



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

“MODELACIÓN Y CONTROL DE UN ROBOT MANIPULADOR AÉREO EN ENTORNOS 3D DE REALIDAD VIRTUAL”

ALEX PATRICIO ERAZO BRAVO

**Trabajo de Titulación modalidad Proyecto de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito para la obtención del grado de**

**MAGÍSTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN
INDUSTRIAL.**

RIOBAMBA – ECUADOR

Octubre 2019

©2019, Alex Patricio Erazo Bravo

Se autoriza la producción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de autor



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo titulado “Modelación y Control de un Robot Manipulador Aéreo en entornos 3D de Realidad Virtual”, de responsabilidad del Ing. Alex Patricio Erazo Bravo ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

Dr. Juan Vargas Guambo M.Sc.

PRESIDENTE

FIRMA

Ing. Jessica Sofía Ortiz Moreno M.Sc.

DIRECTOR

FIRMA

Ing. Jorge Saúl Sánchez Mosquera M.Sc.

MIEMBRO

FIRMA

Ing. Manuel Franklin Silva Monteros M.Sc.

MIEMBRO

FIRMA

Riobamba, Octubre de 2019

DERECHOS INTELECTUALES

Yo, Alex Patricio Erazo Bravo, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

Ing. Alex Patricio Erazo Bravo

C.I. 1803392206

DEDICATORIA

A Dios Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

AGRADECIMIENTO

Le doy gracias a mis padres por apoyarme en todo momento, por los valores que me han inculcado y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida, sobre todo por ser un excelente ejemplo de vida a seguir.

ÍNDICE DE CONTENIDOS

RESUMEN.....	xiv
SUMMARY	xv
CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.1. Planteamiento del Problema	1
1.2. Formulación del problema	2
1.3. Preguntas directrices.....	3
1.4. Justificación	3
1.5. Objetivos de la investigación.....	4
1.5.1. <i>General</i>	4
1.5.2. <i>Específicos</i>	4
1.6. Hipótesis.....	5
1.7 Publicación.....	5
CAPÍTULO II.....	6
2. MARCO TEÓRICO	6
2.1. Antecedentes del Problema	6
2.2. Fundamento Teórico	7
2.3. MARCO CONCEPTUAL	8
2.3.1. <i>Robótica</i>	8
2.3.2. <i>Robots Móviles</i>	9
2.3.3. <i>Robots Aéreos</i>	12
2.3.4. <i>Robots Manipuladores</i>	14
2.3.5. <i>Estructura de los Manipuladores</i>	15
2.3.6. <i>Realidad Virtual</i>	20
2.3.7. <i>Gafas de Realidad Virtual</i>	22
2.3.8. <i>Software Unity</i>	23
2.3.9. <i>Objetos 3D</i>	23
2.3.10. <i>Tipos de Inmersión</i>	25
CAPÍTULO III.....	27
3. MODELACIÓN Y DISEÑO DEL CONTROLADOR	27
3.1. Modelación Cinemática	27
3.1.1. <i>Modelo cinemático del brazo robótico</i>	28
3.1.2. <i>MODELO CINEMÁTICO DEL VEHÍCULO AÉREO NO TRIPULADO, UAV</i>	31
3.1.3. <i>MODELO CINEMÁTICO DEL MANIPULADOR AÉREO</i>	32

3.2. Controlador Cinemático	36
<i>3.2.1. Diseño del Controlador</i>	37
3.3. Estabilidad del esquema de control	38
CAPITULO IV	41
4. DESARROLLO SIMULACIÓN 3D	41
4.1. Simulación 3D	41
<i>4.1.1. Motor Gráfico</i>	41
<i>4.1.2. Modelación</i>	44
<i>4.1.3. Virtualización</i>	46
<i>4.1.4. Jerarquía</i>	49
<i>4.1.5. Pivot virtual</i>	50
<i>4.1.6. Comunicación</i>	51
<i>4.1.7. Animación</i>	52
CAPITULO V	57
5. RESULTADOS EXPERIMENTALES	57
5.1. Controlador del Robot Manipulador Aéreo	57
<i>5.1.1. Resultados de Simulación</i>	57
CONCLUSIONES	
RECOMENDACIONES	
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE FIGURAS

Figura. 2-1: Robótica Industrial.....	8
Figura. 2-2: Tipos de Robots Móviles.	10
Figura. 2-3: Robots Aéreos Flame Wheel e Inspire 2.....	12
Figura. 2-4: Robots manipulador KR 210-2 F Exclusive.	14
Figura. 2-5: Robot manipulador Mats.....	15
Figura. 2-6: Partes de un manipulador.....	16
Figura. 2-7: Tipos de Articulación.....	17
Figura. 2-8: Extremo Operativo tipo pinza.....	18
Figura. 2-9: Semejanza de un brazo robótico y un brazo humano.....	19
Figura. 2-10: Grados de libertad de un brazo robótico.	19
Figura. 2-11: Aplicación Realidad Virtual en Turismo.	21
Figura. 2-12: Aplicación Realidad Virtual en Medicina.....	21
Figura. 2-13: Aplicación industrial de R.V.....	22
Figura. 2-14: Gafas Realidad Virtual Samsung.	23
Figura. 2-15: Terreno con herramienta creado Brushes.....	24
Figura. 2-16: Realidad Virtual Inmersa.	25
Figura. 2-17: Realidad Virtual No Inmersa.	26
Figura. 2-18: Realidad Virtual Semi-Inmersa.....	26
Figura. 3-1: Modelo Cinemático a) Directo, b) Inverso.	27
Figura. 3-2: Manipulador de 4 GDL.....	29
Figura. 3-3: Vehículo aéreo no tripulado.....	31
Figura. 3-4: Manipulador Aéreo.	35
Figura. 3-5: Sistema de trayectoria de Robot Manipulador Aéreo.	37
Figura. 4-1: Interfaz de Usuario.....	42
Figura. 4-2: Gizmo.....	43
Figura. 4-3: Botones de control.	44
Figura. 4-4: Diseño del Manipulador Robot Aéreo SolidWorks.	45
Figura. 4-5: Renderizado Manipulador Robot Aéreo Blender.....	45
Figura. 4-6: Tipo de Textura en UNITY.....	46
Figura. 4-7: Efecto GameObject Light	47
Figura. 4-8: Enfoque de cámaras	48
Figura. 4-9: Entorno con montañas.....	49
Figura. 4-10: Jerarquía en Unity	49

Figura. 4-11: Pivot Virtual.....	50
Figura. 4-12: Proceso de Comunicación.....	52
Figura. 4-13: Proceso de Animación	53
Figura. 4-14: Ejemplo de la función transform.position.....	55
Figura. 4-15: Script adjuntado al objeto	55
Figura. 5-1: Proceso de Simulación.....	58
Figura. 5-2: Robot manipulador Aéreo en Unity	59
Figura. 5-3: Movimiento estroboscópico: Simulación 1 del seguimiento de trayectoria del robot	60
Figura. 5-4: Entorno virtual: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo.....	61
Figura. 5-5: Errores de control: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo.....	62
Figura. 5-6: Velocidades en el cuadricóptero: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo.....	62
Figura. 5-7: Velocidades en el extremo operativo: Simulación 1 del seguimiento de trayectoria del robot	63
Figura. 5-8: Movimiento estroboscópico: Simulación 2 del seguimiento de trayectoria del robot	64
Figura. 5-9: Entorno virtual: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo.....	65
Figura. 5-10: Errores de control: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo.....	66
Figura. 5-11: Velocidades en el cuadricóptero: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo.....	66
Figura. 5-12: Velocidades en el extremo operativo: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo	67
Figura. 5-13: Movimiento estroboscópico: Simulación 3 del seguimiento de trayectoria del robot	68
Figura. 5-14: Entorno virtual: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo.....	69
Figura. 5-15: Errores de control: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo.....	70
Figura. 5-16: Velocidades en el cuadricóptero: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo.....	70
Figura. 5-17: Velocidades en el extremo operativo: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo	71

Figura. 5-18: Movimiento estroboscópico: Simulación 4 del seguimiento de trayectoria del robot	72
Figura. 5-19: Entorno virtual: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo.....	73
Figura. 5-20: Errores de control: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo.....	74
Figura. 5-21: Velocidades en el cuadricóptero: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo.....	74
Figura. 5-22: Velocidades en el extremo operativo: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo	75
Figura. 5-23: Movimiento estroboscópico: Simulación 5 del seguimiento de trayectoria del robot	76
Figura. 5-24: Entorno virtual: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo.....	77
Figura. 5-25: Errores de control: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo.....	78
Figura. 5-26: Velocidades en el cuadricóptero: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo.....	78
Figura. 5-27: Velocidades en el extremo operativo: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo	79

ÍNDICE DE TABLAS

Tabla 3-1: Términos Modelamiento Cinemático.	34
Tabla 3-2: Nomenclatura Partes Robot Manipulador	28
Tabla 3-3: Nomenclatura Velocidades Manipulador Aéreo. ¡Error! Marcador no definido.	

INDICE DE ANEXOS

ANEXO A. Interfaz Unity display 1.....	84
ANEXO B. Interfaz Unity display 2.....	85
ANEXO C. Interfaz Unity display 3	85
ANEXO D. Interfaz Unity display 1 total	86
ANEXO E. Interfaz Unity display 2 total.....	87
ANEXO F. Interfaz Unity display 3 total.....	88

RESUMEN

Modelar y proponer un esquema de control avanzado para realizar tareas de locomoción autónoma de un robot manipulador aéreo conformado por un brazo robótico montado sobre un cuadricóptero; considerando la teoría de control para el análisis de estabilidad y robustez, para la obtención del modelo cinemático del manipulador móvil conformado de un cuadricóptero y brazo robótico en su parte inferior se obtiene inicialmente por separado los modelos cinemáticos del Unidad Manipuladora Aérea (UAV) y del robot manipulador para posterior a su obtención unirlos y analizar el modelo final, de esta manera se garantiza que el modelo cinemático represente de manera más exacta las características físicas del manipulador móvil. De acuerdo con el modelo encontrado se desarrolla el controlador cinemático, dotando así, de un control autónomo al manipulador móvil para cumplir tareas de seguimiento de trayectorias en ambiente no estructurados. Para el análisis de estabilidad del esquema de control se utiliza la función candidata del método de Lyapunov, el cual permite validar el controlador cinemático encontrado, garantizando que el controlador cinemático cumpla con los objetivos planteados. Los resultados obtenidos de los experimentos son simulados en un entorno de realidad virtual 3D en donde se aprecia de mejor manera el correcto funcionamiento del controlador. El sistema de aprendizaje cada día va siendo más interactivo cuyo objetivo es crear al estudiante la sensación de inmersión en entornos de simulación casi reales, se recomienda utilizar como plataforma de realidad virtual Unity, el entorno 3D de realidad virtual permite inmersión del operador humano en donde se puede visualizar de una manera más didáctica y casi real los resultados obtenidos del controlador creado y su estabilidad al seguir una trayectoria establecida.

PALABRAS CLAVES: < TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, CONTROL AUTOMÁTICO>, < LYAPUNOV (MÉTODO) >, <REALIDAD VIRTUAL>, <EXTREMO OPERATIVO >, <ROBOT MANIPULADOR AÉREO>, <VIRTUALIZACIÓN>, <UNITY (SOFTWARE)>, <MEMORIA COMPARTIDA>.

SUMMARY

The modelling and proposal of an advanced control scheme to perform autonomous locomotion tasks of an aerial manipulator robot consisting of a robotic arm mounted on a quadcopter: considering the control theory for stability and robustness analysis to obtain the Cinematic model of the mobile manipulator formed by a quadcopter and robotic arm, the Cinematic models of the Unmanned Aerial Vehicle (UAV) are initially obtained separately and the manipulator robot for subsequently joining them and analyze the final model, this way it is guaranteed that the kinematic model more accurately represents the physical characteristics of the mobile manipulator. In accordance with the model obtained, the kinematic controller is developed, thus providing autonomous control to the mobile manipulator to carry out trajectory monitoring tasks in unstructured environments. For the stability analysis of the control scheme, the candidate function of the Lyapunov method is used, which allows the Cinematic controller found to be validated, ensuring that the kinematic controller meets the stated objectives. The results obtained from the experiments are simulated in a 3D virtual reality environment where the correct operation of the controller is better appreciated. The learning system is becoming more interactive every day whose objective is to create the student's feeling of immersion in almost real simulation environments, it is recommended to use as a Unity virtual reality platform, the 3D virtual reality environment that allows immersion of the human operator where the results obtained from the controller created and its stability can be visualized in a more didactic and almost real way by following an established path.

KEY WORDS: <ENGINEERING TECHNOLOGY AND SCIENCES>, <AUTOMATIC CONTROL>, <LYAPUNOV (METHOD)>, <VIRTUAL REALITY>, <OPERATING EXTREME>, <AERIAL HANDLING ROBOT>, <VIRTUALIZATION>, <UNITY (SOFT)>, <SHARED MEMORY>.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del Problema

Los avances tecnológicos han permitido la implementación de la robótica en diversas áreas de estudio, enfatizando su interés en resolver problemas de locomoción de los robots, estos deben desarrollar tareas predestinadas en ambientes de trabajo no estructurados (Hongtai et al., 2013) (López et al., 2013). Estudios desarrollados definen aplicaciones relevantes dentro de los campos de industria, agrícola, militar, entre otra, desarrollando actividades consideradas como peligrosas, imposibles o inaceptables para el ser humano (Lidoris et al., 2009).

En este contexto es indispensable el análisis del tipo de locomoción de los robots, dentro de esto se encuentra: *i) Robots con ruedas* es el tipo de robot móvil más común, se utilizan frecuentemente para el transporte de materiales, mercancías y/o personas, las ruedas le proporcionan una movilidad con una alta eficiencia, pero suelen estar restringidos a entornos cuya superficie es por lo general plana o con pocas irregularidades ; *ii) Robots acuáticos*, los robots submarinos son obviamente robots que se desenvuelven en el medio acuático marino, los AUV (Autonomous Underwater Vehicles) son robots no tripulados con navegación autónoma, mientras que los ROV (Remotely Operated Vehicles) son robots tripulados desde un barco y suelen tener una dependencia energética. También podemos encontrar, dentro de los AUV, los planeadores o gliders, que son robots que planean entre la superficie marina y debajo del mar para mejorar la eficiencia en cuanto al consumo energético (Wiki , 2005); *iii) Robots aéreos* son robots que vuelan de forma no tripulada UAV (Unmanned Arial Vehicles) o por control remoto RPV (Remotely Piloted Vehicle), se usan principalmente en aplicaciones militares, en los que a diferencia de los misiles, éstos son capaces de mantener un nivel de vuelo controlado y sostenido.

El sistema de locomoción es la principal característica de un robot que se encuentra condicionado por su entorno. De acuerdo a las características del robot manipulador aéreo se generará el control de locomoción en ambientes parcialmente estructurados, para lo cual se desarrolla el control cinemático considerando la posición-orientación del sistema operativo

como un solo sistema. En este contexto, este trabajo propone un sistema que se caracteriza por tener un alto grado de redundancia, debido a que combina la capacidad de manipulación de un brazo robótico de base fija con la navegación de un vehículo aéreo no tripulado con ala rotativa. Estos sistemas permiten realizar las misiones más habituales de los sistemas robóticos que requieren tanto la capacidad de navegación aérea y manipulación.

El presente proyecto de investigación tiene como objetivo proponer y analizar analíticamente la estabilidad al maniobrar de forma coordinada un robot manipulador aéreo, considerando en el desarrollo los efectos negativos que ocasionan los retardos de tiempo, a fin de incrementar la transparencia del sitio remoto a través de herramientas como el software Unity3D-Matlab.

En el plan nacional de Buen Vivir vigente se encuentran detallados doce objetivos nacionales para el buen vivir, de los cuales se hará énfasis en el cuarto objetivo que dice *Fortalecer las capacidades y potencialidades de la ciudadanía*, enfocándose en la Ciencia, tecnología e innovación (Secretaría Nacional de Planificación y Desarrollo , 2013), bajo este contexto en este proyecto se propone el desarrollo de un esquema de control para un robot manipulador aéreo. La propuesta contempla un esquema de control en cascada formada por dos subsistemas: *i) control cinemático* para la ejecución de una tarea a través de un coordinado multi-operador; y *ii) simulación 3D* que permita la inmersión del operador humano para el análisis del desempeño del algoritmo de control propuesto.

El trabajo propuesto tiene cuatro ítems principales: 1) modelar la cinemática de un robot manipulador aéreo; 2) desarrollar un esquema de control basado en la cinemática del robot para tareas de locomoción autónoma de un robot manipulador aéreo en espacios de trabajos parcialmente estructurado; 3) realizar una simulación 3D en realidad virtual que permita la inmersión del operador humano para el análisis del desempeño del algoritmo de control propuesto y finalmente 4) garantizar por simulación la estabilidad y robustez de los sistemas de control propuestos.

1.2. Formulación del problema

¿Cómo se controla un robot manipulador aéreo a través de un coordinado multi-operador?

1.3. Preguntas directrices

¿Cómo se analiza el modelado matemático de un robot manipulador aéreo?

¿Cómo se realiza un esquema de control avanzado?

¿Cómo usar el esquema de control avanzado para tareas de locomoción autónoma de un robot manipulador aéreo?

¿Cómo controlar un robot manipulador aéreo considerando de la teoría de control para el análisis de estabilidad y robustez?

1.4. Justificación

Existen en la actualidad numerosas aplicaciones civiles que utilizan vehículos aéreos para realizar tareas de inspección, vigilancia o monitoreo. Prácticamente la totalidad de las mismas son llevadas a cabo mediante el uso de aviones y helicópteros tripulados. Sin embargo, hay ocasiones en las que el alto costo de operación, el riesgo que corren las personas a bordo del vehículo y/o la incapacidad para acceder a ciertas áreas justifican el uso de vehículos aéreos no tripulados. (Andaluz et al., 2015)

En estudios recientes se ha progresado de forma significativa en la investigación y desarrollo de vehículos aéreos no tripulados (UAVs) para tareas que involucran contacto físico y actuación sobre objetos del entorno tales como el transporte y despliegue de cargas, la toma de muestras, el agarre en vuelo e incluso la manipulación aérea empleando fundamentalmente plataformas aéreas de ala rotatoria. Por lo que para aplicaciones que requieren tanto la capacidad de navegación aérea y manipulación se requiere de robot manipuladores aéreos. Los manipuladores aéreos, se caracterizan por tener un alto grado de redundancia, que combina la capacidad de manipulación de un manipulador de base fija con la navegación de un vehículo aéreo no tripulado con ala fija o rotativa, ver Figura. 2-3. En este sentido, para UAVs de baja capacidad de carga, se emplearon inicialmente garras fijas que se incorporaban bajo la plataforma (Orsag, Stjepan, & Yu, 2014), permitiendo al sistema transportar objetos de poco peso y reducido tamaño. No obstante, la incertidumbre en el posicionamiento durante la maniobra de vuelo a punto fijo, inherente a las plataformas de ala rotatoria, y la escasa maniobrabilidad de la garra dificultaban el agarre preciso de la carga de forma autónoma (Soonkyum & Fink, 2009).

En la mayoría de los trabajos sólo se realizan experimentos con operadores humanos en un entorno de simulación (“Human-in-the Loop” o HITL, por sus siglas en inglés) con retardos de tiempo constantes y/o muy inferiores a los reales. Si bien algunos trabajos han realizado avances en tratar de proveer al operador de señales hápticas para mejorar su conciencia situacional, para que “sienta” la dinámica del vehículo o para indicarle la presencia de obstáculos; pocos estudios se han realizado con el objeto de aumentar la transparencia en el sitio local, coordinar la maniobrabilidad de varios operadores humanos sobre un mismo manipulador aéreo, compensar los efectos negativos de retardos de tiempo y en un entorno virtual 3D.

Se proponen distintas estrategias para maniobrar coordinadamente en ambientes no estructurados el sistema robótico conformado por dos robots (brazo más UAV); se conceptualiza en controlar los movimientos deseados del manipulador móvil a través de especificaciones en función de los atributos de la aplicación, i.e., posición-orientación del UAV y posición-orientación y manipulación del brazo robótico. Estos atributos permiten la selección adecuada de un conjunto de referencias deseadas para el control coordinado del extremo operativo del brazo robótico en función de la posición-orientación del UAV.

1.5. Objetivos de la investigación

1.5.1 General

Modelar y proponer un esquema de control avanzado para realizar tareas de locomoción autónoma de un robot manipulador aéreo conformado por un brazo robótico montado sobre un cuadricóptero; considerando la teoría de control para el análisis de estabilidad y robustez.

1.5.2 Específicos

- Modelar la cinemática de un robot manipulador aéreo conformado por un brazo robótico montado sobre un cuadricóptero en la que se considere la posición-orientación del extremo operativo como un sólo sistema. El modelo debe considerar como entradas velocidades lineales y angular.
- Proponer un esquema de control basado en la cinemática del robot para tareas de locomoción autónoma de un robot manipulador aéreo en espacios de trabajos parcialmente estructurados. Asimismo, se deberá garantizar analíticamente la estabilidad de los algoritmos de control propuestos.

- Desarrollar un entorno 3D en realidad virtual que permita la inmersión del operador humano para el análisis del desempeño del algoritmo de control propuesto.
- Realizar evaluaciones por simulación a fin de evaluar el desempeño de los algoritmos de control propuestos utilizado para aplicaciones de locomoción.

Nota: La simulación del robot manipulador aéreo se realizará en el software Unity3D-Matlab.

1.6. Hipótesis

“Se puede maniobrar un robot manipulador aéreo a través de un control coordinado multi-operador. ”

1.7. Publicación

El siguiente trabajo ha sido publicado durante el desarrollo de esta tesis.

- Jessica S. Ortiz, **Alex P. Erazo**, Christian P. Carvajal, José A. Pérez, Luis E. Proaño, Frankiln M. Silva M. and Víctor H. Andaluz, “**Modeling and Kinematic Control for Trajectory Tracking of Aerial Mobile Manipulators**” In: Zeghloul S., Romdhane L., Laribi M. (eds) Computational Kinematics. Mechanisms and Machine Science, vol 50, pp. 87-95. Springer, Cham, 2017.

CAPÍTULO II

2 MARCO TEÓRICO

2.1. Antecedentes del Problema

La robótica desde décadas atrás tiene gran relevancia con robots que realizaban trabajos repetitivos en ambientes estructurados, varios de estos realizando trabajos en sectores industriales, militares, salud, y más, en los últimos estudios los robots son capaces de trabajar en ambientes no estructurados de forma autónoma o semi-autónoma. En la nueva generación de robots se habla de robots con inteligencia artificial capaces de aprender de su entorno y mejora en su locomoción se desenvuelven en ambientes terrestres, aéreos o acuáticos realizando tareas que suelen ser consideradas como peligrosas para los seres humanos.

La robótica es muy extensa y a la vez abarca en su clasificación a los robots industriales y de servicio, este último se basa principalmente en facilitar la vida a los seres humanos no solo en el ámbito de la comodidad, sino también en realizar tareas peligrosas que muchas veces no pueden ser ejecutadas por el hombre.

Los manipuladores móviles han sido uno de los principales temas de la investigación académica en los últimos años ya que permiten realizar tareas más sofisticadas, especialmente para vehículos aéreos no tripulados (UAVs), la movilidad de estos no se limita a desplazamiento en superficies planas. (Víctor H. Andaluz, 2016)

Algunas de las arquitecturas de control implementadas para el control del manipulador aéreo son las siguientes: i) *Robots teleoperados* son aquellos que son gobernados como un esclavo (controlar su movimiento y la fuerza ejercida) si se encuentra ubicado en una zona remota (puede que el slave esté realmente muy lejos o puede que esté en un entorno hostil del que haya que proteger al operador) a través del manejo de un robot master localizado en el punto de trabajo del operador. ii) *Robots autónomos* pueden operar con un alto grado de autonomía, lo que es particularmente deseable en ambientes desconocidos y peligrosos. iii) *Robots semiautónomos* vienen a ser en si un sistema híbrido entre los robots teleoperados y autónomos que son usados mayormente en ambientes industriales (Bermúdez, 2015). El objetivo de estos controles es brindar robustez y flexibilidad a los sistemas a maniobrar garantizando estabilidad en su función y movilidad.

2.2. Fundamento Teórico

Varios estudios en las últimas décadas han integrado a la robótica, enfocándose en robots autónomos que se muevan en ambientes no estructurados en el área de los manipuladores, por lo que se hace mención de diferentes estudios:

Raúl Cano *et al*, 2014 en su trabajo denominado Diseño mecánico de un manipulador aéreo ligero de 6 Grados de Libertad (GDL) para la construcción de estructuras de barras, en el cual se muestra la metodología que se ha seguido en el diseño mecánico de un manipulador ligero de 6 GDL para la construcción de estructuras de barras desde un UAV de ala rotatoria. La arquitectura del manipulador aéreo se fundamenta en un análisis exhaustivo de su funcionamiento, un estudio del grado de manipulabilidad de las diferentes opciones posibles y una evaluación previa de la motorización requerida. Finalmente, se realiza el cálculo del modelo cinemático del manipulador, para el que se obtiene una expresión analítica.

Andaluz *et al*, 2016, en su trabajo denominado Modeling and Kinematic Nonlinear Control of Aerial Mobile Manipulators, en el cual propone un modelado cinemático y un controlador cinemático no lineal para un robot de manipulación móvil aéreo autónomo que genera órdenes saturando la velocidad de referencia para un seguimiento preciso de la trayectoria.

En la actualidad existen simuladores que permiten analizar el desempeño de diferentes esquemas de control autónomo y / o teleoperado en entornos estructurados, parcialmente estructurados y no estructurados. Para la interacción robot-ambiente se considera software de realidad virtual Unity3D, este software intercambia información con MATLAB para ejecutar diferentes algoritmos de control propuestos a través del uso de memoria compartida. El intercambio de información en tiempo real entre los dos software es esencial porque los algoritmos de control avanzados requieren una retroalimentación de la interacción robot-entorno para cerrar el bucle de control, mientras que el robot simulado actualiza sus parámetros cinemáticos y dinámicos dependiendo de las variables de control calculadas por MATLAB. (Andaluz V.H. et al., 2016)

2.3. Marco Conceptual

2.3.1. Robótica

La robótica es la ciencia que estudia el diseño, control y la construcción de máquinas conocidos como robots, los cuales son capaces de realizar diversas tareas en los diferentes entornos facilitando la vida a los usuarios. En sus inicios la investigación en el campo de la robótica se había concentrado en robots simples o sistemas de solución puntuales que no tomaban en cuenta todas las variables y las ventajas que podía ofrecer la robótica a los procesos intervenidos. Al paso del tiempo y debido al desarrollo de la electrónica y sus diferentes componentes, las investigaciones se expanden en temas de sistemas de robots móviles múltiples. (Luis López Pérez, 2001)

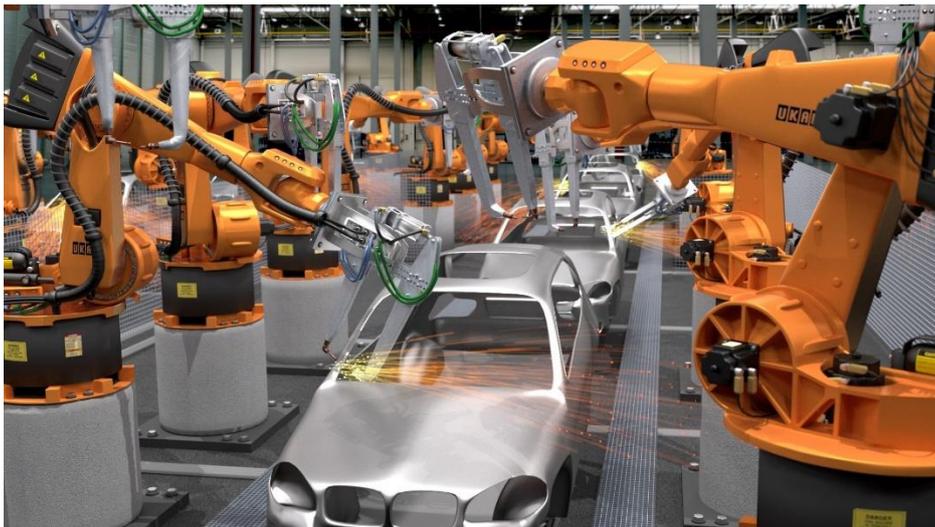


Figura. 2-1: Robótica Industrial.

Fuente: (Diario El Clarín, 2017)

Un robot es, por otra parte, una máquina que puede programarse para que interactúe con objetos, realice tareas que ayuden al desempeño y desarrollo de procesos en ambientes estructurados o que posean la característica de imitar, en cierta forma, el comportamiento humano o animal. Los robots han pasado de máquinas repetitivas a mecanismos físicos autónomos capaces de aprender de su entorno, la evolución por la que ha pasado la robótica permite que los robots posean cierto grado de inteligencia por lo que los tipos de controladores implementados muestran mayor complejidad según el tipo de locomoción y el entorno en donde

desarrollará su funcionalidad. En torno a este contexto y a las aplicaciones de los robots se propone la siguiente clasificación:

- a) **Industriales:** El siguiente paso a la revolución industrial se puede considerar la automatización de las máquinas y la implementación de robots en los diferentes procesos de producción. A pesar de que se los denomina como máquinas que realizan tareas repetitivas y autónomas en un entorno industrial, a lo largo del tiempo, encontramos que las investigaciones se han centrado en poder dotar inteligencia a este tipo de robots para corregir errores comunes y predecibles en los procesos industriales. En la actualidad en las megas industrias (Ford, Coca-Cola, etc.) poseen en su gran mayoría, plantas de procesos automatizadas gracias a la implementación de este tipo de robots que permiten aumentar la velocidad de las cadenas de producción con una gran eficiencia y resultados óptimos. (Graetz & Michaels, 2015)

- b) **Móviles:** Estos tipos de robots se denominan así ya que tienen la capacidad de desplazarse, ya sea por medio de ruedas, patas, hélices o tipo oruga. Su clasificación dependerá de la aplicación y el ambiente en el cual se vaya a desenvolver, para lo cual profundizaremos un poco más sobre este tema en los siguientes puntos a tratar.

2.3.2. Robots Móviles

En el campo de la robótica la movilidad ha sido una de las principales áreas de investigación en todo el mundo. Las diversas investigaciones realizadas en este campo de la robótica han permitido mejorar la inteligencia de los robots móviles, desarrollando nuevas aplicaciones para mejorar su desenvolvimiento de manera autónoma en diferentes áreas de trabajo. (Gabriela Andaluz, 2011). Al mencionar la autonomía del robot se entiende al dominio que tiene éste para determinar su curso de acción, mediante un proceso propio de razonamiento en base a sensores, en lugar de seguir una secuencia fija de instrucciones. y actuar sin la intervención del usuario para realizar las diferentes actividades o tareas propuestas por el operador. (Silva, García, & Bar, 2007)



Figura. 2-2: Tipos de Robots Móviles.

Fuente: (Instituto IDF, 2018)

Los robots móviles han sido de gran ayuda para realizar tareas en ambientes de difícil acceso para seres humanos o en procesos que poseen un alto riesgo de generar lesiones en los operarios. Al mencionar tareas que conlleven demasiado tiempo de operación y supervisión, la robótica móvil ha generado diversas aplicaciones que permiten realizar este tipo de tareas, optimizando de esta manera recursos y disminuyendo errores humanos debido al tiempo excesivo de trabajo que puede tomar supervisar y controlar un proceso en general. De esta manera pueden ser

Además, brindan la posibilidad de navegar en distintos terrenos y tienen aplicaciones como: exploración minera, exploración planetaria, misiones de búsqueda y rescate de personas, limpieza de desechos peligrosos, automatización de procesos, vigilancia, reconocimiento de terreno, y también son utilizados como plataformas móviles que incorporan un brazo manipulador.

- a) **Robots Terrestres:** Los robots móviles terrestres se desplazan mediante ruedas o patas, tienen aplicaciones en rastreo y traslado de objetos, evasión de obstáculos, traslado de instrumental quirúrgico en hospitales, limpieza del área del hogar, ambientes cooperativos y en la industria donde se emplean para análisis e inspección de fisuras en gaseoductos y contenedores de petróleo, por ejemplo se pueden clasificar por el tipo de locomoción utilizado, en general, los tres medios de movimiento son: por ruedas, por patas y orugas. Cabe señalar que aunque la locomoción por patas y orugas han sido ampliamente estudiadas, el mayor desarrollo se presenta en los Robots Móviles con Ruedas (RMR). Dentro de los atributos más relevantes de los RMR, destacan su eficiencia en cuanto a energía en superficies lisas y firmes, a la vez que no causan desgaste en la superficie donde se mueven y requieren un número menor de partes y

menos complejas, en comparación con los robots de patas y de orugas, lo que permite que su construcción sea más sencilla. (Silva, García, & Bar, 2007). A pesar del uso común y las facilidades que presenta este tipo de robots terrestres, están limitados al terreno del ambiente de trabajo, por lo que condiciona de gran manera su movilidad en espacios abiertos y de difícil acceso (quebradas, espacios concurridos).

- b) **Robots Acuáticos:** son robots que se desenvuelven en el medio acuático marino, la principal dificultad de este tipo de robots, especialmente los robots submarinos que al moverse en un medio subacuático son muy distintos los sensores que utilizan al respecto de los utilizados en la robótica terrestre o aérea. (Wiki , 2005)

Constantemente se ha citado que los robots móviles son los más apropiados para aplicaciones en donde un ser humano no podría realizar tareas, el medio acuático es el más propicio ejemplo de estos ambientes, ya que es peligroso, y muchas tareas requieren una operación a largo plazo y profundidad significativa. La robótica móvil está particularmente bien adecuada para aplicaciones subacuáticas como arrecifes o tuberías inspección, vigilancia de poblaciones de peces, observación de la vida marina y evaluación de desastres ambientales. (Georgiades, German, & Hogue, 2005). En este tipo de robots hay que considerar el término biomimético el cual consiste en recrear los sistemas naturales a los sistemas de ingeniería, con el objetivo de un mejor desenvolvimiento por parte del robot en los ambientes en el cual se encuentra el sistema natural, por ejemplo: robots tipo pez para ambientes acuáticos). (Fish, 2006)

- c) **Robots Aéreos:** son aeronaves no tripuladas como helicópteros o pequeños aviones operados a control remoto, pueden proporcionar imágenes aéreas para reconocimiento de terreno y superficie, y son muy útiles en problemas de análisis de tráfico e inspección de edificios. (Marcombo, 2014). En este tipo de robots profundizaremos más en el siguiente apartado.

2.3.3. Robots Aéreos

Estos robots capaces de dominar el espacio aéreo, controlados de forma remota por el usuario, rutas precargadas de vuelo o controladores autónomos que generen planes de vuelo, adaptándose a la aplicación requerida por el usuario, han resultado de gran ayuda en aplicaciones militares, comerciales y de investigación. Los podemos encontrar en diferentes formas de acuerdo a la aplicación en la que se van a desenvolver como robots biomiméticos, drones, cuadricópteros, entre otros. A pesar de que no hay un término que los englobe a todos, en la comunidad científica son más conocidos como UAV (Unmanned Aerial Vehicle), vehículos aéreos no tripulados.

Mientras que robots y los helicópteros tripulados poseen importantes desventajas, los UAV pueden ser enviados a todo tipo de operaciones. Si nos referimos a realizar tareas a baja escala y presupuesto, y en tareas con acciones de vuelo arriesgada, el instinto de supervivencia del piloto limitara las acciones de control, por ende limitando las investigaciones a ser realizadas, en vehículos aéreos tripulados. Los vehículos aéreos no tripulados tienen el potencial de servir en una amplia gama de aplicaciones del espacio aéreo civil. En este caso hay que tomar en cuenta que la tarea del operador de UAV, es diferente y en algunos aspectos, más difícil que la tarea de pilotar un avión tripulado, pero se asegura que la probabilidad de riesgos sea baja en comparación a vehículos aéreos tripulados. (Jason S. McCarley, 2004)



Figura. 2-3: Robots Aéreos Flame Wheel e Inspire 2.
Fuente: (Empresa DJI, 2004)

Aunque el sueño de volar incluye por asociación al ser humano, el vuelo no tripulado simplemente ha sido un paso más en la evolución de esta técnica. Su principal desarrollo fue impulsado para aplicaciones militares, donde los UAV destacan en campos como el reconocimiento aéreo, vigilancia aérea, adquisición de objetivos, evaluación de daños, búsqueda y rescate en zonas de combate, etc. Sin embargo, paralelamente a este desarrollo se han ideado aplicaciones menos belicistas, especialmente en el campo del reconocimiento aéreo e incluso en la exploración espacial. Los UAV también han generado un gran desarrollo en el campo de reconocimiento en terrenos de difícil acceso, que por lo general eran tareas de grandes satélites o aviones tripulados, los UAV han permitido realizar estas tareas abaratando costos y ofreciendo la posibilidad de que este tipo de investigaciones no se concentre en grandes entidades y grupos de investigación. Los UAV también destacan en otras aplicaciones como la entrega de suministros o materiales en zonas sensibles, búsqueda y rescate, turismo, ocio, etc. (Zaloga, 1997)

Aunque la clasificación de los UAV pueda ser muy extensa para métodos de estudio lo clasificaremos de acuerdo al tipo de control que posean, de esta manera tenemos UAV controlados remotamente desde una estación de control en mar o tierra y los UAV con autonomía, que son programados previamente para responder al entorno en el que van a ejecutar las tareas e incluso interactuar en el ambiente a desenvolverse.

- a) **UAV con control remoto:** Al ser un robot controlado desde una base debe poseer características nativas de largo alcance para el control, dependiendo la distancia de la cámara para teleoperar sus trayectos, y opciones adicionales como aterrizaje de emergencia por falta de batería o por pérdida de señal de control. Dependiendo de la aplicación tendrá diferentes objetivos y tareas a desarrollar, una de las más habituales es el estudio de la aerodinámica del UAV en conjunto con las habilidades presentadas por el operador. Las desventajas tomando el caso anterior como ejemplo es que existe un factor variable que determinará los estudios, como lo es la habilidad del operario para tomar acciones de control en el UAV, a esto se le puede sumar tiempo de respuesta del operario a eventos fortuitos (caída de objetos, ráfaga de viento) que pueden afectar al control del UAV, dependiendo de esto se obtendrá los resultados de los casos de estudio, tomando en cuenta todos estos parámetros variables. (Yiming, Jingchao, & Wenping, 2006)

- b) **UAV con autonomía programada:** A diferencia del anterior tipo, este UAV posee un algoritmo de control que le ayuda a pilotar en las tareas a ser ejecutadas sin necesidad

de un operario. Dependiendo del algoritmo de control el UAV podrá responder a diferentes eventos que presente el ambiente en donde ejecutará las tareas pre programadas por el usuario final. Su campo de aplicación es amplio, desde vigilancia y supervisión, en espacios controlados, seguimiento de trayectorias, e incluso como robots de servicio para facilitar tareas a la comunidad (Transporte de equipos, control cooperativo, etc). Cabe recordar que en el país todavía no existen restricciones para este tipo de vuelos preprogramados, pero que alrededor del mundo se está regularizando debido a la invasión de propiedad, privacidad, entre otros.

2.3.4. Robots Manipuladores

Al hablar de robots manipuladores encontraremos diversos conceptos de este tipo de robot, pero tomaremos en cuenta la definición adoptada por la AFNOR (Asociación Francesa de Normalización) que separa los términos y nos define a un manipulador como un: “mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.” (AFNOR Normalisation, 2018)



Figura. 2-4: Robots manipulador KR 210-2 F Exclusive.
Fuente: (Empresa KUKA, 2018)

En un principio han sido proyectados con el objeto de eliminar los esfuerzos provocados por el levantamiento de la mercancía y están destinados a campos varios y diversificados, a su vez se distinguen por la sencillez de uso y por su estructura resistente y sólida, características que

otorgan a dichos productos un alto nivel de fiabilidad. Debido a su estructura se los puede encontrar en la mayoría de industrias de producción masiva.

A los manipuladores también los podemos encontrar en proyectos de investigación enfocados en suplir deficiencias en personas con discapacidad. Un ejemplo claro es la investigación realizada por (Balaguer, Gimenez , & Jardon, 2005), en el cual al robot denominado MATS lo programaban para ejecutar tareas de asistencia al usuario, en diferentes ambientes de trabajo estructurado, colocándolo incluso en una silla robótica para tareas de alimentación a la persona discapacitada.



Figura. 2-5: Robot manipulador Mats
Fuente: (Carlos Balaguer, Universidad Carlos III de Madrid, 2005)

El desarrollo de la electrónica ha permitido que este tipo de investigaciones sigan evolucionando y generando aplicaciones más eficientes capaces de cumplir uno de los objetivos fundamentales de la robótica, que es facilitar las tareas realizadas por las personas. Para entender mejor las funciones de los manipuladores, trataremos de describir la estructura general de los mismos, tomando en cuenta sus formas más comunes.

2.3.5. Estructura de los Manipuladores

Si nos referimos de forma general un robot manipulador consta de una secuencia de elementos estructurales, denominados enlaces o eslabones, conectados entre sí mediante juntas o articulaciones, que permiten el movimiento relativo de cada uno de sus eslabones. Hay que tomar en cuenta que el material de los eslabones de un manipulador debe ser rígidos, ya que al ser una serie de enlaces que dependen del movimiento del enlace anterior puede provocar vibraciones en su extremo operativo.

Lo fundamental de la estructura de un manipulador es considerarlo como una cadena cinemática abierta. Se dice que una cadena cinemática es abierta si cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente eslabón, exceptuando el primero, que se suele fijar a un soporte, y el último, cuyo extremo final queda libre. A éste se puede conectar un elemento terminal o actuador final: una herramienta especial que permite al robot de uso general realizar una aplicación particular, que debe diseñarse específicamente para dicha aplicación: una herramienta de sujeción, de soldadura, de pintura, etc. El punto más significativo del elemento terminal se denomina extremo operativo (EO), para cumplir con esta definición el robot debe poseer articulaciones en las juntas de los eslabones que le permiten moverse respecto a un plano. (Lorenzo Sciavicco, 2012) En la Figura. 2-6, se presenta las partes de un manipulador con 3 grados de libertad.

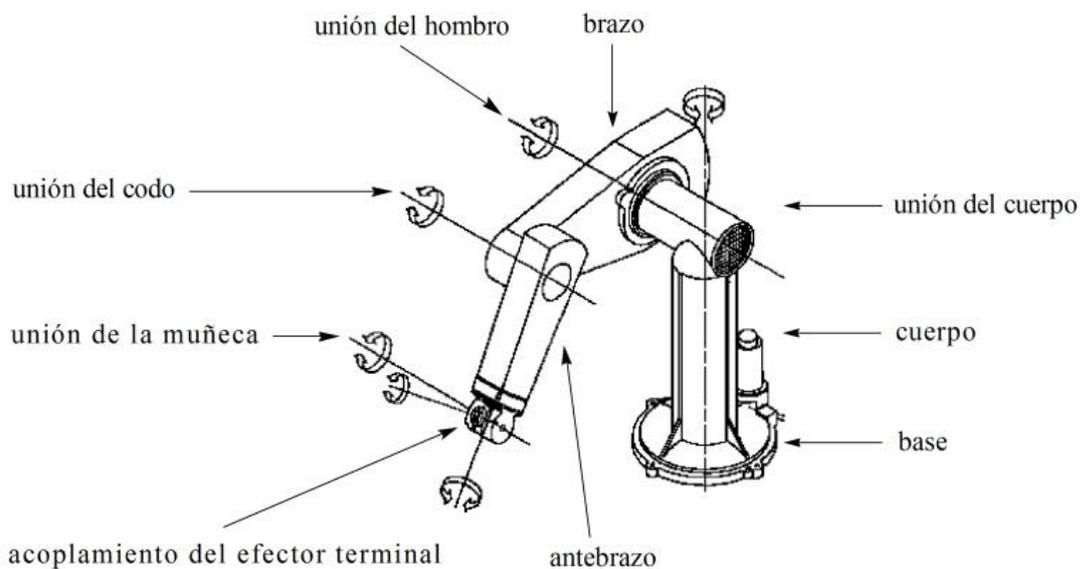


Figura. 2-6: Partes de un manipulador

Fuente: (Antonio Romero, Morfología de los robots industriales, 2012)

Además, que la cantidad de movimientos que pueda efectuar dependerá de acuerdo a los grados de libertad que posea, este término lo denominamos al movimiento que puede realizar un eslabón respecto al eje de un plano cartesiano, siempre hay que tomar en cuenta que los grados de libertad que posee un manipulador no dependen de la cantidad de actuadores que posea el robot ni de los eslabones del mismo. A pesar de que la mayoría de los casos los grados de libertad del manipulador coinciden con el número de eslabones y la cantidad de actuadores que posee. (Sciavicco & Siciliano, 2012)

De acuerdo con las articulaciones que posee un robot manipulador la podemos clasificar por su tipo de movimiento que realice respecto al plano cartesiano, por lo que tenemos:

- Articulación Lineal: También denominada deslizante, traslacional o prismática, lo consideramos lineal, si un eslabón se desliza sobre un eje respecto al eslabón anterior, ver Figura. 2-7.
- Articulación Rotacional: Para este caso nos centramos en el giro que realiza un eslabón en torno a un eje respecto al eslabón anterior, se debe tomar en cuenta que para los dos casos se tiene como referencia el eslabón anterior ver Figura. 2-7.

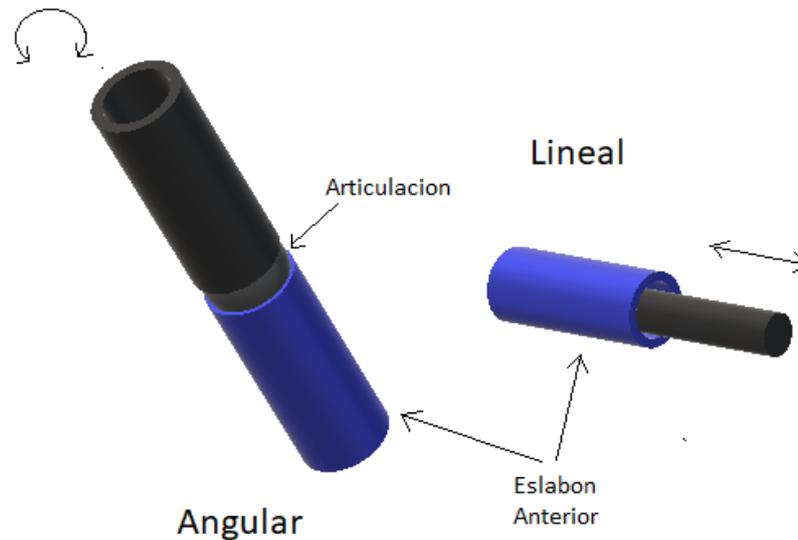


Figura. 2-7: Tipos de Articulación.
Fuente: Erazo A. 2019

Al conjunto de eslabones articulaciones lo denominamos una cadena cinemática, si posee un extremo operativo es una cadena cinemática abierta, o en su caso contrario si el eslabón final se conecta al eslabón inicial, generando una retroalimentación lo denominamos cadena cinemática cerrada. (Sciavicco & Siciliano, 2012)

Si nos referimos a una cadena cinemática abierta tendremos un extremo operativo final, el cual ejecutara las tareas programadas, ya sea tareas de soldadura, ajuste, traslado de objetos, entre otros.

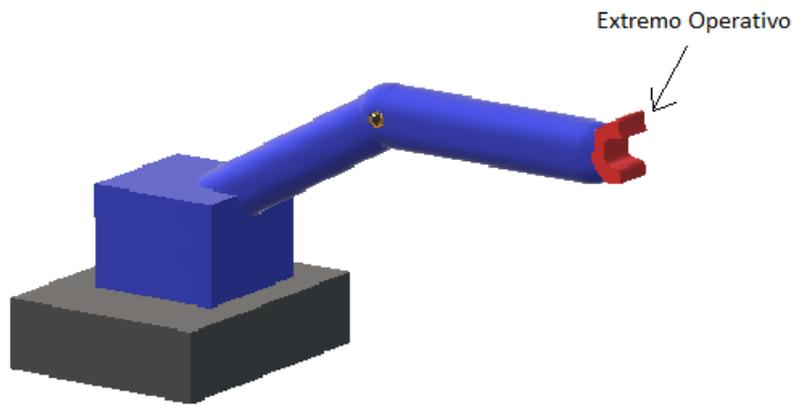


Figura. 2-8: Extremo Operativo tipo pinza.
Fuente: Erazo A. 2019

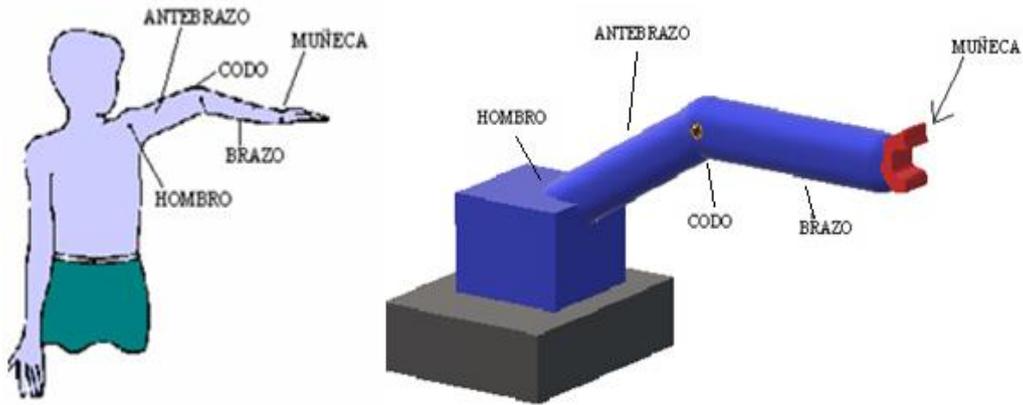
El extremo operativo (EO) de un manipulador lo pueden dividir en dos categorías:

- EO tipo pinzas (*gripper*)
- EO tipo herramienta

Las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarre de la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquillos de sujeción, imanes, ganchos, y cucharas.

Una herramienta se utiliza como actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.

A los manipuladores robóticos se les suele denominar también brazos robóticos por la analogía que se puede establecer, en muchos casos, con las extremidades superiores del cuerpo humano, como se puede ver en la Figura. 2-9.



(a) Brazo Humano

(b) Brazo robótico

Figura. 2-9: Semejanza de un brazo robótico y un brazo humano.
Fuente: Erazo A. 2019

Para profundizar más sobre el tema de los Grados de libertad (GDL) lo denominaremos a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos). Normalmente, en cadenas cinemáticas abiertas, el conjunto eslabón-articulación tiene un solo grado de libertad, ya sea de rotación o de traslación. Pero una articulación podría tener dos o más GDL que operan sobre ejes que se cortan entre sí, como lo mencionamos al principio de este capítulo.

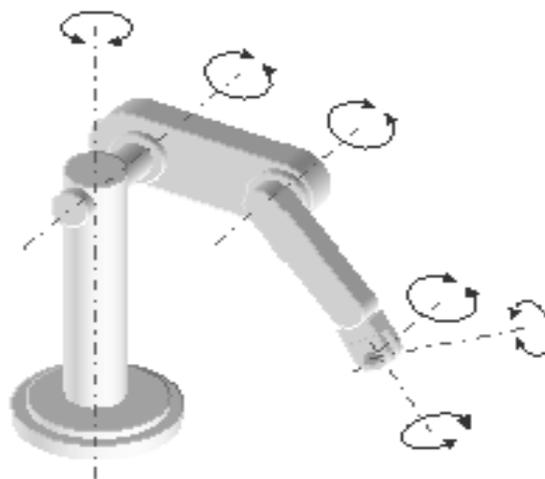


Figura. 2-10: Grados de libertad de un brazo robótico.
Fuente: Erazo A. 2019

Para describir y controlar el estado de un brazo de robot es preciso determinar:

- La posición del punto terminal (o de cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, denominado el sistema mundo.
- El movimiento del brazo cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal. (Platea, 2007)

2.3.6. Realidad Virtual

La realidad virtual se podría definir como una interfaz de usuario final que involucra, la simulación en tiempo real y la interacción de múltiples canales sensoriales, como visión, tacto, olfato, sonidos, y sentido del gusto. La simulación que hace la realidad virtual se puede referir a escenas virtuales, creando un mundo virtual que sólo existe en el ordenador de lugares u objetos que existen en la realidad. También permite capturar la voluntad implícita del usuario en sus movimientos naturales proyectándolos en el mundo virtual que estamos generando, proyectando en el mundo virtual movimientos reales. (Burdea Grigore, (1994))

Además, también nos permite hundirnos completamente en un mundo virtual, desconectando los sentidos completamente de la realidad teniendo la sensación la persona que está dentro de que la realidad corresponde en el mundo virtual.

Su desarrollo ha sido progresivo en los últimos años debido a las diferentes aplicaciones a las que se puede orientar este tipo de tecnologías. Así encontramos aplicaciones en campos:

Turístico: Se pueden recrear ambientes y transmitirlos en sistemas simulados, ofreciendo experiencias inmersivas a potenciales turistas, en diferentes partes del mundo.



(a) Realidad virtual parque



(b) Realidad virtual cascada

Figura. 2-11: Aplicación Realidad Virtual en Turismo.

Fuente: (Virtual Reality on e-Turism, Washington Quevedo, 1998)

Medicina: Existe aplicaciones que simulan cirugías de alto riesgo, con variaciones de complejidad y recrean los diferentes eventos que puede ocurrir mediante las operaciones quirúrgicas.



Figura. 2-12: Aplicación Realidad Virtual en Medicina.

Fuente: (MindMaze, 1998)

Electrónica: En este campo en especial encontramos mayor diversidad de aplicaciones, desde entornos virtuales donde simulan laboratorios, operaciones en procesos industriales, y simulan los diferentes comportamientos de equipos específicos (robots móviles, manipuladores, maquinas, etc) para su posterior implementación en ambiente reales, reduciendo de esta manera daños significativos en los equipos y asegurando el buen funcionamiento de las máquinas o robots testados en el ambiente simulado virtualmente.

Industria: En este campo se utiliza la R.V. para conceptos como entrenamiento de personal nuevo y pruebas piloto de uso adecuado de la maquinaria relacionado en el proceso industrial, capacitaciones en el área de seguridad industrial, control y producción, además permite la simulación de proceso industrial, guías de uso para maquinaria industrial.



Figura. 2-13: Aplicación industrial de R.V.
Fuente: (ResearchGate, 1998)

Los sistemas de realidad virtual necesariamente deben constar de los equipos adecuados para su desarrollo, de esto tenemos que las gafas de realidad virtual con sus debidos controles, y el software en donde se desarrollara el ambiente de simulación, también hay que tomar en cuenta el espacio físico donde se situara el usuario final que no necesariamente debe ser de gran amplitud, sino de un rango considerable para su desenvolvimiento sin chocar con obstáculos que limiten su movimiento.

2.3.7. Gafas de Realidad Virtual

Es el visor mediante el cual tendremos la experiencia inmersiva en los ambientes virtuales, dependiendo el modelo podremos, encontrarlos con audífonos incluidos, o cámaras que permitan realizar no solo ambientes virtuales sino combinarlos con ambientes reales, gracias a la configuración de la cámara incluida



Figura. 2-14: Gafas Realidad Virtual Samsung.
Fuente: (Samsung, 2013)

Este tipo de dispositivos, son conectados generalmente a un computador mediante cables, por donde se transmitirá la información del ambiente virtual. Las gafas o casco como también se los llama posee sus controles dinámicos los cuales permiten una iteración más completa. Usualmente los controles se usan para movimiento de objetos y la simulación de herramientas.

2.3.8 *Software Unity*

Inicialmente fue desarrollado como el motor de videojuegos para multiplataformas creado por Unity Technologies. Posterior a su popularización, se lo considera como una herramienta para virtualización de ambientes, gracias a sus diferentes herramientas que permiten exportar e importar los datos de los ambientes a diferentes tipos de software.

Hay que tomar en cuenta que los computadores donde va a desarrollar este tipo de aplicaciones necesitan un motor o tarjeta gráfica de gran capacidad, ya que al ser un software que trabajan en ambientes 3D tiene más requerimientos que los programas normales.

2.3.9 *Objetos 3D*

Los objetos 3D son objetos "secundarios" se podría decirse, que se agregan a diferentes escenarios. Los objetos o modelos 3D son clasificados por polígonos al igual que todo elemento que tenga una composición tridimensional.

Low-Poly: Traducido como "Polígonos bajos" son modelos que su composición de polígonos es baja lo cual es probable que el modelo tenga un muy mal detalle y no se obtenga un resultado favorable. Estos modelos se usan para optimizar el rendimiento del videojuego y el "Low-Poly" es efectivo en modelos que no requieren mucho detalle (Ej, una caja o un jabón)

Mid-Poly: Traducido como "Polígonos medios" son modelos que su composición de polígonos es media y logran dar mejor detalle que los "Low-Poly" aunque su velocidad de procesamiento es más tardada. Estos modelos son más usados para modelos que requieren un poco más de detalle (Ej, Un arma, una casa)

High-Poly: Traducido como "Polígonos altos" son modelos que su composición de polígonos es alta y llegan a dar un detalle magnífico pero su procesamiento es más complejo y tiende a ralentizar el ordenador, dependiendo de la potencia que el Hardware de la computadora o consola cuente. Estos modelos son usados para modelos que precisan de un buen grado de detalle (Ej Un personaje, un vehículo).

Para la creación de terrenos se tienen diferentes herramientas las cuales permiten dar textura sombras e iluminación a este tipo de ambientes como la presencia de formas rugosas para simular un efecto en el campo, en la Figura. 2-13 se observa un tipo de terreno simulado en Unity.



Figura. 2-15: Terreno con herramienta creado Brushes.
Fuente: (Unity, 2018)

2.3.10 Tipos de Inmersión

Realidad Virtual Inmersiva: se basa en la simulación de un ambiente tridimensional en el que el usuario percibe a través de estímulos sensoriales y se siente dentro del mundo virtual que está explorando. Lo hace a través de dispositivos como guantes, gafas, cascos o trajes especiales, y todos ellos capturan la posición y rotación de diferentes partes del cuerpo. Este tipo de sistemas son ideales para aplicaciones de entrenamiento o capacitación.



Figura. 2-16: Realidad Virtual Inmersa.
Fuente: (Steemit, 2013)

Realidad Virtual No inmersiva: es aquella en la que la visualización de los elementos virtuales se hace a través de una pantalla. La interacción se hace a través de accesorios como el teclado, el ratón o el micrófono, que se convierten en el medio de entretenimiento. Se aprovecha de los medios que ofrece Internet para poder interactuar en tiempo real con diferentes personas o ambientes que, realmente, no existen. Este tipo es el más idóneo para visualizaciones científicas, aunque lo más extendido es como medio de entretenimiento.

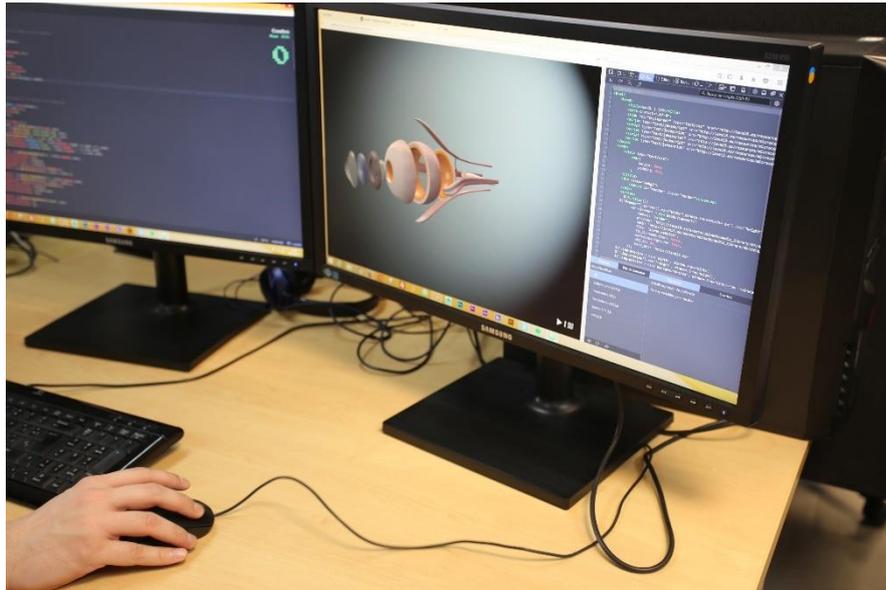


Figura. 2-17: Realidad Virtual No Inmersa.
Fuente: (inmediastudio, 2013)

Realidad Virtual Semi-Inmersiva: se caracteriza por estar formados por cuatro pantallas en forma de cubo, tres para las paredes y otra para el suelo, que rodean al usuario. Este necesita de unas gafas y un dispositivo de seguimiento de movimientos en la cabeza. Son usados principalmente para aquellas visualizaciones donde se requiere que el usuario se mantenga en contacto con elementos del mundo real.



Figura. 2-18: Realidad Virtual Semi-Inmersa.
Fuente: (ojulearning, 2013)

CAPÍTULO III

3 MODELACIÓN Y DISEÑO DEL CONTROLADOR

3.1 Modelación Cinemática

En la modelación cinemática nos podemos encontrar con modelación directa e inversa, al referirnos a directo generamos velocidades y a través del modelo obtenemos la posición del robot, en cambio en modelación inversa es todo lo contrario, le proveemos al modelo de posición (x,y,z,ψ) y el modelo nos generara velocidades (lineal, angular y rotación) con el objetivo de seguir un camino trayectoria o posicionarse en algún punto en el plano. (Ortiz & Erazo , 2018)

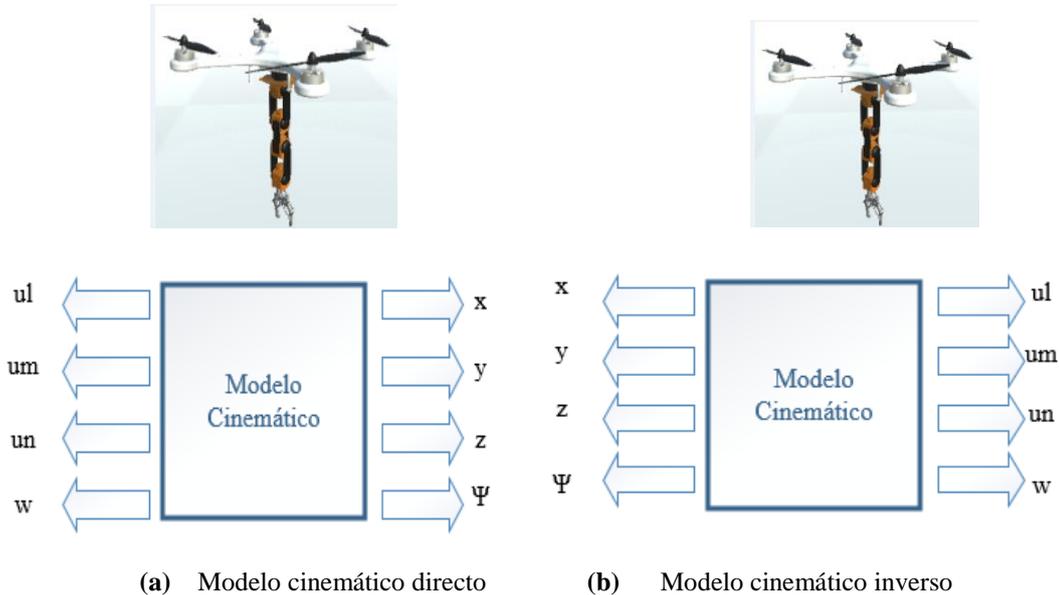


Figura. 3-1: Modelo Cinemático a) Directo, b) Inverso.

Fuente: Erazo A. 2019

Acorde al robot móvil que se analice, logra un desplazamiento en los distintos ejes del plano cartesiano, en este caso nos enfocaremos en los UAV y en los robots manipuladores, para un

mejor entendimiento se analiza de forma separada, y así posteriormente unir los resultados y obtener el modelo cinemático del manipulador aéreo.

A pesar de las ventajas que ofrecen los UAV respecto a los robots móviles terrestres hay que tomar en cuenta que el modelamiento de este tipo de robots resulta ser más extenso, en gran parte a la velocidad de elevación y descenso que ha sido añadida.

3.1.1. Modelo cinemático del brazo robótico

El robot manipulador se posiciona en el centro del robot UAV con su extremo operativo con dirección al piso. El manipulador consta de tres grados de libertad y especificando sus movimientos respecto a los ejes en la Tabla 3-2.

Tabla 3-1: Nomenclatura Partes Robot Manipulador

Nomenclatura	Descripción
l_1	Longitud eslabón 1
l_2	Longitud eslabón 2
l_3	Longitud eslabón 3
q_1	Angulo de giro
q_2	Angulo de giro
q_3	Angulo de giro

Fuente: Erazo A. 2019

La configuración del manipulador robótico a ser modelado se considera el extremo operativo del brazo o un punto de interés $\mathbf{h} = (h_x, h_y, h_z)$, donde $\{R_a\}$ representa el sistema de referencia de coordenadas locales para el brazo robótico, a ser modelado matemáticamente. Así se puede representar un manipulador basado en posición de sus eslabones como se muestra en la Figura. 3-2.

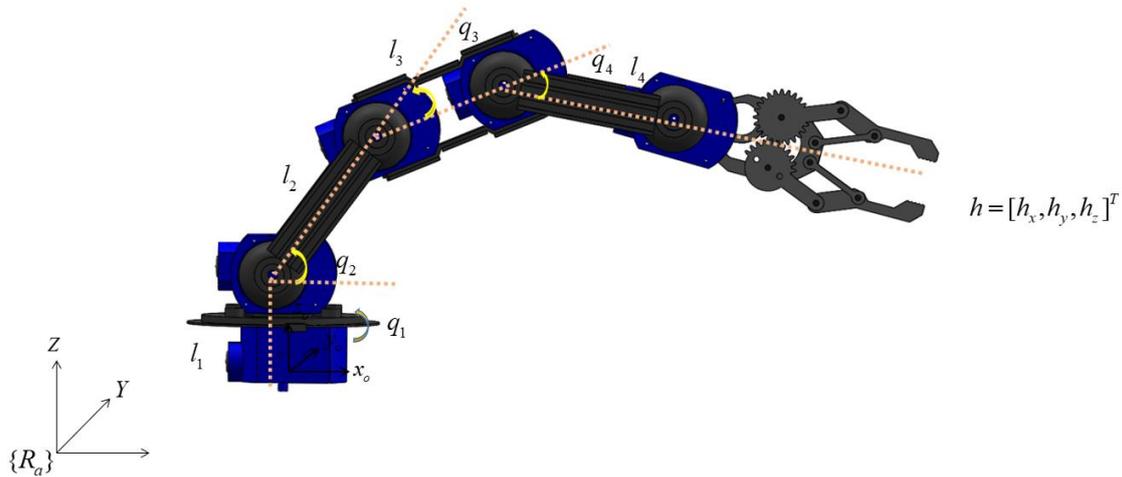


Figura. 3-2: Manipulador de 4 GDL
Fuente: Erazo A. 2019

Para el estudio que se realiza en este proyecto se utiliza un brazo robótico de 3 grados de libertad.

La interpretación geométrica de cada eslabón permite determinar las distancias de los tres ejes según el ángulo de cada eslabón expresado en:

$$\begin{cases} h_x = l_2 \cos(q_2) \cos(q_1) + l_3 \cos(q_2 + q_3) \cos(q_1) \\ h_y = l_2 \cos(q_2) \sin(q_1) + l_3 \cos(q_2 + q_3) \sin(q_1) \\ h_z = l_1 + l_2 \sin(q_2) + l_3 \sin(q_2 + q_3) \end{cases} \quad (1)$$

Donde:

- l_1, l_2 , y l_3 son dimensiones del brazo robótico
- q_1, q_2 y q_3 son los ángulos de giro para cada grado de libertad del robot manipulador.

Una vez encontradas las ecuaciones de posición del extremo operativo se procede a derivar con respecto al tiempo a fin de obtener las velocidades lineales para cada dimensión de manipulador, según muestra.

$$\begin{cases} h'_x = -l_2[\sin(q_2)q'_2 \cos(q_1) + \cos(q_2) \sin(q_1) q'_1] - \dots \\ \quad \dots - l_3[\sin(q_2 + q_3)(q'_2 + q'_3) \cos(q_1) + \sin(q_1) \cos(q_2 + q_3) q'_1] \\ h'_y = -l_2[\sin(q_2)q'_2 \sin(q_1) - \cos(q_1)q'_1 \cos(q_2)] - \dots \\ \quad \dots - l_3[\sin(q_2 + q_3)(q'_2 + q'_3) \sin(q_1) - \cos(q_1)q'_1 \cos(q_2 + q_3)] \\ h'_z = -l_2 \cos(q_2)q'_2 + l_3 \cos(q_2 + q_3)(q'_2 + q'_3) \end{cases}$$

(2)

Usando las ecuaciones anteriores se expresa el modelo cinemático de forma compacta mediante un lenguaje matricial.

$$h'_{(t)} = J(q)q' \quad (3)$$

Donde, $J(q)$ es la matriz Jacobina, la cual define un mapeo lineal entre el vector de velocidad del manipulador q' y el vector de velocidades del extremo operativo $h'_{(t)}$. Las velocidades lineales dependen directamente de las velocidades angulares en cada uno de los eslabones del brazo manipulador. Además, utilizando la pseudo-inversa de la matriz Jacobiana se puede producir el proceso inverso como se indica.

$$q = J^\# h' \quad (4)$$

Siendo la matriz Jacobiana una transformación lineal que realiza el mapeo de las variables de estado a las variables de salida. A continuación, se presenta los términos de la matriz Jacobiana.

$$\begin{aligned} J_{11} &= -\sin(q_1)[l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] \\ J_{12} &= \cos(q_1)[l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] \\ J_{13} &= 0 \\ J_{21} &= -\cos(q_1)[l_1 \sin(q_2) + l_2 \sin(q_2 + q_3)] \\ J_{22} &= -\sin(q_1)[l_1 \sin(q_2) + l_2 \sin(q_2 + q_3)] \\ J_{23} &= l_1 \cos(q_2) + l_2 \cos(q_2 + q_3) \\ J_{31} &= -l_2 \cos(q_1) \sin(q_2 + q_3) \\ J_{32} &= -l_2 \sin(q_1) \sin(q_2 + q_3) \\ J_{33} &= l_2 \cos(q_2 + q_3) \end{aligned}$$

(5)

3.1.2. MODELO CINEMÁTICO DEL VEHÍCULO AÉREO NO TRIPULADO, UAV

Se realiza la modelación de un vehículo aéreo no tripulado, en caso de la presente investigación se hace uso de un cuadricóptero el cual está compuesto por un conjunto de cuatro velocidades representadas en el sistema espacial $\langle Q \rangle$ como se ilustra en Figura. 3-3. El desplazamiento del cuadricóptero consta de tres velocidades lineales u_{ql} , u_{qm} y u_{qn} y una velocidad angular ω_q definidas en un marco espacial $\langle Q \rangle$.

Cada velocidad lineal se orienta en uno de los ejes de la estructura del robot aéreo con respecto al sistema espacial $\langle Q \rangle$ unido al centro de gravedad del cuadricóptero.

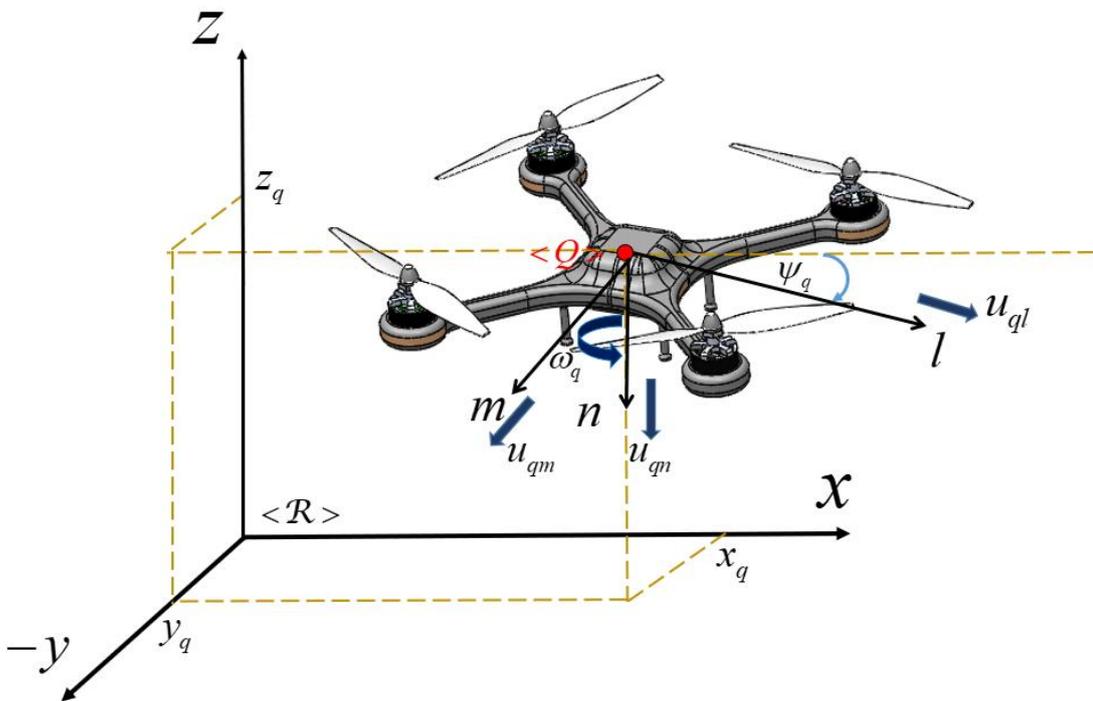


Figura. 3-3: Vehículo aéreo no tripulado

Fuente: Erazo A. 2019

Donde, \mathbf{u}_{ql} es la velocidad lineal que apunta hacia la velocidad lineal; \mathbf{u}_{qm} es la velocidad lineal que apunta al lateral izquierdo; \mathbf{u}_{qn} es la velocidad lineal que apunta hacia arriba; mientras que ω_q es la velocidad angular que gira en sentido anti-horario, alrededor del eje Z_q . En otras palabras, el movimiento del cuadricóptero se realiza con respecto al sistema de referencia $\{\mathbf{R}_q\}$ el cual se define como:

$$\begin{bmatrix} x'_q \\ y'_q \\ z'_q \\ \psi'_q \end{bmatrix} = \begin{bmatrix} \cos\psi_q & -\sin\psi_q & 0 & 0 \\ \sin\psi_q & \cos\psi_q & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{ql} \\ \mathbf{u}_{qm} \\ \mathbf{u}_{qn} \\ \omega_q \end{bmatrix}$$

La expresión de forma compacta del modelo cinemático del robot es definida de forma matricial,

$$\mathbf{h}'_q = J(\psi_q)\mathbf{u}_q$$

(6)

Donde, $\mathbf{h}'_q \in \mathfrak{R}^n$ con $n = 4$ representa el vector del eje de las velocidades del sistema $\{\mathbf{R}_q\}$; $J(\psi_q) \in \mathfrak{R}^{n \times n}$ es una matriz singular y el control de maniobrabilidad del cuadricóptero es definida $\mathbf{u}_q \in \mathfrak{R}^{n \times n}$.

3.1.3 MODELO CINEMÁTICO DEL MANIPULADOR AÉREO

El tipo de UAV al que se aplicará el modelo cinemático es un cuadricóptero con sus cuatro velocidades de operación: delantera, lateral, arriba/abajo, y su respectiva orientación. Se considera el punto de interés en el centro del UAV, para facilidad de modelamiento ya que posteriormente se añadirá el modelo del robot manipulador y este se situará en el centro del UAV.

Para efectos de representación de fórmulas y cálculos a encontrar en el modelamiento se definen los términos y su respectiva nomenclatura de las distintas variables, a ser consideradas en este proyecto, como observamos en la Tabla 3-1 presenta estos términos basándose en la Figura. 2-3 la cual representa al manipulador aéreo en un espacio 3D.

El modelo cinemático de un manipulador aéreo proporciona la ubicación del extremo operativo \mathbf{h} en función de la configuración del brazo robótico y de la ubicación de la plataforma móvil aérea.

$$\begin{aligned} f: \mathcal{N}_a \times \mathcal{M}_h &\rightarrow \mathcal{M} \\ (\mathbf{q}_h, \mathbf{q}_a) &\rightarrow \mathbf{h} = f(\mathbf{q}_h, \mathbf{q}_a) \end{aligned} \quad (7)$$

Donde, \mathcal{N}_a es el espacio vectorial para la configuración del brazo robótico, \mathcal{M}_h es el espacio vectorial de operación de la plataforma móvil aérea.

El modelo cinemático instantáneo de un manipulador aéreo proporciona la derivada de la ubicación del extremo operativo en función de los derivados tanto de la configuración del brazo robótico como de la ubicación de la plataforma móvil aérea.

$$\mathbf{h}' = \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}_h, \mathbf{q}_a) \mathbf{v} \quad (8)$$

Siendo, $\mathbf{h}' = [\mathbf{h}'_1 \ \mathbf{h}'_2 \ \dots \ \mathbf{h}'_m]^T$ el vector velocidad del efector final, $\mathbf{v} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_{\delta_n}]^T = [\mathbf{v}_h^T \ \mathbf{v}_a^T]^T$ es el vector de movimiento de la plataforma móvil aérea. La dimensión es $\delta_n = \delta_{nh} + \delta_{na}$ donde δ_{nh} y δ_{na} son las dimensiones del vector de control de movimiento de la plataforma móvil aérea y el brazo robótico.

Reemplazar $\mathbf{J}(\mathbf{q}) = \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}_h, \mathbf{q}_a)$ en (8), se obtiene.

$$\mathbf{h}'(\mathbf{t}) = \mathbf{J}(\mathbf{q}) \mathbf{v}(\mathbf{t}) \quad (9)$$

Donde, $\mathbf{J}(\mathbf{q})$ es la matriz Jacobiana que define la representación lineal entre el vector velocidad del manipulador aéreo móvil $\mathbf{v}(\mathbf{t})$ y el vector velocidad del extremo operativo $\mathbf{h}'(\mathbf{t})$. La matriz Jacobiana está en función de $\mathbf{q}(\mathbf{t})$.

Cuadróptero y Brazo Robótico

El modelo cinemático del manipulador móvil aéreo está compuesto por un conjunto de siete velocidades representadas en el marco de referencia espacial $\langle H \rangle$. El desplazamiento del manipulador aéreo está guiado por las tres velocidades lineales \mathbf{u}_l , \mathbf{u}_m y \mathbf{u}_n , y se define en un marco de referencia espacial giratorio con la mano derecha $\langle H \rangle$ y la velocidad angular $\boldsymbol{\omega}$, como se muestra en la Figura. 3-4.

Tabla 3-2: Términos Modelamiento Cinemático.

Nomenclatura	Termino
u_l	Velocidad Lateral
u_m	Velocidad Frontal
u_n	Velocidad elevación y descenso
ψ	Rotación
l	Vector velocidad lateral
m	Vector velocidad frontal
n	Vector velocidad de elevación y descenso
x	Eje x
y	Eje y
z	Eje z
h_x	Proyección punto de interés en eje x
h_y	Proyección punto de interés en eje y
h_z	Proyección punto de interés en eje z
H	Punto de interés
w	Velocidad de rotación
l_1, l_2, l_3	Longitud eslabones

q_1, q_2, q_3	Angulo de operación de cada eslabón
-----------------	-------------------------------------

Fuente: Erazo A. 2019

La Figura. 3-4. representa los términos utilizados en la Tabla 3-1 en el robot manipulador aéreo.

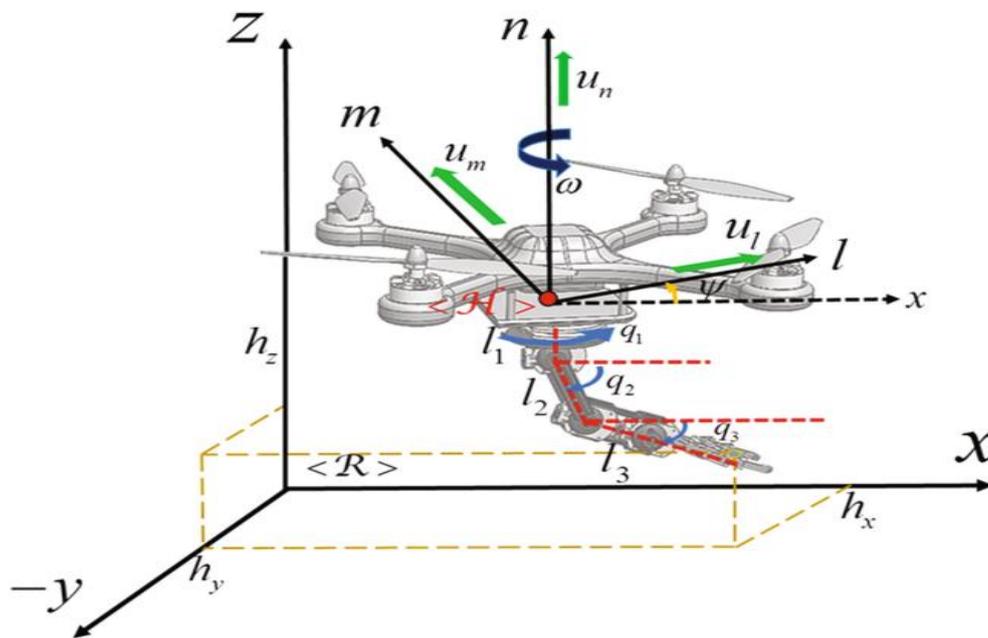


Figura. 3-4: Manipulador Aéreo

Fuente: Erazo A. 2019

Cada velocidad lineal se orienta como uno de los ejes de la estructura $\langle H \rangle$ acoplado al centro de gravedad del cuadricóptero: u_l apunta a la dirección frontal; u_m apunta a la dirección lateral izquierda y u_n apunta hacia arriba. La velocidad angular ω gira el sistema referencial $\langle H \rangle$ en sentido contrario a las agujas del reloj, alrededor del eje H_z (considerando la vista superior). Mientras que la maniobrabilidad del brazo robótico está definida por tres velocidades angulares con respecto al sistema de referencia $\langle H \rangle$, es decir, q'_1 rota con respecto al eje n y q'_2 , q'_3 rotan con respecto al eje del sistema de referencia $\langle H \rangle$. En otras palabras, el movimiento cartesiano del manipulador móvil aéreo en el marco inercial $\langle \mathcal{R} \rangle$ se define como.

$$\begin{cases} h'_x = u_1 \cos(\psi) - u_m \sin(\psi) + l_2 \sin(q_2) \cos(\psi + q_1) q'_2 + l_2 \cos(q_2) \sin(\psi + q_1) (\psi' + q'_1) + \\ \quad l_3 \sin(q_2 + q_3) \cos(\psi + q_1) (q'_2 + q'_3) + l_3 \cos(q_2 + q_3) \sin(\psi + q_1) (\psi' + q'_1) \\ h'_y = u_1 \sin(\psi) - u_m \cos(\psi) + l_2 \sin(q_2) \sin(\psi + q_2) q'_2 - l_2 \cos(q_2) \cos(\psi + q_1) (\psi' + q'_1) + \\ \quad l_3 \sin(q_2 + q_3) \sin(\psi + q_1) (q'_2 + q'_3) - l_3 \cos(q_2 + q_3) \cos(\psi + q_1) (\psi' + q'_1) \\ h'_z = u_n - l_1 - l_2 \sin(q_2) - l_3 \sin(q_2 + q_3) \end{cases} \quad (10)$$

El punto de interés (cuya posición está siendo controlada) es el extremo operativo del manipulador aéreo real. El sistema de ecuaciones (2) puede ser escrito en forma compacta como $\mathbf{h}' = f(\mathbf{h}, \mathbf{q})\mathbf{u}$, es decir.

$$\mathbf{h}'(\mathbf{t}) = J(\mathbf{q}, \psi)\mathbf{v}(\mathbf{t})$$

$$(11)$$

donde, $J(\mathbf{q}, \psi) \in \mathfrak{R}^{m \times n}$ con $m = 3$ y $n = 7$ representa la matriz Jacobiana que define un mapeo lineal entre el vector velocidad del manipulador aéreo móvil $\mathbf{v} \in \mathfrak{R}^n$ donde $\mathbf{v} = [u_1 \ u_m \ u_n \ \psi \ q'_1 \ q'_2 \ q'_3]^T$ y el vector de velocidad del extremo operativo $\mathbf{h}' \in \mathfrak{R}^n$ donde $\mathbf{h}' = [h'_x \ h'_y \ h'_z]^T$.

3.2 Controlador Cinemático

El controlador cinemático permite ejecutar tareas de control al manipulador aéreo mediante el modelo cinemático encontrado, para realizar tareas de ejecución autónoma en ambientes no estructurados, dado que la trayectoria está representada por una variación en el tiempo de la trayectoria deseada $\mathbf{h}_d(\mathbf{t})$ y a sus derivadas sucesivas representadas por $\mathbf{h}'_d(\mathbf{t})$ describe la velocidad deseada del manipulador aéreo.

Donde $\mathbf{h}_d(\mathbf{t}) = [h_{xd}, h_{yd}, h_{zd}]^T$

- Es el vector que define la trayectoria deseada del extremo operativo de robot manipulador aéreo
- Esta dada en $\mathfrak{R}(x, y, z)$

Tomar en cuenta que la trayectoria deseada no depende ni considera, una posición instantánea del extremo operativo del robot manipulador aéreo, de hecho está definida por la variación en el tiempo de la trayectoria deseada.

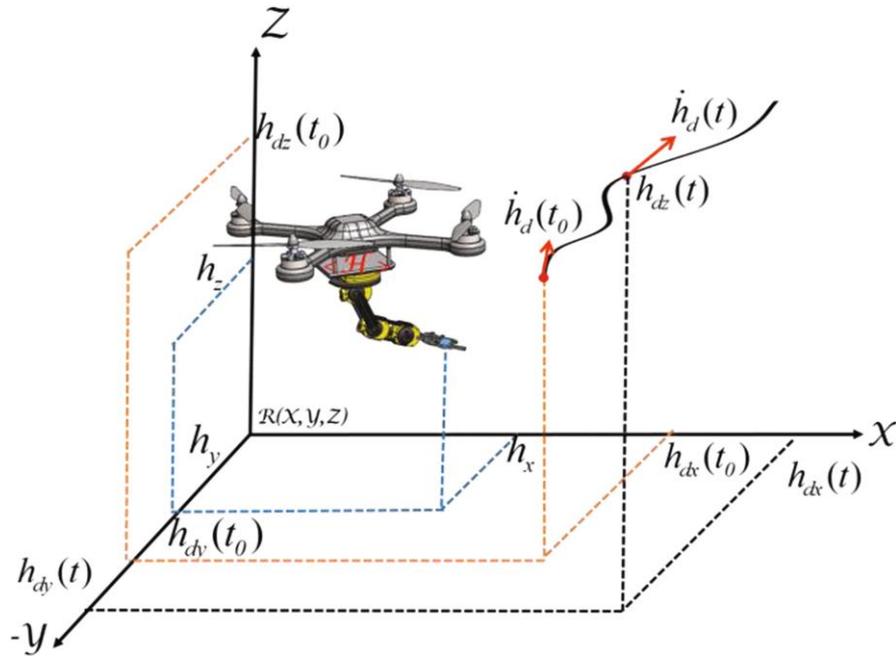


Figura. 3-5: Sistema de trayectoria de Robot Manipulador Aéreo.
Fuente: Erazo A. 2019

3.2.1 Diseño del Controlador

El sistema de seguimiento del dron será sometido al seguimiento de trayectorias, el cual considera como extremo operativo el final del eslabón 3 del robot manipulador, tomando en cuenta que el manipulador aéreo se desplaza en un sistema de referencia (x, y, z) , como se observa la Figura. 3-5, el controlador determinará la velocidad a la que tendrá que desplazarse el manipulador aéreo para que su extremo operativo cumpla la trayectoria deseada.

El controlador propuesto pretende resolver el problema de seguimiento de trayectoria del robot manipulador aéreo, este se basa en el modelo cinemático encontrado anteriormente. A continuación, se presenta la ley de control a usarse,

$$\mathbf{v} = \mathbf{J}^\#(\mathbf{h}'_d + \mathbf{L}_K \tanh(\mathbf{L}_K^{-1} \mathbf{K} \tilde{\mathbf{h}})) \quad (11)$$

Donde:

\mathbf{h}'_d : La velocidad del manipulador aéreo para el controlador.

$$\mathbf{J}^\# = \mathbf{W}^{-1} \mathbf{J}^{-1} (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \quad (12)$$

$\mathbf{J}^\#$: La matriz cinemática pseudoinversa para el manipulador aéreo

\mathbf{W} : Es una matriz definida positiva que pondera las acciones de control del sistema

$\tanh()$: Es descrita como la función de saturación de las velocidades de maniobrabilidad del manipulador aéreo

Finalmente $\mathbf{l}_x, \mathbf{k}_x, \mathbf{l}_y, \mathbf{k}_y, \mathbf{l}_z, \mathbf{k}_z$, son constantes de ganancia cuya única condición es pertenecer a los $\Re +$, y las cuales ponderan principalmente el control de errores respecto al sistema inercial \Re .

Por otro lado, para el análisis del comportamiento de control de errores del extremo operativo, $\mathbf{h}(\mathbf{t})$ se asume como una velocidad de trayectoria perfecta, para efectos de análisis, de este modo si se sustituye (11) en (10) y se obtiene la ecuación de lazo cerrado.

$$\mathbf{h}'_d + \mathbf{L} \tanh(\tilde{\mathbf{h}}) = \mathbf{0} \quad (13)$$

De acuerdo con el diseño del controlador propuesto es necesario comprobar y analizar la estabilidad del esquema de control.

3.3 Estabilidad del esquema de control

La estabilidad se determinará por el método de Lyapunov, el cual se describe como la estabilidad de los puntos de equilibrio dependiendo si todas las soluciones que se proponen en las cercanías del punto de equilibrio permanecen cercanas alrededor del mismo, caso contrario se dice que el punto de equilibrio es inestable de acuerdo al teorema de Lyapunov.

Además, se puede considerar también que un punto de equilibrio es asintóticamente estable si todas las soluciones que se inicien en las cercanías del punto de equilibrio no sólo permanecen en las cercanías del punto de equilibrio, sino que además tienden hacia el equilibrio a medida que el tiempo se aproxima a infinito.

Los teoremas de estabilidad de Lyapunov dan condiciones suficientes para estabilidad de puntos de equilibrio, por lo cual nos enfocaremos en la función candidata de Lyapunov que está dada por (14), el cual validará el algoritmo de control y lo someterá a los procesos experimentales adecuados para comprobar la eficiencia completa del mismo.

$$V(\tilde{\mathbf{h}}) = \frac{1}{2} \tilde{\mathbf{h}}^T \tilde{\mathbf{h}} \tag{14}$$

Su respectiva derivada sobre la trayectoria del sistema es:

$$V'(\tilde{\mathbf{h}}) = -\tilde{\mathbf{h}}^T L_K \tanh(L_K^{-1} K \tilde{\mathbf{h}}) \tag{15}$$

La única condición necesaria para que $\dot{V}(\tilde{\mathbf{h}}) < 0$ sería,

$$\tilde{\mathbf{h}}^T L_K \tanh(L_K^{-1} K \tilde{\mathbf{h}}) > 0 \tag{16}$$

Considerando que:

- K es definida como una diagonal positiva

El vector de control de error es,

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{h}}(t) = \mathbf{0}, \text{ asintóticamente.}$$

CAPITULO IV

4 DESARROLLO SIMULACIÓN 3D

Para este Proyecto se plantea el uso de un motor gráfico para simular los resultados obtenidos en Matlab. Este nos permite observar de una forma más ilustrativa mediante un objeto diseñado en SolidWorks y animado en Unity dentro de un entorno en donde el usuario siente la sensación de inmersión en un ambiente casi real.

4.1 Simulación 3D

4.1.1 *Motor Gráfico*

Unity es un motor gráfico 3D para PC y Mac multiplataforma creado por Unity Technologie, que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D y tiempo real. Unity puede publicar contenido para múltiples plataformas como PC, Mac, Nintendo Wii y iPhone. El motor también puede publicar juegos basados en la web usando el plugin Unity web player, que es un componente que permite interpretar código y los mapas de archivos 3D (AFNOR Normalisation, 2018).

Unity está disponible como plataforma de desarrollo para Microsoft Windows y OS.

Se asocia con 3DS Max, Maya, Blender, Cinema 4D, ya que los cambios realizados a los objetos creados con este software se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente (Andaluz et al., 2016).

Características

1. Interfaz de Usuario

La Interfaz de Usuario de Unity tiene 5 áreas principales de trabajo que pueden ser organizadas acorde a la facilidad del usuario, cada una tiene su funcionalidad específica y son detalladas a continuación en la Figura. 4-1:

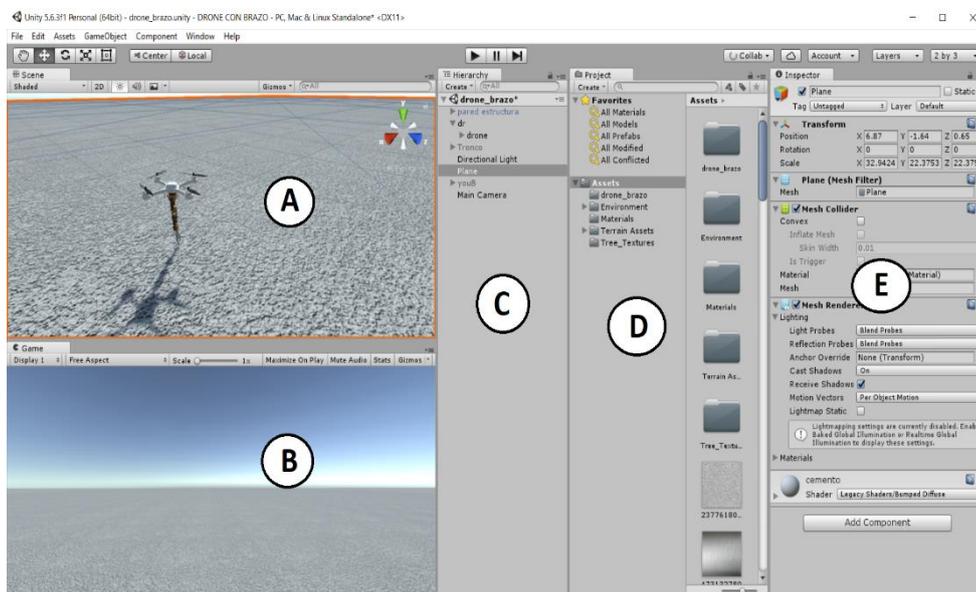


Figura. 4-1: Interfaz de Usuario
Fuente: Erazo A. 2019

- a. **Vista de Escena:** La escena es el área de construcción de Unity donde se construye visualmente cada escena.
- b. **Vista de Juego:** En la vista de juego se obtiene una previsualización de una animación. En cualquier momento se puede reproducir una visualización en esta vista.
- c. **Vista de Jerarquía:** La vista de jerarquía contiene todos los objetos en la escena actual.
- d. **Vista de Proyecto:** Esta es la librería de assets para el modelamiento. Se puede importar objetos 3D de distintas aplicaciones a la librería, se puede importar texturas y crear otros objetos como Scripts o Prefabs que se almacenarán aquí.
- e. **Vista de Inspector:** La vista de inspector sirve para varias cosas. Si se seleccionan objetos entonces mostrará las propiedades de ese objeto donde se pueden personalizar varias características del objeto. También contiene la configuración para ciertas herramientas.

2. Modos de Visualización

Por defecto la vista de escena tiene una perspectiva 3D. Se puede cambiar esto por un número de vistas Ortográficas: 1) top-down; 2) side; 3) front. En la parte derecha de la vista de la escena se puede ver un “Gizmo” que parece una caja con conos que salen de ella, Figura. 4-2.

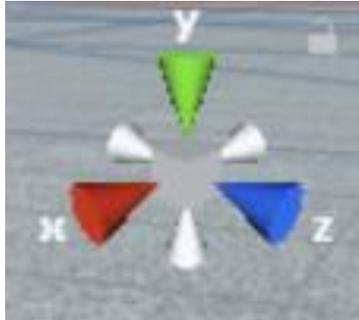


Figura. 4-2: Gizmo
Fuente: Erazo A. 2019

a. Perspectiva 3D top-down (arriba-abajo): En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono verde del “Gizmo” de perspectiva.

b. Perspectiva 3D side (derecha): En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono rojo del “Gizmo” de perspectiva.

c. Perspectiva 3D front (frontal): En este modo se visualiza la escena y sus objetos vistos desde arriba y desde abajo, para lo cual se hace clic en el cono azul del “Gizmo” de perspectiva.

3. Botones de Control

Debajo de las opciones de visualización se tiene una fila con 4 botones, como se muestra en la Figura. 4-3. Se puede usar Q, W, E, R para alternar entre cada uno de los controles, que se detalla a continuación:



Figura. 4-3: Botones de control.

Fuente: Erazo A. 2019

a) Hand Tool (Q): Este control permite moverse alrededor en la vista de escena. Mantener ALT permitirá rotar, COMMAND/CTRL permitirá hacer zoom y SHIFT incrementa la velocidad de movimiento mientras se usa la herramienta

b) Translate Tool (W): Permite mover cualquier objeto seleccionado en la escena en los ejes X, Y y Z.

c) Rotate Tool (E): Permite rotar cualquier objeto seleccionado en la escena.

d) Scale Tool (R): Permite escalar cualquier objeto seleccionado en la escena.

4.1.2 Modelación

En el modelado del dron y su respectivo brazo robótico se requiere de un proceso para poder importar a Unity con todas las propiedades necesarias, en el cual intervienen los siguientes softwares:

- Creación del diseño en el software SolidWorks, este programa permite modelar piezas y conjuntos y extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la producción en masa. SolidWorks se basa en técnicas de modelado con sistemas CAD. El concepto principal que se maneja es que es el usuario pueda diseñar virtualmente su pieza o conjunto.

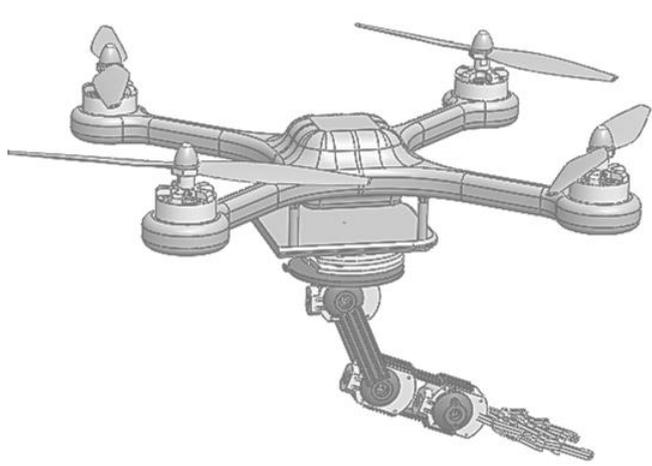


Figura. 4-4: Diseño del Manipulador Robot Aéreo SolidWorks.
Fuente: Erazo A. 2019

- Blender es un programa dedicado a modelación, renderizado, animación y creación de gráficos tridimensionales, contiene una distribución de software libre. Blender nos permite renderizar a un objeto dándole así una característica más realística, también nos permite modificar las propiedades con las que el objeto fue importado desde SolidWorks.

De igual manera Blender nos da la facilidad de exportar nuestro objeto a cualquier otro formato requerido, como en el caso de Unity que solo admite objetos con extensión (.fbx).



Figura. 4-5: Renderizado Manipulador Robot Aéreo Blender
Fuente: Erazo A. 2019

4.1.3 Virtualización

Para tener un plano de referencia se crea un objeto 3D "Plane" en Unity, que sirve como base en el cual van ubicados todos los objetos requeridos.

Con las herramientas de transformación que dispone el software para poder girar, mover o ampliar los objetos de acuerdo con cada eje (x-y-z), se van ubicando cada objeto.

Los gráficos de texturas descargados son guardados dentro de la raíz del proyecto, la carpeta "Assets". Estos gráficos se los pueden adjuntar a un material creado a través de los shaders los cuales son scripts que contienen algoritmos para procesar el color de cada pixel renderizado acorde a la configuración del material. Este material se lo puede adherir fácilmente a cada objeto para dar un aspecto más real como se muestra en la Figura. 4-6.

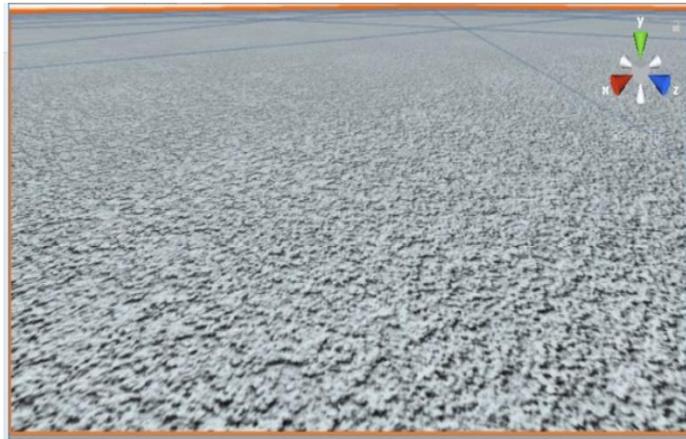


Figura. 4-6: Tipo de Textura en UNITY
Fuente: Erazo A. 2019

Para hacer aún más real la animación Unity posee entre sus GameObject lo que se puede asimilar al mismo efecto de la luz solar por un lado claridad y por la otra oscuridad, dependiendo de la posición del objeto con respecto a la ubicación de la luz y su dirección (Figura. 4-7).



(a) Sin GameObject Light



(b) Con GameObject Light

Figura. 4-7: Efecto GameObject Light
Fuente: Erazo A. 2019

Unity nos da la facilidad de manipular el enfoque para visualizar de mejor manera la animación por medio de la única cámara que contiene cada escena. Sin embargo, nos permite poner tantas cámaras según el usuario vea necesario. De igual forma cada cámara es como un objeto al cual se le puede rotar o mover entre los diferentes ejes (Figura. 4-8).

Las cámaras pueden permanecer estáticas al ubicarlas en un punto en específico sin pertenecer a ningún objeto en la vista de jerarquía, al ingresar la cámara en un objeto cualquiera está hereda la animación del objeto al cual pertenece, de esta forma la cámara puede seguir al objeto el cual se desea enfocar sin tener que programar ningún script a la cámara.



(a) Enfoque frontal



(b) Enfoque superior

Figura. 4-8: Enfoque de cámaras

Fuente: Erazo A. 2019

Unity posee herramientas que nos ayudan a dar características únicas a los objetos como generar relieves en el caso de una montaña, esta herramienta se llama Brushes. Los Brushes nos facilitan el diseño del entorno cuando se requiere crear montañas sin tener que importar montañas ya diseñadas desde otro programa.

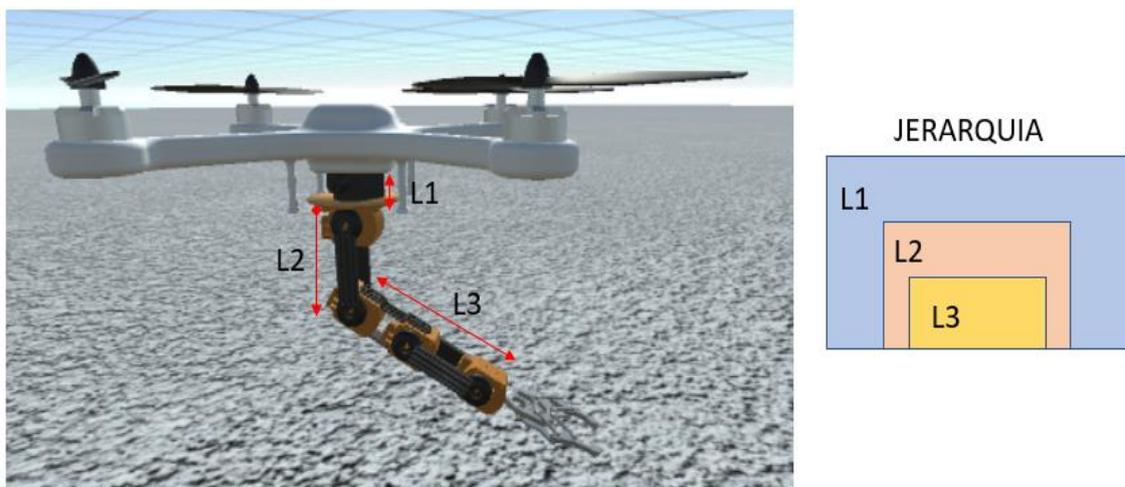


Figura. 4-9: Entorno con montañas
Fuente: Erazo A. 2019

4.1.4 Jerarquía

Unity maneja un concepto llamado “Parenting” que permite a un objeto ser hijo de otro el cual pasará a ser objeto padre con solo arrastrar el un objeto sobre el otro objeto.

Este concepto nos ayuda para lo que son animaciones ya que cuando un objeto llega a ser hijo automáticamente hereda las propiedades y animaciones del padre.



(a) Diagrama del Manipulador Móvil

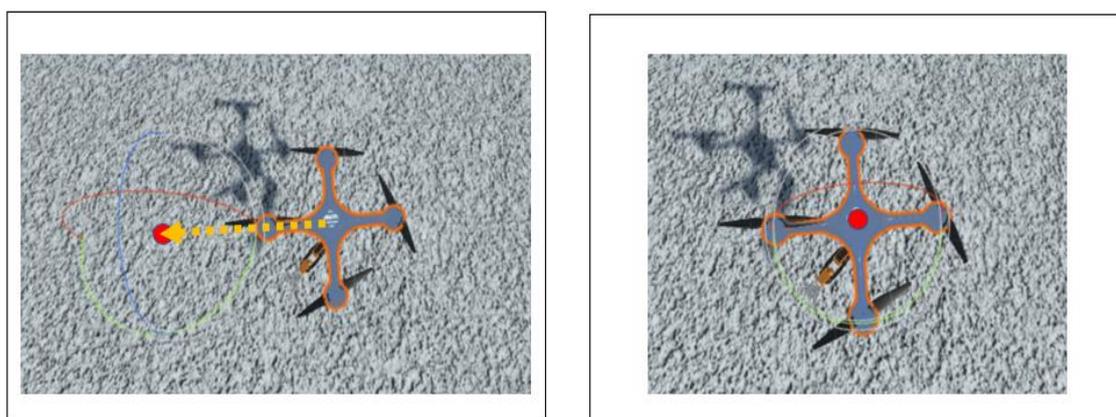
(b) Jerarquía del brazo robótico

Figura. 4-10: Jerarquía en Unity
Fuente: Erazo A. 2019

Un objeto puede llegar a tener un sin número de hijos pudiendo este pasar a ser padre de otros objetos. Al realizar animaciones es muy usual armar una estructura completa manejando una jerarquía como se muestra en la Figura. 4-10, donde se tiene el brazo robótico con sus tres eslabones estructurados, donde L3 es hijo de L2, L2 es hijo de L1, y por ende, L3 es hijo de L1 también; lo que resulta de gran ayuda al momento de aplicar una propiedad de rotación en el eslabón L1 ya que rotan también L2 y L3 manteniendo su posición con respecto a sus propios ejes, muy diferente en el caso que rotemos el eslabón L3 puesto que solo este va a rotar dado que no tiene ningún objeto hijo.

4.1.5 *Pivot virtual*

Algunas veces al exportar los objetos desde los softwares de diseño o renderizado se cambia el valor del centro del pivote del objeto que hace referencia respecto a qué punto va a girar. Aplicando el mismo concepto de “Parenting” Unity nos permite modificar este pivot de acuerdo con nuestro requerimiento, haciendo que este objeto sea hijo de un objeto en blanco al cual se lo llama pivot virtual. Al crear un objeto en blanco en la vista de jerarquía nosotros podemos colocar la posición que llegará a ser el nuevo centro de pivot para nuestro objeto como podemos observar en la Figura. 4-11.



(a) Centro de Pivot Original

(b) Centro de Pivot Virtual

Figura. 4-11: Pivot Virtual

Fuente: Erazo A. 2019

4.1.6 Comunicación

Unity posee librerías para realizar comunicación serial con cualquier otra aplicación, pero tiene un retardo al momento de leer los datos enviados desde Matlab haciendo visualizar en la simulación un movimiento no tan continuo.

Una biblioteca de enlace dinámico o más comúnmente conocido como DLL contiene archivos con código programado que se cargan al ser llamados por un programa que está corriendo en un sistema operativo, lo cual facilita el aprovechamiento de la memoria del sistema puesto que, al estar compartidas, es suficiente con mantener una copia en memoria para todos los programas que la utilicen.

La comunicación entre Matlab y Unity se la realiza mediante un DLL la cual es llamada en ambas aplicaciones para el envío y recepción de datos. El funcionamiento no es muy complejo; se carga la librería dentro de cada aplicación, se inicia para que se asigne un espacio en memoria y procedemos a setear los datos que vamos a enviar, en Unity leemos esos datos cargados en los espacios de memoria. La comunicación establecida mediante este tipo de archivos es en tiempo real.

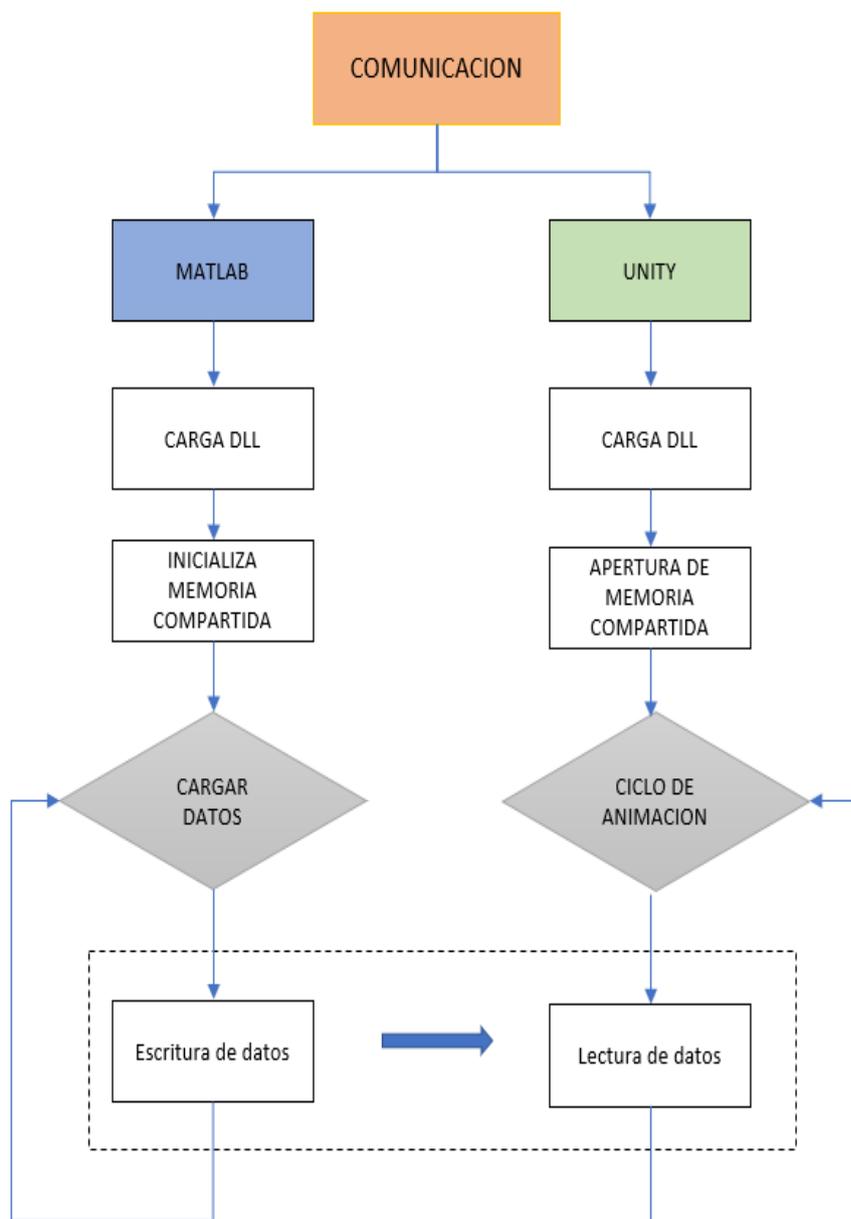


Figura. 4-12: Proceso de Comunicación
 Fuente: Erazo A. 2019

4.1.7 Animación

La programación de scripts es realizada en la propia herramienta MonoDevelop de Unity que utiliza lenguaje C# y es obligatoria para realizar animaciones en el entorno, a fin de que los eventos sean ejecutados en el momento deseado. Estos scripts permiten manipular las propiedades que conforman cada objeto (i.e. crear movimiento, desaparecer el objeto).

En el script se crean como variables los GameObject que vayamos a necesitar para nuestra animación lo que nos da la facilidad que desde un mismo script podamos animar varios objetos a la vez dando el efecto de continuidad en el proceso de animación.

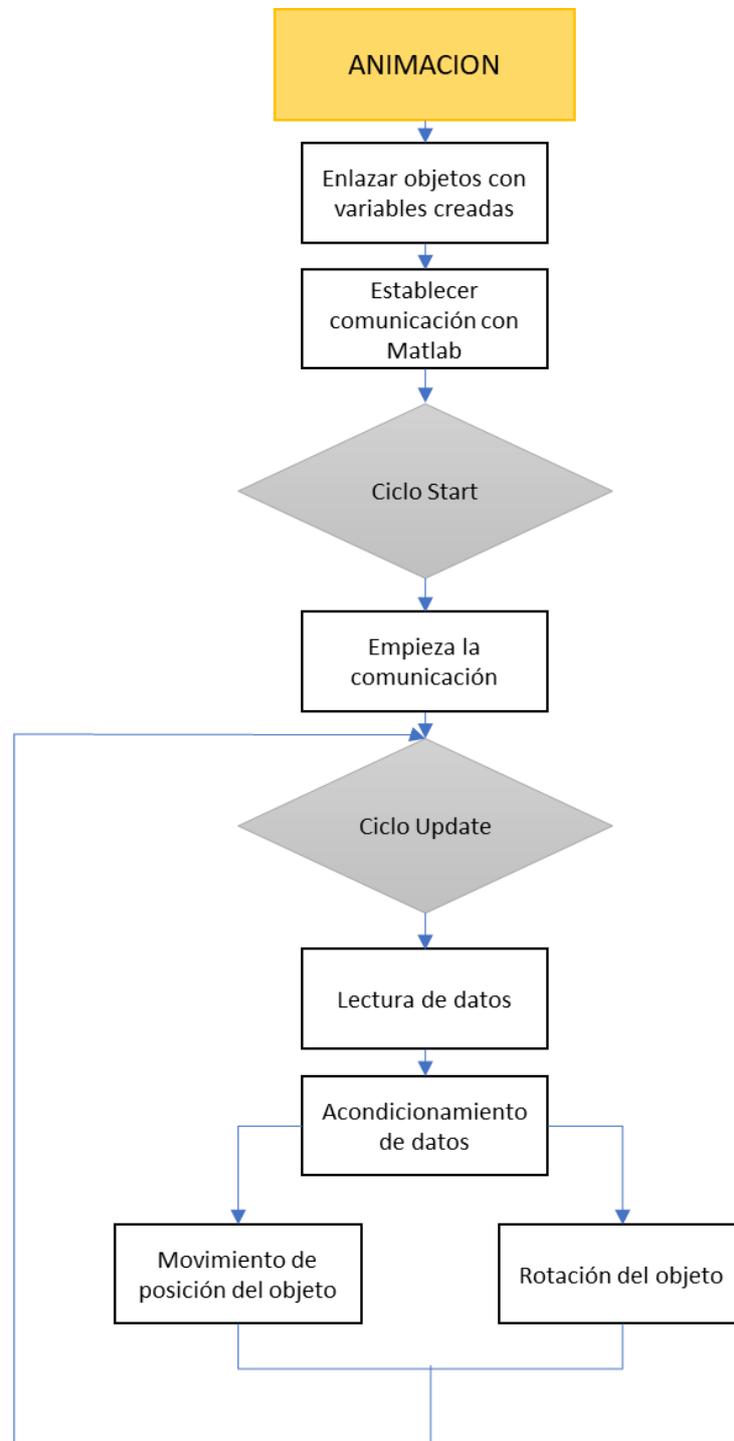


Figura. 4-13: Proceso de Animación
Fuente: Erazo A. 2019

Al generar el script automáticamente se genera la estructura básica que consta de la clase con el nombre que nosotros creamos, dentro de esta se crean dos funciones que no devuelven ningún valor, la primera que se llama “Start” y la segunda que se llama “Update”.

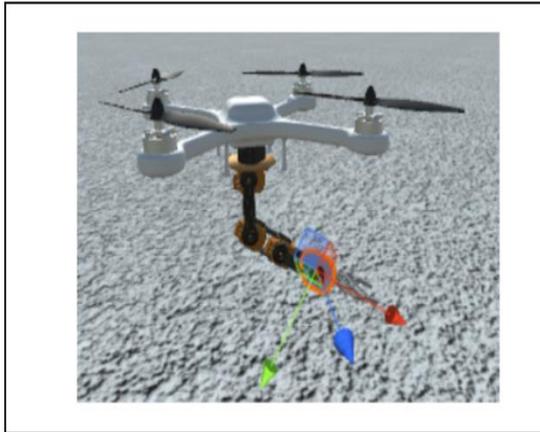
La función “Start” es llamada al iniciar la primera vez el script siempre y cuando este script se encuentre enlazado a un objeto y habilitado, esta se utiliza para inicializar algún valor de una variable o para enviar la sincronización con otros scripts.

La función “Update” se llama de forma automática por cada frame que avanza al cargar la animación, lo que quiere decir que, si nuestra animación está funcionando a 50 fps la función “Update” se ejecutará 50 veces cada segundo. Como este es un ciclo repetitivo hasta finalizar la animación, dentro de éste se encuentra el código para realizar el movimiento de posición y la rotación del objeto con los datos obtenidos desde Matlab.

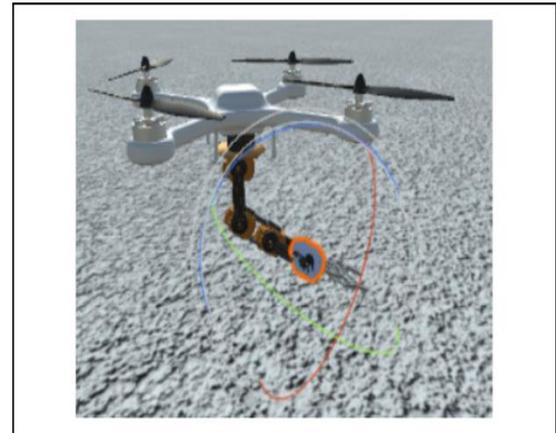
Primero se adquiere los datos enviados desde Matlab para posteriormente acondicionarlos, una vez acondicionados los datos utilizamos funciones propias que hacen referencia a las propiedades que tienes cada objeto en Unity.

Cada objeto dentro del editor de Unity 3D posee propiedades como Transform (position, rotation y scale), que representan los valores de las variables de cada uno de los objetos, los mismos que pueden modificarlos mediante un script utilizando la función “transform.localEulerAngles (float xAngle, float yAngle, float zAngle)”. Esta función permite rotar al objeto de acuerdo con los ejes x-y-z al mismo tiempo o individualmente poniendo cero en los otros valores de los ejes. El sentido de la rotación depende del signo con el que se encuentre la variable positiva (horario) o negativo (anti horario).

La función “transform.position” actualiza la posición del objeto generando una animación de movimiento vertical, horizontal o diagonal al modificar los valores de las variables que se encuentra en un vector tridimensional “Vector3” que contiene las posiciones en x,y,z del objeto.



(a) Movimiento de posición



(b) Rotación

Figura. 4-14: Ejemplo de la función transform.position
Fuente: Erazo A. 2019

El archivo del script compilado debe ser arrastrado desde la vista de proyecto hasta el objeto en vista de jerarquía que va a ser animado.

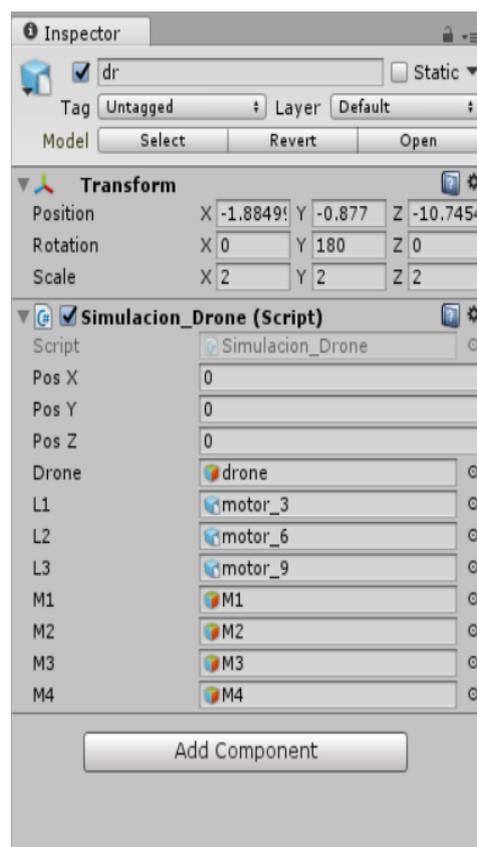


Figura. 4-15: Script adjuntado al objeto
Fuente: Erazo A. 2019

En la Figura se muestra nuestro script adjunto al objeto, dentro del script se encuentran especificadas las variables que utilizamos para nuestra animación, como se crearon variables de tipo objeto debemos enlazar con los objetos que tenemos dentro de nuestra carpeta Assets, buscamos y seleccionamos todos los objetos necesarios.

Finalmente, la animación empezará a correr con los datos recibidos desde Matlab acorde al tiempo programado.

CAPITULO V

5 RESULTADOS EXPERIMENTALES

5.1 Controlador del Robot Manipulador Aéreo

5.1.1 *Resultados de Simulación*

El objetivo principal de las pruebas es comprobar la estabilidad y el desempeño del controlador cinemático no lineal propuesto, en un ambiente simulado 3D, el cual ha sido desarrollado en el software Unity, la Figura. 5-1, describe el proceso y las etapas de simulación al cual ha sido sometido el robot manipulador aéreo y su respectivo control cinemático no lineal

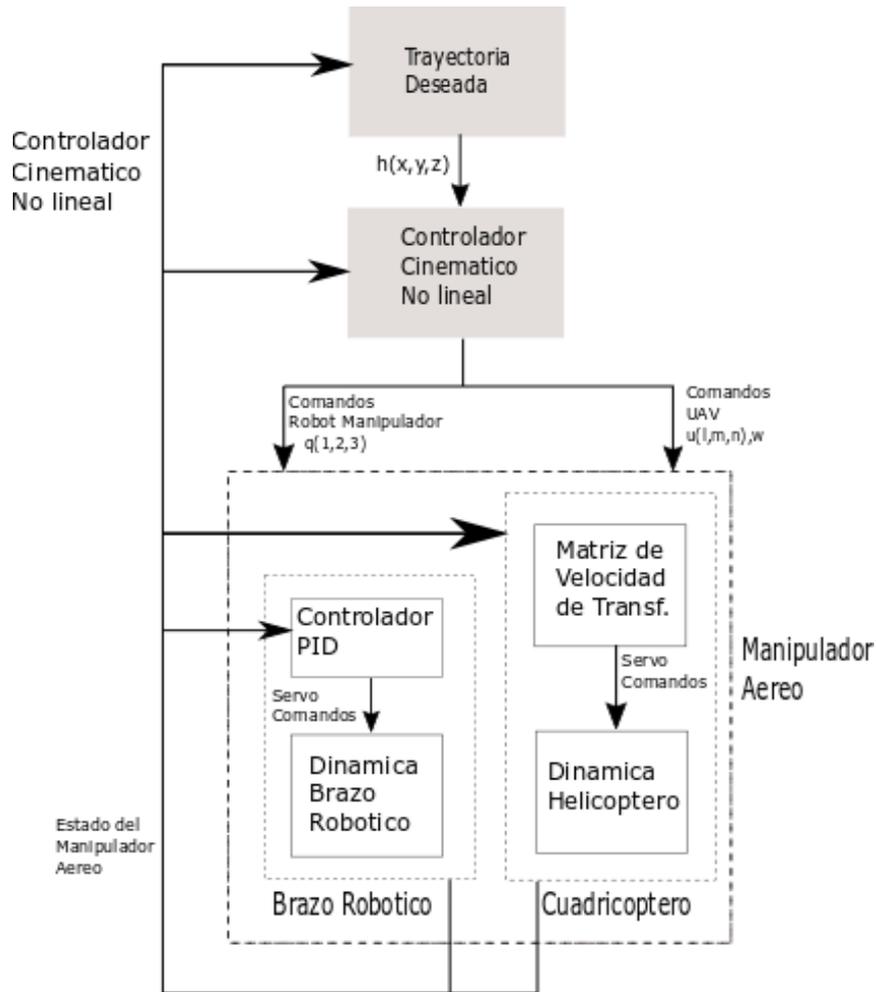


Figura. 5-1: Proceso de Simulación
Fuente: Erazo A. 2019

Tomar en consideración que los datos como: posición inicial del robot manipulador aéreo, trayectorias deseadas, controlador cinemático no lineal son desarrolladas en el software Matlab, en el entorno Unity se reciben todos estos datos y se los simula en un entorno 3D, de acuerdo a los parámetros recibidos y se los aplica al robot manipulador animado en este entorno, en la Figura. 5-2 se puede observar el robot en el entorno creado en Unity.



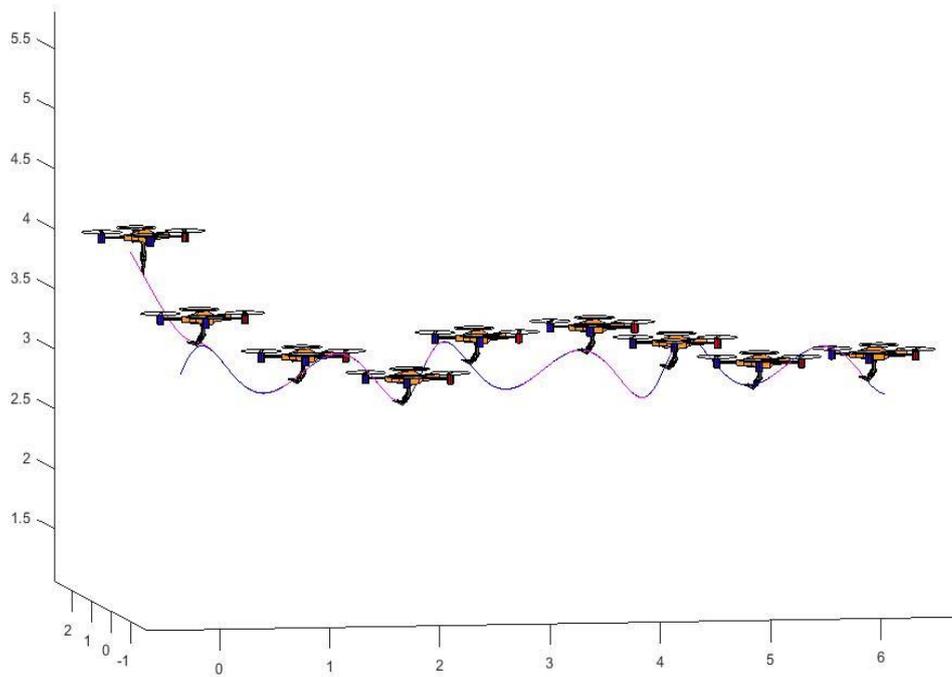
Figura. 5-2: Robot manipulador Aéreo en Unity

Fuente: Erazo A. 2019

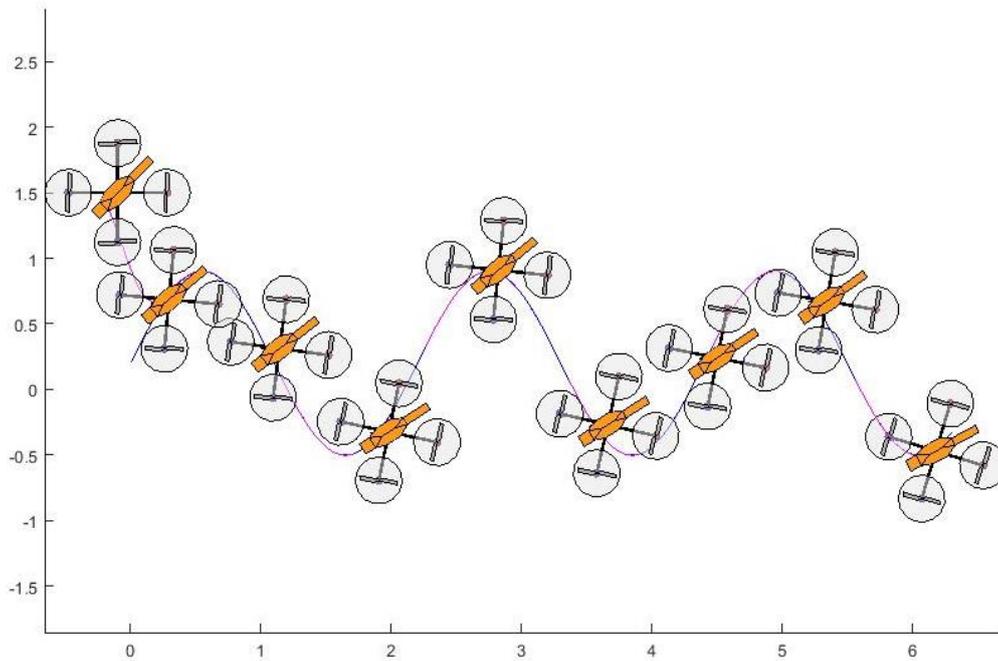
Para efectos de comprobación experimental se analiza el control de manipulador aéreo en varias trayectorias distintas las cuales cumplen con el seguimiento trazado. Las trayectorias han sido seleccionadas de tal manera que cumplan movimientos en todos los ejes y así poder comprobar de forma íntegra que el controlador cinemático no lineal cumple con los objetivos planteados.

Simulación 1

La trayectoria a seguir es una función seno, en las Figura. 5-3 y 5-4 se presenta el movimiento estroboscópico del robot y la evolución del seguimiento de trayectoria en el entorno virtual de Unity, respectivamente.

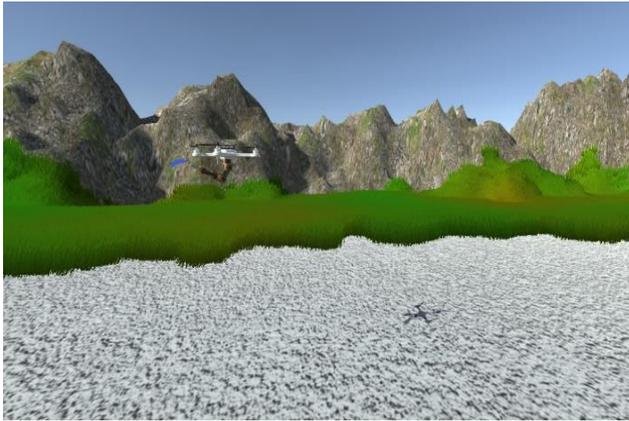


a) Simulación 1 de seguimiento trayectoria vista frontal

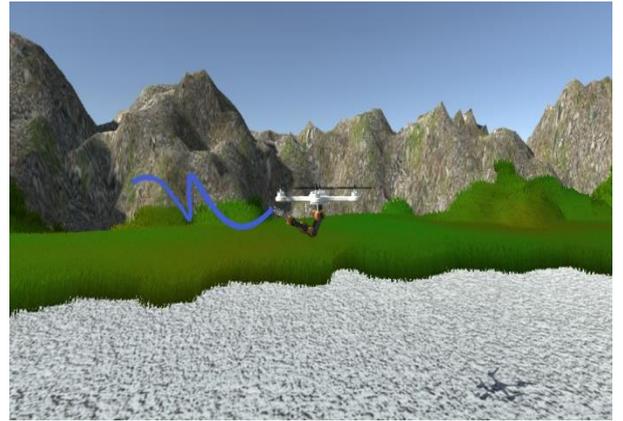


b) Simulación 1 de seguimiento trayectoria vista superior

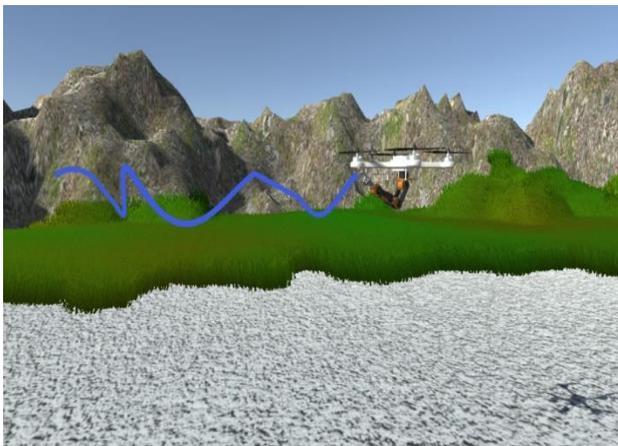
Figura. 5-3: Movimiento estroboscópico: Simulación 1 del seguimiento de trayectoria del robot Manipulador aéreo.
Fuente: Erazo A. 2019



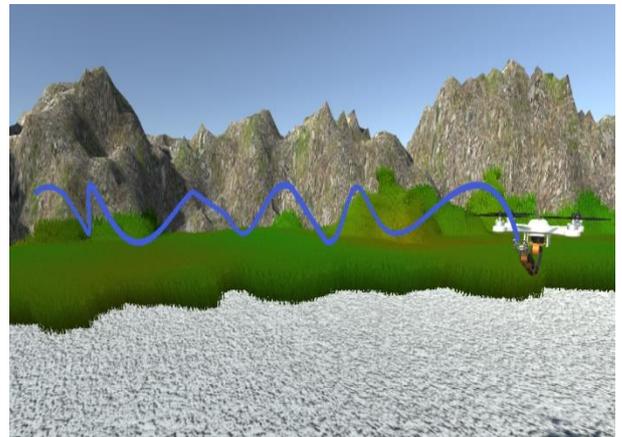
(a) Vista inicial



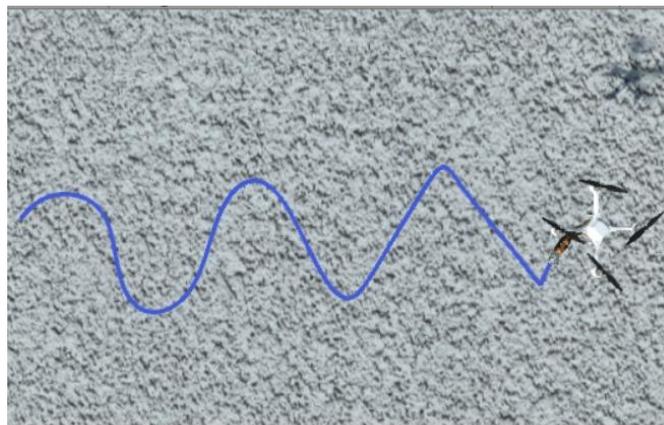
(b) Vista nivel 1



(c) Vista nivel 2



(d) Vista final



(e) Vista superior

Figura. 5-4: Entorno virtual: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo
Fuente: Erazo A. 2019

La Figura. 5-5 y 5-6, 5-7 corresponden a los errores de seguimiento y las velocidades de control, respectivamente, observando que los errores convergen a cero y las velocidades que permiten el seguimiento de la trayectoria.

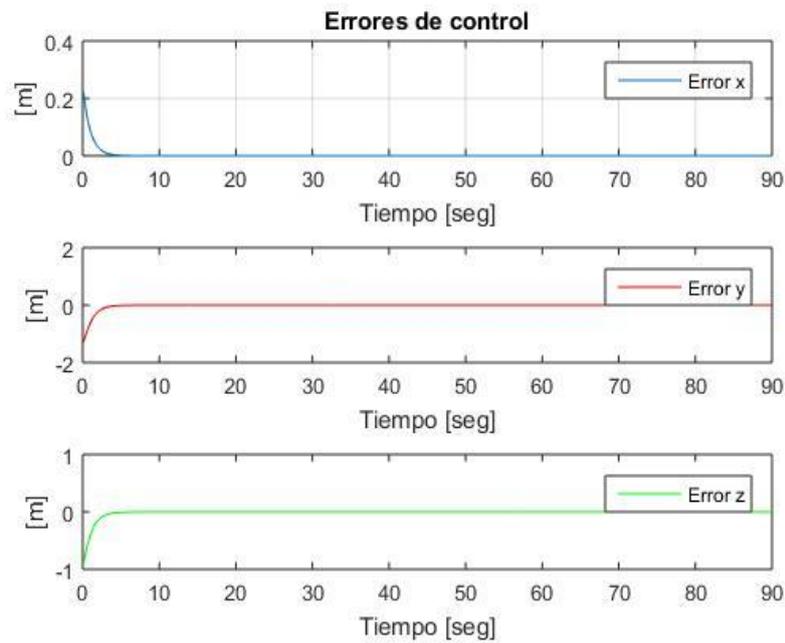


Figura. 5-5: Errores de control: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

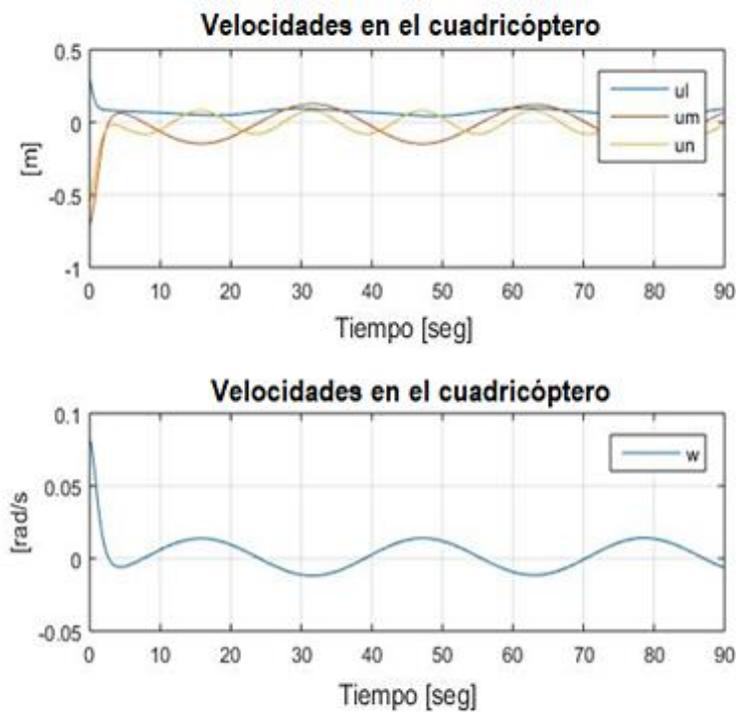


Figura. 5-6: Velocidades en el cuadricóptero: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

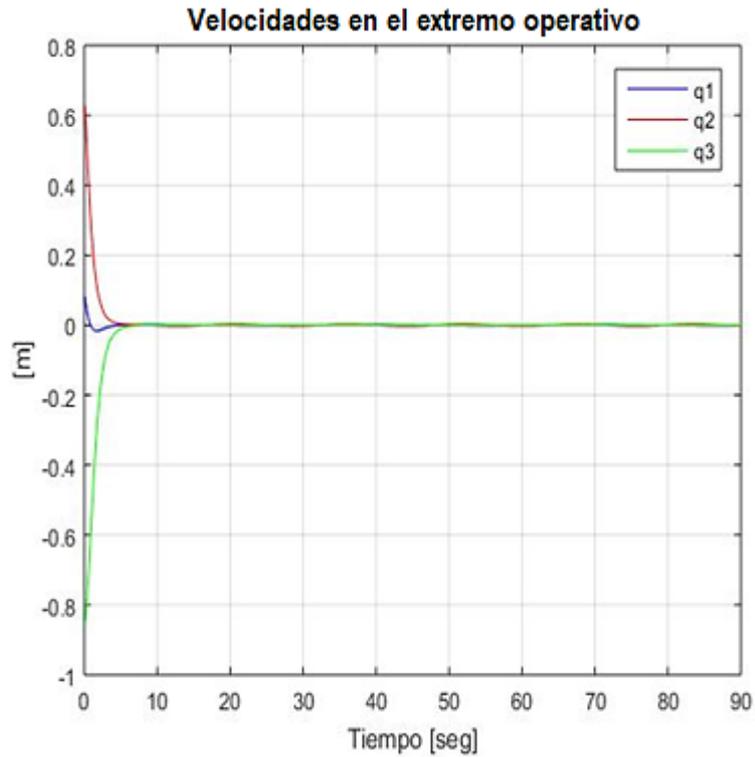
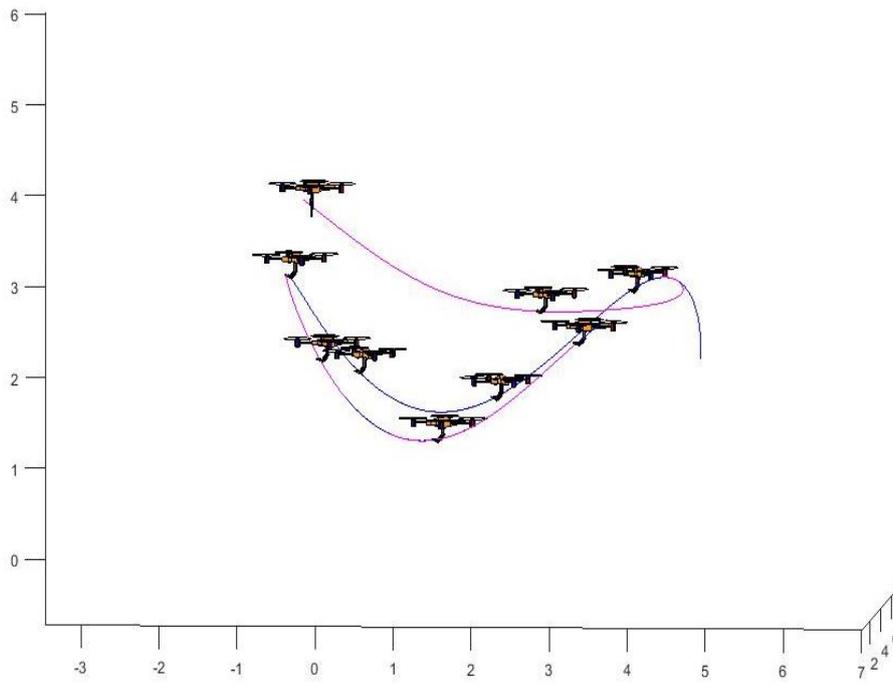


Figura. 5-7: Velocidades en el extremo operativo: Simulación 1 del seguimiento de trayectoria del robot manipulador aéreo

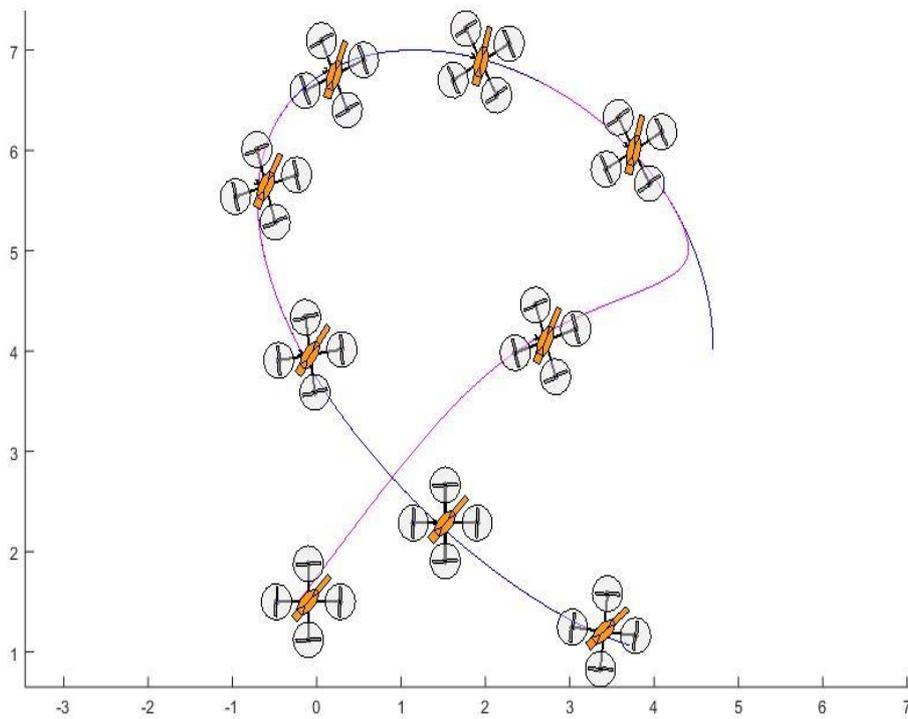
Fuente: Erazo A. 2019

Simulación 2

La segunda simulación corresponde a la función silla de montar, en las Figura. 5-8 y 5-9 se presenta el movimiento estroboscópico del robot y la evolución del seguimiento de trayectoria en el entorno virtual de Unity, respectivamente.



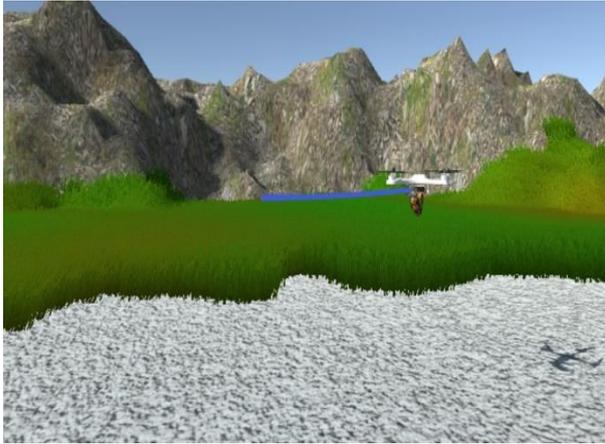
(a) Simulación 2 de seguimiento trayectoria vista frontal



(b) Simulación 2 de seguimiento de trayectoria vista superior

Figura. 5-8: Movimiento estroboscópico: Simulación 2 del seguimiento de trayectoria del robot Manipulador aéreo.

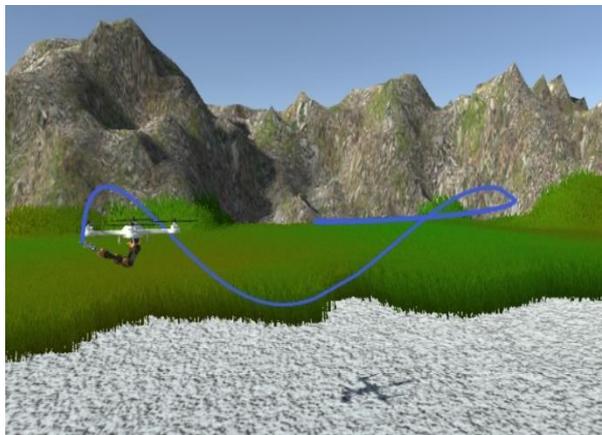
Fuente: Erazo A. 2019



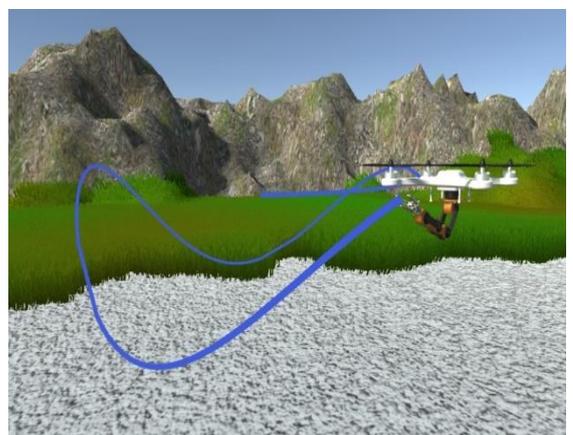
(a) Vista inicial



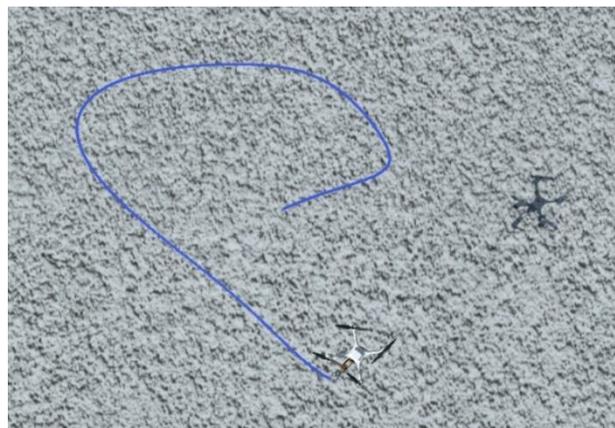
(b) Vista nivel 1



(c) Vista nivel 2



(d) Vista final



(e) Vista superior

Figura. 5-9: Entorno virtual: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

La Figura. 5-10 y 5-11, 5-12 corresponden a los errores de seguimiento y las velocidades de control, respectivamente, observando que los errores convergen a cero y las velocidades que permiten el seguimiento de la trayectoria.

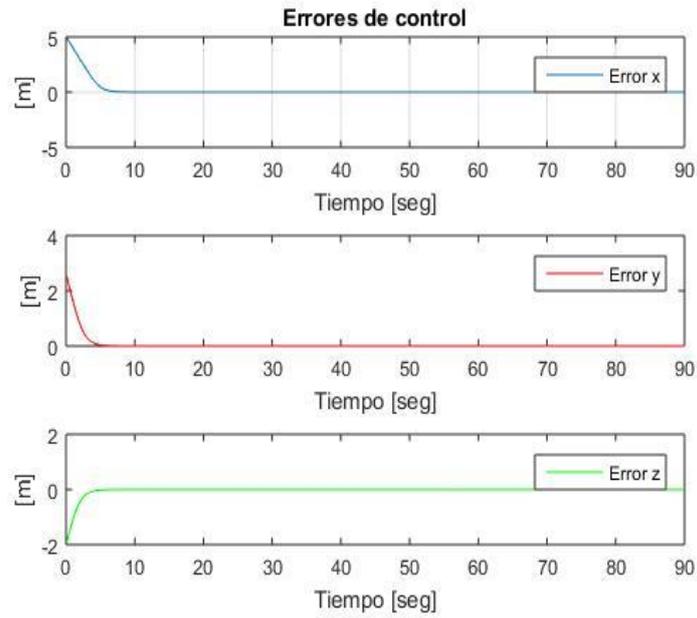


Figura. 5-10: Errores de control: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

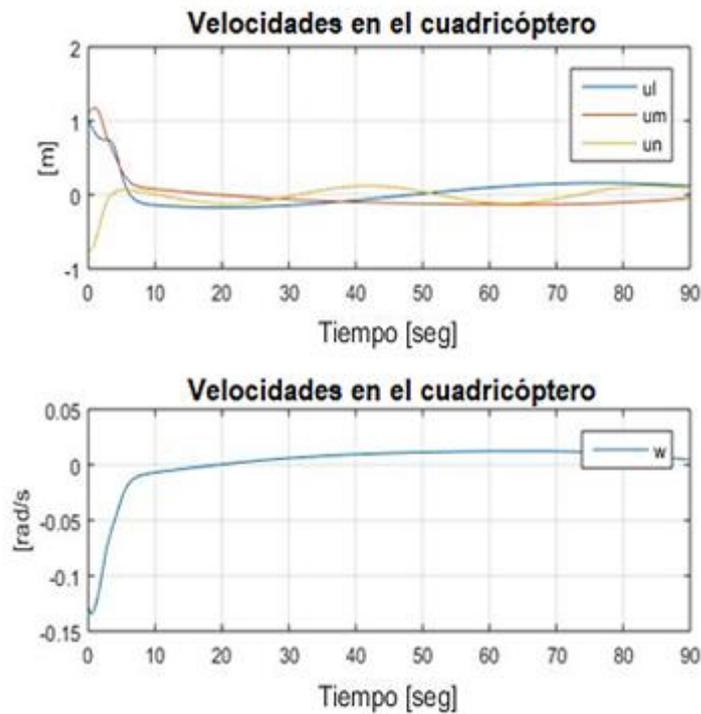


Figura. 5-11: Velocidades en el cuadricóptero: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

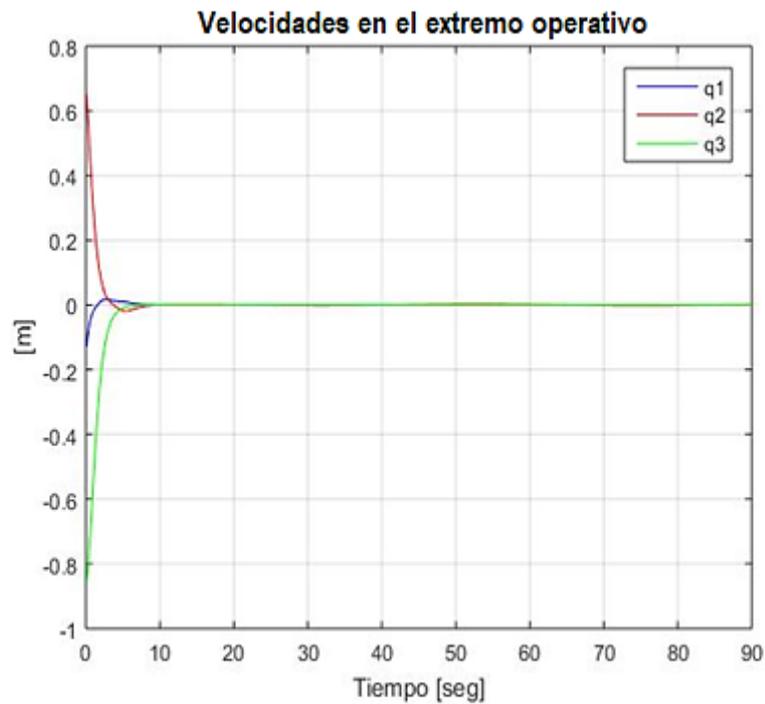
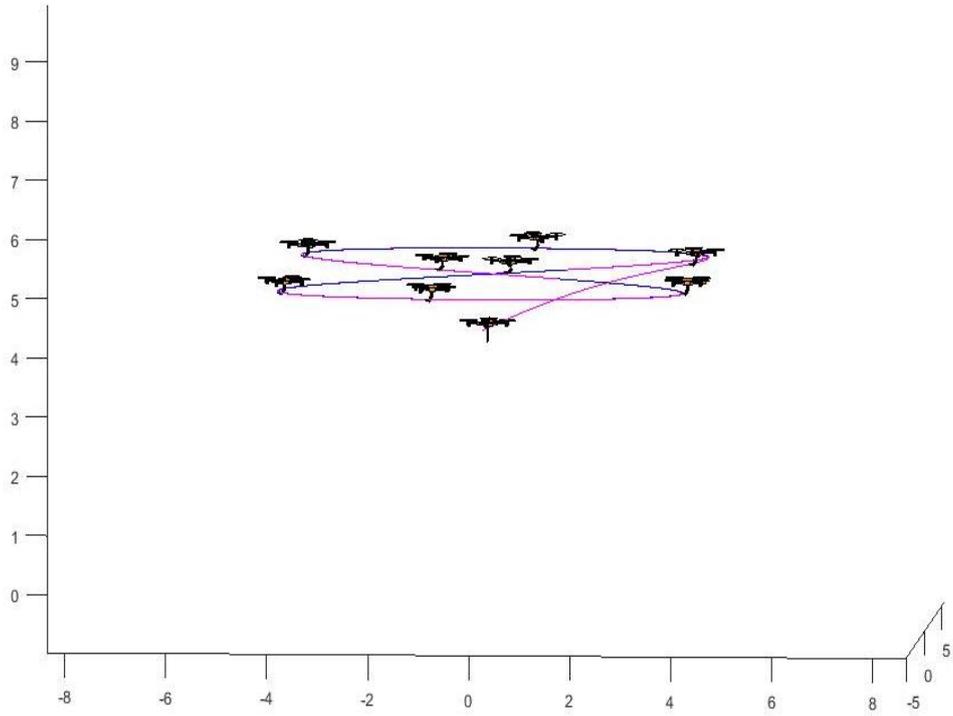


Figura. 5-12: Velocidades en el extremo operativo: Simulación 2 del seguimiento de trayectoria del robot manipulador aéreo

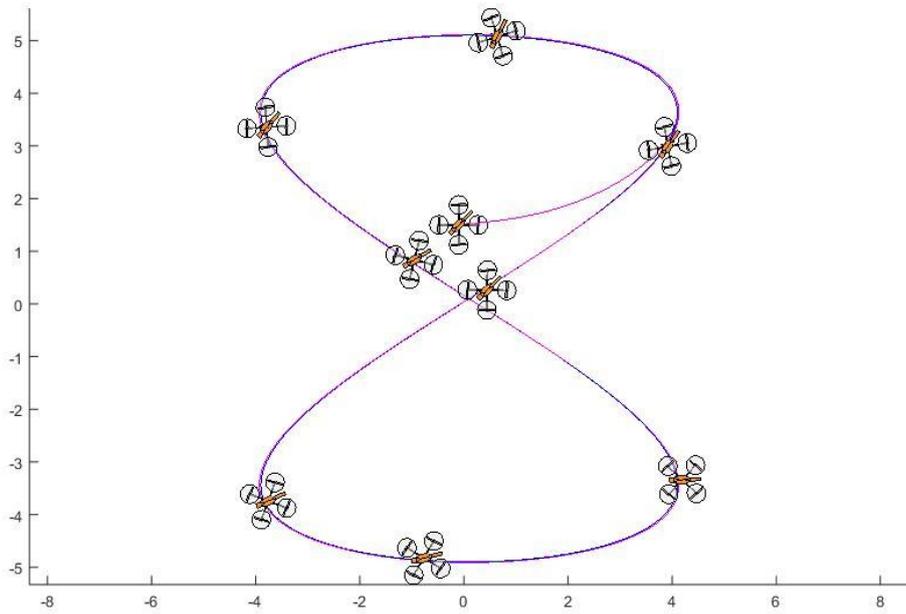
Fuente: Erazo A. 2019

Simulación 3

Esta simulación corresponde a la función que forma un número 8, en las Figuras 5-13 y 5-14 se presenta el movimiento estroboscópico del robot y la evolución del seguimiento de trayectoria en el entorno virtual de Unity, respectivamente.



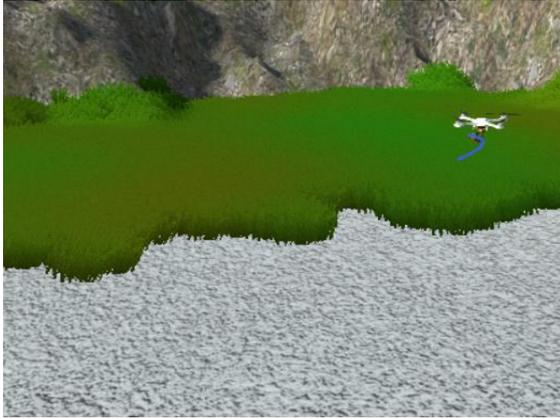
(a) Simulación 3 de seguimiento trayectoria vista frontal



(b) Simulación 3 de seguimiento trayectoria vista superior

Figura. 5-13: Movimiento estroboscópico: Simulación 3 del seguimiento de trayectoria del robot Manipulador aéreo.

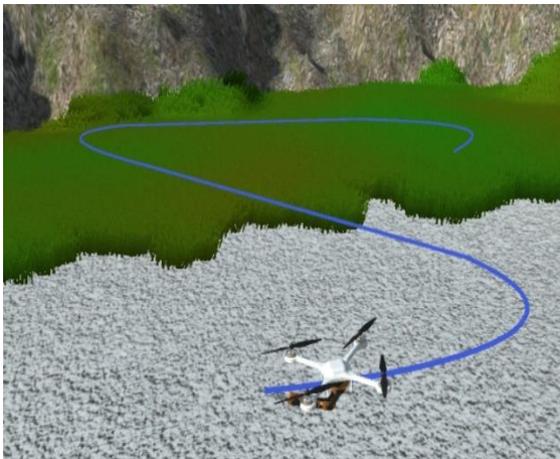
Fuente: Erazo A. 2019



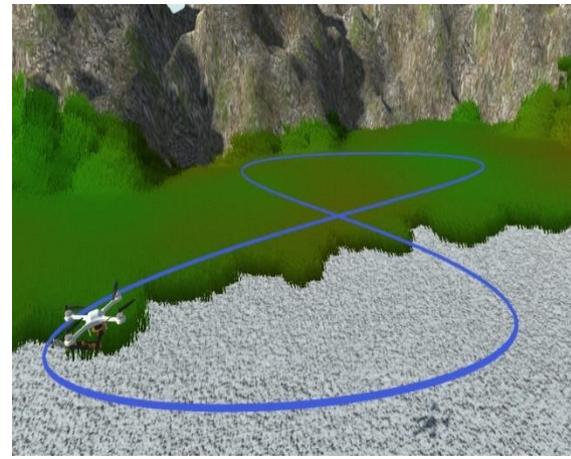
(a) Vista inicial



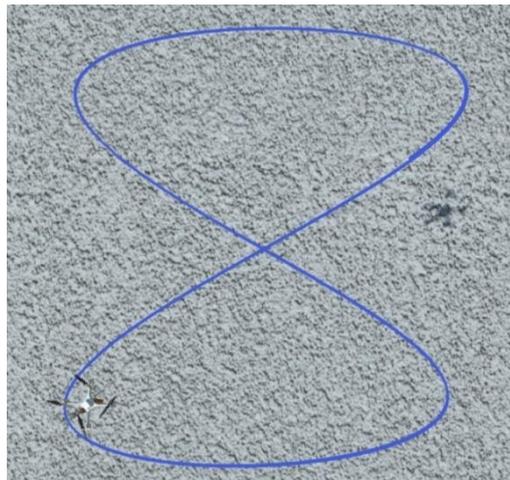
(b) Vista nivel 1



(c) Vista nivel 2



(d) Vista final



(e) Vista superior

Figura. 5-14: Entorno virtual: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo
Fuente: Erazo A. 2019

La Figura. 5-15 y 5-16, 5-17 corresponden a los errores de seguimiento y las velocidades de control, respectivamente, observando que los errores convergen a cero y las velocidades que permiten el seguimiento de la trayectoria.

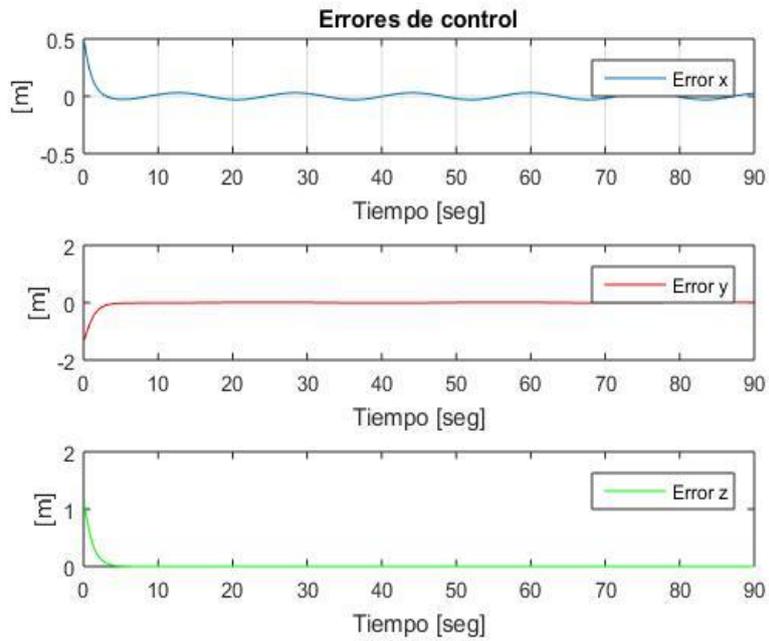


Figura. 5-15: Errores de control: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

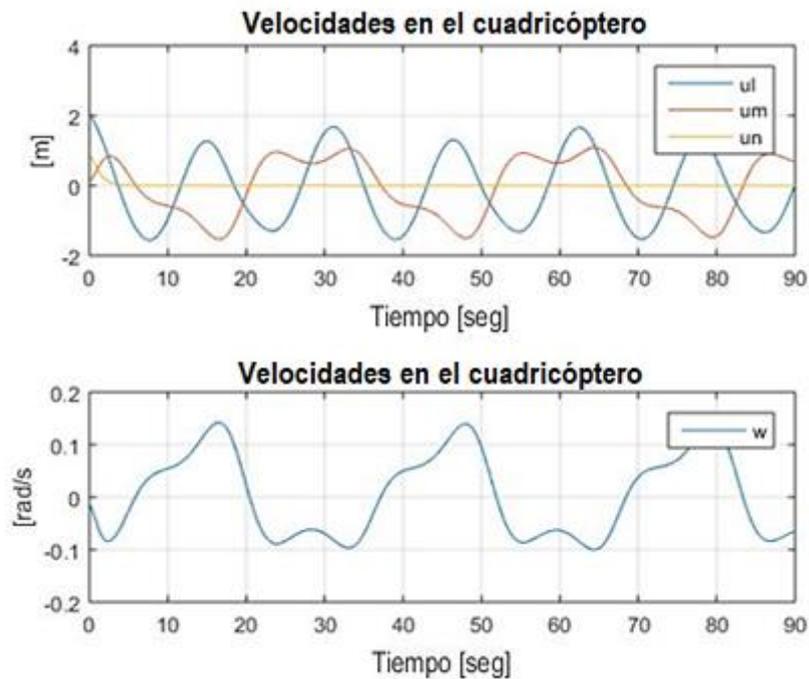


Figura. 5-16: Velocidades en el cuadricóptero: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

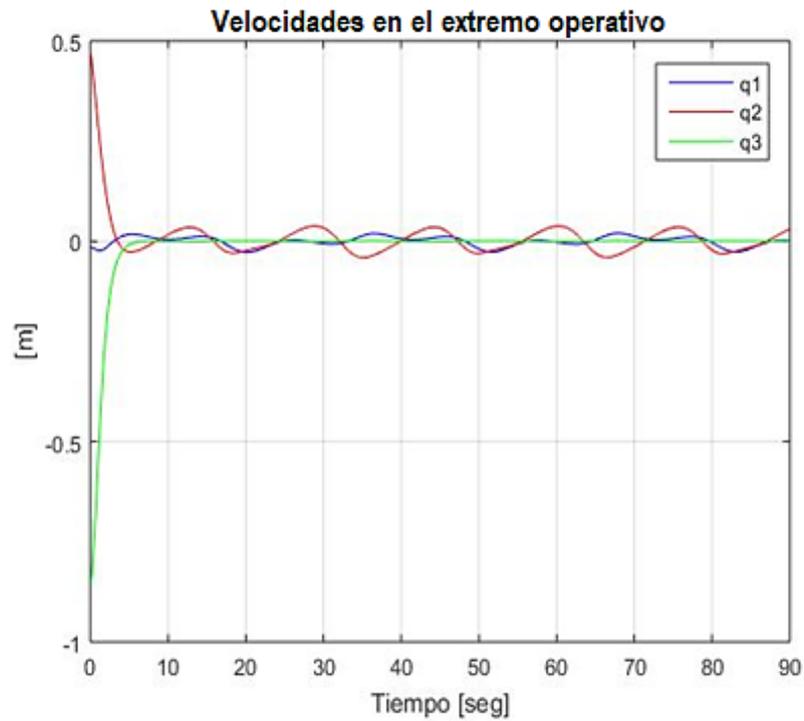
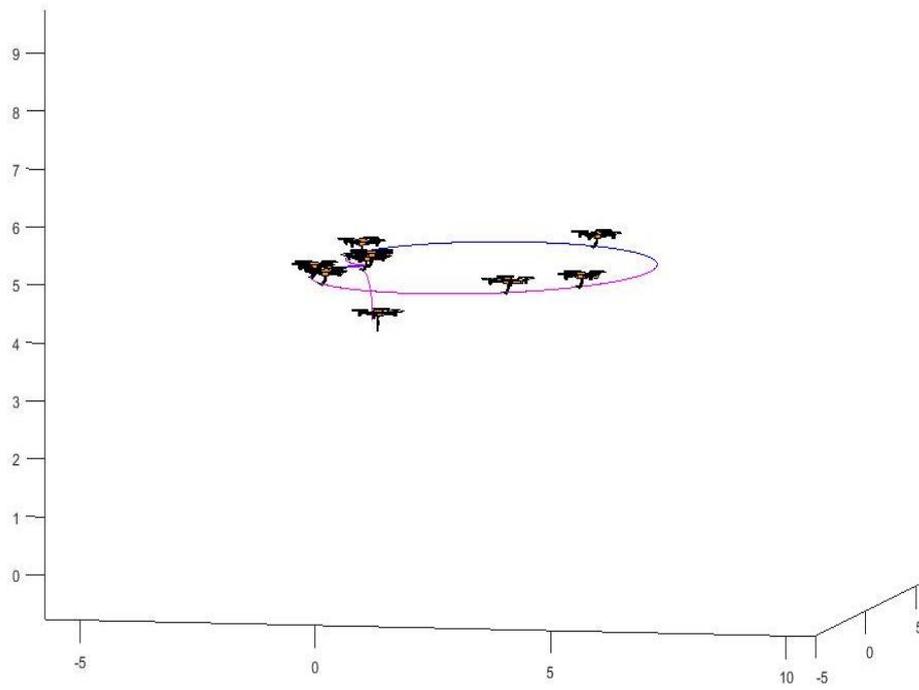


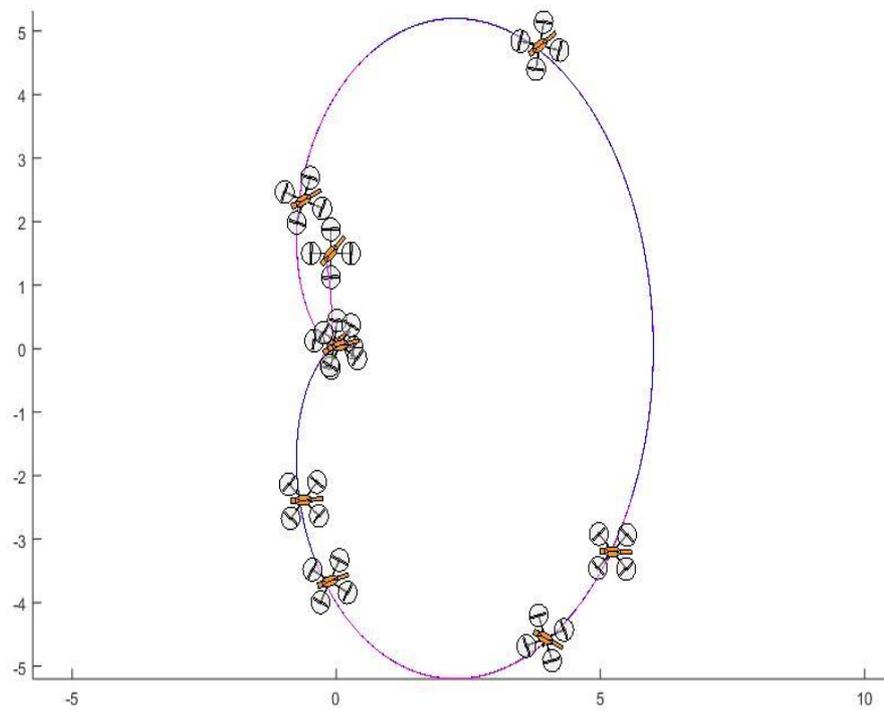
Figura. 5-17: Velocidades en el extremo operativo: Simulación 3 del seguimiento de trayectoria del robot manipulador aéreo
Fuente: Erazo A. 2019

Simulación 4

Esta simulación corresponde a la función que forma de un corazón, en las Figuras 5-18 y 5-19 se presenta el movimiento estroboscópico del robot y la evolución del seguimiento de trayectoria en el entorno virtual de Unity, respectivamente.



(a) Simulación 4 de seguimiento trayectoria vista frontal



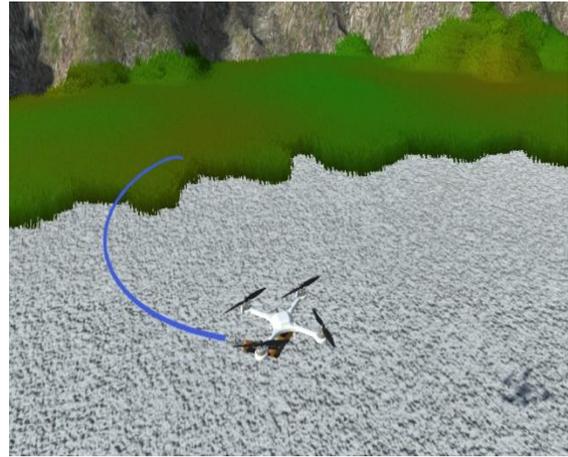
(b) Simulación 4 de seguimiento trayectoria vista superior

Figura. 5-18: Movimiento estroboscópico: Simulación 4 del seguimiento de trayectoria del robot Manipulador aéreo.

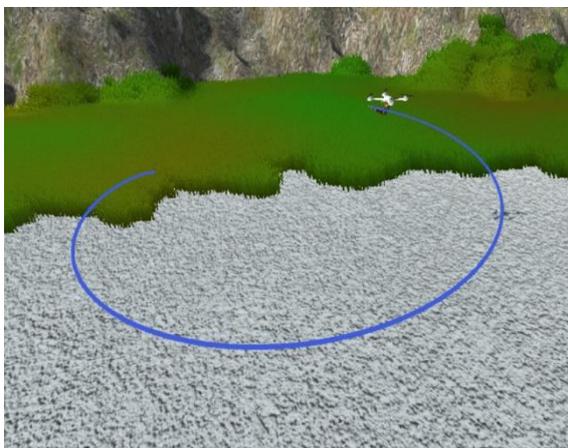
Fuente: Erazo A. 2019



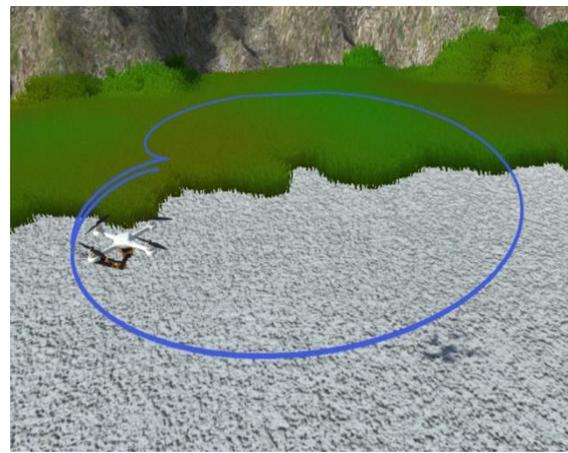
(a) Vista inicial



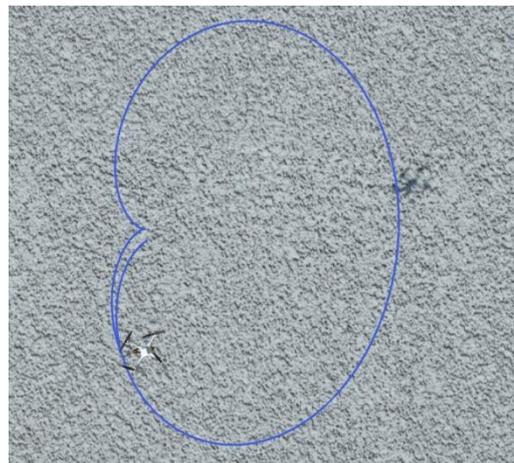
(b) Vista nivel 1



(c) Vista nivel 2



(d) Vista final



(e) Vista superior

Figura. 5-19: Entorno virtual: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo
Fuente: Erazo A. 2019

La Figura. 5-20 y 5-21, 5-22 corresponden a los errores de seguimiento y las velocidades de control, respectivamente, observando que los errores convergen a cero y las velocidades que permiten el seguimiento de la trayectoria.

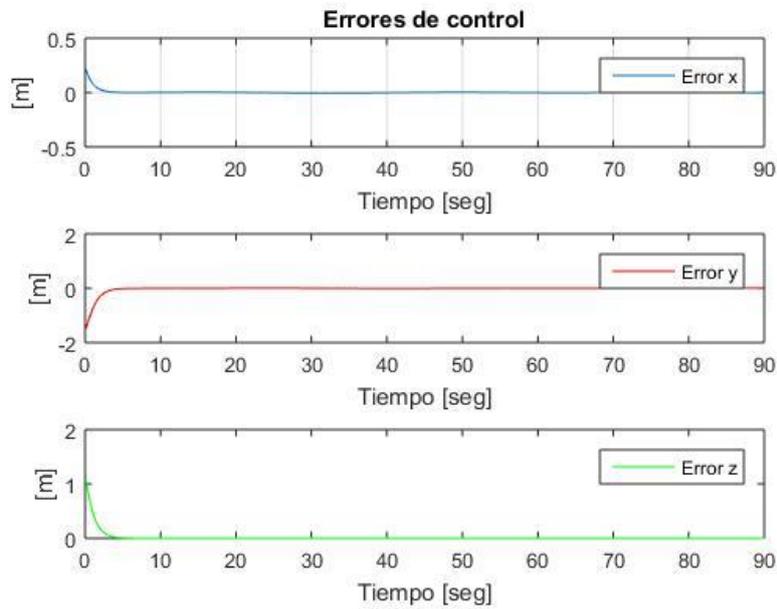


Figura. 5-20: Errores de control: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

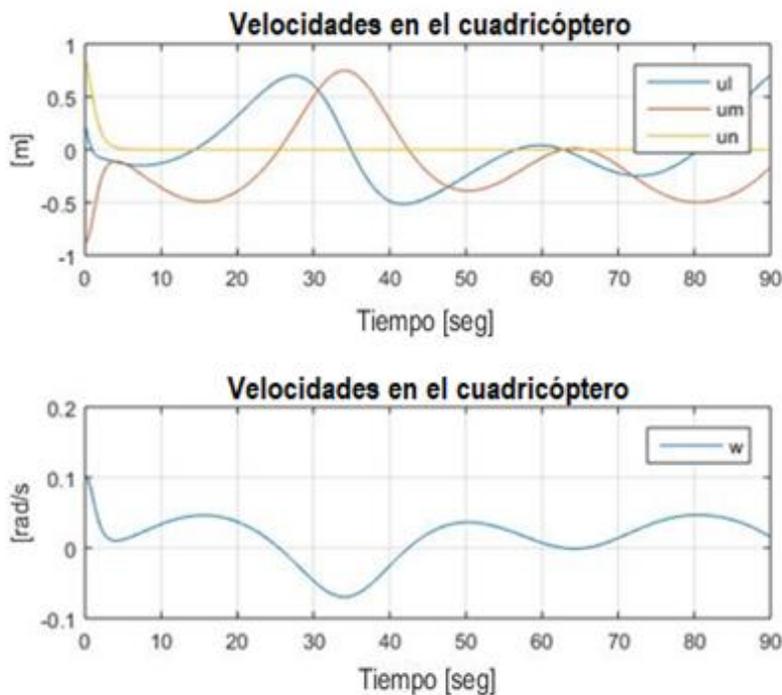


Figura. 5-21: Velocidades en el cuadricóptero: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

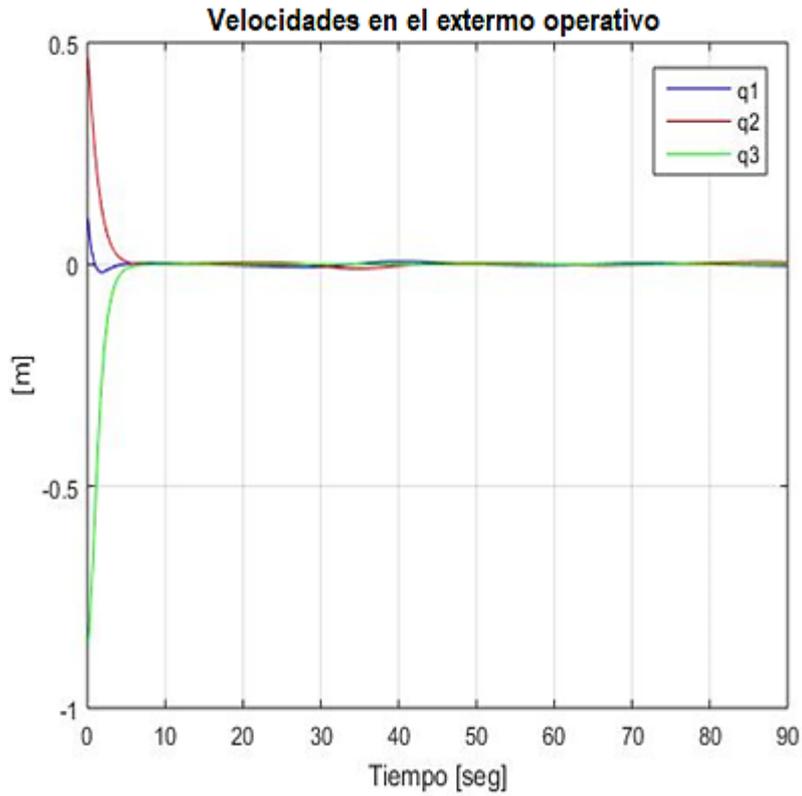
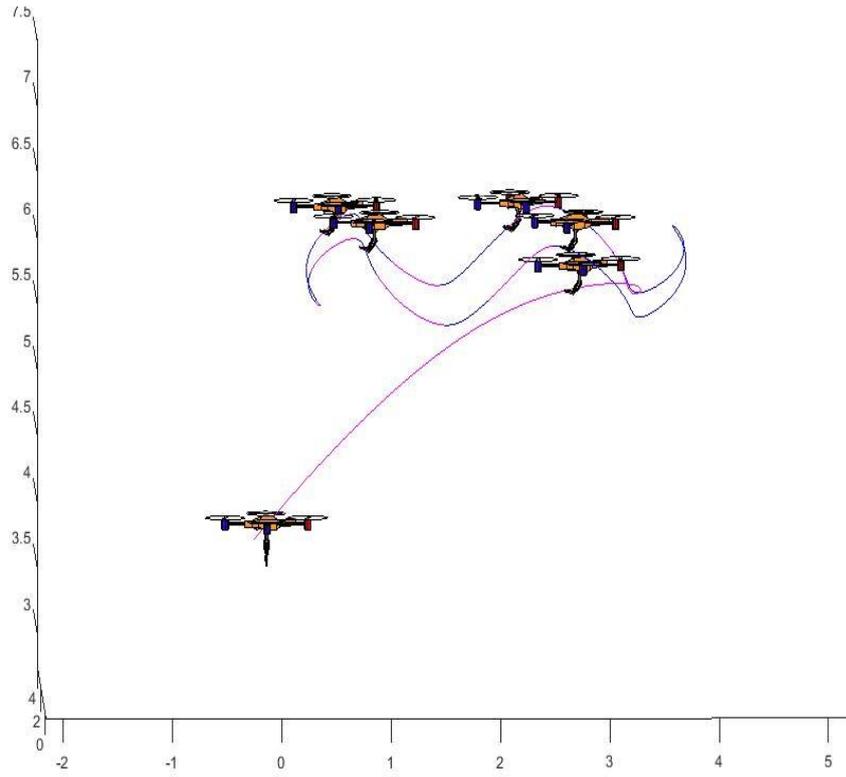


Figura. 5-22: Velocidades en el extremo operativo: Simulación 4 del seguimiento de trayectoria del robot manipulador aéreo

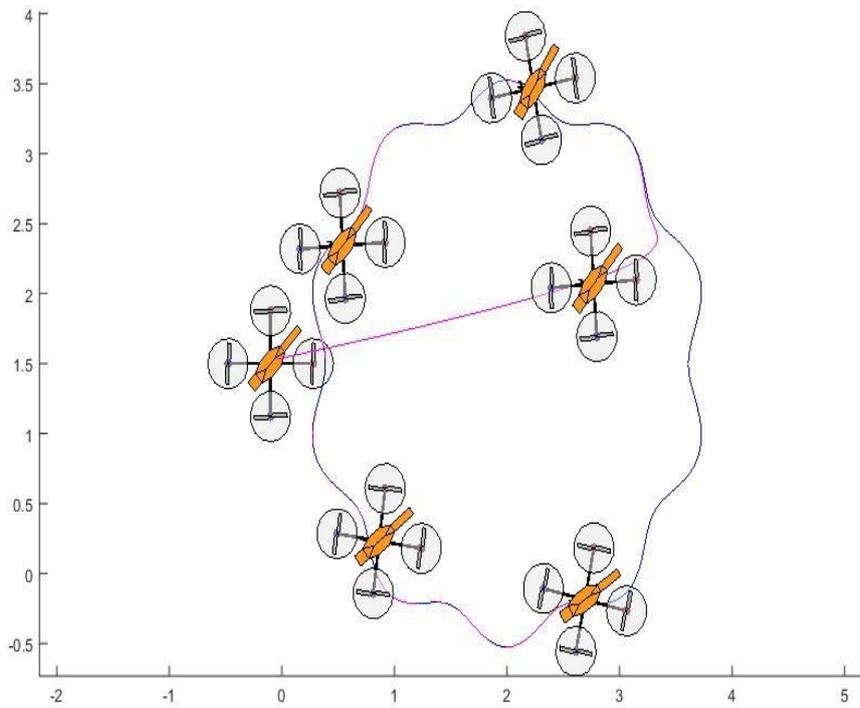
Fuente: Erazo A. 2019

Simulación 5

Esta simulación corresponde a la función que forma de una flor, en las Figura. 5-23 y 5-24 se presenta el movimiento estroboscópico del robot y la evolución del seguimiento de trayectoria en el entorno virtual de Unity, respectivamente.



(a) Simulación 5 de seguimiento trayectoria vista frontal



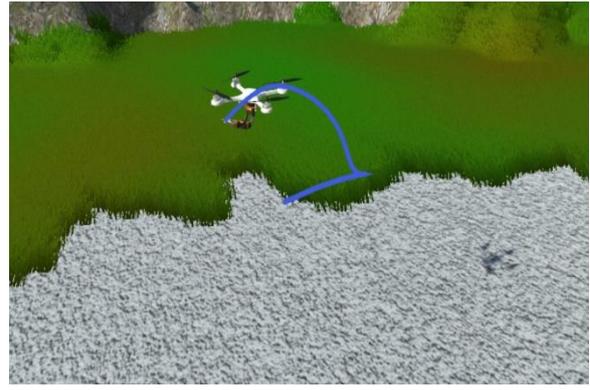
(b) Simulación 5 de seguimiento trayectoria vista superior

Figura. 5-23: Movimiento estroboscópico: Simulación 5 del seguimiento de trayectoria del robot Manipulador aéreo.

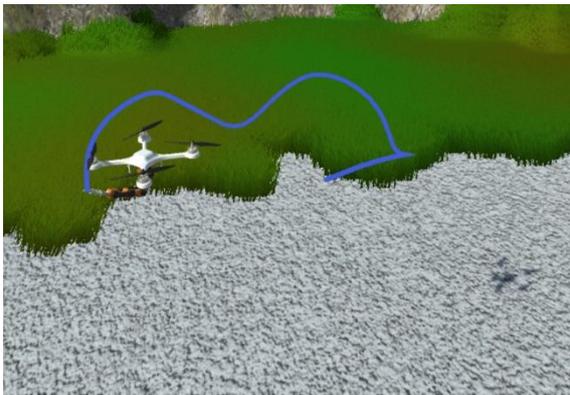
Fuente: Erazo A. 2019



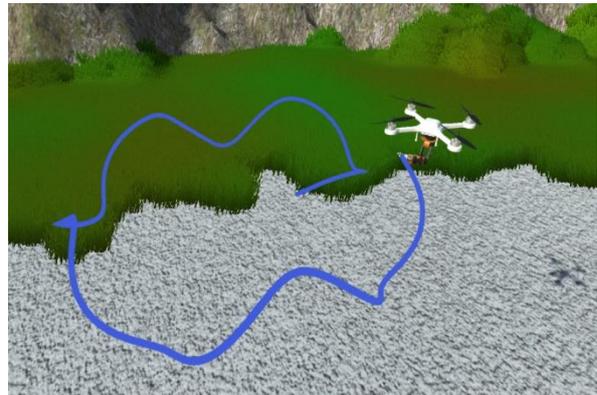
(a) Vista inicial



(b) Vista nivel 1



(c) Vista nivel 2



(d) Vista Final



(e) Vista Superior

Figura. 5-24: Entorno virtual: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo
Fuente: Erazo A. 2019

La Figura. 5-25 y 5-26, 5-27 corresponden a los errores de seguimiento y las velocidades de control, respectivamente, observando que los errores convergen a cero y las velocidades que permiten el seguimiento de la trayectoria.

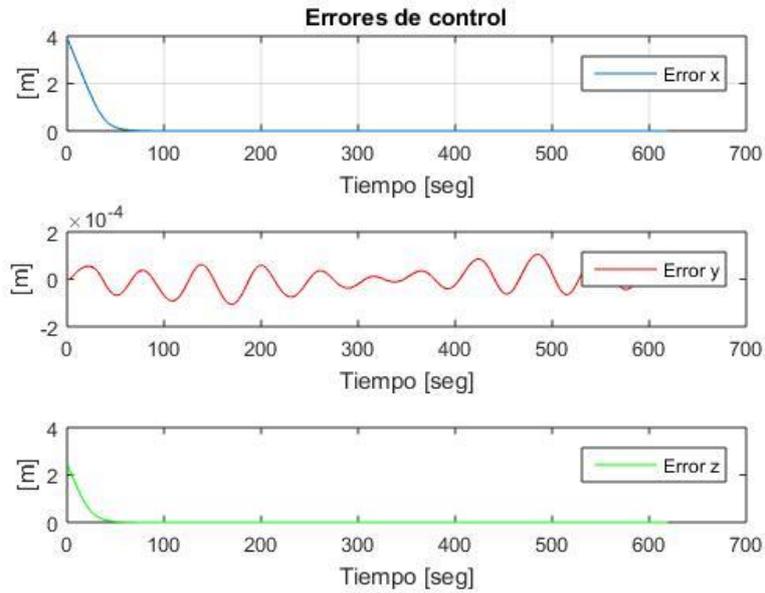


Figura. 5-25: Errores de control: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

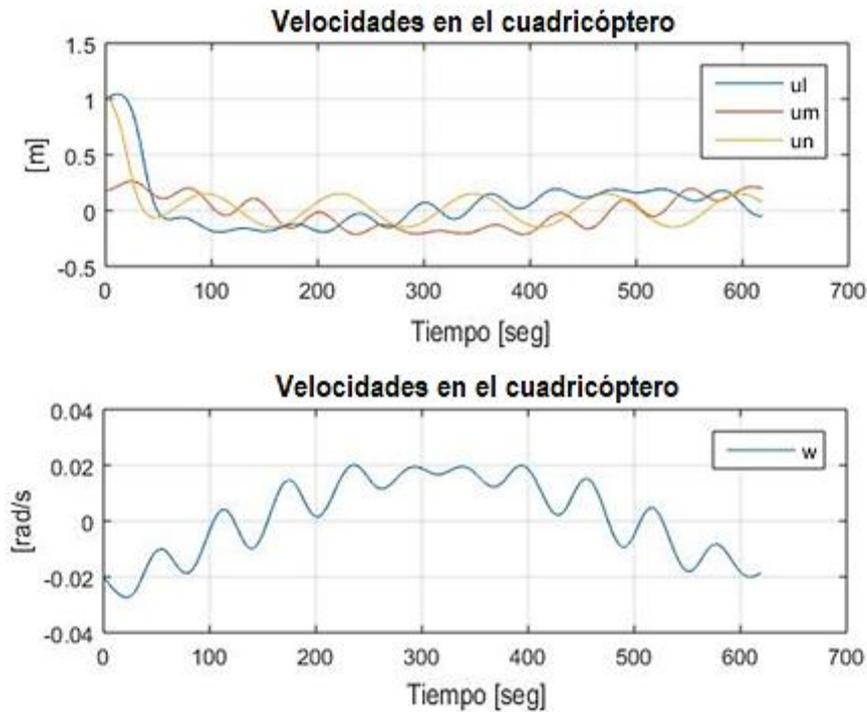


Figura. 5-26: Velocidades en el cuadricóptero: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

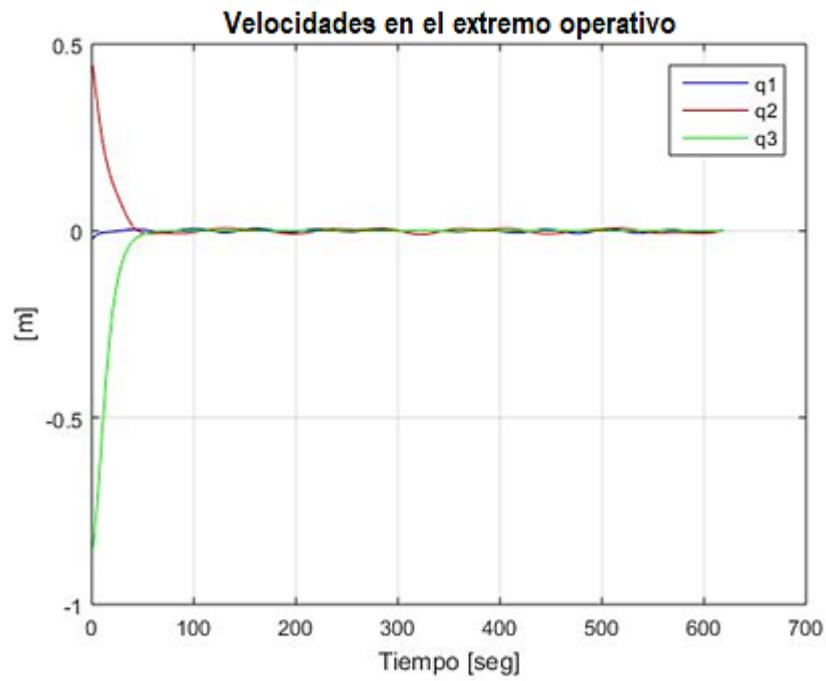


Figura. 5-27: Velocidades en el extremo operativo: Simulación 5 del seguimiento de trayectoria del robot manipulador aéreo

Fuente: Erazo A. 2019

CONCLUSIONES

1) El modelo cinemático del robot manipulador aéreo satisface las necesidades de control, las cuales posteriormente serán de gran ayuda para el diseño del controlador cinemático no lineal y su objetivo principal de ejecución de tareas de seguimiento de trayectoria en ambientes no estructurados.

2) El controlador cinemático no lineal propuesto para el robot manipulador aéreo satisface el problema de seguimiento de trayectorias en ambientes no estructurados. Su principal ventaja es la facilidad de implementación de acuerdo con el sistema de análisis propuesto y la posibilidad de ser probado analíticamente para su posterior simulación o ejecución en ambientes reales.

3) Para trabajos futuros se recomienda probar el controlador cinemático no lineal en un robot manipulador aéreo real, para constatar que cumpla con los objetivos planteados y los resultados se asemejen a los ya obtenidos en las simulaciones.

4) El entorno 3D de realidad virtual permite inmersión del operador humano en donde se puede visualizar de una manera más didáctica y casi real los resultados obtenidos del controlador creado y su estabilidad al seguir una trayectoria establecida.

5) En las simulaciones se pudo apreciar un adecuado funcionamiento del sistema al comprobar que los algoritmos de control propuestos cumplen satisfactoriamente la locomoción ante diferentes trayectorias propuestas tendiendo el error a cero.

6) La estabilidad de los controladores es garantizada por el método de Lyapunov, demostrando que los errores de posición tienden a cero a medida que el tiempo se aproxima a infinito.

RECOMENDACIONES

- 1) Se sugiere utilizar dispositivos de manipulación o motores de control gestual como leap motion u otros dispositivos, para que sea la simulación más interactiva y real ya que al trabajar con este motor de video juegos permite realizar este tipo de adaptaciones.
- 2) Se recomienda tener muy en cuenta las dimensiones y posiciones iniciales utilizadas en Maltab para que la simulación en Unity sea totalmente idéntica.
- 3) La memoria compartida envía solo valores enteros; para que no se pierdan los decimales se recomienda acondicionar los datos para el envío a Unity.
- 4) Al iniciar la simulación en el entorno virtual busca los datos almacenados en memoria, si la librería no encuentra datos, automáticamente la simulación se cerrará, se recomienda ejecutar primero MatLab para que no se genere este error.
- 5) Unity es un motor de video juegos más completa que existe, facilita el diseño de interfaz y animación con efectos casi reales, es multiplataforma por lo que no es necesario hacer ajustes para adaptar para dispositivos móviles. Se recomienda Unity como plataforma de realidad virtual por todas sus bondades que ofrece y la facilidad de comunicación con diferentes gadgets.

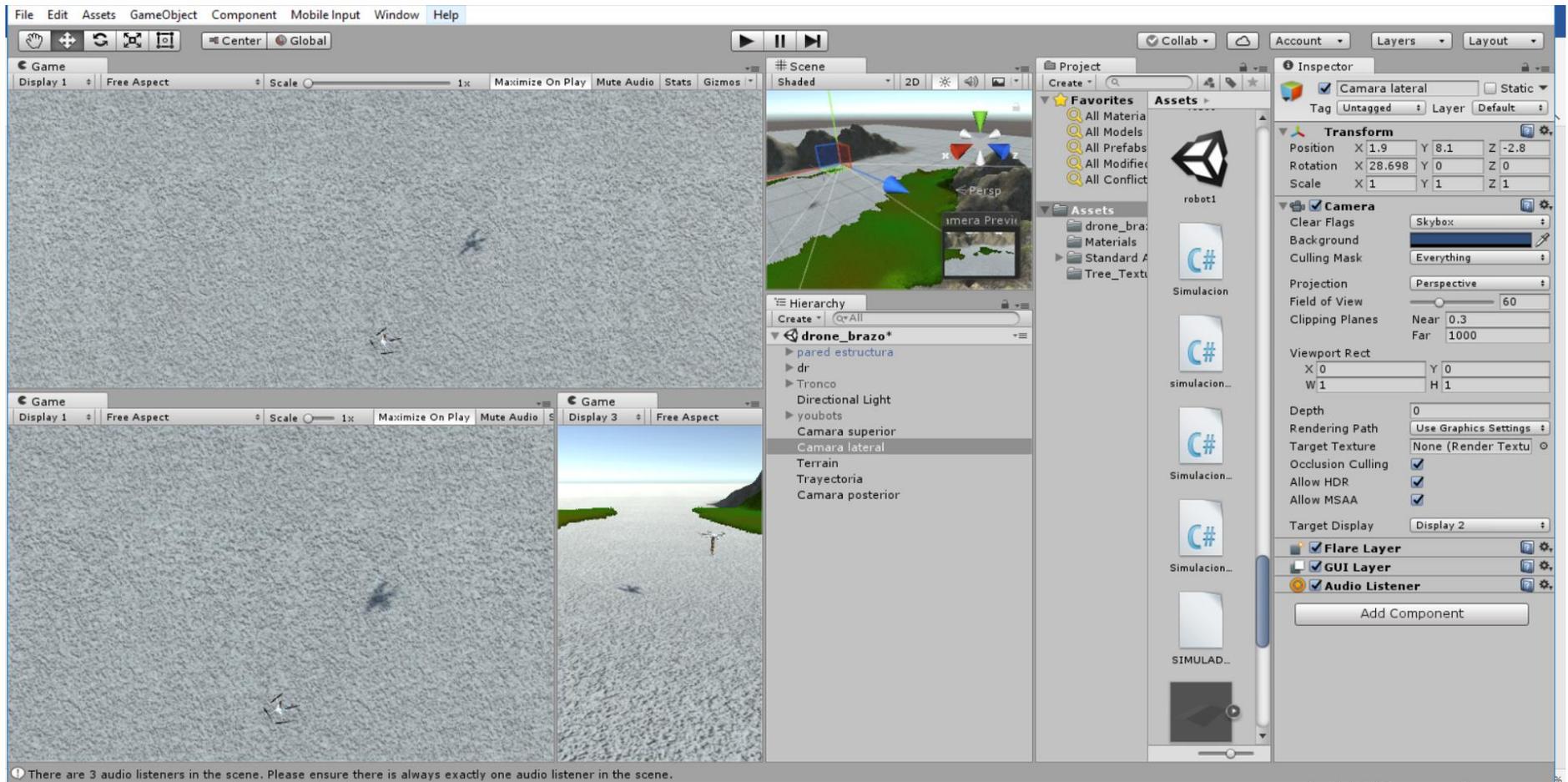
BIBLIOGRAFÍA

1. Andaluz , V., & Perez, L. (2016). *Modeling and Kinematic Nonlinear Control of Aerial Mobile Manipulators*.
2. Andaluz et al. (2015). Nonlinear Controller of Quadcopters for Agricultural Monitoring. *Advances in Visula Computing*, 9474, 476-487.
3. Andaluz et al. (2016). Unity3D-MatLab Simulator in Real Time for Robotics Applications. 50-62.
4. Andaluz, G., Andaluz, V., & Rosales, A. (2011). Modelación, Identificación y Control de Robots Móviles. *Escuela Politécnica Nacional*, 30-35.
5. Balaguer, C., Gimenez , A., & Jardon, A. (2005). Live experimentation of the service robot applications for elderly people care in home environments. *In Intelligent Robots and Systems*, 2345-2350.
6. Bermúdez, G. (2015). ROBOTS MÓVILES.TEORIA, APLICACIONES Y EXPERIENCIAS. *ResearchGate*, 80-90.
7. Burdea , G., & Coiffet, P. (1994). *Virtual reality technology*. Londres: Wiley-Interscience.
8. Fish, F. (2006). What does biomimetics have to offer to aquatic robots. *Applied Bionics and Biomechanics*, 49-60.
9. Georgiades, C., German, A., & Hogue, H. (2005). AQUA: an aquatic walking robot. *IEEE Xplore*, 3525-3531.
10. Graetz, G., & Michaels, G. (2015). Robots at work. *CEPR Discussion Paper No. DP10477*, 2-5.
11. Hongtai et al. (2013). Autonomous robot teaching using a smart robot in production line. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1772-1777.
12. Lidoris et al. (2009). The Autonomous City Explorer (ACE) project — mobile robot navigation in highly populated urban environments. *IEEE Internacional Conference on Robotics and Automation*, 1416-1422.
13. López et al. (2013). Aprendizaje de y con robótica, algunas experiencias. *Revista Educación*, 43-63.
14. López, L., & Mata, E. (2001). Estado del arte en robotica movil autonoma distribuida. *Conciencia Tecnológica*, 23-26.
15. Marcombo, A. (2014). *Marcombo*. Retrieved from http://www.marcombo.com/Descargas/9788426717450%20-%20ROB%D3TICA/EXTRACTO%20DEL%20LIBRO_ROBOTICA.pdf
16. McCarley, J., & Wickens, C. (2004). Human factors concerns in UAV flight. *University of Illinois at Urbana-Champaign*, 13-16.
17. Normalisation, A. (2018). *AFNOR Normalisation*. Retrieved from <https://normalisation.afnor.org/>

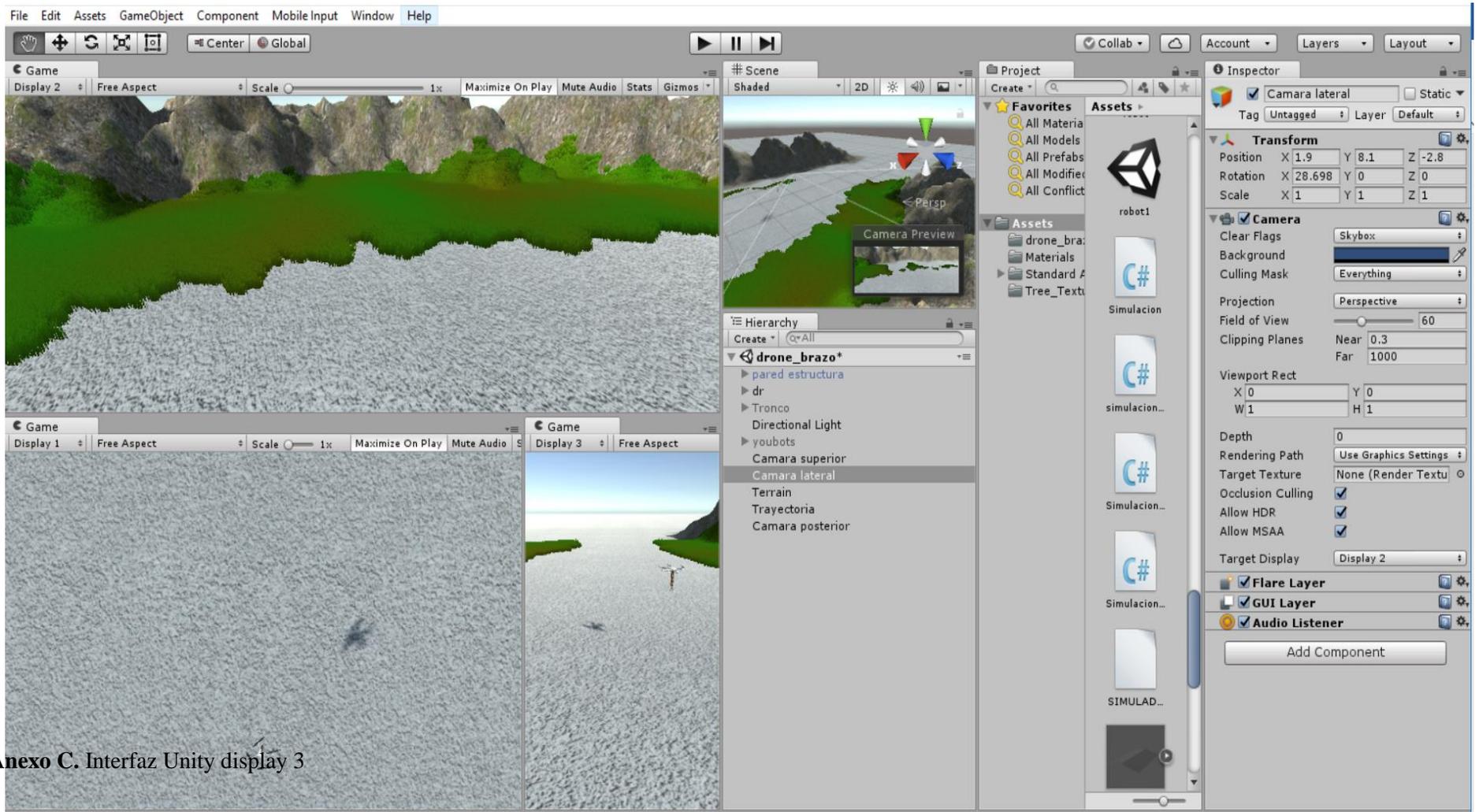
18. Orsag, M., Stjepan, K., & Yu, P. (2014). *Hybrid Adaptive Control for Aerial Manipulation*. J Intell Robot Systems.
19. Ortiz , J., & Erazo , A. (2018). Modeling and Kinematic Nonlinear Control. *Computational Kinematics*, 87-95.
20. Platea, M. (2007). *Platea.pntic.mec*. Retrieved from http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.4.htm
21. Sciavicco, L., & Siciliano, B. (2012). *Modelling and control of robot manipulators*. Londres: Springer Science & Business Media.
22. Secretaría Nacional de Planificación y Desarrollo . (2013). *Plan Nacional para el Buen Vivir 2013 - 2017*. Quito, Ecuador: Senplades.
23. Silva, R., García, J., & Bar, V. (2007). Una panorámica de los robots móviles. *Revista Electrónica de Estudios Telemáticos*, 34-37.
24. Soonkyum, K., & Fink, J. (2009). *Kinematics And Statics Of Cooperative Multi-Robot Aerial Manipulation With Cables*. ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.
25. Wiki , R. (2005). *wiki.robotica.webs.upv*. Retrieved from <http://wiki.robotica.webs.upv.es/wiki-de-robotica/introduccion/clasificacion-de-robots/>
26. Yiming, Y., Jingchao, J., & Wenping, P. (2006). Real-time Processing Research of UAV Remote Control and Test Data. *Computer Measurement & Control*, 41-46.
27. Zaloga, Z. (1997). UAV Military Future Deemed Promising. *Aviation Week and Space Technology* 13, 89-90.

ANEXOS

Anexo A. Interfaz Unity display 1



Anexo B. Interfaz Unity display 2



Anexo C. Interfaz Unity display 3



Anexo D. Interfaz Unity display 1 total



Anexo E. Interfaz Unity display 2 total



Anexo F. Interfaz Unity display 3 total

