiment.

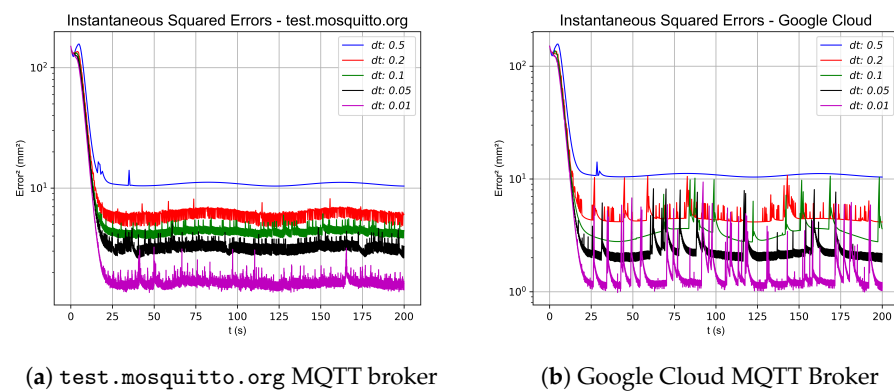


Figure 35. Instantaneous squared error in the 4 diamond-shape Robot experiment.

It is also important to mention that although the Google Cloud broker has a higher standard deviation in transmission time and occasionally experiences significant delays of 0.5, 1, and 2 s, the control strategy remains robust enough to benefit from its lower average latency and faster sampling times. Figures 27, 31, and 35 display the instantaneous sum of squared errors for both brokers in the three different topologies; for `test.mosquitto.org`, the error is relatively constant with occasional peaks due to communication delays and platform limitations, whereas the Google Cloud broker exhibits a general lower error value but is less consistent and more erratic. However, when integrating the absolute and squared errors to compute the L_1 and L_2 norms, the values are lower for the Google

Cloud broker compared to the commercial alternative. Regarding convergence speed, the control model and the numerical simulation results, illustrated in Tables 1–3 for the different topologies and formations, as well as the simulations conducted with various controller gains values (k_c, k_r, k_v) , show that higher controller gains yield a faster and more pronounced convergence. Thus, increasing the gain results in a more rapid convergence of the system. Another critical factor is the latency introduced by the MQTT broker communications and the delays associated with the computer vision system. As shown in the numerical simulations and the results presented in Tables 4–6 for the different topologies and formations, as well as the simulations conducted with various delay values at τ , a lower delay, ideally approaching zero, leads to a faster and more pronounced convergence. Finally, additional factors that affect the convergence speed in the experiments include the maximum velocity of the robot. As explained in Section 5.1, given the maximum speed of 33 RPM and a radius of the wheels of $r = 17$ mm, the control system limits the maximum angular velocity of the wheels to $W_{\max} = 2.5$ rad/s. Moreover, the initial positions of the robots play a significant role; starting from positions that are distant from the desired trajectory or formation will increase the convergence time due to the imposed speed limitations.

Another important detail to consider is that the errors observed in the real robots are smaller than those measured in the virtual robots in the experimental data. This difference is attributed to the inherent communication delays affecting the virtual robots whose positional data include the latency of data transmission, whereas the real robots only experienced the delay associated with the computer vision-based positioning system. Moreover, due to the formation and relative vectors, the virtual robots in workspace 1 incurred a double communication delay: the first delay originated from the virtual robots in workspace 2, and the second arose from the additional latency when transmitting the relative position data of the real robots in workspace 2 back to workspace 1. This results in two accumulated delays in the overall experimental cycle.

6. Conclusions

This paper presents a comprehensive model of cyber–physical formations, offering clear insights into their design, control, and experimentation. The approach is given by mobile robots represented as a single integrator moving in a plane. These robots move in two different workspaces, sharing their postures through the internet. The control law is designed to ensure the convergence of the robots to a formation pattern as if they moved in a single workspace. The control law is designed for a general case of n robots.

The control strategy uses virtual agents that converge to the physical robots moving in the opposite workspace. Two cases were studied: first, a formation convergence without communication delays in measuring the robot's postures, and second, the effect of communication delays when the posture of robots is transmitted over the internet. The latter case is particularly relevant for real-world applications where workspaces are distant from each other and information transmission can be significantly delayed. Other practical drawbacks can generate these delays as part of the cyber–physical setups different from formations made in a single workspace. The convergence of the control law is analyzed and validated through numerical simulations.

In addition, as a proof of concept, the control law is extended to the case of unicycle-type robots. A cost-effective experimental platform is then designed and assembled using commercial electronics and common 3D-printer-based manufacturing techniques. The robots are placed in two different physical workspaces with internet communication using Bluetooth wireless communication and the MQTT protocol. This setup allows for the experimental testing of two distinct control scenarios: one involving delays and the other

without delays. The experiment aims to calculate the inherent packet delivery time of the MQTT server. The experiments demonstrate the performance of the control law and its applicability to practical problems.

It should be noted that the approach of virtual agents can be extended to the case of agents moving in 3D in cyber–physical scenarios including more than two workspaces. The control scheme can be applied to different practical multi-robot tasks, where the coordinated motion of robots in different workspaces or internet networks offers some advantages. For instance, it can emulate mirror effects of robots moving outdoors with a physical remote telepresence in a control station room. Additionally, it can be utilized in formations of robots that are not necessarily in different spaces but connected via separate routers and internet networks, as is typical in factory plants and similar environments. It is also noted that the virtual and real agents can be strategically switched, e.g., when a real robot fails, and it might be convenient to switch to a virtual robot in order to maintain the formation pattern. A significant contribution of this paper is the demonstration of a minimal setup with a single model of robots in a cyber–physical system as the foundation for future extensions and applications.

In future work, improvements will focus on the dynamical modeling of robots, as well as the integration of heterogeneous robot formations, including unicycle-type, omnidirectional, car-like, and other common vehicle models. While the primary focus of this work is on achieving convergence in multi-robot formations, the integration of these diverse models is crucial for enhancing the versatility and efficacy of robotic systems. However, collision avoidance is a critical aspect for ensuring safe operations, especially during transient phases. Therefore, some robust collision-avoidance mechanisms will be investigated in the multi-robot coordination framework. Also, other drawbacks related to internet communication can be addressed, such as intermittency effects, non-fixed delay times, and other expected effects that commonly occur in cyber–physical environments.

Author Contributions: Conceptualization, E.G.H.-M. and J.-J.F.-G.; methodology, E.D.F.-V. and E.G.H.-M.; software, H.G.-N. and A.M.-J.; validation, H.G.-N.; formal analysis, E.D.F.-V.; investigation, M.R.-N.; resources, E.G.H.-M.; writing—original draft preparation, G.F.-A. and H.G.-N.; writing—review and editing, A.M.-J. and J.-J.F.-G.; visualization, E.D.F.-V. and H.G.-N.; project administration, G.F.-A.; funding acquisition, E.G.H.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by project DINVP-051 from the Institute of Applied Research and Technology (INIAT) at Universidad Iberoamericana Ciudad de México, México and by project 5152-25805 from the Institute of Design and Innovation Technology (IDIT) at Universidad Iberoamericana Puebla, México.

Data Availability Statement: The data that support the findings of this study are available from the author, Huber Giron-Nieto, upon reasonable request to huber.giron2@iberopuebla.mx.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Canudas de Wit, C.; Siciliano, B.; Bastin, G. (Eds.) *Theory of Robot Control*; Springer: London, UK, 1996.
2. Valentim, T.; Cunha, R.; Oliveira, P.; Cabecinhas, D.; Silvestre, C. Multi-vehicle Cooperative Control for Load Transportation. *IFAC-PapersOnLine* **2019**, *52*, 358–363. [[CrossRef](#)]
3. Matsuo, R.; Yasuda, S.; Kumagai, T.; Sato, N.; Yoshida, H.; Yairi, T. Residual reinforcement learning for logistics cart transportation. *Adv. Robot.* **2022**, *36*, 404–421. [[CrossRef](#)]
4. Lopes, S.L.; Ferreira, A.I.; Prada, R.; Schwarzer, R. Social robots as health promoting agents: An application of the health action process approach to human-robot interaction at the workplace. *Int. J. Hum.-Comput. Stud.* **2023**, *180*, 103124. [[CrossRef](#)]

5. Gasparino, M.V.; Higuti, V.A.; Sivakumar, A.N.; Velasquez, A.E.; Becker, M.; Chowdhary, G. CropNav: A Framework for Autonomous Navigation in Real Farms. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 11824–11830. [\[CrossRef\]](#)
6. Vichare, S.; Katare, S.; Pawar, A. Wireless Military Defense Robot. *Int. Res. J. Eng. Technol. (IRJET)* **2020**, *7*, 88–95.
7. Benazir Begam, R.; Yogalakshmi, V.; Sanjai Nagaraaja, M.; Sanjay Kumar, A.; Santhosh Priyan, A. IoT based Multipurpose Agricultural Robot. In Proceedings of the 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 5–7 June 2024; pp. 1755–1761. [\[CrossRef\]](#)
8. Do, K.D. Formation tracking control of unicycle-type mobile robots. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2391–2396.
9. Fei, Y.; Shi, P.; Lim, C.P.; Yuan, X. Finite-time observer-based formation tracking with application to omnidirectional robots. *IEEE Trans. Ind. Electron.* **2022**, *70*, 10598–10606. [\[CrossRef\]](#)
10. Hernandez-Martinez, E.G.; Foyo-Valdes, S.A.; Puga-Velazquez, E.S.; Meda-Campaña, J.A. Hybrid architecture for coordination of AGVs in FMS. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 41. [\[CrossRef\]](#)
11. Farrugia, J.L.; Fabri, S.G. Swarm Robotics for Object Transportation. In Proceedings of the 2018 UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, 5–7 September 2018; pp. 353–358.
12. Ferreira-Vazquez, E.D.; Hernández-Martínez, E.G.; Flores-Godoy, J.J.; Fernandez-Anaya, G.; Paniagua-Contro, P. Distance-based formation control using angular information between robots. *J. Intell. Robot. Syst.* **2016**, *83*, 543–560. [\[CrossRef\]](#)
13. Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [\[CrossRef\]](#)
14. Lopez-Gonzalez, A.; Ferreira, E.; Hernández-Martínez, E.G.; Flores-Godoy, J.J.; Fernandez-Anaya, G.; Paniagua-Contro, P. Multi-robot formation control using distance and orientation. *Adv. Robot.* **2016**, *30*, 901–913. [\[CrossRef\]](#)
15. Keung, K.L.; Lee, C.K.M.; Ji, P.; Ng, K.K.H. Cloud-Based Cyber-Physical Robotic Mobile Fulfillment Systems: A Case Study of Collision Avoidance. *IEEE Access* **2020**, *8*, 89318–89336. [\[CrossRef\]](#)
16. Dang, T.V.; Bui, N.T. Obstacle avoidance strategy for mobile robot based on monocular camera. *Electronics* **2023**, *12*, 1932. [\[CrossRef\]](#)
17. Escobar, L.; Moyano, C.; Aguirre, G.; Guerra, G.; Allauca, L.; Loza, D. Multi-Robot platform with features of Cyber-physical systems for education applications. In Proceedings of the 2020 IEEE ANDESCON, Quito, Ecuador, 13–16 October 2020; pp. 1–6.
18. Samie, F.; Bauer, L.; Henkel, J. IoT technologies for embedded computing: A survey. In Proceedings of the 2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Pittsburgh, PA, USA, 2–7 October 2016; pp. 1–10.
19. Nemade, B.; Phadnis, N.; Desai, A.; Mungekar, K.K. Enhancing connectivity and intelligence through embedded Internet of Things devices. *ICTACT J. Microelectron.* **2024**, *9*, 1670–1674. [\[CrossRef\]](#)
20. Deng, Y.; Léchappé, V.; Moulay, E.; Chen, Z.; Liang, B.; Plestan, F.; Han, Q.L. Predictor-based control of time-delay systems: A survey. *Int. J. Syst. Sci.* **2022**, *53*, 2496–2534. [\[CrossRef\]](#)
21. Kharitonov, V. *Time-Delay Systems: Lyapunov Functionals and Matrices*; Springer Science & Business Media: Boston, MA, USA, 2012.
22. Mondie, S.; Kharitonov, V.L. Exponential estimates for retarded time-delay systems: An LMI approach. *IEEE Trans. Autom. Control.* **2005**, *50*, 268–273. [\[CrossRef\]](#)
23. Trinh, H. Exponential stability of time-delay systems via new weighted integral inequalities. *Appl. Math. Comput.* **2016**, *275*, 335–344.
24. Notomista, G.; Cai, X.; Egerstedt, M. Passivity-Based Decentralized Control of Multi-Robot Systems With Delays Using Control Barrier Functions. *arXiv* **2019**, arXiv:1904.04801.
25. Karafyllis, I.; Malisoff, M.; Mazenc, F.; Pierdomenico Pepe, E. Recent Advances in Nonlinear Delay Control Systems. In *Advances in Delays and Dynamics*; Springer: Cham, Switzerland, 2016.
26. Lian, Y.; Xie, W.; Yang, Q.; Zhang, L.; Lin, D.; Zhou, Y. A Novel Multi-Warehouse Mobile Robot Hierarchical Scheduling Strategy Based on Industrial Cyber-Physical System. In Proceedings of the 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Virtual Conference, 10–12 May 2021; pp. 263–269.
27. Pikner, H.; Sell, R.; Karjust, K.; Malayjerdi, E.; Velsker, T. Cyber-physical Control System for Autonomous Logistic Robot. In Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021; pp. 699–704.
28. Kruglova, T.; Schmelev, I.; Sushkov, I.; Filatov, R. Cyber-physical System of the Mobile Robot's Optimal Trajectory Planning with taking into account Electric Motors Deterioration. In Proceedings of the 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 1–4 October 2019; pp. 1–5.
29. Guan, J.; Zhou, W.; Kang, S.; Sun, Y.; Liu, Z. Robot Formation Control Based on Internet of Things Technology Platform. *IEEE Access* **2020**, *8*, 96767–96776. [\[CrossRef\]](#)

30. de Morais, H.R.F.; de Almeida, J.P.L.S.; de Arruda, L.V.R. Running Cooperative Tasks in a Multi-Robot System with Cyber-Physical Features. In Proceedings of the 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), Sao Paulo, Brazil, 18–21 October 2022; pp. 217–222.
31. Li, W.; Wu, J.; Long, C. Formation Control for Unmanned Surface Vehicles Based on Integrative APF and MPC. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Xishuangbanna, China, 5–9 December 2022; pp. 201–206.
32. Kivrak, H.; Karakusak, M.Z.; Watson, S.; Lennox, B. Cyber–physical system architecture of autonomous robot ecosystem for industrial asset monitoring. *Comput. Commun.* **2024**, *218*, 72–84. [\[CrossRef\]](#)
33. Xie, J.; Liu, S.; Wang, X. Framework for a closed-loop cooperative human Cyber-Physical System for the mining industry driven by VR and AR: MHCPs. *Comput. Ind. Eng.* **2022**, *168*, 108050. [\[CrossRef\]](#)
34. Kim, K.J.; Suh, I.H.; Kim, S.H.; Oh, S.R. A novel real-time control architecture for internet-based thin-client robot; simulacrum-based approach. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 4080–4085.
35. Nanu, L.; Colangelo, L.; Novara, C.; Montenegro, C.P. Embedded model control of networked control systems: An experimental robotic application. *Mechatronics* **2024**, *99*, 103160. [\[CrossRef\]](#)
36. Lo, W.T.; Liu, Y.H.; Elhajj, I.; Xi, N.; Shi, Y.; Wang, Y. Co-operative control of internet based multi-robot systems with force reflection. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Taipei, Taiwan, 14–19 September 2003; Volume 3, pp. 4414–4419.
37. Ishikawa, T.; Masuda, M.; Uda, S.; Yokoyama, K.; Hamamoto, K.; Kogiso, K. Constrained remote control of construction machine with time-varying delay and packet loss. *Adv. Robot.* **2023**, *37*, 46–60. [\[CrossRef\]](#)
38. Shahzad, A.; Roth, H. Bilateral telecontrol of AutoMerlin mobile robot with fix communication delay. In Proceedings of the 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2016; pp. 1–6.
39. Irawan, A.; Abas, M.F.; Hasan, N. Robot local network using TQS protocol for land-to-underwater communications. *J. Telecommun. Inf. Technol.* **2019**, *75*, 23–30. [\[CrossRef\]](#)
40. Wang, H.; Li, Z.; Liu, T.; Liang, X. Infinite-horizon optimal control for networked control systems with Markovian packet losses. *Optim. Control. Appl. Methods* **2024**, *45*, 29–44. [\[CrossRef\]](#)
41. Taheri, H.; Zhao, C.X. Omnidirectional mobile robots, mechanisms and navigation approaches. *Mech. Mach. Theory* **2020**, *153*, 103958. [\[CrossRef\]](#)
42. Gonzalez-Sierra, J.; Dzul, A.; Martinez, E. Formation control of distance and orientation based-model of an omnidirectional robot and a quadrotor UAV. *Robot. Auton. Syst.* **2022**, *147*, 103921. [\[CrossRef\]](#)
43. Giron-Nieto, H.; Ochoa-Garcia, O.; Hernandez-Martinez, E.G.; Ramirez-Neria, M.; Fernández Anaya, G.; Ferreira, E.D.; Flores-Godoy, J.J. Cyber–physical Multi-robot Formation: Virtual Agents Approach and Low-Cost Experiments. In Proceedings of the Congreso Nacional de Control Automático, Acapulco, Guerrero, México, 23–27 October 2023; pp. 507–512. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.