



# **ESCUELA POLITÉCNICA NACIONAL**



## **FACULTAD DE INGENIERÍA MECÁNICA**

### **VALIDACIÓN DE LA CINEMÁTICA DIRECTA DEL ROBOT CARTESIANO CLASIFICADOR DE OBJETOS FERROMAGNÉTICOS POR PESO, MEDIANTE UN SISTEMA DE ANÁLISIS AUTOMÁTICO Y EN TIEMPO REAL DE MOVIMIENTOS POR FOTOGRAMETRÍA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO MECÁNICO**

**JOHN ALEXIS BARRIONUEVO FREIRE**

**john.barrionuevo@epn.edu.ec**

**NELSON ALEXANDER GALLO CHASI**

**nelson.gallo01@epn.edu.ec**

**DIRECTOR: OSCAR IVÁN ZAMBRANO, M.Sc.**

**ivan.zambrano@epn.edu.ec**

**CO-DIRECTORA: DAYSI ALEXANDRA BAÑO, Mgtr.**

**dbano@ups.edu.ec**

**Quito, 2020**

## **CERTIFICACIÓN**

Certificamos que el presente trabajo fue desarrollado por los señores John Alexis Barrionuevo Freire y Nelson Alexander Gallo Chasi bajo nuestra supervisión.

---

Ing. Oscar Iván Zambrano. M.Sc.  
Director del Proyecto de Titulación

---

Ing. Daysi Alexandra Baño. Mgtr.  
Co-Directora del Proyecto de Titulación

## DECLARACIÓN

Nosotros, John Alexis Barrionuevo Freire y Nelson Alexander Gallo Chasi, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado previamente para ningún grado; y que hemos consultado las referencias bibliográficas incluidas en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos inherentes a este trabajo; según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

John Alexis Barrionuevo Freire

---

Nelson Alexander Gallo Chasi

## DEDICATORIA

A mis padres, José y María, quienes me acogieron desde el nacimiento, me enseñaron el valor de la vida, me enseñaron a siempre luchar por los sueños, ser humilde y respetuoso, gracias por brindarme tanto amor.

A mis hermanos, Washington, Norma, Milton, Javier, quienes estuvieron en cada etapa de mi vida apoyándome incondicionalmente, este peldaño de mi vida es por ustedes.

A mi angelito que está en el cielo mi mamá Lucila, aunque no te conocí físicamente sé que siempre me acompañas en mi corazón, me diste la vida para vivirla con una gran familia a la que amo con todo mi corazón, esto es por ti mami.

John Barrionuevo

## DEDICATORIA

Dedico el presente trabajo a mi padre Nelson Germánico, quien supo implantar en mí su ideología. Tú me pusiste en la competencia y me enseñaste a ser un ganador, además me ayudas a despejar las incógnitas de la vida como si se tratase de una ecuación. Todos mis logros desde mi niñez hasta la presente fecha son tuyos.

Para ti madre Laura, por tus atenciones y correcciones a lo largo del camino, y por enseñarme a ser una persona con buenos valores y sentimientos, siempre tengo presente tu palabra favorita hacia mí, "Campeón".

Para mi hermana Estéfani, mi pequeña amiga ver la manera como luchas por tus sueños me motivan a conseguir los míos. Tienes mi apoyo incondicional siempre.

Para mis abuelitos Julio y Zoila, por todo su amor fraternal. Quiero que sepan que no han arado en el mar.

Para mis tíos Luis, Segundo, José, Ignacio, Héctor, Julio y Cristóbal, por brindarme su cariño, enseñarme a jugar fútbol y ser mis amigos. A mis tías Blanca, Gladys, Isabel, Sandra y Amparo por abrirme las puertas de sus casas y tratarme como a un hijo.

Para mis primos Marco, Cristiann, Carolina, Katherine, Heidy, Karla, Shirley, Daniela, Danilo, Bryan, Majo, Giancarlo y en especial para William, Diego y Christopher, por permitirme ser parte de sus vidas.

Para todas mis amistades que me han apoyado en mi trayecto colegial y universitario, para todas las personas que me han ayudado con un consejo o una palabra de aliento y para la estrella que desde el cielo me alumbra, te amaré eternamente 8.

Nelson A. Gallo

## **AGRADECIMIENTO**

A mi mamá, María, por aconsejarme en los momentos más difíciles de mi vida, por ser la persona que me enseñó a ser luchador.

A mi papá, José, por enseñarme a ser humilde en las victorias, por ese apoyo incondicional en los proyectos de vida que me propuse.

A mi hermana, Norma, por ser la persona que siempre me alienta a ser mejor cada día y ser mi confidente.

A mi hermano, Washington, quien me enseñó lo duro de la vida profesional y que siempre hay que ser luchador.

A mi hermano, Milton, quien me enseñó que por más duro que nos golpee la vida siempre hay que levantarse y seguir para adelante.

A mi hermano, Javier, por aconsejarme en los buenos y malos momentos, por apoyarme en las locuras.

A mis amigos, Diego y Alexander, son como mis hermanos, crecimos juntos jugando fútbol en la cancha del barrio y me brindaron su apoyo incondicional.

A mi amigo, Nelson, con quien compartí toda la vida universitaria, me enseñó el valor de la lealtad.

A los Ingenieros Iván Zambrano y Daysi Baño quienes me guiaron en la elaboración del presente trabajo.

**Gracias**

John Barrionuevo

## AGRADECIMIENTO

A Dios, por la existencia.

A mi padre Nelson, por ser mi director técnico, mi docente y el motivo de mi amor por las matemáticas.

A mi madre Laura, por ser mi confidente y haberme dado muestras incansables de afecto y cariño durante toda mi vida.

A mi hermana Estéfani, por la mesura que me transmite en momentos difíciles.

A mi amado colegio Juan Pío Montúfar, por darme las herramientas necesarias para defenderme en mi vida universitaria.

A mis tíos Luis y Gladys y a mis primos William y Carolina, por el apoyo y la guía que me han brindado en mi formación personal y profesional.

A John, por su amistad, compañerismo y dedicación en la elaboración de este trabajo de titulación.

A todos los profesores de la Facultad de Ingeniería Mecánica, en especial al Ing. Iván Zambrano y a la Ingeniera Daysi Baño por la guía y tiempo prestado para la consecución de este proyecto.

Nelson A. Gallo

# ÍNDICE

INTRODUCCIÓN.....	1
Objetivo general.....	3
Objetivos específicos.....	3
1 MARCO TEÓRICO .....	4
1.1 Robot cartesiano clasificador de objetos ferromagnéticos por peso.....	5
1.2 Matlab.....	6
1.2.1 Robotics Toolbox for Matlab.....	6
1.2.2 Comandos del Robotics Toolbox usados en el modelado .....	7
1.3 MSC Adams.....	9
1.3.1 Justificación .....	9
1.3.2 Fundamento.....	9
1.3.3 Enfoque de MDB .....	9
1.3.4 Adams/View.....	10
1.4 Fotogrametría.....	11
1.4.1 Definición .....	11
1.4.2 Fundamento.....	12
1.4.3 Aplicaciones.....	13
1.4.4 Tecnologías cinemáticas de análisis de movimientos.....	13
1.4.5 Sistema de análisis de movimiento Kinescan/IBV.....	14
1.5 Cinemática del robot.....	18
1.5.1 Obtención de los parámetros de Denavit-Hartenberg para el robot cartesiano .....	20
1.5.2 Cuaternión unitario y cuaternión dual .....	22
1.5.3 Cinemática diferencial .....	25
1.5.4 Cinemática diferencial del robot cartesiano .....	29
1.6 Métodos de validación .....	37
1.6.1 Cálculo del error .....	38
2 METODOLOGÍA.....	40
2.1 Modelo matemático en Matlab .....	40
2.2 Simulación en MSC Adams/View .....	46
2.3 Captura del movimiento mediante el sistema de análisis de Kinescan/IBV (Fotogrametría) .....	51
2.4 Comparativa de resultados.....	59
3 RESULTADOS Y DISCUSIÓN .....	60
3.1 Resultados .....	60



3.1.1	Desplazamiento del efector final .....	60
3.1.2	Velocidad del efector final .....	61
3.1.3	Aceleración del efector final .....	61
3.1.4	Comparación Adams-Kinescan/IBV .....	62
3.1.5	Comparación Matlab-Kinescan/IBV .....	67
3.2	Discusión.....	72
3.2.1	Adams-Fotogrametría.....	72
3.2.2	Matlab-Fotogrametría .....	73
4.	CONCLUSIONES.....	75
	Referencias Bibliográficas.....	76
	Anexos.....	78

## ÍNDICE DE FIGURAS

Figura i. Manipulador cartesiano.....	1
Figura ii. Esquema del robot cartesiano.....	2
Figura 1.1 Disposición de ejes del robot cartesiano.....	5
Figura 1.2 Ejemplo de codificación del comando Link.....	7
Figura 1.3 Ejemplo de aplicación del comando SerialLink.....	7
Figura 1.4 Aplicación de la cinemática directa.....	9
Figura 1.5. Pantalla principal de Adams.....	11
Figura 1.6 Esquema del principio fundamental usado en la fotogrametría.....	13
Figura 1.7 Aplicaciones de la fotogrametría.....	13
Figura 1.8 Uso del sistema de seguimiento MoCap en un golfista.....	14
Figura 1.9 Principio de acción de una cámara digital.....	15
Figura 1.10 Tecnología PoE.....	17
Figura 1.11 Varilla de calibración MoCap.....	18
Figura 1.12 Representación de una junta de revolución y prismática.....	20
Figura 1.13 Notación D-H en las articulaciones del robot cartesiano.....	21
Figura 2.1 Diagrama de flujo para el análisis en Matlab.....	40
Figura 2.2 Current Folder.....	41
Figura 2.3 Activación de la librería.....	41
Figura 2.4 Definición de variables articulares.....	41
Figura 2.5 Tabla D-H del robot.....	42
Figura 2.6 Función fkine.....	42
Figura 2.7 Definición de variables articulares.....	42
Figura 2.8 Generación de la trayectoria.....	43
Figura 2.9 Función endEffectorVelocityDQ.....	43
Figura 2.10 Cálculo de la velocidad.....	44
Figura 2.11 Función endEffectorAccelerationDQ.....	44
Figura 2.12 Cálculo de la aceleración.....	45
Figura 2.13 Creación de las gráficas de desplazamiento.....	45
Figura 2.14 Gráfica de desplazamiento.....	45
Figura 2.15 Diagrama de flujo para la simulación en MSC Adams.....	46
Figura 2.16 Modelo CAD del robot en SolidWorks.....	47
Figura 2.17 Modelo CAD del robot en Adams/View.....	48
Figura 2.18 Movimientos del robot en Adams/View.....	48
Figura 2.19 Mediciones del efector final en Adams/View.....	49
Figura 2.20 Simulación del robot en Adams/View.....	49
Figura 2.21 Gráficas del efector final en Adams/View.....	50
Figura 2.22 Exportación de datos a Excel.....	50
Figura 2.23 Selección de gráficas a exportar.....	51
Figura 2.24 Diagrama de flujo para la captura de movimiento por fotogrametría.....	52
Figura 2.25 Verificación de la Calibración.....	52
Figura 2.26 Grabación con el puntero posicionado.....	53
Figura 2.27 Verificación de la Calibración.....	53
Figura 2.28 Marcador móvil (izquierda) y marcador fijo (derecha).....	54
Figura 2.29 Nombre del sujeto o prueba a realizarse.....	54
Figura 2.30 Adición de estudio P1 en el Gestor de Estudios.....	55
Figura 2.31 Creación del segmento del modelo.....	55

Figura 2.32 Definición de los puntos del segmento.....	56
Figura 2.33 Definición de la variable posición en el eje x para el marcador m1.....	56
Figura 2.34 Asignación de la cinemática para cada punto.....	57
Figura 2.35 Variables de m1 y m2 para definir la posición. ....	57
Figura 2.36 Interfaz de grabación de escena. ....	58
Figura 2.37 Gráficas de las variables en función del tiempo.....	58
Figura 2.38 Exportación de Variables. ....	59
Figura 2.39 Cálculo del error.....	59
Figura 3.1 Desplazamientos para los tres análisis en los ejes coordenados. ....	60
Figura 3.2 Velocidades para los tres análisis en los ejes coordenados. ....	61
Figura 3.3 Aceleraciones para los tres análisis en los ejes coordenados. ....	62
Figura 3.4 Desplazamiento en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo) .	62
Figura 3.5 Desplazamiento en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo) .	63
Figura 3.6 Desplazamiento en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo) .	63
Figura 3.7 Velocidad en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo).....	64
Figura 3.8 Velocidad en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo).....	64
Figura 3.9 Velocidad en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo).....	65
Figura 3.10 Aceleración en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo).....	65
Figura 3.11 Aceleración en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo).....	66
Figura 3.12 Aceleración en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo).....	66
Figura 3.13 Desplazamiento en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo)	67
Figura 3.14 Desplazamiento en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo)	68
Figura 3.15 Desplazamiento en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo)	68
Figura 3.16 Velocidad en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	69
Figura 3.17 Velocidad en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	69
Figura 3.18 Velocidad en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	70
Figura 3.19 Aceleración en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	70
Figura 3.20 Aceleración en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	71
Figura 3.21 Aceleración en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo) .....	71
Figura 3.22 Error porcentual en el eje Y para el comparativo Adams-Fotogrametría .	73
Figura 3.23 Error porcentual en el eje z para el comparativo Adams-Fotogrametría..	73
Figura 3.24 Error porcentual en el eje Y para el comparativo Matlab-Fotogrametría .	74
Figura 3.25 Error porcentual en el eje Y para el comparativo Matlab-Fotogrametría .	74

## ÍNDICE DE TABLAS

Tabla 1.1 Especificaciones técnicas para el eje x y el eje y del robot cartesiano. ....	6
Tabla 1.2 Especificaciones técnicas para el eje z del robot cartesiano.....	6
Tabla 1.3 Opciones de visualización del área gráfica del robot.....	8
Tabla 1.4 Elementos físicos conformantes del sistema Kinescan/IBV. ....	16
Tabla 1.5 Requerimientos del Equipo. ....	17
Tabla 1.6 Parámetros D-H para el robot cartesiano.....	20
Tabla 3.1 Error cuadrático medio entre Adams y Fotogrametría.....	67
Tabla 3.2 Error cuadrático medio entre Matlab y Fotogrametría. ....	72

## RESUMEN

El presente trabajo de titulación realiza un análisis integral de la cinemática directa del robot cartesiano clasificador de objetos ferromagnéticos por peso, de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional mediante tres metodologías de obtención de curvas de posición, velocidad y aceleración en función del tiempo. La primera metodología consistió en la creación del modelo matemático de los movimientos que presenta el robot cartesiano, mediante un código en Matlab fundamentado en las ecuaciones cinemáticas de posición, velocidad y aceleración obtenidas a partir de cuaterniones duales. En la segunda metodología se recurrió a una simulación del modelo CAD del robot en MSC Adams, y se sometió a un análisis dinámico multicuerpo para obtener las curvas cinemáticas. Para la tercera metodología se obtuvieron curvas bidimensionales de la cinemática mediante un sistema de captura de movimientos (Kinescan/IBV) empleando los principios de la fotogrametría, dicha captura se produjo mediante cámaras de grabación capaces de digitalizar imágenes en tiempo real. Se realizó un análisis comparativo para demostrar la validez del modelo matemático, mediante un contraste entre las gráficas obtenidas en Matlab, Adams y el sistema Kinescan/IBV; el error cuadrático medio fue el criterio de validación empleado, de esta manera se generaron curvas de error a lo largo de toda la trayectoria para cada variable (posición, velocidad y aceleración) en los tres ejes respectivos. Los resultados evidenciaron errores menores en comparación a los trabajos de Pulloquina y Misyurin que confirmaron la validez del modelo matemático y se demostró la versatilidad del sistema Kinescan/IBV en cuanto a tiempo, didáctica y tratamiento de datos.

**Palabras Clave:** modelo matemático, fotogrametría, robot cartesiano, cinemática directa.

## ABSTRACT

The present degree work makes an integral analysis of the direct kinematics of the Cartesian robot classifier of ferromagnetic objects by weight, of the Facultad de Ingeniería Mecánica of the Escuela Politécnica Nacional by means of three methodologies to obtain curves of position, velocity and acceleration as a function of time. The first methodology consisted in the creation of the mathematical model of the movements presented by the Cartesian robot, through a Matlab code based on the kinematic equations of position, speed and acceleration obtained from dual quaternions. In the second methodology it was resorted to a simulation of the CAD model of the robot in MSC Adams, and it was submitted to a dynamic analysis multibody to obtain the kinematic curves. For the third methodology, two-dimensional curves of the kinematics were obtained by means of a motion capture system (Kinescan/IBV) using the principles of photogrammetry. This capture was produced by means of recording cameras capable of digitizing images in real time. A comparative analysis was carried out to demonstrate the validity of the mathematical model by means of a contrast between the graphics obtained in Matlab, Adams and the Kinescan/IBV system; the mean square error was the validation criterion used. In this way, error curves were generated along the entire trajectory for each variable (position, velocity and acceleration) on the three respective axes. The results showed minor errors in comparison to the works of Pulloquina and Misyurin, which confirmed the validity of the mathematical model and demonstrated the versatility of the Kinescan/IBV system in terms of time, didactics and data processing.

**Keywords:** mathematical model, photogrammetry, Cartesian robot, direct kinematic.

# VALIDACIÓN DE LA CINEMÁTICA DIRECTA DEL ROBOT CARTESIANO (RCO), MEDIANTE UN SISTEMA DE ANÁLISIS AUTOMÁTICO Y EN TIEMPO REAL DE MOVIMIENTOS POR FOTOGRAMETRÍA.

## INTRODUCCIÓN

La descripción idealizada de un determinado fenómeno requiere de la formulación de un modelo matemático, este proceso de plasmar las ideas en ecuaciones no es necesariamente lineal y sencillo y a menudo se puede caer en ciertas contradicciones por tratar de englobar todos los eventos conformantes del fenómeno.

En el ámbito de la robótica se trata de solventar la multifuncionalidad de un fenómeno mediante la convergencia de todos sus subsistemas para que cumpla determinado fin, y según el Instituto de Robots de América (*RIA* por sus siglas en inglés), un robot es un manipulador reprogramable diseñado para mover materiales, piezas o herramientas mediante movimientos variables programados para cumplir dicha tarea (Spong, 2006).

Dicho esto, se selecciona como objeto de estudio el robot cartesiano clasificador de objetos ferromagnéticos de la Facultad de Ingeniería Mecánica (FIM) de la Escuela Politécnica Nacional (EPN). Este robot tiene tres articulaciones de tipo prismático (PPP) (ver Figura i). Las variables de las articulaciones son las coordenadas cartesianas del efector final respecto a la base, lo cual conlleva una simplicidad en su descripción cinemática.

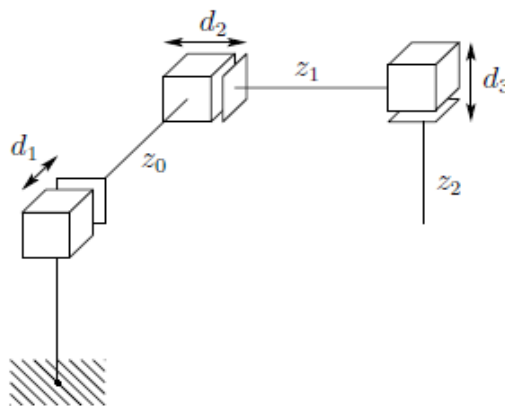


Figura i. Manipulador cartesiano.  
(Fuente: Spong, 2006)

Esta simplicidad permite enfocarse en el análisis de la cinemática del robot, el método para definir completamente su geometría espacial y el establecimiento del modelo matemático que represente fehacientemente la realidad; para tal cometido se usan tres

enfoques diferentes de análisis: Matlab, Adams y el sistema Kinescan/IBV de Fotogrametría, lo cual ofrece un mayor espectro de discernimiento al momento de validar las ecuaciones que se obtengan.

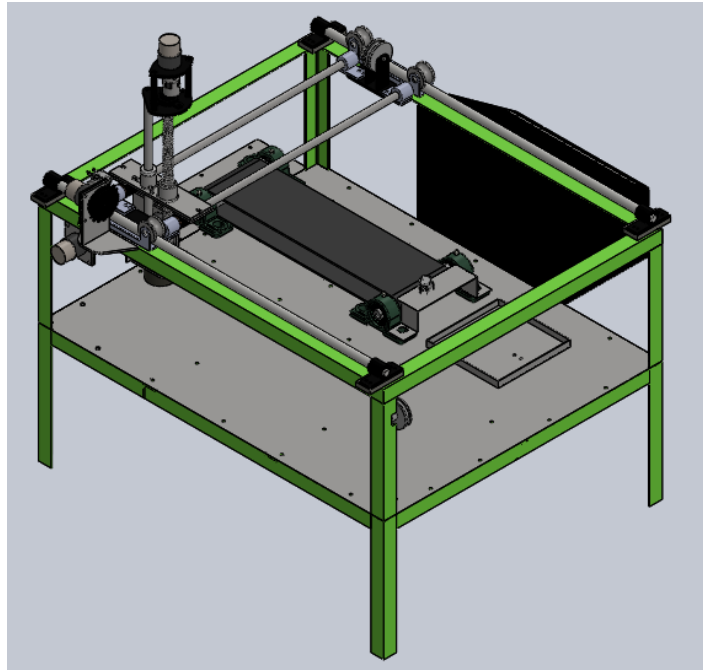


Figura ii. Esquema del robot cartesiano  
(Fuente: Cazco, 2010)

El software Matlab se usa como un método tradicional, ya que es un lenguaje con una interfaz amigable, destinado a fines didácticos y de investigación, más específicamente se recurre al Robotics Toolbox de Peter Corke que proporciona funciones relacionadas a la cinemática, didáctica y generación de trayectorias que son temas útiles en la robótica (Corke, 2017).

Por otra parte, el software MSC Adams es un simulador de dinámica multicuerpo que optimiza prototipos reales virtuales mediante la variación del diseño. El análisis comienza a partir del número de cuerpos en el sistema mecánico que está siendo modelado, hasta la generación de las ecuaciones matemáticas descriptivas del sistema mecánico. El software reduce el número de prototipos físicos usados, aumenta la calidad de diseño y disminuye el tiempo de desarrollo, mejorando así el nivel de productividad de un proyecto (Hernández, 2011).

Otra opción de validación es el sistema de análisis de movimientos en tiempo real y de forma automática mediante fotogrametría llamado Kinescan/IBV que se encuentra en el laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica (FIM). Está basado



en la tecnología video digital con preprocesamiento, lo cual permite calcular las posiciones de los segmentos corporales de forma automática a través de datos relativos a las posiciones de puntos y articulaciones, así como las variables cinemáticas y cinéticas derivadas inmediatamente después de la realización del gesto.

Es así como, además de los análisis mediante Adams y Matlab se incluye el análisis del robot cartesiano mediante fotogrametría como un análisis preliminar y experimental que sirva como un precedente para el naciente laboratorio en posteriores proyectos enfocados en la biomecánica.

### **Objetivo General**

Validar la cinemática directa del robot cartesiano clasificador de objetos ferromagnéticos por peso, mediante un sistema de análisis automático y en tiempo real de movimientos por fotogrametría.

### **Objetivos Específicos**

- Generar un código en Matlab que permita conocer la cinemática directa del robot cartesiano.
- Realizar una simulación del funcionamiento del robot cartesiano en el software Adams.
- Validar el modelo matemático del robot cartesiano mediante el equipo de fotogrametría.
- Realizar un análisis comparativo que englobe las tres alternativas de solución del modelo matemático del robot cartesiano y genere un porcentaje de aceptabilidad de cada alternativa.

# 1 MARCO TEÓRICO

La validación de un sistema de ecuaciones que represente a un fenómeno requiere de un correcto tratamiento de los eventos que engloban a dicho fenómeno y de las herramientas que se van a usar para la construcción del sistema. Es importante optimizar todo lo relacionado al proceso de búsqueda del modelo matemático, el cual debe ser balanceado, sin caer en exceso ni escasez de argumentos.

## Generalidades

El establecimiento del modelo matemático gobernante en el funcionamiento del robot cartesiano manipulador de objetos requiere de la confluencia de varios de sus sistemas, para tal caso se debe conocer morfológicamente al robot, entendiéndose esto como un análisis estructural que defina sus limitaciones físicas y permita establecer un control cinemático y dinámico adecuado.

## Modelo matemático de un robot

El diseño y control de un robot precisa el cálculo de varios modelos matemáticos, tal es el caso de:

- Modelos de transformación entre el espacio articular (en el que se define la configuración del robot) y el espacio de trabajo (en el que se especifica la ubicación del efector final). Los modelos de transformación son muy importantes ya que los robots se controlan en el espacio conjunto, mientras que las tareas se definen en el espacio de trabajo. Dos clases de modelos son considerados:
  - Modelos geométricos directos e inversos, que proporcionan la ubicación del efector final como una función de las variables articulares del mecanismo y viceversa.
  - Modelos cinemáticos directos e inversos que dan la velocidad del efector final en función de las velocidades de las articulaciones y viceversa.
- Modelos dinámicos que proporcionan las relaciones entre los pares de entrada o fuerzas de los actuadores y las posiciones, velocidades y aceleraciones de las articulaciones (Cardoso, 2011).

## 1.1 Robot cartesiano clasificador de objetos ferromagnéticos por peso

El robot cartesiano que se escoge como objeto de estudio solventa conocimientos iniciales referentes a la automatización de procesos y sienta un precedente para futuras aplicaciones industriales. La incorporación de un sistema de mando manual optimiza el control del movimiento y le da al robot un amplio espectro de análisis cinemático (Narváez, 2019).

Una correcta formulación de la casa de la calidad permite al diseñador escoger y discernir las especificaciones técnicas del robot cartesiano y también definir la funcionalidad de la máquina a través de divisiones modulares; la elección de cada módulo se realiza mediante evaluación de alternativas respecto a criterios predefinidos (diseño, economía, mantenimiento, adaptabilidad) (Riba, 2002).

La disposición de los ejes y la inclusión de un tornillo de potencia y un electroimán que optimiza el movimiento en el eje z, ver Figura 1.1, constituyen los cambios más significativos respecto al diseño inicial de (Cazco, 2010).

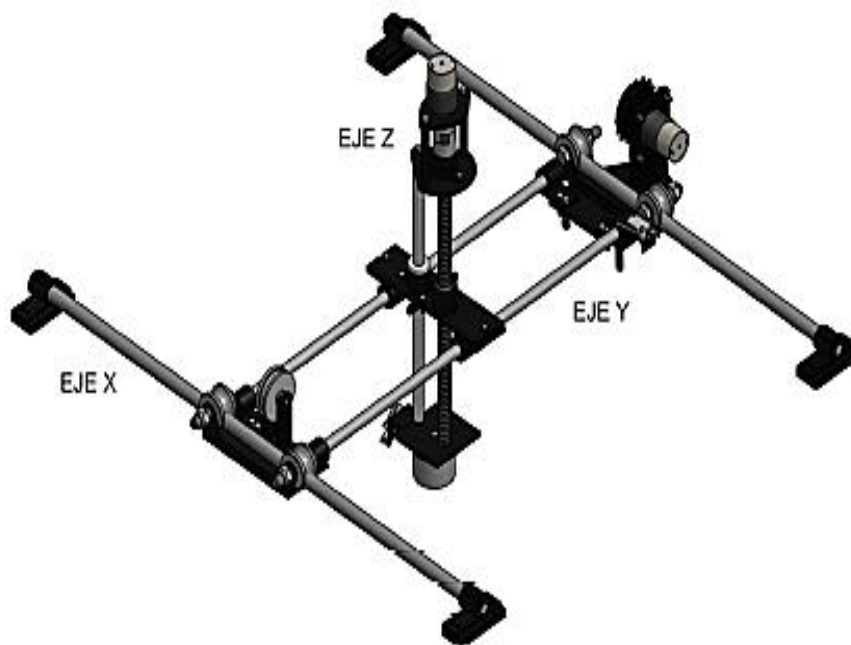


Figura 1.1 Disposición de ejes del robot cartesiano.  
(Fuente: Narváez, 2019)

Una descripción generalizada de las especificaciones técnicas del robot se encuentra en las Tablas 1.1 y 1.2.

Tabla 1.1 Especificaciones técnicas para el eje x y el eje y del robot cartesiano.

Parámetro	Unidad	Eje X	Eje Y
Material (Sy)	(MPa)	AISI C1010 (379)	AISI C1010 (379)
Peso aproximado que soporta	kg	12,5	7,5
Longitud del eje requerida	m	0,76	0,55
Diámetro del eje	mm	16	12,5
Deflexión máxima	mm	-0,21	-0,12
Sistema de Transmisión	n/a	Cadena-Catarina	Cadena-Catarina
Velocidad requerida	rpm	14,3	14,3
Potencia requerida	W	0,2	0,13
Motor-Reductor seleccionado	n/a	POLOLU 100:1	POLOLU 100:1

(Fuente: Narváez, 2019)

Tabla 1.2 Especificaciones técnicas para el eje z del robot cartesiano.

Elemento	Parámetro	Unidad	Eje Z
Tornillo de Potencia	Material (Sy)	(MPa)	A36 (250)
	Diámetro Exterior	mm	15,6
	Paso	in	0,2
	Velocidad requerida	rpm	175
	Eficiencia	%	42
	Peso máximo a levantar	kg	3
Tuerca	Material (Sy)	(MPa)	Bronce (69)
	Longitud	mm	25
	Motor-Reductor seleccionado	n/a	POLOLU 100:1

(Fuente: Narváez, 2019)

## 1.2 Matlab

Matlab es un lenguaje diseñado para la computación técnica que puede ser utilizado para el modelado, simulación, análisis y procesamiento de datos, representación gráfica y también en el desarrollo de algoritmos. Además de las herramientas comunes que posee el programa se añaden los denominados toolboxes que son colecciones de funciones especializadas que resuelven aplicaciones específicas (Gilat, 2006).

### 1.2.1 Robotics Toolbox for Matlab

El *Robotic Toolbox* es una herramienta con funciones útiles para el estudio y la simulación de robots, ha sido desarrollada por Peter Corke durante más de veinticinco años alcanzando un nivel sustancial de madurez y plasmada en el libro *Robotics Vision & Control: Fundamental Algorithms in Matlab*, un práctico manual con los comandos más importantes. En el caso de robots manipuladores las funciones del toolbox se especializan en la cinemática, la generación de trayectorias, la dinámica y el control (Corke, 2017).

## 1.2.2 Comandos del Robotics Toolbox usados en el modelado

El manual de Peter Corke presenta varias categorías con sus respectivos comandos (funciones), por ejemplo: transformaciones homogéneas en 2D y 3D, puntos y líneas homogéneas, movimiento diferencial, generación de trayectorias, manipulador de enlaces en serie, cinemática, dinámica, robot móvil y planificación de trayectorias.

En la caracterización del robot cartesiano manipulador de objetos se usaron los siguientes comandos:

### Link

Representa parámetros cinemáticos, inerciales, motrices y de transmisión del eslabón del robot.

```
L = Link([0 1.2 0.3 pi/2]);
L = Link('revolute', 'd', 1.2, 'a', 0.3, 'alpha', pi/2);
L = Revolute('d', 1.2, 'a', 0.3, 'alpha', pi/2);
```

Figura 1.2 Ejemplo de codificación del comando Link.  
(Fuente: Corke 2017)

### SerialLink

Agrupar los vínculos existentes en un robot en serie. Cada eslabón y articulación de la cadena está descrito por un Link usando los parámetros de Denavit-Hartenberg (estándar o modificado). Este comando define al robot mediante el número y tipo de Link que posea, se incluyen las subclases: revolución, prismático, revolución con D-H modificado, prismático con D-H modificado.

```
>> clone = SerialLink(twolink, 'name', 'bob')
clone =
bob (2 axis, RR, stdDH)
+-----+-----+-----+-----+
| j |   theta |       d |       a |   alpha |
+-----+-----+-----+-----+
| 1 |     q1 |       0 |       1 |       0 |
| 2 |     q2 |       0 |       1 |       0 |
+-----+-----+-----+-----+

grav =    0   base = 1 0 0 0   tool = 1 0 0 0
          0           0 1 0 0           0 1 0 0
          9.81        0 0 1 0           0 0 1 0
                   0 0 0 1           0 0 0 1
```

Figura 1.3 Ejemplo de aplicación del comando SerialLink  
(Fuente: Corke, 2017)

## SerialLink.plot

R.plot(q,options) muestra una animación gráfica de un robot sustentada en el modelo cinemático. Se define un argumento **q** que representa donde se muestra el robot ya sea en un ángulo de unión (1xN), o si es una matriz (MxN) se anima a medida que la trayectoria se mueve por el punto M. Las opciones se presentan en la Tabla 1.3.

Tabla 1.3 Opciones de visualización del área gráfica del robot.

Opciones	Descripción
workspace, W	Tamaño del espacio de trabajo del robot en 3D, W = [xmn, xmx ymn ymx zmn zmx]
floorlevel, L	Coordenada Z del plano (por defecto -1)
delay, D	Retraso entre fotogramas para la animación (s)
Fps	Número de fotogramas por segundo para la visualización, inverso a la opción "delay".
[no]loop	Bucle sobre la trayectoria
[no]raise	Autoenfoco de la figura
movie, M	Guardar una animación en la película M
trail, L	Dibuja una línea registrando la trayectoria de la punta, con el estilo de línea L
scale, S	Factor de escala de anotación
zoom, Z	Reducir el tamaño del espacio de trabajo autocomputado en Z, hace que el robot parezca más grande
Ortho	Vista ortográfica
Perspective	Vista de perspectiva (por defecto)
view, V	Especifique la vista V='x', 'y', 'top' o [az el] para elevaciones laterales, vista en planta o vista general por el azimut y el ángulo de elevación
Top	Vista desde arriba
[no]shading	Habilitar el sombreado Gouraud (por defecto SI)
lightps, L	Posición de la fuente de luz (por defecto [0 0 20])
[no]name	Mostrar el nombre del robot
[no]tiles	Activar suelo de baldosas (por defecto SI)
tilesize, S	Longitud lateral de las baldosas cuadradas en el suelo (por defecto 0.2)
tile1color,C	Color de las baldosas par [r g b] (por defecto [0,5 1 0,5] verde claro)
tile2color,C	Color de las baldosas impares [r g b] (por defecto [1 1 1] blanco)

(Fuente: Corke 2017)

## SerialLink.fkine

R.fkine= (q,options) representa la cinemática directa del robot, éste método imprime la transformación homogénea que define la posición del efector final del robot como un objeto perteneciente a R3, es decir una matriz 4x4 donde la columna 4 presenta tradicionalmente el resultado en coordenadas cartesianas. Se presenta un ejemplo de aplicación teniendo como entrada q1=q2=0 como se ve en la Figura 1.4.

```
>> twolink.fkine([0 0])
ans =
    1     0     0     2
    0     1     0     0
    0     0     1     0
    0     0     0     1
```

Figura 1.4 Aplicación de la cinemática directa.  
(Fuente: Corke, 2017)

### 1.3 MSC Adams

Es un programa desarrollado por MSC Software Corporation, se enfoca en el estudio de la dinámica de las partes en movimiento y la forma en que las cargas y fuerzas están distribuidas a través de los sistemas mecánicos.

#### 1.3.1 Justificación

Al momento de diseñar un producto es importante entender oportunamente el rendimiento del sistema que lo gobierna. Para validar el sistema en general es importante hacerlo en función de los requerimientos de los subsistemas mecánicos, eléctricos y demás, a menudo la demora de esta validación desencadena en varios procesos de rediseño que a su vez incurren en costos adicionales no deseables.

#### 1.3.2 Fundamento

El software utiliza la dinámica multicuerpo MBD (*Multibody Dynamics*, por sus siglas en inglés) que incrementa la eficiencia ingenieril al validar el diseño a nivel de sistema tempranamente. La vasta capacidad de análisis y el entorno informático de Adams permite la gestión de las interacciones interdisciplinarias entre las que se tiene el movimiento, las estructuras, sensores, actuadores; con el fin de optimizar el rendimiento, fiabilidad y accesibilidad del diseño de determinado producto.

La tecnología de solución de MBD, con enfoque lagrangiano, ejecuta la dinámica no lineal en un menor tiempo que la solución mediante el análisis por elementos finitos FEA (*Finite Elements Analysis*, por sus siglas en inglés), que a su vez tiene un enfoque newtoniano. Las cargas y fuerzas que se compilan en la simulación de Adams tienen un mayor grado de refinamiento que el FEA al realizar una mejor evaluación mediante un amplio espectro de movimientos y entornos operativos (Adams, 2005).

#### 1.3.3 Enfoque de MDB

Basado en principio de la dinámica Lagrangiana, Adams genera y resuelve numéricamente el sistema de ecuaciones respectivo como función del tiempo. Dichas

ecuaciones son algebraicas, diferenciales y no lineales, generalmente. El enfoque básico de MDB usa la segunda formulación de Lagrange.

$$F_j = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} + \sum_{i=1}^m \frac{\partial \Phi_i^T}{\partial q_j} \lambda_i - Q_j \quad \text{donde } \Phi_i = 0 \quad \text{para } j = 1, \dots, n$$

Ecuación 1.1

Donde:

q: coordenada generalizada.

F: la ecuación de equilibrio en dirección de la coordenada generalizada q.

L: el Lagrangiano (T-V), donde T=energía cinética y V=energía potencial.

$\Phi$ : ecuaciones algebraicas de restricción.

$\lambda$ : multiplicador de Lagrange.

Q: fuerza generalizada.

n: número de coordenadas generalizadas.

m: número de ecuaciones de restricción. (< o = al número de n).

Generalmente las fuerzas generalizadas pueden estar en función de sus coordenadas y del tiempo. Las  $\Phi$  son constantes e independientes del tiempo (escleronómicas) o continuas, las limitaciones algebraicas varían con el tiempo (holonómicas). Las soluciones que presenta Adams son generalmente iterativas. Si  $m=n$  todo el movimiento es predeterminado y solamente las ecuaciones de restricción son adecuadas para resolver el problema, con cualquier fuerza resultante siendo re-calculada. Si el problema es invariante en el tiempo, pero  $n>m$ , la solución, de ser factible, es iterada además es quasi-estática con Q invariante en el tiempo teniendo participación en la solución (McConville, 2015).

#### 1.3.4 Adams/View

Es el preprocesador estándar del software Adams que solventa proyectos de gran complejidad mediante la creación de subrutinas e interacción con otros programas. Adams/View importa y gestiona archivos CAD procedentes de AutoCAD, Solidworks, etc., relacionados al diseño mecánico, estructural y afines.

La interfaz gráfica del programa esta provista de menús desplegables que permiten la visualización de las herramientas de simulación dinámica, engloba la simulación y la



posible animación de un sistema o mecanismo, la rapidez con la que se despliegan los resultados mediante tablas y gráficos dinámicos con iconos modificables, evidencia la ventaja del uso de esta herramienta (Martínez, 2018).

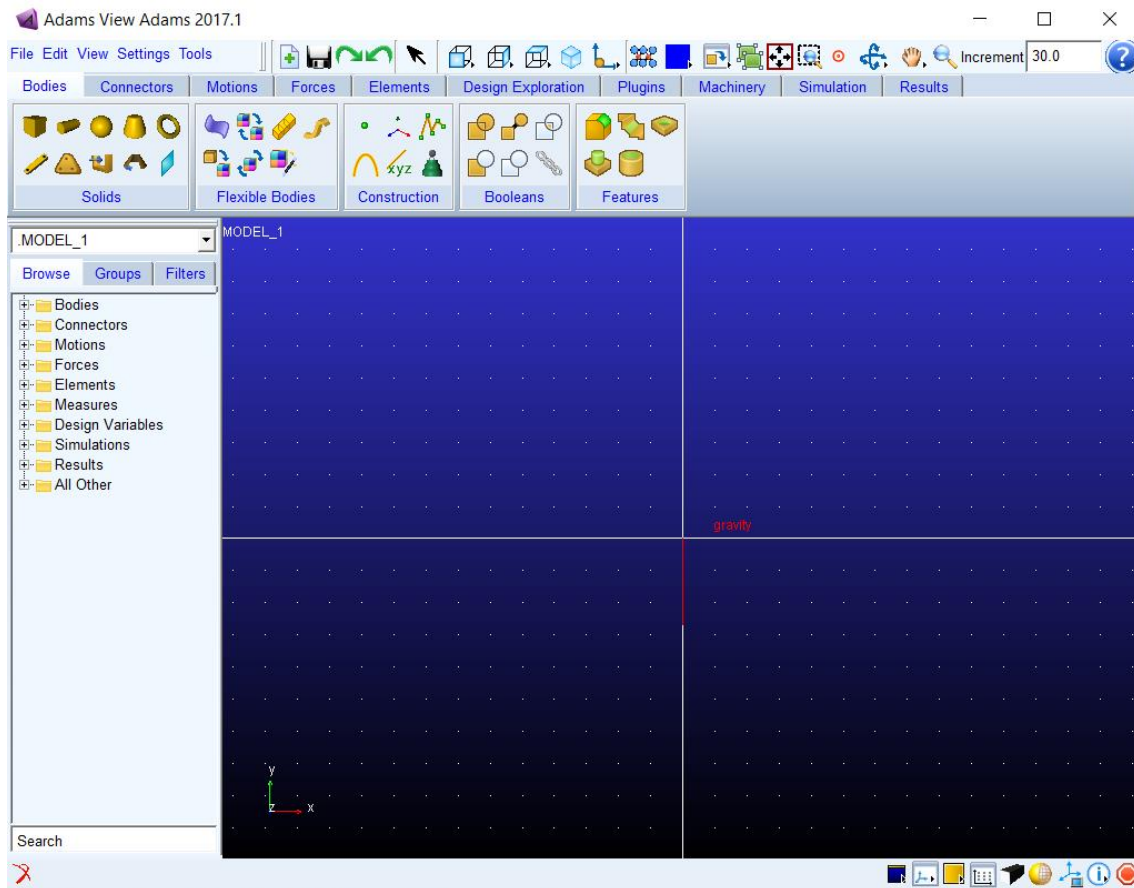


Figura 1.5. Pantalla principal de Adams  
(Fuente: Autores)

## 1.4 Fotogrametría

### 1.4.1 Definición

La palabra fotogrametría se deriva del griego: *photos* = luz, *gramma* = dibujo, *metrón* = medida. Por lo tanto, la fotogrametría es la ciencia de realizar medidas a partir de fotografías.

La definición que proporciona **la Sociedad Americana de Fotogrametría y Teledetección (ASPRS)** es “el arte, ciencia y tecnología para la obtención de medidas fiables de objetos físicos y su entorno, a través de grabación, medida e interpretación de imágenes y patrones de energía electromagnética radiante u otros fenómenos” (Rutty, 2019).

La fotogrametría permite reconstruir la posición, orientación, forma y tamaño de los objetos a partir de imágenes, las cuales pueden originarse como imágenes fotoquímicas (fotografía convencional) o como imágenes fotoeléctricas (fotografía digital).

En años recientes se originan las imágenes de escáner láser que proporcionan información de distancia asociada con cada elemento de la imagen. Los resultados de un análisis fotogramétrico pueden ser (Krauss, 2007):

- Números: coordenadas de puntos separados en un sistema de coordenadas tridimensional.
- Dibujos (análogo): mapas y planos con detalles planimétricos y líneas de contorno.
- Modelos geométricos (digital): que alimentan a los sistemas de información.
- Imágenes (análogo y/o digital): Sobre todo fotografías rectificadas y los llamados fotomodelos tridimensionales, los cuales son modelos CAD texturizados.

#### **1.4.2 Fundamento**

El principio fundamental utilizado por la fotogrametría es la triangulación. Mediante la toma de fotografías superpuestas de al menos dos ubicaciones espaciales diferentes se pueden apuntar líneas de visión desde cada cámara a puntos en el objeto de estudio, ver Figura 1.6.

En la fotogrametría de múltiples imágenes, el número de imágenes puede ser ilimitado. Por lo tanto, la fotogrametría utiliza la posición de la cámara a medida que se mueve a través del espacio 3D para calcular las coordenadas (x, y, z) de los objetos; para eso también se conoce como estructura de fotogrametría de movimiento.

En la práctica, se puede crear un modelo 3D preciso y a escala real de un objeto (por ejemplo, una habitación, un automóvil, un cuchillo, un cuerpo) a partir de una serie de imágenes superpuestas tomadas desde diferentes posiciones.

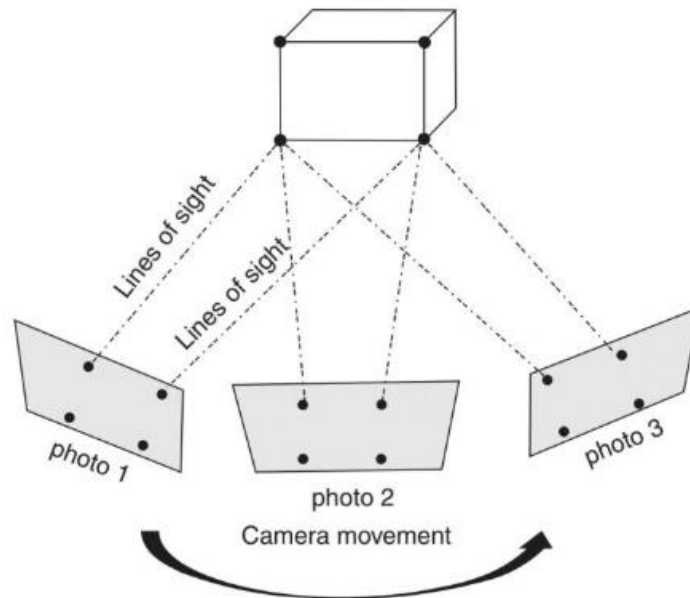


Figura 1.6 Esquema del principio fundamental usado en la fotogrametría.  
(Fuente: Ruddy, 2019)

### 1.4.3 Aplicaciones

La fotogrametría en la actualidad es aplicada en diferentes campos debido a su crecimiento dinámico, los cuales se pueden apreciar en la Figura 1.7.

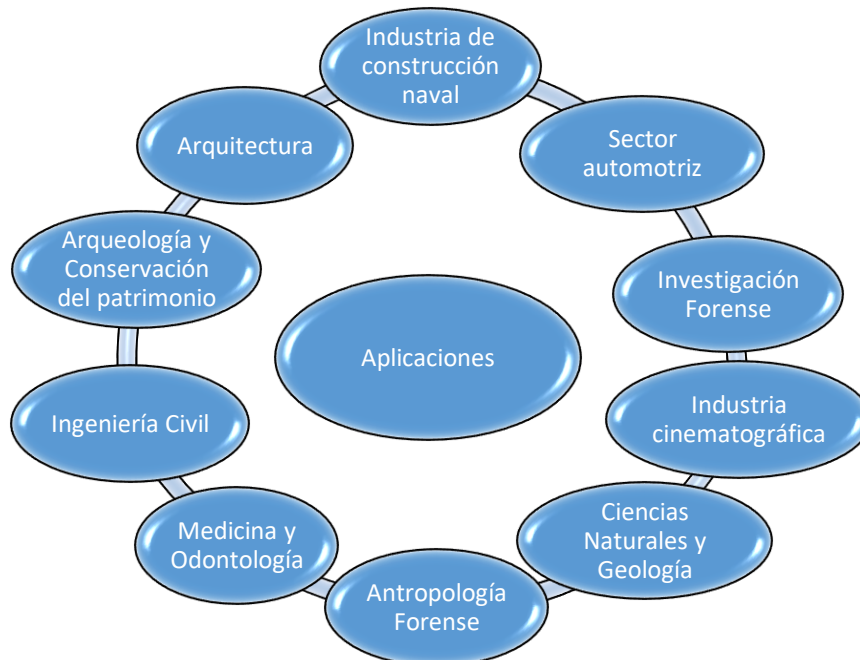


Figura 1.7 Aplicaciones de la fotogrametría.  
Fuente: (Autores)

### 1.4.4 Tecnologías cinemáticas de análisis de movimientos

El término Mocap (Motion capture o captura del movimiento) engloba técnicas de grabación de movimiento del cuerpo humano, que capturan datos espaciotemporales y

se representan digitalmente. Mocap permite la captura de parámetros de movimiento lineales y coordenadas angulares, velocidades y aceleraciones para extremidades y articulaciones. El hardware Mocap se sustenta en la tecnología de captura de movimiento (marcador o sin marcador), usando sensores con diferentes principios físicos: cámaras 2D/3D para detección óptica, sensores magnéticos con imanes permanentes y receptores de bobina para registrar cambios en el campo magnético, esqueletos mecánicos para rastreo directo de ángulos de articulación, o basados en sensores inerciales como acelerómetros y giróscopos (Gómez, 2018).

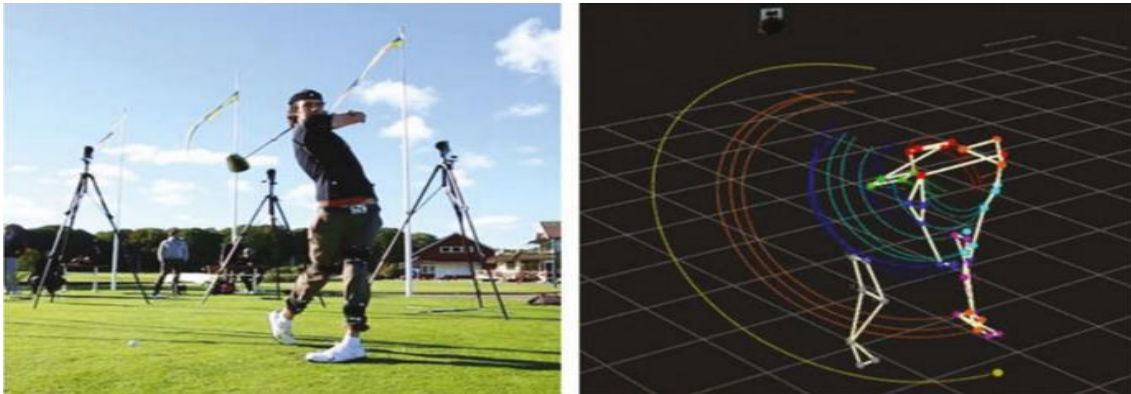


Figura 1.8 Uso del sistema de seguimiento MoCap en un golfista.  
(Fuente: Luhmann, 2013)

#### **1.4.5 Sistema de análisis de movimiento Kinescan/IBV**

Es un sistema que realiza un análisis de movimientos en tiempo real basado en la tecnología de fotogrametría video 3D, dicho análisis se basa en el registro de movimientos y en el posterior desglose de la cinemática de dicho movimiento a partir de los marcadores o pines colocados en el objeto de estudio (Kinescan/IBV, 2014).

Una descripción detallada del sistema Kinescan/IBV requiere de la introducción de algunos conceptos acerca de la fotogrametría que interviene en el equipo, es decir, lo concerniente a las cámaras y sus estándares de grabación:

#### **Requerimientos de video**

Kinescan/IBV trabaja con formato de video digital AVI, el cual es una especie de acumulador en el cual convergen varias señales de audio y video, convirtiéndolas en un archivo multimedia. La razón por la cual se escoge este formato es su versatilidad a la hora de ser comprimido sin que la imagen pierda calidad (Quintero, 2015). La cámara de video debe cumplir con ciertos requisitos que garanticen la precisión de los resultados:

## Resolución

La resolución está dada por el número de columnas multiplicado por el número de filas que posea el sensor generalmente expresado en píxeles, que es un elemento de imagen y representa a la unidad de resolución.

Un chip o sensor CCD (*Charge Couple Device*) es un acumulador de fotones que reproduce una matriz de puntos similar a la de un escáner e influye directamente en la calidad de imagen de grabación (Dembowski, 2003).

Los fotodiodos de un CCD generan una carga eléctrica al ser inducidos por un fotón de luz. Dicha carga eléctrica que genera esta interacción se transmite mediante flujo de electrones, método conocido como de transferencia interlineal. Cuando el obturador empieza la exposición se inicia el trabajo de acumulación del CCD, el proceso termina con el cierre del obturador. El periodo en el cual el obturador se encuentra en reposo es ideal para la digitalización de las cargas ubicadas en los píxeles. El CCD del sistema Kinescan/IBV es de 832x832 píxeles.

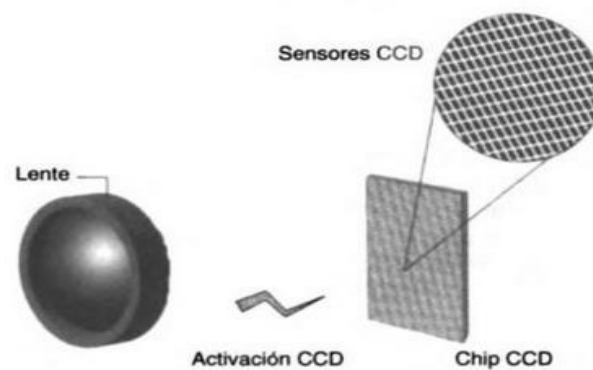


Figura 1.9 Principio de acción de una cámara digital.  
(Fuente: Dembowski, 2003)

## Velocidad de obturación y luz

Es preciso establecer una diferenciación entre la velocidad de obturación y la velocidad de grabación, la primera establece el tiempo en el cual el chip CCD está expuesto a la luz mientras que la segunda viene dada por el número de fotogramas por segundo que grabe la cámara. Por lo mencionado, la velocidad de obturación incide en cuán nítida o borrosa sale una imagen en la grabación. Para el sistema Kinescan/IBV la velocidad de obturación es ajustable por software de 10 a 1000  $\mu$ s.

Respecto a la luz ocurre que, mientras se quiera asegurar nitidez con velocidades de obturación altas, el tiempo de exposición será pequeño y por consiguiente menos luz incidirá en los sensores CCD dando como resultado una imagen oscura. Por tal razón la iluminación mínima necesaria de las cámaras debe ser lo más baja posible para garantizar un desempeño óptimo en condiciones deficientes de iluminación. (Kinescan/IBV, 2014).

### **Descripción de los elementos del sistema Kinescan/IBV**

El sistema Kinescan/IBV instalado en el Laboratorio de Biomecánica de la Facultad de Ingeniería Mecánica de la EPN, dispone de seis cámaras de análisis dispuestas estratégicamente para un buen enfoque y correcta detección de movimientos, también de un concentrador Ethernet con tecnología PoE, una cámara WiFi de documentación, un PC maestro que interpreta las grabaciones y permite el análisis cinemático, varillas y marcadores de calibración, ver Tabla 1.4. Además, el sistema es compatible para la adquisición de señales de electromiografía (referentes al sistema nervioso y muscular) y señales analógicas.

Tabla 1.4 Elementos físicos conformantes del sistema Kinescan/IBV.

<b>Elemento</b>	<b>Descripción</b>
Cámaras de Análisis	Cámaras Giga-Ethernet con procesador On-Board, con frecuencia de adquisición de: 30 a 250 fps.
Concentrador Ethernet POE	Permite la comunicación entre las cámaras y el PC.
Cámaras de Documentación	Son del tipo IP con conexión inalámbrica y almacenan la información visual del movimiento.
Utillaje de Calibración	Sirven para la calibración y el establecimiento de sistemas de referencia locales, incluye varilla y marcadores.

(Fuente: Kinescan/IBV, 2014)

La tecnología PoE (Powerover Ethernet) que utiliza el sistema Kinescan/IBV permite la alimentación de dispositivos Ethernet a través del propio cable de red, tiene la ventaja de no necesitar de cableado adicional para la alimentación DC de otros sistemas. Para el correcto funcionamiento del sistema es necesario que todos los elementos (cámaras, switches, etc) soporten la tecnología PoE (Castillo, 2014), ver Figura 1.10.

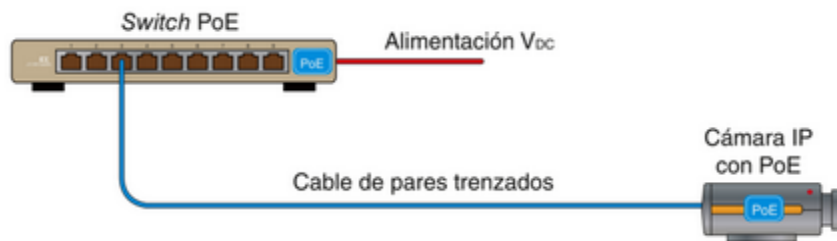


Figura 1.10 Tecnología PoE.  
(Fuente: Castillo, 2014)

## Marcador o pin

La ubicación o posición de un punto en el espacio se encuentra representada físicamente por un marcador de material refractante que facilita su detección y posterior digitalización. Los diámetros disponibles son: 9,5mm, 15,9mm y 25,4mm los cuales se adhieren fácilmente al cuerpo y a la ropa mediante la aplicación de cinta doble faz.

## Requerimientos de ubicación del equipo

Es importante garantizar el correcto funcionamiento del equipo por lo cual se deben cumplir con los requerimientos de la Tabla 1.5.

Tabla 1.5 Requerimientos del Equipo.

Requerimiento	Descripción
Alimentación	Tensión de 200 a 230 V alterna y monofásica. Frecuencia 50 Hz ( $\pm 5\%$ ).
Toma de Tierra	La no conexión a la toma de tierra o a una resistencia superior a 10 $\Omega$ supone errores de funcionamiento.
Fuentes de Ruido	Equipo debe estar aislado de fuentes eléctricas generadoras de ruido.
Entorno de Trabajo	Ambientes exentos de polvo y partículas similares, se debe evitar ubicarse cerca de calefactores e incidencia directa de los rayos solares.
Temperatura	Condiciones normales de trabajo: 10-40 °C y 10-95% de humedad relativa.

(Fuente: Kinescan/IBV, 2014)

Es necesario también asegurar que la zona de trabajo o análisis sea espaciosa, oscura y esté libre de elementos reflectantes y brillantes que interfieran con la detección de los marcadores detallada en el apartado 2.3.1.

## Calibración

La calibración del sistema no implica mayor dificultad. Se usa un dispositivo de calibración, como una varilla provista de marcadores predefinidos llamados también pines de marcación. Los pines se colocan distanciados uno de otro, de modo que cuando el software vea los marcadores en una cámara, podrá discernir qué es qué. Cuando varias cámaras ven todos los marcadores, simultáneamente empezarán a averiguar dónde están en relación con las otras cámaras (Figura 1.11). Ya ubicadas por las cámaras, éstas pueden triangular un marcador que ven dos o más cámaras. Algunos sistemas también tienen un dispositivo secundario que establece los ejes y el origen del sistema de coordenadas global (Kitagawa, 2012).



Figura 1.11 Varilla de calibración MoCap.  
(Fuente: Kitagawa, 2012)

## 1.5 Cinemática del robot

En el aseguramiento de la localización del efector final o de cualquier eslabón conformante del robot cartesiano es necesario especificar su posición y orientación, para lo cual existen algunos medios matemáticos enfocados en:

- Representar la posición: el sistema cartesiano de referencia en coordenadas cartesianas, polares, cilíndricas o esféricas.
- Representar la orientación: matrices de rotación, ángulos de Euler, cuaternios.

Para una correcta localización en un espacio  $n$ -dimensional se debe asegurar la representación conjunta de posición y orientación usando coordenadas de un espacio  $(n+1)$  dimensional, la dimensión adicional añadida representa un factor de escala.



Surge el concepto de matriz de transformación homogénea representada por la letra T en la ecuación 1.2, dicha matriz tiene un tamaño 4x4, está formada por cuatro submatrices y se debe aplicar sobre vectores con 4 dimensiones que representen las coordenadas homogéneas de un vector tridimensional, explicadas en el apartado 1.3 (Barrientos, 2007).

$$T = \begin{pmatrix} R_{3x3} & p_{3x1} \\ f_{1x3} & w_{1x1} \end{pmatrix} = \begin{pmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{pmatrix}$$

Ecuación 1.2

La cinemática del sistema mecánico de un robot plantea la determinación de la posición, orientación, velocidad y aceleración del extremo final del robot y de cada eslabón que lo conforma en función del tiempo, e intenta expresarlo respecto a un sistema de referencia. Este análisis inicial obvia los pares y fuerzas que actúan sobre los eslabones conformantes.

Por consiguiente, surgen dos problemas fundamentales que deben ser solucionados:

- El problema cinemático directo que se centra en la localización del efector final del robot conocidos los valores de las articulaciones.
- El problema cinemático inverso trata sobre la transformación del espacio cartesiano al espacio articular, es decir, conocida la posición y orientación del efector final se descubre la configuración de las articulaciones conformantes (Barrientos, 2007).

Derivado de lo anterior también es de suma importancia encontrar las relaciones de velocidades entre las articulaciones y el extremo, para lo cual se emplea el uso de un modelo diferencial mediante la matriz Jacobiana.

Al ser el robot objeto de estudio de tipo PPP, la variable de las articulaciones prismáticas o deslizantes que lo conforman será la extensión que alcancen sus eslabones; es importante resaltar la diferencia con una junta de revolución en cuyo caso la variable es el ángulo que forman los eslabones. (ver Figura 1.12).

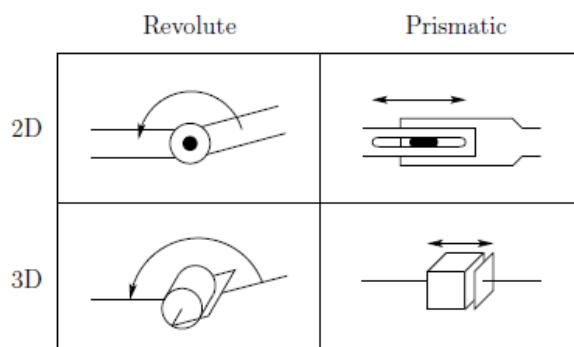


Figura 1.12 Representación de una junta de revolución y prismática.  
(Fuente: Spong, 2006)

El modelado del robot cartesiano requiere de un método apropiado para describir su geometría, el más difundido es el método de Denavit-Hartenberg para sistemas en serie, el cual consiste en encontrar una matriz de transformación homogénea que relacione espacialmente dos elementos rígidos.

### 1.5.1 Obtención de los parámetros de Denavit-Hartenberg para el robot cartesiano

Este algoritmo es empleado en la resolución de la cinemática directa del robot y consta de la determinación de la relación existente entre las articulaciones y el efector final, usando la menor cantidad de parámetros posibles al darle una notación característica a cada articulación y eslabón (ver Figura 1.13).

Tabla 1.6 Parámetros D-H para el robot cartesiano.

Articulación	$\theta$	D	A	$\alpha$
1	0	0	a1	-90°
2	0	d2	0	90°
3	0	-d3	0	0

(Fuente: Cazco, 2010)

Donde:

$a_i$ : Longitud del eslabón

$\alpha_i$ : Giro del eslabón

$d_i$ : Desplazamiento del eslabón

$\theta_i$ : Ángulo de la articulación

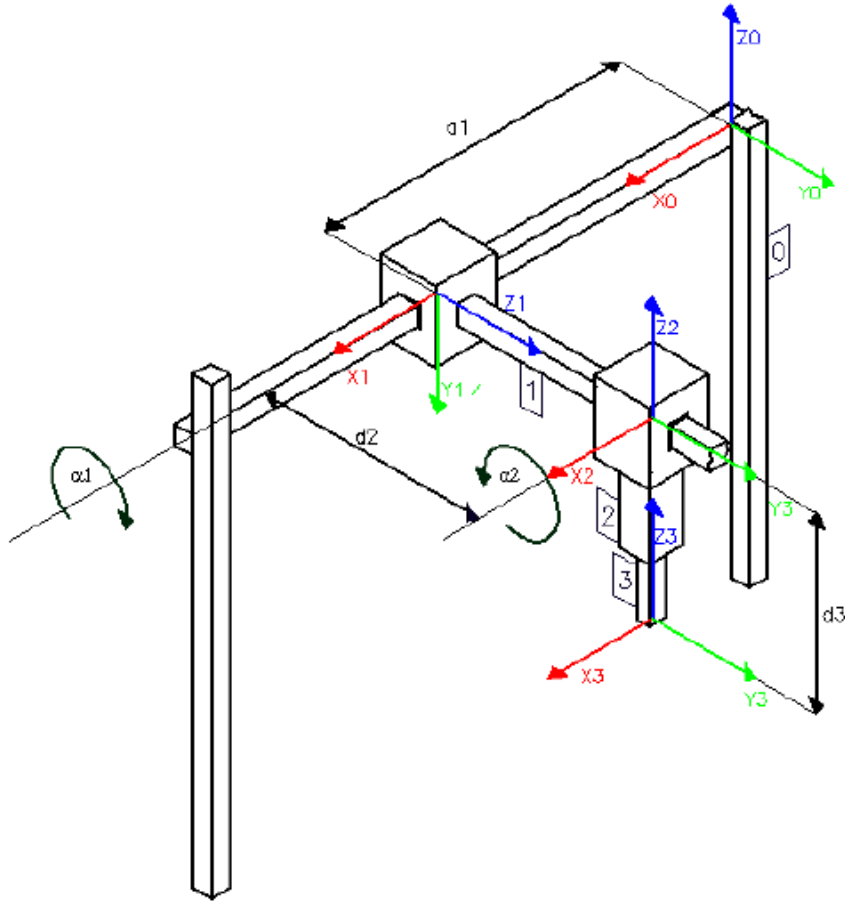


Figura 1.13 Notación D-H en las articulaciones del robot cartesiano.  
(Fuente: Cazco, 2010)

Al desarrollar el método se tiene que cada transformación homogénea  $A_i$  (ver ecuación 1.3) permite relacionar el sistema de referencia de un elemento "i" con el sistema del elemento "i+1" (Barrientos, 2007).

La transformación homogénea resulta de la multiplicación de cuatro transformaciones básicas.

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

Ecuación 1.3

Desarrollando el producto matricial detallado en el trabajo de Spong (2006), se tiene que:

$$A_i = \begin{pmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ecuación 1.4

En el caso del robot cartesiano al reemplazar los valores de la Tabla 1.6 en la ecuación 1.3 se tienen tres matrices "A":

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Las cuales se multiplican entre sí para hallar la matriz de transformación T global que indica la localización del sistema del efector final con respecto al sistema inicial (0, 0, 0).

$$T = A_1 A_2 A_3$$

Ecuación 1.5

$$T = \begin{pmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T es una matriz homogénea de traslación que representa la cinemática del robot cartesiano con articulaciones prismáticas cuyo efector final no rota respecto al sistema inicial (Cazco, 2010).

### 1.5.2 Cuaternión unitario y cuaternión dual

El álgebra de cuaterniones permite expresar la posición, orientación y/o movimientos de un cuerpo en el espacio tetradimensional utilizando números hipercomplejos (Adorno, 2011). Los cuaterniones unitarios fueron introducidos por Hamilton en el siglo XIX y pueden considerarse una extensión de números complejos, donde las tres unidades imaginarias  $\hat{i}$ ,  $\hat{j}$  y  $\hat{k}$  están definidas y, tal como se describe en (Adorno, 2011), tienen las siguientes propiedades (Hamilton, 1884):

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$$

Ecuación 1.6

Entonces, el conjunto  $\mathbb{Q}$  de cuaterniones unitarios se define como:

$$\mathbb{Q} = \{h_1 + h_2\hat{i} + h_3\hat{j} + h_4\hat{k} : h_1, h_2, h_3, h_4 \in \mathbb{R}\}$$

Tal como sucede con los números complejos, los cuaterniones pueden dividirse en una parte real  $Re(h) = h_1$  y una parte hipercompleja  $Im(h) = h_2\hat{i} + h_3\hat{j} + h_4\hat{k}$ .

Algunas propiedades aritméticas son:

- Conmutativo en la suma y resta:  $q_1 \pm q_2 = q_2 \pm q_1$
- Asociativo y distributivo, pero no conmutativo en la multiplicación:  $q_1q_2 \neq q_2q_1$

Además, dado un cuaternión unitario  $q = h_1 + h_2\hat{i} + h_3\hat{j} + h_4\hat{k}$ , este puede representarse de manera vectorial de la siguiente manera:

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Ecuación 1.7

Esta representación no permite la multiplicación directa entre dos o más cuaterniones unitarios, para ello, estos vectores deben asociarse a una matriz. Esto se logra utilizando operadores izquierdo y derecho, los cuales se encargan de transformar linealmente el vector  $[\cdot]_L, [\cdot]_R: \mathbb{R}^{4 \times 1} \rightarrow \mathbb{R}^{4 \times 4}$ :

$$[q]_L = \begin{bmatrix} q_0 & -\bar{q}^T \\ \bar{q} & q_0\mathbb{I} + [\bar{q}]^\times \end{bmatrix}$$

Ecuación 1.8

$$[q]_R = \begin{bmatrix} q_0 & -\bar{q}^T \\ \bar{q} & q_0\mathbb{I} - [\bar{q}]^\times \end{bmatrix}$$

Ecuación 1.9

Donde  $\mathbb{I} \in \mathbb{R}^{3 \times 3}$  es una matriz identidad, mientras que  $\bar{q} \in \mathbb{R}^{3 \times 1}$  representa la parte imaginaria del cuaternión  $q$ . Además, el operador  $[\cdot]^\times: \mathbb{R}^{3 \times 1} \rightarrow \mathbb{R}^{3 \times 3}$  permite la transformación lineal de un vector a una matriz simétrica de la siguiente manera:

$$[\bar{q}]^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

Ecuación 1.10

Si se utiliza al cuaternión unitario como una representación de traslaciones y rotaciones de un robot, se tendrían solo cuatro elementos y su interpretación podría ser compleja. Esto puede solucionarse al utilizar cuaterniones duales.

Tal como ocurre con los cuaterniones unitarios, los cuaterniones duales poseen las unidades imaginarias  $\hat{i}, \hat{j}$  y  $\hat{k}$ , sin embargo, también incorpora la unidad dual  $\epsilon$ . De acuerdo con (Salazar & Tsiotras, 2018), este término fue introducido por (Clifford, 1871) y tiene las siguientes propiedades:

$$\epsilon \neq 0, \epsilon^2 = 0$$

Entonces, un cuaternión dual  $\hat{q} \in \hat{\mathbb{Q}}$  podría expresarse de la siguiente forma:

$$\hat{q} = a + \epsilon b$$

Ecuación 1.11

Donde  $a, b \in \mathbb{Q}$  son cuaterniones unitarios. A continuación, se enuncian algunas propiedades aritméticas de los cuaterniones duales:

- Conmutativo en la suma y resta:  $\hat{q}_1 \pm \hat{q}_2 = \hat{q}_2 \pm \hat{q}_1$
- Asociativo y distributivo, pero no conmutativo en la multiplicación:  $\hat{q}_1 \hat{q}_2 \neq \hat{q}_2 \hat{q}_1$

Al igual que los cuaterniones unitarios, los cuaterniones duales pueden expresarse en forma vectorial de la siguiente manera:

$$\hat{q} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Asimismo, los operadores izquierdo y derecho pueden ser extendidos al espacio de los cuaterniones duales de la siguiente manera:

$$\llbracket \hat{q} \rrbracket_L = \begin{bmatrix} [a]_L & 0_{4 \times 4} \\ [b]_L & [a]_L \end{bmatrix}$$

Ecuación 1.12

$$\llbracket \hat{q} \rrbracket_R = \begin{bmatrix} [a]_R & 0_{4 \times 4} \\ [b]_R & [a]_R \end{bmatrix}$$

Ecuación 1.13

Siendo  $[[\cdot]]_L, [[\cdot]]_R: \mathbb{R}^{8 \times 1} \rightarrow \mathbb{R}^{8 \times 8}$ . Su uso en el producto de dos cuaterniones duales, se ejemplifica a continuación:

$$\hat{q}_1 \hat{q}_2 = [[\hat{q}_1]]_L \hat{q}_2 = [[\hat{q}_2]]_R \hat{q}_1$$

Ecuación 1.14

Aunque un cuaternión dual contiene más elementos que un cuaternión unitario, estos pueden ser utilizados para representar la orientación y posición de un cuerpo en el espacio. De acuerdo con (Zenteno-Lara & Valverde-Salazar), la parte real  $q_r \in \mathbb{R}^{4 \times 1}$  tiene información sobre la orientación de un marco de referencia, mientras que la parte dual  $q_d \in \mathbb{R}^{4 \times 1}$  muestra su posición; ambas respecto a un marco de referencia predefinido (comúnmente el inercial):

$$\hat{q}_{y/x}^z = q_r + \frac{\epsilon}{2}(q_d)$$

Ecuación 1.15

Donde  $q_r$  representa la orientación de un marco de referencia, definida como:

$$q_r = \cos\left(\frac{\theta}{2}\right) + \hat{n} \sin\left(\frac{\theta}{2}\right)$$

Ecuación 1.16

Donde  $\hat{n}$  representa el eje de giro imaginario  $\hat{i}, \hat{j}$  o  $\hat{k}$ . Por otra parte,  $q_d$  representa la posición del marco de referencia, definida como:

$$q_d = p_x \hat{i} + p_y \hat{j} + p_z \hat{k}$$

Ecuación 1.17

Gracias a lo establecido anteriormente, se podrá resolver el problema de la cinemática diferencial utilizando, cuaterniones duales.

### 1.5.3 Cinemática diferencial

La derivada en función del tiempo de una representación matemática de la postura (posición) de un marco de referencia representa su velocidad angular y lineal. Entonces, si un cuaternión dual  $\hat{q}_{y/x}^z \in \mathbb{R}^{8 \times 1}$  representa la postura de un marco de referencia,

puede derivarse en función del tiempo y así conocer la velocidad del marco de referencia (Valverde-Salazar & Tsiotras, 2018):

$$\dot{\hat{q}}_{y/x}^z = \frac{1}{2} \hat{q}_{y/x}^z \hat{\omega}_{y/x}^y = \frac{1}{2} \hat{\omega}_{y/x}^x \hat{q}_{y/x}^z$$

Ecuación 1.18

Siendo  $\hat{\omega}_{y/x}^y, \hat{\omega}_{y/x}^x \in \mathbb{R}^{8 \times 1}$  la velocidad dual vista desde diferentes marcos de referencia. La primera parte de la ecuación anterior se asocia con la derivada temporal del marco de referencia del cuerpo  $y$ , mientras que la segunda parte se relaciona con el marco de referencia  $x$ . Con esto, se puede conocer la velocidad en el efector final de un robot.

Consideremos que el cuaternión dual  $\hat{q}_{n/I}^I \in \mathbb{R}^{8 \times 1}$  denota la representación de la postura del  $n$  – ésimo marco de referencia, visto desde el marco inercial  $I$ . Entonces, se deriva la cinemática directa de la siguiente manera:

$$\dot{\hat{q}}_{n/I}^I = \sum_{i=0}^{n-1} (\hat{q}_{i/0}^0 \dot{\hat{q}}_{i+1/i}^0 \hat{q}_{n/i+1}^0)$$

Ecuación 1.19

Si  $\dot{\hat{q}}_{i+1/i}^0 = \frac{1}{2} \hat{\omega}_{i+1/i}^i \hat{q}_{i+1/i}^0$  entonces

$$\dot{\hat{q}}_{n/I}^I = \sum_{i=0}^{n-1} \left[ \hat{q}_{i/0}^0 \left( \frac{1}{2} \hat{\omega}_{i+1/i}^i \hat{q}_{i+1/i}^0 \right) \hat{q}_{n/i+1}^0 \right]$$

$$\dot{\hat{q}}_{n/I}^I = \frac{1}{2} \sum_{i=0}^{n-1} \left[ \hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i \hat{q}_{i+1/i}^0) \hat{q}_{n/i+1}^0 \right]$$

$$\dot{\hat{q}}_{n/I}^I = \frac{1}{2} \sum_{i=0}^{n-1} \left[ \hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i) \hat{q}_{i+1/i}^0 \hat{q}_{n/i+1}^0 \right]$$

$$\dot{\hat{q}}_{n/I}^I = \frac{1}{2} \sum_{i=0}^{n-1} \left[ \hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i) \hat{q}_{n/i}^0 \right]$$

Ecuación 1.20



Estableciendo que  $\hat{q}_{n/0}^0 = \hat{q}_{i/0}^0 \hat{q}_{n/i}^0$ , entonces  $(\hat{q}_{i/0}^0)^* \hat{q}_{n/0}^0 = \hat{q}_{n/i}^0$ , siendo  $(\cdot)^*: \mathbb{R}^{8 \times 1} \rightarrow \mathbb{R}^{8 \times 1}$  el operador conjugado que permite invertir un cuaternión dual. Posteriormente, se sustituye este resultado en la ecuación anterior, simplificando de la siguiente manera:

$$\begin{aligned}\dot{\hat{q}}_{n/I}^I &= \frac{1}{2} \sum_{i=0}^{n-1} [\hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i) (\hat{q}_{i/0}^0)^* \hat{q}_{n/0}^0] \\ \dot{\hat{q}}_{n/I}^I &= \frac{1}{2} \sum_{i=0}^{n-1} [\hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i) (\hat{q}_{i/0}^0)^*] \hat{q}_{n/0}^0 \\ 2\dot{\hat{q}}_{n/I}^I (\hat{q}_{n/0}^0)^* &= \sum_{i=0}^{n-1} [\hat{q}_{i/0}^0 (\hat{\omega}_{i+1/i}^i) (\hat{q}_{i/0}^0)^*] \\ \hat{\omega}_{n/0}^0 &= \sum_{i=0}^{n-1} \left[ \left[ \hat{q}_{i/0}^0 \right]_L \left[ (\hat{q}_{i/0}^0)^* \right]_R \hat{\omega}_{i+1/i}^i \right]\end{aligned}$$

Ecuación 1.21

En la última ecuación,  $\hat{\omega}_{i+1/i}^i$  representa la velocidad dual relativa entre los marcos de referencia  $i$  e  $i+1$ , la cual se genera por la acción de una articulación (sobre un eje definido) acoplada a un cuerpo rígido. Debido a que existen diferentes tipos de articulaciones,  $\hat{\omega}_{i+1/i}^i$  se define como:

$$\hat{\omega}_{i+1/i}^i = \xi_i \dot{\alpha}_i$$

Ecuación 1.22

Siendo  $\xi_i \in \mathbb{R}^{8 \times 1}$  el operador tornillo que permite representar, de forma dual, el eje de acción de cualquier tipo de articulación; asimismo,  $\dot{\alpha}_i$  representa la  $i$ -ésima coordenada generalizada del sistema. Entonces, podemos definir la velocidad dual inercial de la siguiente manera:

$$\hat{\omega}_{n/0}^0 = \sum_{i=0}^{n-1} \left[ \left[ \hat{q}_{i/0}^0 \right]_L \left[ (\hat{q}_{i/0}^0)^* \right]_R \xi_i \dot{\alpha}_i \right]$$

Ecuación 1.23

$$\hat{\omega}_{n/0}^0 = \left[ \left[ \hat{q}_{1/0}^0 \right]_L \left[ \left( \hat{q}_{1/0}^0 \right)^* \right]_R \xi_1 \quad \cdots \quad \left[ \hat{q}_{n/0}^0 \right]_L \left[ \left( \hat{q}_{n/0}^0 \right)^* \right]_R \xi_n \right] \begin{bmatrix} \dot{\alpha}_1 \\ \vdots \\ \dot{\alpha}_n \end{bmatrix}$$

$$\hat{\omega}_{n/0}^0 = J^I(\hat{q}_{n/0}^0, \xi) \dot{\alpha}$$

Ecuación 1.24

Por lo tanto, se ha establecido una matriz jacobiana relacionada con el marco de referencia inercial de la representación dual de un robot. Ahora es posible encontrar la ecuación que define la velocidad angular y lineal del n-ésimo marco de referencia. Se considera que la velocidad dual también puede representarse de la siguiente manera:

$$\hat{\omega}_{n/0}^0 = \omega_{n/0}^0 + \epsilon(v_{n/0}^0 - \omega_{n/0}^0 \times r_{n/0}^0)$$

Ecuación 1.25

Este también puede expresarse en forma vectorial de la siguiente manera:

$$\hat{\omega}_{n/0}^0 = \begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 - \omega_{n/0}^0 \times r_{n/0}^0 \end{bmatrix} = \begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 + [r_{n/0}^0]^\times \omega_{n/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ [r_{n/0}^0]^\times & \mathbb{I} \end{bmatrix} \begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 \end{bmatrix}$$

Ecuación 1.26

Posteriormente se igualan ambas expresiones de velocidad dual y se obtiene lo siguiente:

$$\hat{\omega}_{n/0}^0 = \hat{\omega}_{n/0}^0$$

$$\begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ [r_{n/0}^0]^\times & \mathbb{I} \end{bmatrix} \begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 \end{bmatrix} = J^I(\hat{q}_{n/0}^0, \xi) \dot{\alpha}$$

$$\begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{n/0}^0]^\times & \mathbb{I} \end{bmatrix} J^I(\hat{q}_{n/0}^0, \xi) \dot{\alpha}$$

Ecuación 1.27

Esto permite conocer las velocidades angular y lineal del n – ésimo marco de referencia en una cadena cinemática o robot. Además, considera que  $r_{n/0}^0 \in \mathbb{R}^{4 \times 1}$  representa la posición del n – ésimo marco de referencia, visto desde el inercial; asimismo, el operador

$[\cdot]^\times : \mathbb{R}^{4 \times 1} \rightarrow \mathbb{R}^{4 \times 4}$  permite la transformación lineal de un cuaternión a una matriz simétrica de la siguiente manera:

$$[r_{n/0}^0]^\times = \begin{bmatrix} 0 & 0_{1 \times 3} \\ 0_{3 \times 1} & [\bar{r}_{n/0}^0]^\times \end{bmatrix}$$

$$[\bar{r}_{n/0}^0]^\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

Ecuación 1.28

Partiendo de esto,  $-\omega_{n/0}^0 \times r_{n/0}^0 = [r_{n/0}^0]^\times \omega_{n/0}^0 \because a \times b = -[b]^\times a = [a]^\times b$

#### 1.5.4 Cinemática diferencial del robot cartesiano

Dada la configuración del robot con tres articulaciones prismáticas, es posible definir la matriz jacobiana del sistema para así conocer la velocidad del efector final en forma dual.

##### Posición

Para representar la posición del efector final se utiliza la definición del cuaternión, entonces queda expresado de la forma:

$$r_{d_3/0}^0 = \begin{bmatrix} 0 \\ a_1 \\ d_2 \\ -d_3 \end{bmatrix}$$

Ecuación 1.29

Considere que la velocidad de las articulaciones puede establecerse a partir del vector de coordenadas generalizadas:

$$\dot{\vec{a}} = \begin{bmatrix} \dot{a}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}$$

Ecuación 1.30

Además, los operadores tornillo para cada articulación, de acuerdo con la configuración del robot, son:

$$\xi_{a_1} = [0_{1 \times 5} \quad 1 \quad 0_{1 \times 2}]^T$$

Ecuación 1.31

$$\xi_{d_2} = \xi_{d_3} = [0_{1 \times 7} \quad 1]^T$$

Ecuación 1.32

Con esto, la cinemática diferencial del robot puede generalizarse de la siguiente manera:

$$\begin{bmatrix} \omega_{n/0}^0 \\ v_{n/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{n/0}^0]^\times & \mathbb{I} \end{bmatrix} J^l(\hat{q}_{n/0}^0, \xi) \ddot{\alpha}$$

Ecuación 1.33

$$\begin{bmatrix} \omega_{d_3/0}^0 \\ v_{d_3/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} \left[ \begin{array}{ccc} \llbracket \hat{q}_{a_1/0}^0 \rrbracket_L \llbracket (\hat{q}_{a_1/0}^0)^* \rrbracket_R \xi_{a_1} & \llbracket \hat{q}_{d_2/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_2/0}^0)^* \rrbracket_R \xi_{d_2} & \llbracket \hat{q}_{d_3/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_3/0}^0)^* \rrbracket_R \xi_{d_3} \end{array} \right] \begin{bmatrix} \dot{a}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}$$

Ecuación 1.34

## Velocidad

Si se calcula la cinemática directa hasta cada marco de referencia, se obtienen los siguientes cuaterniones duales:

$$\hat{q}_{a_1/0}^0 = \left[ \cos\left(\frac{\theta_x}{2}\right) \quad -\sin\left(\frac{\theta_x}{2}\right) \quad 0 \quad 0 \quad 0.5a_1 \sin\left(\frac{\theta_x}{2}\right) \quad 0.5a_1 \cos\left(\frac{\theta_x}{2}\right) \quad 0 \quad 0 \right]^T$$

Ecuación 1.35

$$\hat{q}_{d_2/0}^0 = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{a_1}{2} \quad \frac{d_2 \sin \theta_x}{2} \quad \frac{d_2 \cos \theta_x}{2} \right]^T$$

Ecuación 1.36

$$\hat{q}_{d_3/0}^0 = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{a_1}{2} \quad \frac{d_2 \sin \theta_x}{2} \quad \frac{d_2 \cos \theta_x}{2} - \frac{d_3}{2} \right]^T$$

Ecuación 1.37

Siendo  $\theta_x \in \mathbb{R}$  el ángulo de giro sobre el eje x establecido en la matriz de parámetros Denavit – Hartenberg. Así, la forma conjugada de cada uno es:

$$(\hat{q}_{a_1/0}^0)^* = \left[ \cos\left(\frac{\theta_x}{2}\right) \quad \sin\left(\frac{\theta_x}{2}\right) \quad 0 \quad 0 \quad 0.5a_1 \sin\left(\frac{\theta_x}{2}\right) \quad -0.5a_1 \cos\left(\frac{\theta_x}{2}\right) \quad 0 \quad 0 \right]^T$$

Ecuación 1.38

$$(\hat{q}_{d_2/0}^0)^* = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\frac{a_1}{2} \quad -\frac{d_2 \sin \theta_x}{2} \quad -\frac{d_2 \cos \theta_x}{2} \right]^T$$

Ecuación 1.39

$$(\hat{q}_{d_3/0}^0)^* = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\frac{a_1}{2} \quad -\frac{d_2 \sin \theta_x}{2} \quad \frac{d_3}{2} - \frac{d_2 \cos \theta_x}{2} \right]^T$$

Ecuación 1.40

Así, cada término en la matriz jacobiana es definido de la siguiente manera:

$$\llbracket \hat{q}_{a_1/0}^0 \rrbracket_L \llbracket (\hat{q}_{a_1/0}^0)^* \rrbracket_R \xi_{a_1} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T$$

$$\llbracket \hat{q}_{d_2/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_2/0}^0)^* \rrbracket_R \xi_{d_2} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \sin(\theta_x) \quad \cos(\theta_x)]^T$$

$$\llbracket \hat{q}_{d_3/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_3/0}^0)^* \rrbracket_R \xi_{d_3} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^T$$

Los cuales se concatenan en la matriz que se presenta a continuación:

$$J^l(\hat{q}_{d_3/0}^0, \xi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \sin(\theta_x) & 0 \\ 0 & \cos(\theta_x) & 1 \end{bmatrix}$$

Ecuación 1.41

Por último,  $r_{d_3/0}^0$  puede definirse de la siguiente manera:

$$r_{d_3/0}^0 = \begin{bmatrix} 0 \\ a_1 \\ d_2 \\ -d_3 \end{bmatrix}$$

Ecuación 1.42

Al expandir términos, se obtiene la velocidad dual del efector final

$$\begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} J^I(\hat{q}_{d_3/0}^0, \xi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \sin(\theta_x) & 0 \\ 0 & \cos(\theta_x) & 1 \end{bmatrix}$$

$$\begin{bmatrix} \omega_{d_3/0}^0 \\ v_{d_3/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} J^I(\hat{q}_{d_3/0}^0, \xi) \dot{\alpha}$$

$$= [0 \ 0 \ 0 \ 0 \ 0 \ \dot{a}_1 \ \dot{d}_2 \sin(\theta_x) \ \dot{d}_3 + \dot{d}_2 \cos(\theta_x)]^T$$

Ecuación 1.43

Nótese que la velocidad está definida solo para el término lineal  $v_{d_3/0}^0$ , dado que no existen articulaciones rotatorias. Entonces, la velocidad en el efector final está relacionada directamente con la velocidad en cada articulación prismática, esto es:

$$v_{d_3/0}^0 = \begin{bmatrix} \dot{a}_1 \\ \dot{d}_2 \sin(\theta_x) \\ \dot{d}_3 + \dot{d}_2 \cos(\theta_x) \end{bmatrix}$$

Ecuación 1.44

Si se calcula la cinemática directa hasta cada marco de referencia, se obtienen los siguientes cuaterniones duales:

$$\hat{q}_{a_1/0}^0 = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \quad 0 \quad 0 \quad 0 \quad \frac{a_1}{4\sqrt{2}} \quad \frac{a_1}{4\sqrt{2}} \quad 0 \right]^T$$

Ecuación 1.45

$$\hat{q}_{d_2/0}^0 = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{a_1}{2} \quad \frac{d_2}{2} \quad 0 \right]^T$$

Ecuación 1.46

$$\hat{q}_{d_3/0}^0 = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \frac{a_1}{2} \quad \frac{d_2}{2} \quad -\frac{d_3}{2} \right]^T$$

Ecuación 1.47

Así, la forma conjugada de cada uno es:

$$(\hat{q}_{a_1/0}^0)^* = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \quad 0 \quad 0 \quad 0 \quad -\frac{a_1}{4\sqrt{2}} \quad -\frac{a_1}{4\sqrt{2}} \quad 0 \right]^T$$

Ecuación 1.48

$$(\hat{q}_{d_2/0}^0)^* = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\frac{a_1}{2} \quad -\frac{d_2}{2} \quad 0 \right]^T$$

Ecuación 1.49

$$(\hat{q}_{d_3/0}^0)^* = \left[ 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\frac{a_1}{2} \quad -\frac{d_2}{2} \quad \frac{d_3}{2} \right]^T$$

Ecuación 1.50

Así, cada término en la matriz jacobiana es definido de la siguiente manera:

$$\llbracket \hat{q}_{a_1/0}^0 \rrbracket_L \llbracket (\hat{q}_{a_1/0}^0)^* \rrbracket_R \xi_{a_1} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T$$

$$\llbracket \hat{q}_{d_2/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_2/0}^0)^* \rrbracket_R \xi_{d_2} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]^T$$

$$\llbracket \hat{q}_{d_3/0}^0 \rrbracket_L \llbracket (\hat{q}_{d_3/0}^0)^* \rrbracket_R \xi_{d_3} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^T$$

$$J^I(\hat{q}_{d_3/0}^0, \xi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 1.51

Por último,  $r_{d_3/0}^0$  puede definirse de la siguiente manera:

$$r_{d_3/0}^0 = \begin{bmatrix} 0 \\ a_1 \\ d_2 \\ -d_3 \end{bmatrix}$$

Ecuación 1.52

Al expandir términos, se obtiene la velocidad dual del efector final

$$\begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} J^I(\hat{q}_{d_3/0}^0, \xi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \omega_{d_3/0}^0 \\ v_{d_3/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} J^I(\hat{q}_{d_3/0}^0, \xi) \dot{\alpha} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ a_1 \ d_2 \ d_3]^T$$

Ecuación 1.53

Nótese que la velocidad está definida solo para el término lineal  $v_{d_3/0}^0$ , dado que no existen articulaciones rotatorias. Entonces, la velocidad en el efector final está relacionada directamente con la velocidad en cada articulación prismática, esto es:

$$v_{d_3/0}^0 = \begin{bmatrix} \dot{a}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}$$

Ecuación 1.54



## Aceleración

Debido a la extensión de la demostración para obtener las ecuaciones de aceleración, solo se muestran los resultados finales; sin embargo, puede ser consultada en (Zenteno-Lara & Valverde-Salazar).

Considere la siguiente ecuación:

$$\begin{bmatrix} \dot{\omega}_{n/0}^0 \\ \dot{v}_{n/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbf{0}_{4 \times 4} \\ -[r_{n/0}^0]^\times & \mathbb{I} \end{bmatrix} \left[ J^I(\hat{q}_{n/0}^0, \xi) \ddot{\alpha} + K^I(\hat{q}_{n/0}^0, \hat{\omega}_{n/0}^n, \xi, \dot{\xi}) \dot{\alpha} \right. \\ \left. + \begin{bmatrix} \mathbf{0}_{4 \times 1} \\ \omega_{n/0}^0 \times (\omega_{n/0}^0 \times r_{n/0}^0) \end{bmatrix} + g \right]$$

Ecuación 1.55

Donde  $K^I(\hat{q}_{n/0}^0, \hat{\omega}_{n/0}^n, \xi, \dot{\xi}) \in \mathbb{R}^{8 \times n}$  es una matriz de aceleraciones para elementos esféricos. Debido a que el sistema analizado contiene solamente articulaciones prismáticas, esta matriz es nula. Además,  $\omega_{n/0}^0 \times (\omega_{n/0}^0 \times r_{n/0}^0)$  representa la aceleración centrífuga del sistema. Por último,  $g \in \mathbb{R}^{8 \times 1}$  es un cuaternión dual con información sobre la aceleración gravitatoria del sistema.

Con los resultados obtenidos con el cálculo de la velocidad, es posible conocer la aceleración del sistema:

$$J^I(\hat{q}_{d_3/0}^0, \xi) \ddot{\alpha} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \ddot{a}_1 \quad \ddot{d}_2 \sin(\theta_x) \quad \ddot{d}_3 + \ddot{d}_2 \cos(\theta_x)]^T$$

Ecuación 1.56

$$K^I(\hat{q}_{d_3/0}^0, \hat{\omega}_{d_3/0}^{d_3}, \xi, \dot{\xi}) \dot{\alpha} = \mathbf{0}_{8 \times 1}$$

Ecuación 1.57

$$\begin{bmatrix} \mathbf{0}_{4 \times 1} \\ \omega_{d_3/0}^0 \times (\omega_{d_3/0}^0 \times r_{d_3/0}^0) \end{bmatrix} = \mathbf{0}_{8 \times 1}$$

Ecuación 1.58

$$g = [0_{1 \times 7} \quad -9.80665]^T$$

Ecuación 1.59

Por lo tanto, la aceleración del efector final es:

$$\begin{bmatrix} \dot{\omega}_{d_3/0}^0 \\ \dot{v}_{d_3/0}^0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0_{4 \times 4} \\ -[r_{d_3/0}^0]^\times & \mathbb{I} \end{bmatrix} [J^I(\hat{q}_{n/0}^0, \xi) \ddot{\alpha} + g]$$

Ecuación 1.60

$$\begin{bmatrix} \dot{\omega}_{d_3/0}^0 \\ \dot{v}_{d_3/0}^0 \end{bmatrix} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \ddot{a}_1 \quad \ddot{d}_2 \sin(\theta_x) \quad \ddot{d}_3 + \ddot{d}_2 \cos(\theta_x) - 9.80665]^T$$

Ecuación 1.61

Nótese que la aceleración está definida solo para el término lineal  $\dot{v}_{d_3/0}^0$ , dado que no existen articulaciones rotatorias. Entonces, la aceleración en el efector final está relacionada directamente con la aceleración en cada articulación prismática, esto es:

$$\dot{v}_{d_3/0}^0 = \begin{bmatrix} \ddot{a}_1 \\ \ddot{d}_2 \sin(\theta_x) \\ \ddot{d}_3 + \ddot{d}_2 \cos(\theta_x) - 9.80665 \end{bmatrix}$$

Ecuación 1.62

Con el resultado anterior es posible calcular la aceleración de forma numérica. Considere que  $\theta_x = \frac{\pi}{2}$ , entonces:

$$\begin{bmatrix} \dot{\omega}_{d_3/0}^0 \\ \dot{v}_{d_3/0}^0 \end{bmatrix} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \ddot{a}_1 \quad \ddot{d}_2 \quad \ddot{d}_3 - 9.80665]^T$$

Ecuación 1.63

Nótese que la aceleración está definida solo para el término lineal  $\dot{v}_{d_3/0}^0$ , dado que no existen articulaciones rotatorias. Entonces, la aceleración en el efector final está relacionada directamente con la aceleración en cada articulación prismática, esto es:

$$\dot{v}_{d_3/0}^0 = \begin{bmatrix} \ddot{a}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix}$$

Ecuación 1.64

## Ecuaciones representativas del robot cartesiano

Posición del efector final:

$$r_{d_3/0}^0 = \begin{bmatrix} 0 \\ a_1 \\ d_2 \\ -d_3 \end{bmatrix}$$

Velocidad del efector final:

$$v_{d_3/0}^0 = \begin{bmatrix} \dot{a}_1 \\ \dot{d}_2 \sin(\theta_x) \\ \dot{d}_3 + \dot{d}_2 \cos(\theta_x) \end{bmatrix}$$

Aceleración del efector final:

$$\dot{v}_{d_3/0}^0 = \begin{bmatrix} \ddot{a}_1 \\ \ddot{d}_2 \sin(\theta_x) \\ \ddot{d}_3 + \ddot{d}_2 \cos(\theta_x) \end{bmatrix}$$

## 1.6 Métodos de validación

Los métodos de validación de modelos tienden a responder las siguientes interrogantes:

- ¿El modelo se adapta suficientemente a los datos observados?
- ¿El modelo es suficientemente adecuado para nuestro objetivo?

Para ello se necesita tener la mayor cantidad de información posible sobre el sistema real, entonces para establecer un criterio de validación existen técnicas que permiten determinar si el modelo es representativo del sistema real (Kunusch, 2003).

**Validación del modelo respecto al propósito:** Consiste en verificar si el modelado que se obtuvo a partir del problema planteado puede ser solucionado con exactitud con el modelo obtenido.

**Viabilidad de parámetros físicos:** Para estructuras en función de parámetros físicos, se evalúa la sensibilidad del comportamiento entrada-salida, para ello se compara los valores estimados con varianzas, con valores obtenidos del sistema real basados en el conocimiento de este.

**Consistencia del modelo entrada-salida:** Para modelos lineales que presentan características de comportamiento de entrada-salida (caja negra). Se evalúa mediante diagramas de Bode para determinar si el sistema dinámico ha sido modelado correctamente.

**Reducción del modelo:** Este procedimiento reduce el modelo sin afectar las propiedades entrada-salida.

**Simulación:** Una técnica muy utilizada que simula el sistema con una entrada actual y compara la salida medida con la simulada.

### 1.6.1 Cálculo del error

Para el cálculo del error se establecen comparativos entre: los datos simulados en Adams versus los datos de la captura de movimiento por fotogrametría, y también entre los datos del modelado de Matlab con los datos fotogramétricos.

Se utiliza la estimación de residuos mediante el error cuadrático medio (ECM) (Pulloquina, 2018), que es el modelo más adecuado para comparaciones entre diferentes clases de modelos probados a la tendencia de una misma serie (Aguirre, 1994), para lo cual se emplean las siguientes fórmulas:

$$\varepsilon(t, \theta) = y(t) - y_e(t, \theta)$$

Ecuación 1.65

Donde:

$\varepsilon$  : Residuo

$\theta$ : Parámetro del modelo

$y(t)$ : Respuesta de obtención de datos por fotogrametría en el tiempo t.

$y_e(t, \theta)$ : Respuesta de simulación por Adams o modelado por Matlab, en el tiempo t.

El tamaño residual se estima mediante el error medio cuadrático:

$$ECM(\theta) = \sqrt{\frac{1}{N} \sum_1^N \varepsilon^2(t, \theta)}$$

Ecuación 1.66

Donde:

N: Número de elementos simulados.

Se determina la correlación entre los residuos y la entrada.

$$R_{\varepsilon u}(\tau) = \frac{1}{N} \sum_t^N \varepsilon(t + \tau, \theta) u(t, \theta)$$

Ecuación 1.67

Donde:

$\tau$ : Retardos de correlación

Para que el modelo sea más exacto los términos de correlación debe acercarse a cero, en el rango de  $\pm 3\sqrt{P_r}$  donde  $P_r$  se determina:

$$P_r = \frac{1}{N} \sum_k^N R_{\varepsilon}(k) R_u(k)$$

Ecuación 1.68

Donde:

$P_r$ : Probabilidad de correlación

$R_{\varepsilon}(k)$  Covarianza de  $\varepsilon(t, \theta)$

$R_u(k)$  Covarianza de  $u(t, \theta)$

Para obtener una confiabilidad del 99% los términos de correlación deben estar en los rangos de  $\pm 3\sqrt{P_r}$  que es igual a  $\pm 0,3$  (Pulloquina, 2018). Los límites de aceptación del ECM dependen de la aplicación en la que se trabaja, para el caso de un robot exoesquelético los resultados de simulación de la cinemática de sus articulaciones admiten un ECM menor a 0,8 (Misyurin, 2020).

## 2 METODOLOGÍA

En esta sección se presentan tres procedimientos para la validación de la cinemática del robot cartesiano, primero el análisis en Matlab, seguido de la simulación en MSC Adams y finalmente la obtención de los movimientos cinemáticos por fotogrametría mediante el uso del sistema Kinescan/IBV.

### 2.1 Modelo matemático en Matlab

El problema de la cinemática directa del robot cartesiano se analiza mediante el software Matlab, el cual permite determinar las velocidades y aceleraciones del efector final del robot, a partir de las variables articulares ( $\theta, d, \alpha, a$ ). En la Figura 2.1 se muestra el proceso que se lleva a cabo en el programa para la resolución de la cinemática directa.

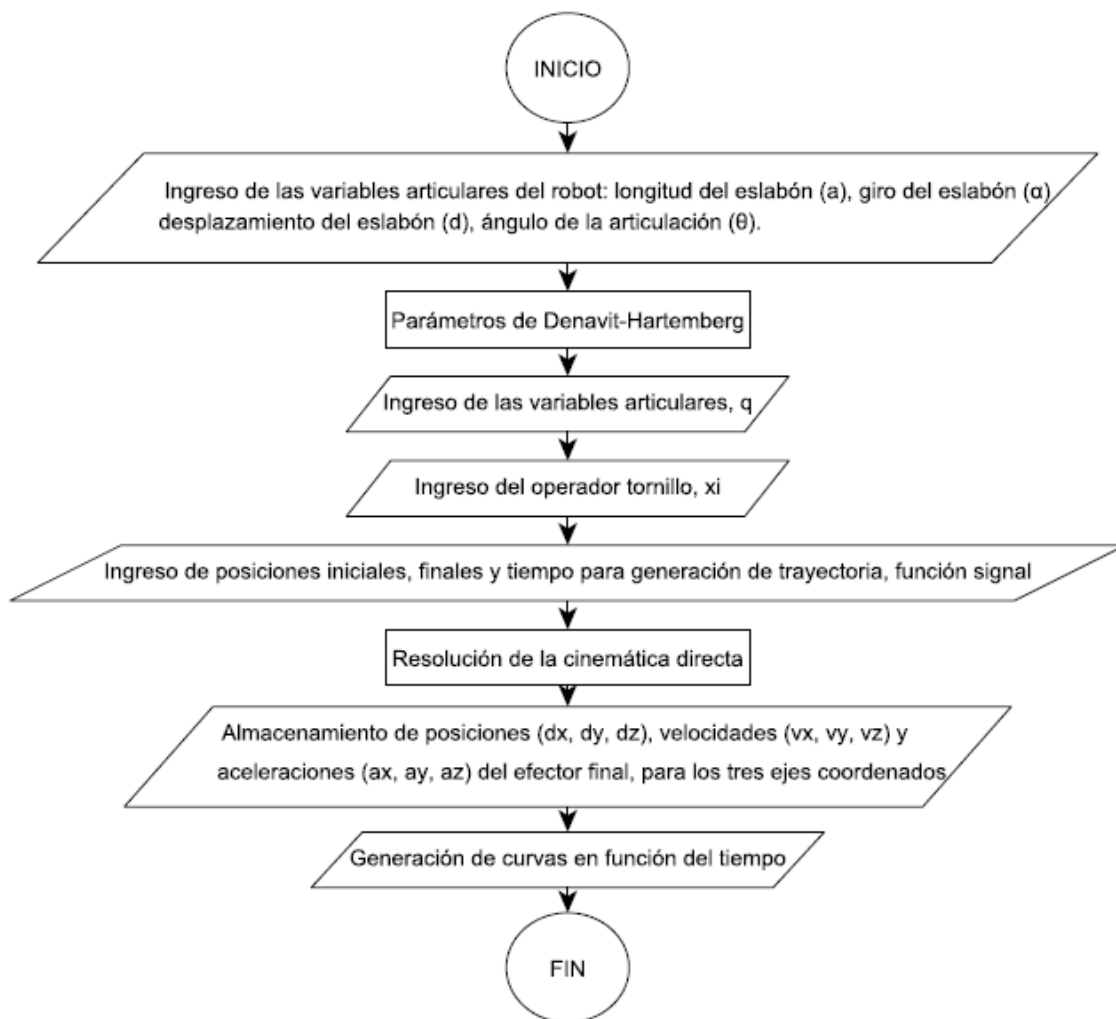


Figura 2.1 Diagrama de flujo para el análisis en Matlab.  
(Fuente: Autores)

Se descarga la librería *Robotics Toolbox* de Peter Corke de la página <https://www.petercorke.com/RTB/>, la cual se mantiene abierta en la “Current Folder”, como se evidencia en la Figura 2.2.



Figura 2.2 Current Folder.  
(Fuente: Autores)

Se despliega la carpeta *rvctools* y se corre el archivo *startup\_rvc.m* el cual activa todas las funciones y herramientas relacionadas a la robótica, la ventana de comando de Matlab se aprecia en la Figura 2.3.

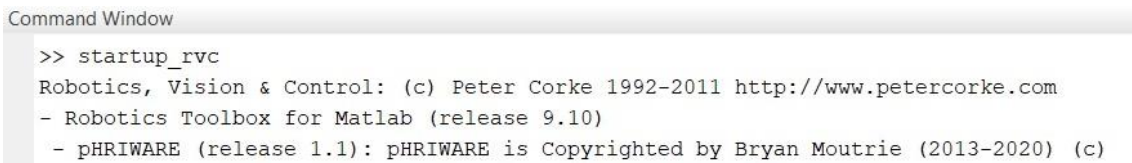


Figura 2.3 Activación de la librería.  
(Fuente: Autores)

Se crea un nuevo script que define el robot, se utiliza la función *Link* para nombrar las variables articulares ( $\theta, d, \alpha, a$ ) del robot expresadas en la Tabla 1.4., especificando los ángulos en radianes, la letra P evidencia la naturaleza prismática de las articulaciones del robot cartesiano, como se observa en la Figura 2.4.

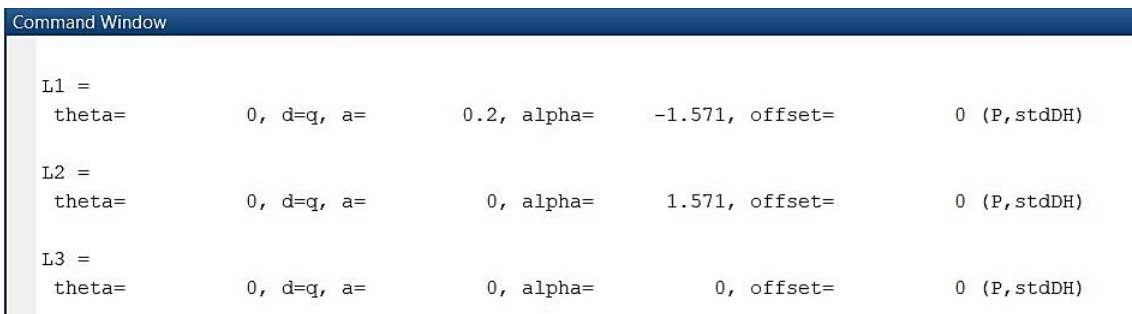


Figura 2.4 Definición de variables articulares.  
(Fuente: Autores)

Se nombra al robot con las siglas RCO (Robot Clasificador de Objetos) además se utiliza el comando SerialLink para agrupar las articulaciones participantes y crear la tabla D-H automáticamente (ver Figura 2.5).

```
RCO =
noname (3 axis, PPP, stdDH, fastRNE)
```

i	j	theta	d	a	alpha	offset
1	1	0	q1	0.2	-1.571	0
1	2	0	q2	0	1.571	0
1	3	0	q3	0	0	0

Figura 2.5 Tabla D-H del robot.  
(Fuente: Autores)

Se evalúa la cinemática directa del efector final del RCO con el comando fkine (ver Figura 2.6), se debe definir previamente la variable  $q1 = [0 \ 0,35 \ -0,1]$ , que representa los desplazamientos en el eje z.

```
% EVALUACIÓN DEL PUNTO FINAL DEL EFECTOR
T=RCO.fkine(q1)
```

Figura 2.6 Función fkine.  
(Fuente: Autores)

Se define el vector de variables articulares con el operador tornillo para calcular la cinemática directa del robot. Ver Figura 2.7.

```
% Variables articulares
q = [0
     0
     0];

% Screw vector
xi = [0 0 0
      0 0 0
      0 0 0
      0 0 0
      0 0 0
      1 0 0
      0 0 0
      0 1 1];

% Creación de la Matriz Denavit - Hartenberg
DH = denavitHartenberg(q, 0);
```

Figura 2.7 Definición de variables articulares.  
(Fuente: Autores)



Se genera la trayectoria óptima (q) del efector final con la función “signal”, al conocer la posición inicial, posición final, el vector tiempo en el cual transcurre este desplazamiento, estos valores se referencian siempre al sistema de coordenadas global o fijo. Ver Figura 2.8.

```

% POSICIÓN DEL EFECTOR FINAL
%Creación de la trayectoria
q = [signal([0 0.2 0.2], [0 0.9 16.8], 0.0354)
      signal([0 0 0.35 0.35], [0 0.9 1.8 15], 0.0354)
      signal([0 0 -0.1], [0 2.7 15], 0.0354)];
t1=0:0.0354:17.7;
dx=q(1,:);
dy=q(2,:);
dz=q(3,:);

```

Figura 2.8 Generación de la trayectoria.  
(Fuente: Autores)

Para obtener la velocidad del efector final se utiliza la función “endEffectorVelocityDQ” la cual calcula la velocidad del efector final utilizando el método generalizado, coordenadas «q» y su respectiva derivada temporal «qd»; también necesita la matriz Denavit-Hartenberg del sistema y el número de marco referencia a analizar cómo se ve en la Figura 2.9.

```

function [ W ] = endEffectorVelocityDQ ( q , qd , DH , xi , m )
%{
    Esta función calcula la velocidad del efector final utilizando el método generalizado
    coordenadas «q» y su respectiva derivada temporal «qd»; también necesita
    la matriz Denavit-Hartenberg del sistema y el número de marco referencia
    a analizar
%}
Q = forwardKinematicsDQ(DH, m);
r = positionDQ(Q);
s = quaternionsCrossOperator(r(5 : 8));
J = velocityJacobianMatrixDQ(DH, q, xi);
M = [eye(4) zeros(4)
      s eye(4)];
W = (M \ J) * qd;
end

```

Figura 2.9 Función endEffectorVelocityDQ  
(Fuente: Autores)

Se crea una matriz W en la cual se va guardando los valores de las velocidades en cada instante del movimiento del efector final. Ver Figura 2.10.

```

% VELOCIDAD DUAL DEL EFECTOR FINAL
[~, n] = size(q);
W = zeros(8, n);
for i = 2 : n
    W(:, i) = endEffectorVelocityDQ(q(:, i), [diff(q(1, i - 1 : i))
        diff(q(2, i - 1 : i)) diff(q(3, i - 1 : i))]', denavitHartenberg(q(:, i), 0), xi, 4);
end
vx=W(6,:);
vy=W(7,:);
vz=W(8,:);

```

Figura 2.10 Cálculo de la velocidad.  
(Fuente: Autores)

Para obtener la aceleración se utiliza la función “endEffectorAccelerationDQ”. Esta función calcula la aceleración del efector final utilizando el método generalizado, coordenadas «q» y sus respectivas derivadas temporales «qd» y «qdd»; eso también necesita la matriz Denavit-Hartenberg del sistema y el número de los marcos de referencia a analizar cómo se observa en la Figura 2.11.

```

function [ Wd ] = endEffectorAccelerationDQ ( q , qd , qdd , DH , xi , m )
|
    Q = forwardKinematicsDQ(DH, m);
    r = positionDQ(Q);
    s = quaternionsCrossOperator(r(5 : 8));
    M = [eye(4) zeros(4)
        s eye(4)];
    J = velocityJacobianMatrixDQ(DH, q, xi);
    K = accelerationMatrixDQ(DH, q, qd, xi);
    W = endEffectorVelocityDQ(q, qd, DH, xi, m);
    w = W(1 : 4);
    x = [zeros(4, 1)
        (quaternionsCrossOperator(w)) * (quaternionsCrossOperator(w) * r(5 : 8))];
    g = [0 0 0 0 0 0 0 0]';
    Wd = M \ ((J * qdd) + (K * qd) + x + g);
end

```

Figura 2.11 Función endEffectorAccelerationDQ.  
(Fuente: Autores)

Se genera una matriz A para guardar los valores de velocidad instantánea. Se ingresa como variables las derivadas de la velocidad, la matriz de Denavit-Hartenberg, el operador tornillo y el marco de referencia. Ver Figura 2.12.

```

% ACELERACIÓN DUAL DEL EFECTOR FINAL
acx = diff(q(1, :), 2);
acy = diff(q(2, :), 2);
acz = diff(q(3, :), 2);
[~, n] = size(acx);
A = zeros(8, n+2);
for i = 2 : n
    A(:, i) = endEffectorAccelerationDQ(q(:, i), [diff(q(1, i - 1 : i)) diff(q(2, i - 1 : i))
        diff(q(3, i - 1 : i))]', [acx(i) acy(i) acz(i)]', denavitHartenberg(q(:, i), 0), xi, 4);
end
ax=A(6,:);
ay=A(7,:);
az=A(8,:);

```

Figura 2.12 Cálculo de la aceleración.  
(Fuente: Autores)

Para visualizar las gráficas se recurre al comando plot además de los comandos secundarios que mejoran la visualización de las gráficas, como se observa en la Figura 2.13.

```

%GRÁFICAS DE DESPLAZAMIENTO DEL EFECTOR FINAL
%POSICIÓN EN X

figure (1)
subplot(3,1,1)
plot(Tiempo, Pos_x, Time, Displacement_X, t1, dx)
grid on
title('Desplazamiento en X')
legend('q_F', 'q_A', 'q_M')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([0 0.25])
subplot(3,1,2)

```

Figura 2.13 Creación de las gráficas de desplazamiento.  
(Fuente: Autores)

Las gráficas obtenidas se representan como se observa en la Figura 2.14.

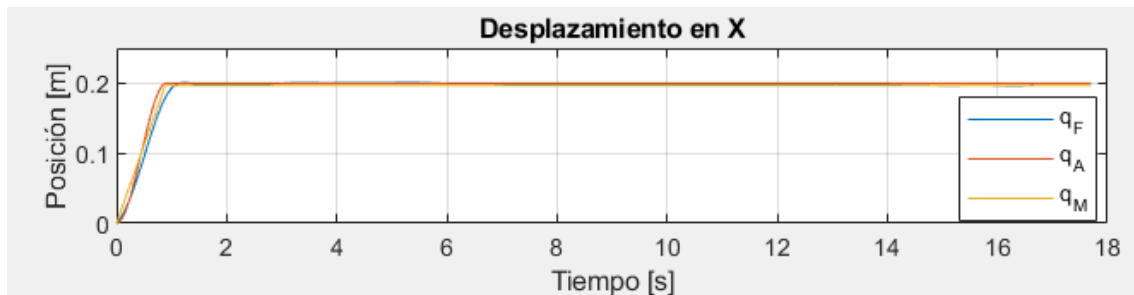


Figura 2.14 Gráfica de desplazamiento.  
(Fuente: Autores)

## 2.2 Simulación en MSC Adams/View

En esta sección se emplea como herramienta de validación del modelo matemático del robot cartesiano clasificador de objetos ferromagnéticos el software MSC Adams/View, el cual permite determinar la posición  $(x, y, z)$ , las velocidades  $(\dot{x}, \dot{y}, \dot{z})$  y las aceleraciones  $(\ddot{x}, \ddot{y}, \ddot{z})$  del efector final del robot. En la Figura 2.15 se observa el diagrama de flujo que se sigue en MSC Adams para la obtención de las gráficas de las variables requeridas.

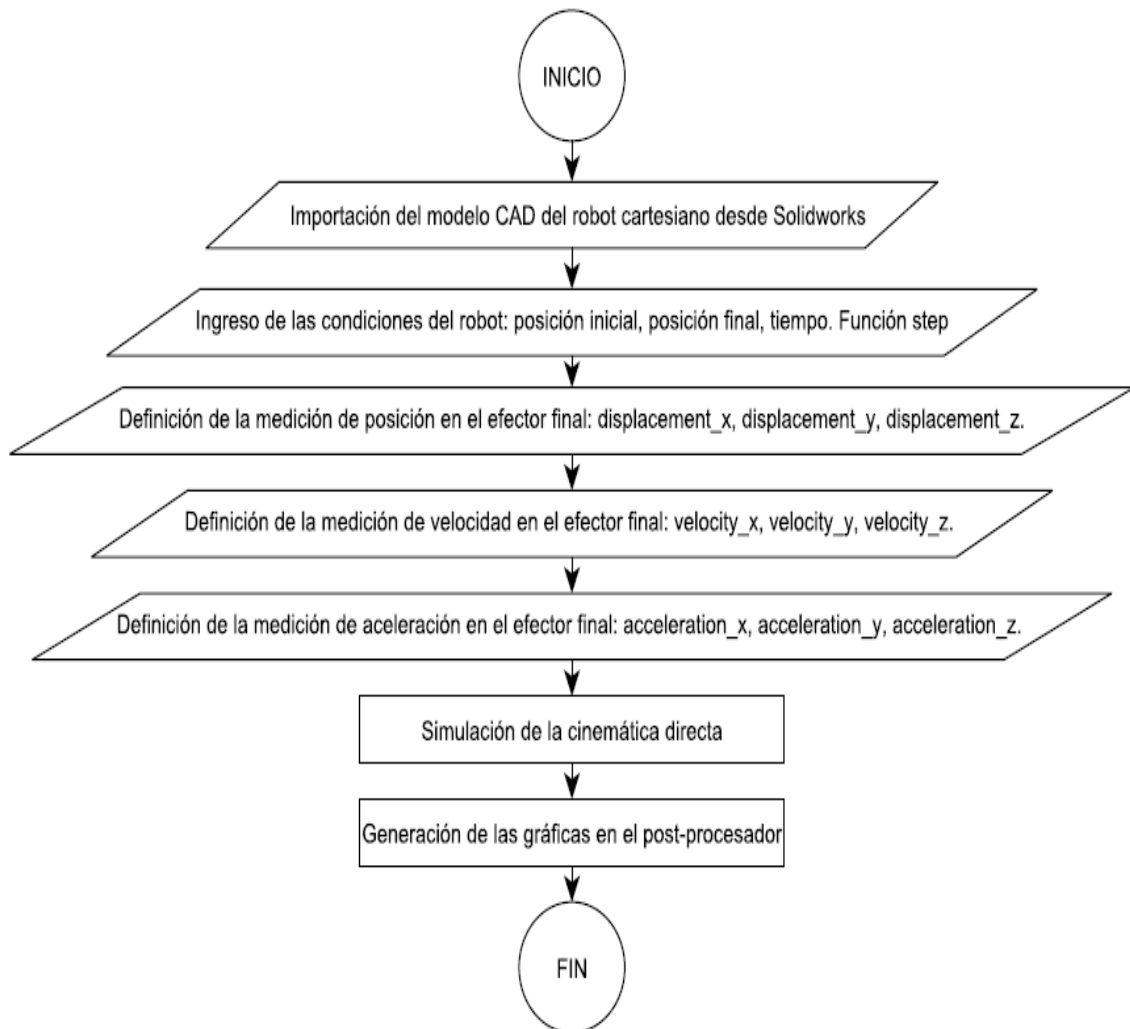


Figura 2.15 Diagrama de flujo para la simulación en MSC Adams.  
(Fuente: Autores)

En el software SolidWorks se realiza el modelo CAD del robot clasificador de objetos ferromagnéticos y un análisis de movimiento (Figura 2.16), el cual se exporta a Adams.

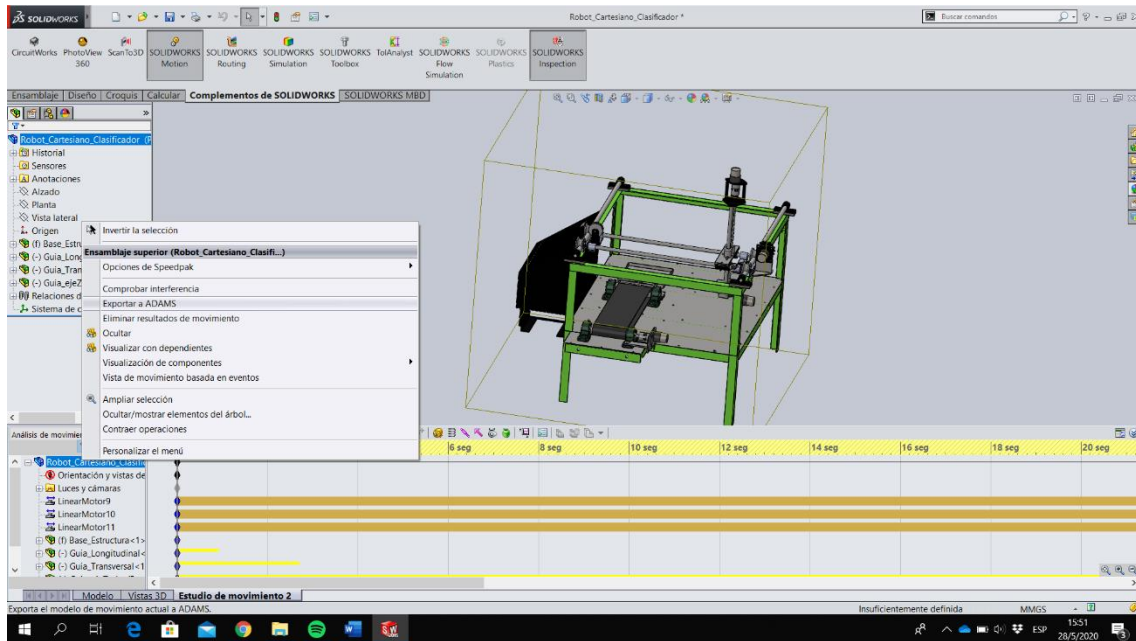
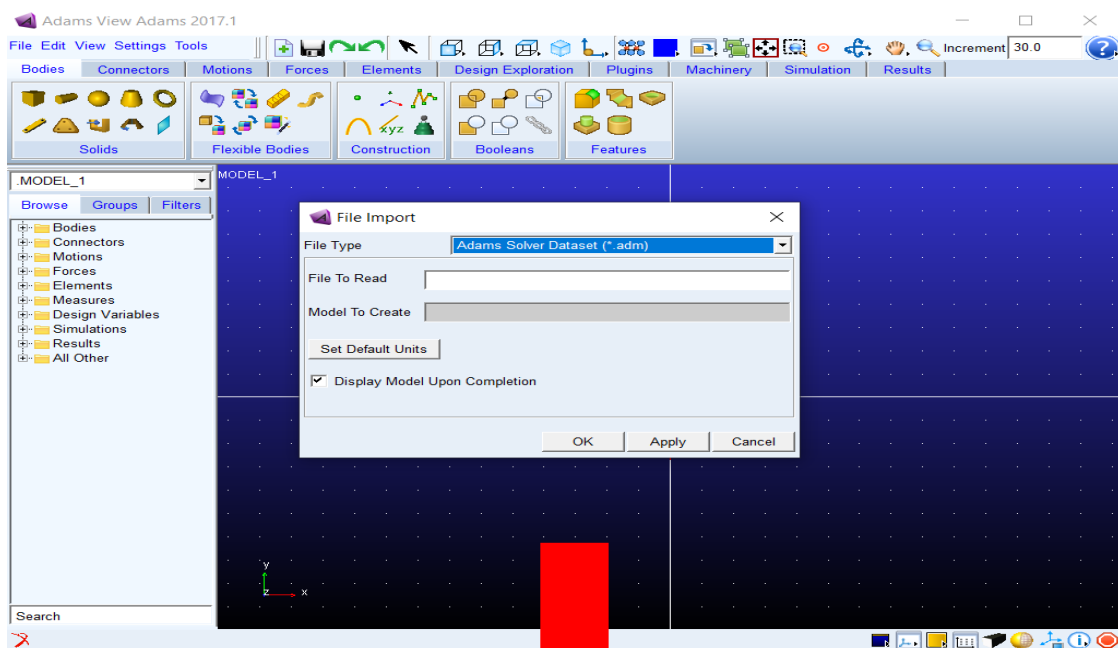


Figura 2.16 Modelo CAD del robot en SolidWorks.  
(Fuente: Autores)

En Adams/View se importa el análisis de movimiento, pinchando en la pestaña file como se puede observar en la Figura 2.17. En el cuadro de importación se escoge el tipo de archivo que es '.adm', en la pestaña de file to read se busca el archivo que se exportó desde Solidworks, por último, dar click en ok y se visualiza el robot en la pantalla de trabajo de Adams.



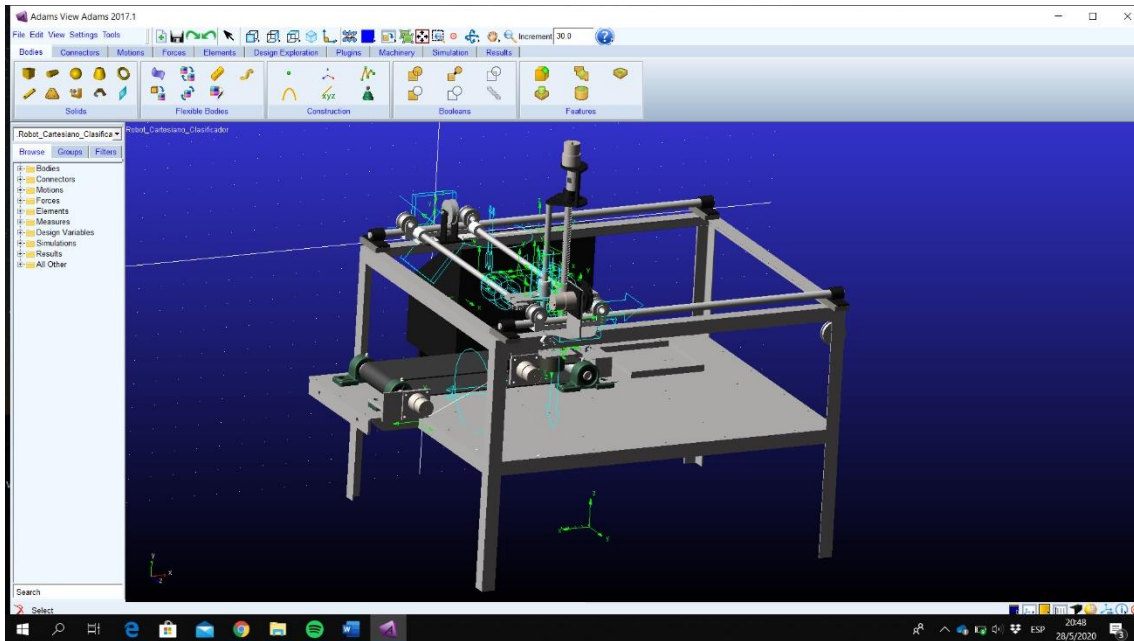


Figura 2.17 Modelo CAD del robot en Adams/View.  
(Fuente: Autores)

Los movimientos de traslación de las juntas prismáticas se instauran mediante la herramienta de movimiento de punto como se indica en la Figura 2.18.

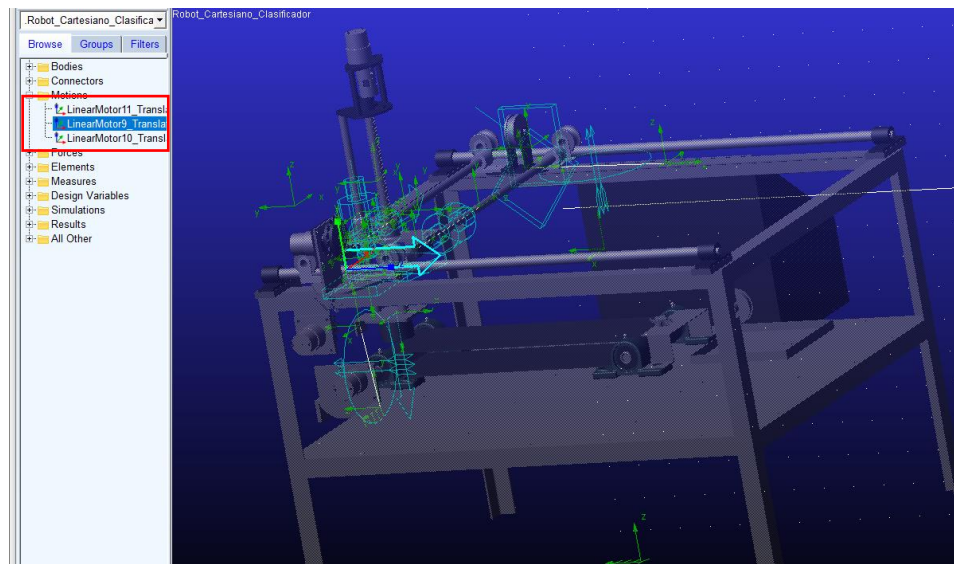


Figura 2.18 Movimientos del robot en Adams/View.  
(Fuente: Autores)

Después de configurar los movimientos se procede a definir las mediciones de cada articulación, en este caso son mediciones de posición, velocidad y aceleración, en la Figura 2.19 se puede observar la medición de la posición del efector final, de forma análoga se define las mediciones de velocidad y aceleración de las articulaciones.

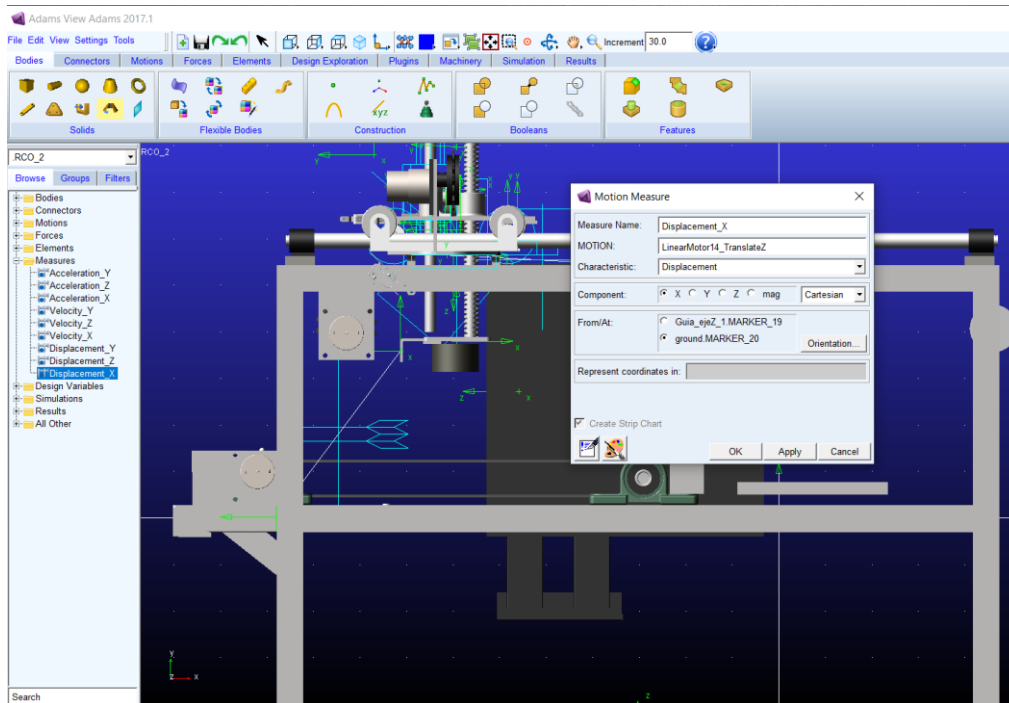


Figura 2.19 Mediciones del efector final en Adams/View.  
(Fuente: Autores)

En la pestaña de simulación se abre la herramienta de control de simulación en la cual se ingresa el tiempo de simulación y los pasos de simulación (ver Figura 2.20). A continuación, se obtienen las gráficas de posición, velocidad y aceleración del efector final en el postprocesador que se encuentra en la pestaña de resultados (ver Figura 2.21).

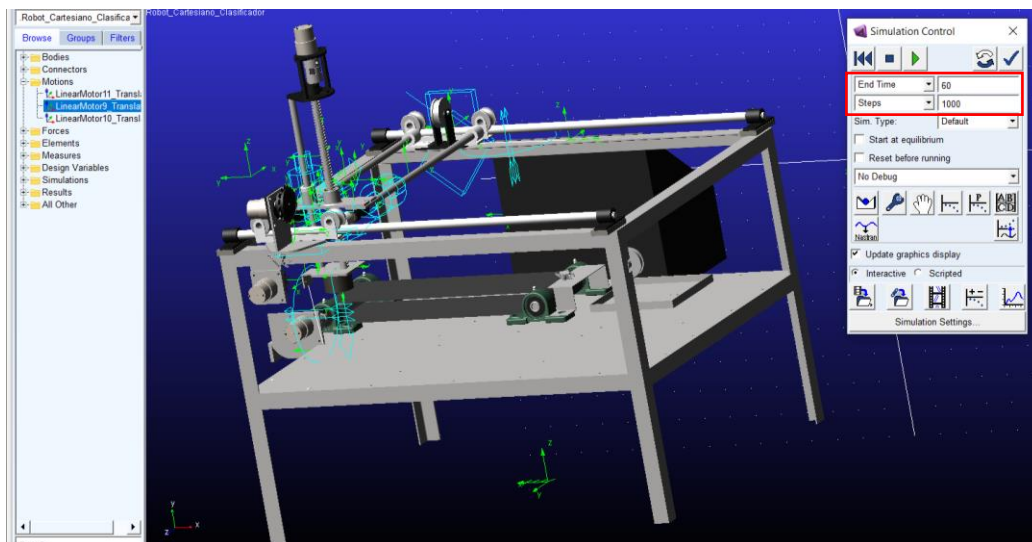


Figura 2.20 Simulación del robot en Adams/View.  
(Fuente: Autores)

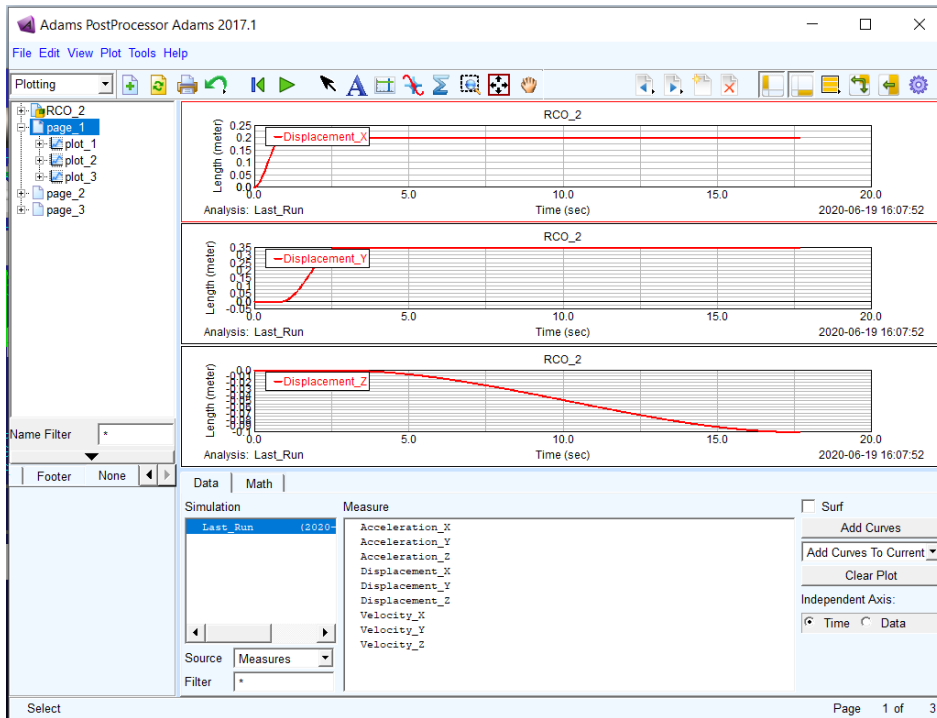


Figura 2.21 Gráficas del efector final en Adams/View.  
(Fuente: Autores)

Una vez obtenidas todas las gráficas se exporta a Excel. Se da clic a la pestaña de File, se abre la sección Export y el cursor en Table como se indica en la Figura 2.22.

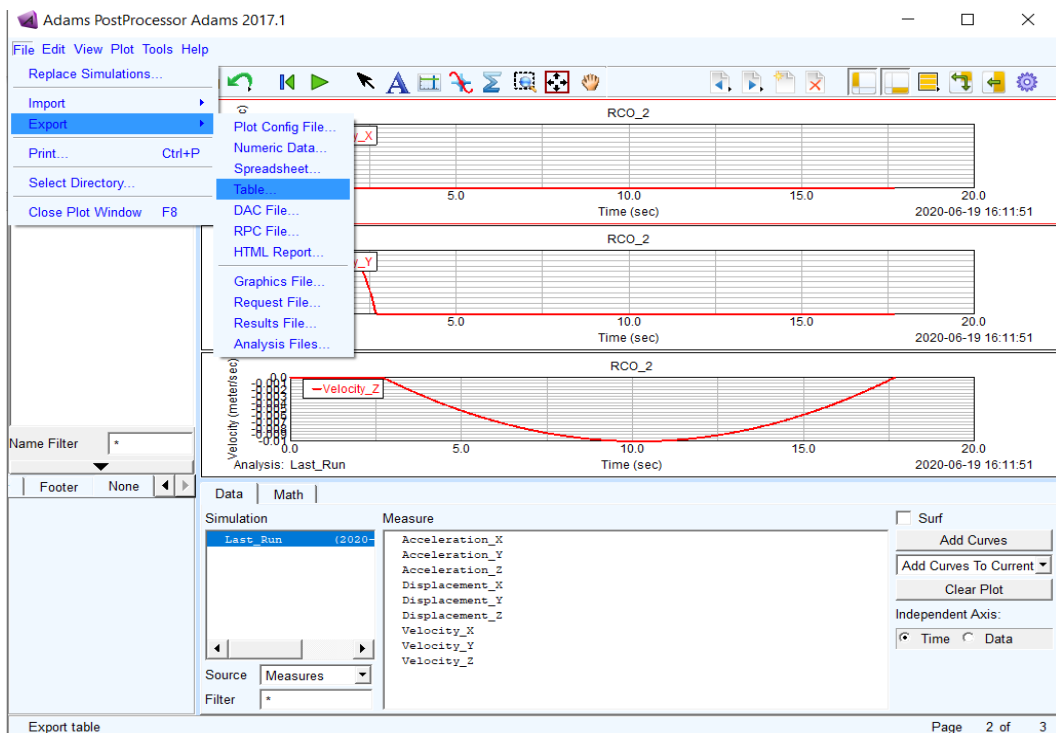


Figura 2.22 Exportación de datos a Excel.  
(Fuente: Autores)



Se despliega un cuadro de dialogo, donde se nombra al archivo en el label de plot, doble clic y se abre el Database Navigator en el cual se escoge el plot que contiene los datos del desplazamiento en el eje x del efector final (ver Figura 2.23). Se repite el mismo proceso para las gráficas de velocidad y aceleración.

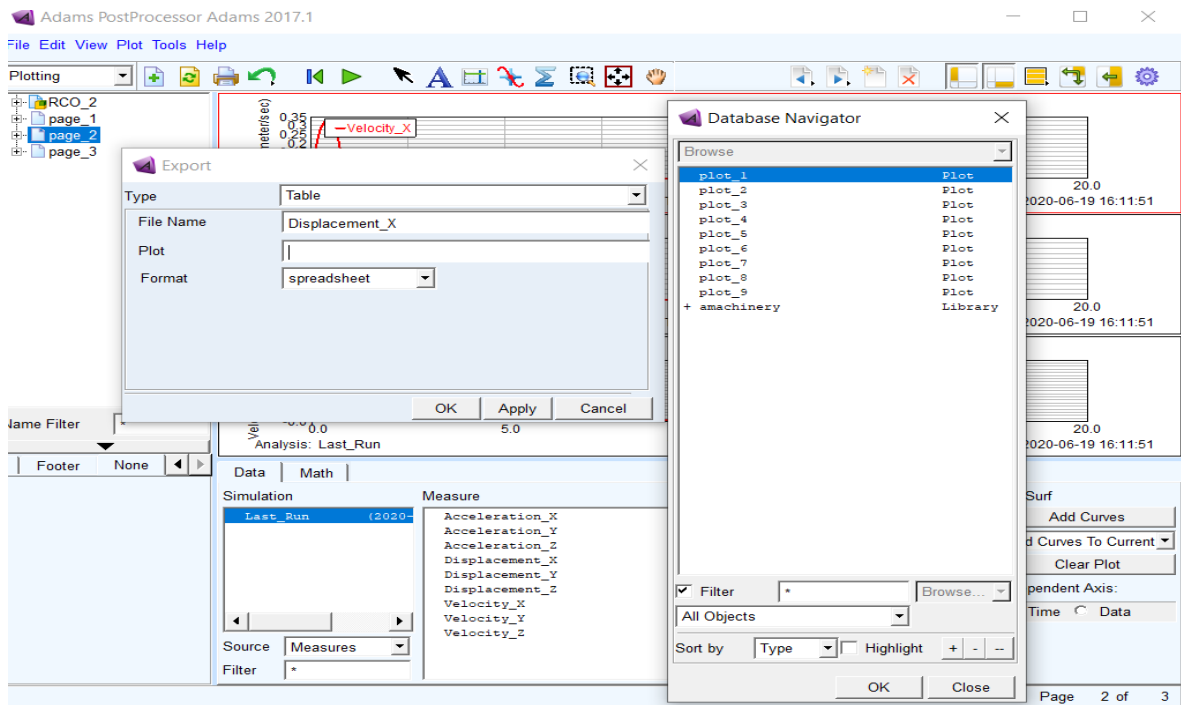


Figura 2.23 Selección de gráficas a exportar.  
(Fuente: Autores)

## 2.3 Captura del movimiento mediante el sistema de análisis de Kinescan/IBV (Fotogrametría)

Como tercera opción para validar la cinemática del robot se tiene el software Kinescan/IBV, el cual requiere del uso de marcadores que fungen como representaciones digitales de los objetos estudiados. En la Figura 2.24 se observa el diagrama de flujo para la obtención de gráficas que interpreten en 2D los movimientos generados en la realidad.

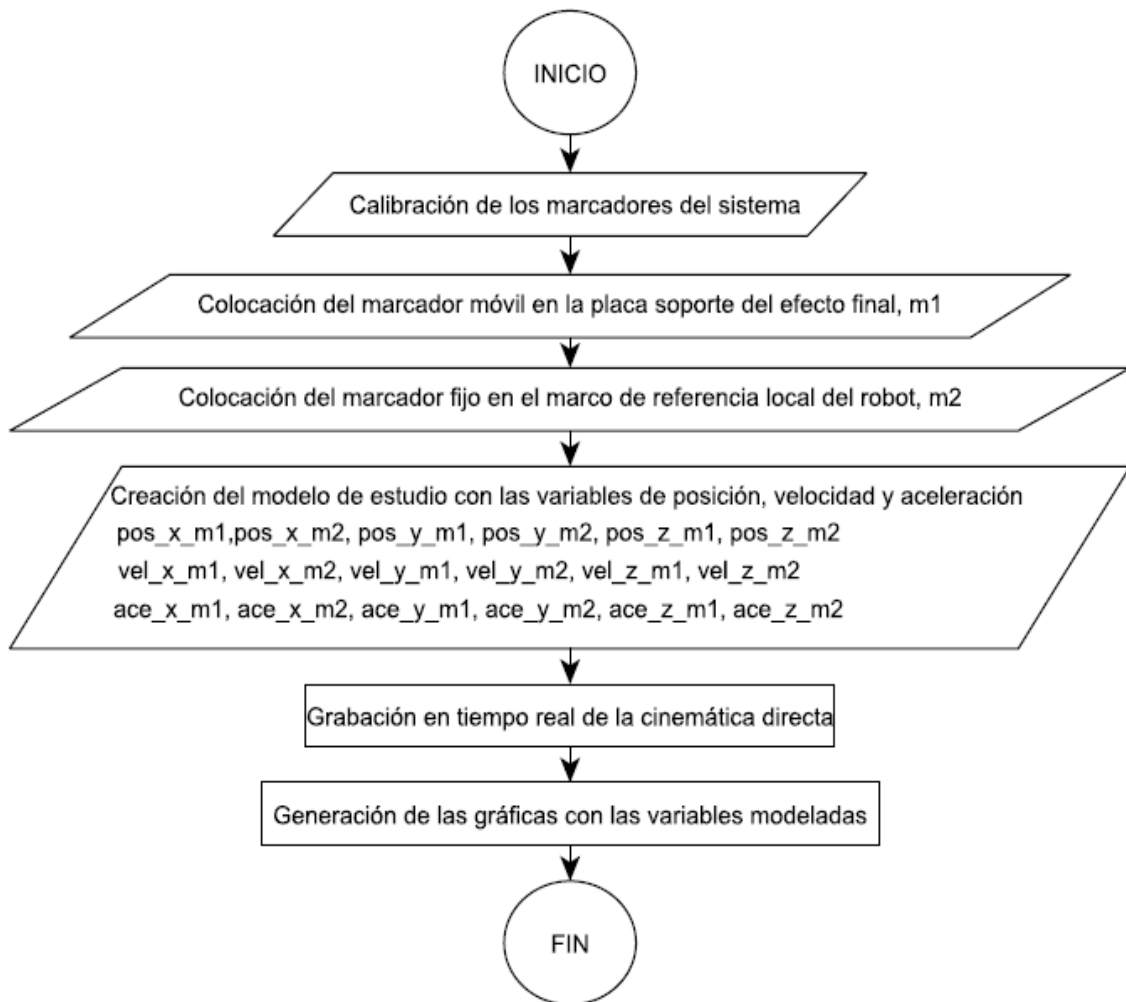


Figura 2.24 Diagrama de flujo para la captura de movimiento por fotogrametría.  
(Fuente: Autores)

### 2.3.1 Calibración de las cámaras

Previo a la captura de movimientos se debe calibrar las cámaras a partir del movimiento de los marcadores para establecer una correspondencia entre el sistema de grabación y el posicionamiento de los objetos en la realidad.



Figura 2.25 Verificación de la Calibración.  
Fuente: Autores

Se procede a grabar el puntero y automáticamente empieza la configuración de las cámaras, al detectarse el puntero se procede a iniciar grabación como se muestra en la Figura 2.26, para lo cual la persona que va a realizar la verificación debe estar posicionada en un lugar visible para las seis cámaras existentes en el laboratorio, después de presionar iniciar grabación la persona con el puntero debe realizar desplazamientos aleatorios a lo largo de los tres ejes imaginarios tratando de abarcar la mayor cantidad de espacio posible para validar la calibración.



Figura 2.26 Grabación con el puntero posicionado.  
(Fuente: Autores)

Luego de que se realice la captura correctamente, se procede a realizar la verificación y si el icono de calibración correcta se pinta de verde se procede a salir, ver Figura 2.27.



Figura 2.27 Verificación de la Calibración.  
(Fuente: Autores)

### 2.3.2 Simulación

Después de realizar la calibración se procede a colocar los marcadores en las zonas que van a ser objeto de estudio, se coloca un marcador en la placa soporte del efector final para simularlo, por tanto, este marcador será móvil, ver Figura 2.28 (izquierda) y se

ubica otro marcador fijo en la esquina superior más cercana a la posición inicial del robot cartesiano para establecer un sistema de coordenadas local. Ver Figura 2.28 (derecha).

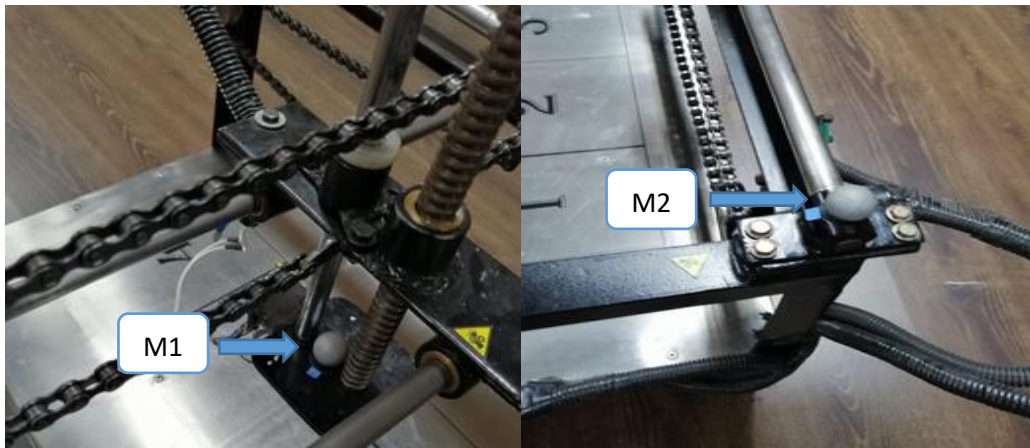


Figura 2.28 Marcador móvil (izquierda) y marcador fijo (derecha).  
(Fuente: Autores)

Realizado el posicionamiento de los marcadores, en el software se crea un nuevo estudio, dando clic en la hoja en blanco y posteriormente en el icono añadir sujeto donde se graba el nombre de la prueba a realizarse. Ver Figura 2.29.

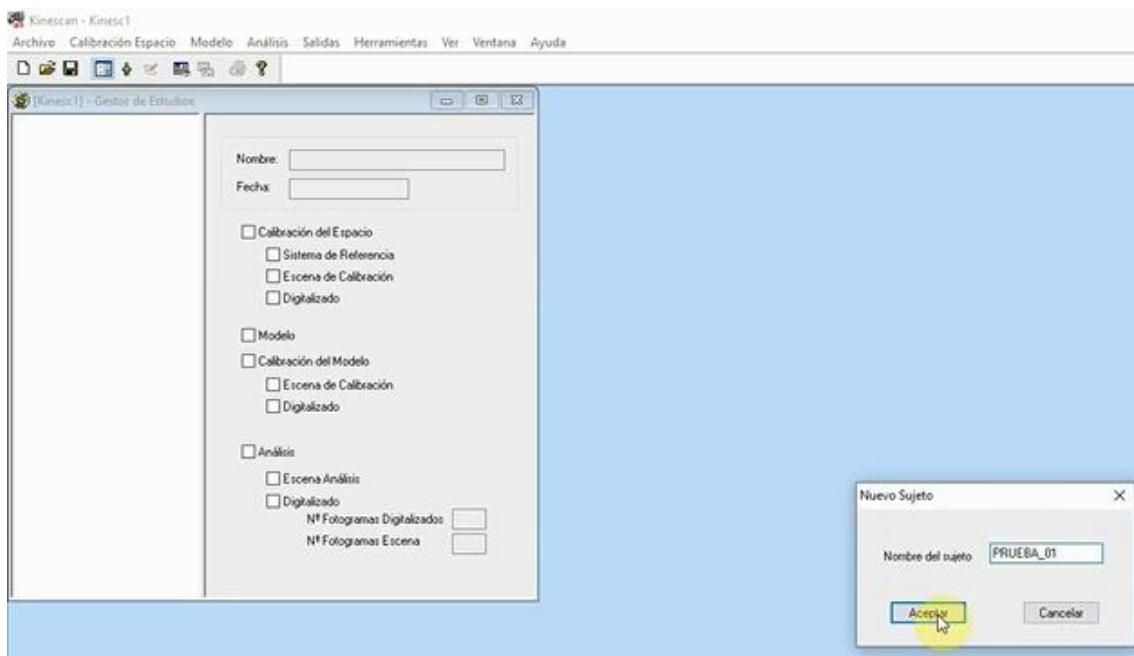


Figura 2.29 Nombre del sujeto o prueba a realizarse.  
(Fuente: Autores)

Posteriormente, se da clic en añadir estudio (Figura 2.30) y se le da un nombre, a continuación, aparece una ventana que pregunta si se quiere utilizar un patrón de estudio a lo cual se responde no, ya que se va a crear un patrón particular para el caso.

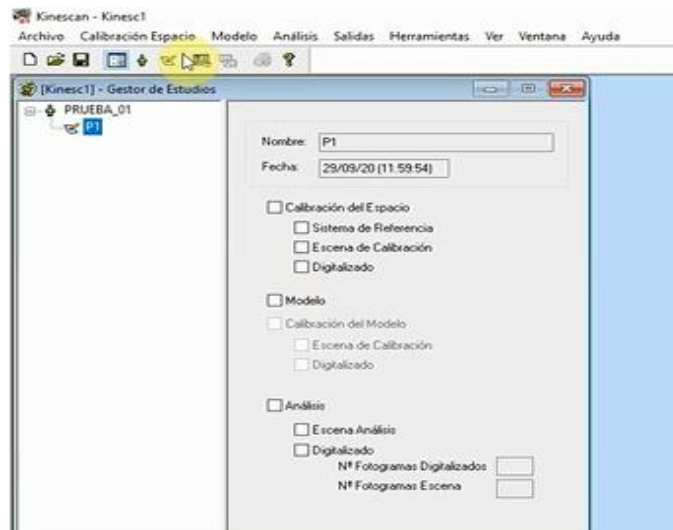


Figura 2.30 Adición de estudio P1 en el Gestor de Estudios.  
(Fuente: Autores)

Se dirige el cursor a la pestaña modelo en donde se elige crear modelo, se despliega una ventana y en la pestaña segmentos clic en el botón crear, a continuación, se despliega otra ventana en donde se nombra al segmento del modelo, como se ve en la Figura 2.31.

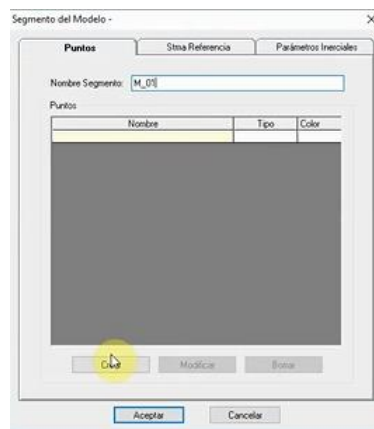


Figura 2.31 Creación del segmento del modelo.  
(Fuente: Autores)

Para crear el segmento del modelo es necesario definir los puntos conformantes del segmento, el número de puntos a utilizarse depende de la complejidad y la funcionalidad del modelo de estudio. Se da un nombre al punto, en este caso m1 y se define como un

marcador real, ver Figura 2.32. Se sigue un proceso análogo para el resto de puntos que se quieran incluir.

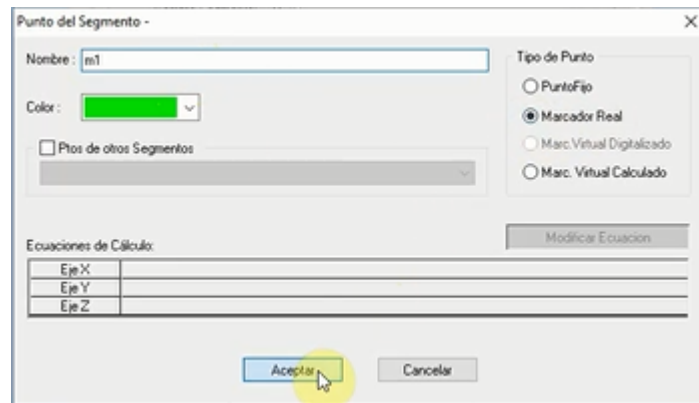


Figura 2.32 Definición de los puntos del segmento.  
(Fuente: Autores)

De vuelta en la pestaña modelo se dirige hacia la pestaña variables, luego crear y define la variable del modelo, ver Figura 2.33. Para el presente estudio se seleccionan posiciones, velocidades y aceleraciones de dos marcadores, un móvil (m1) y un fijo (m2), ver Figura 2.28.

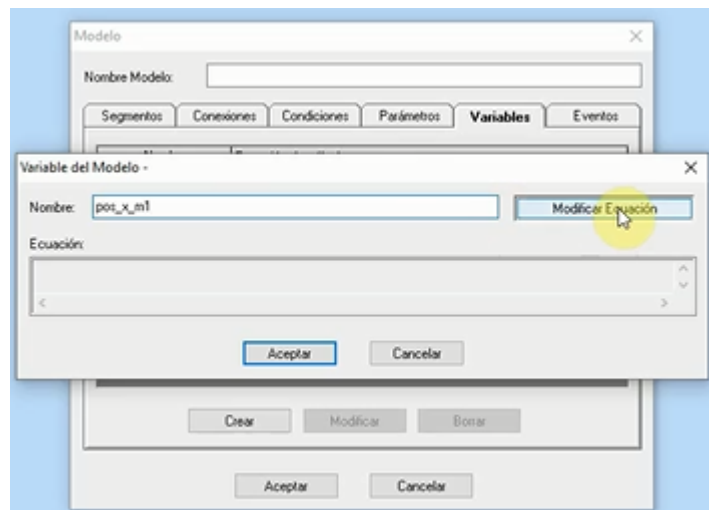


Figura 2.33 Definición de la variable posición en el eje x para el marcador m1.  
(Fuente: Autores)

Entonces, se presiona modificar ecuación como se observa en la figura anterior, se despliega una ventana llamada ecuación de la variable, se dirige a cinemática de punto y se selecciona posición, velocidad, aceleración y la componente según sea el caso, ver Figura 2.34. Siempre se refiere al sistema de referencia global SRG.

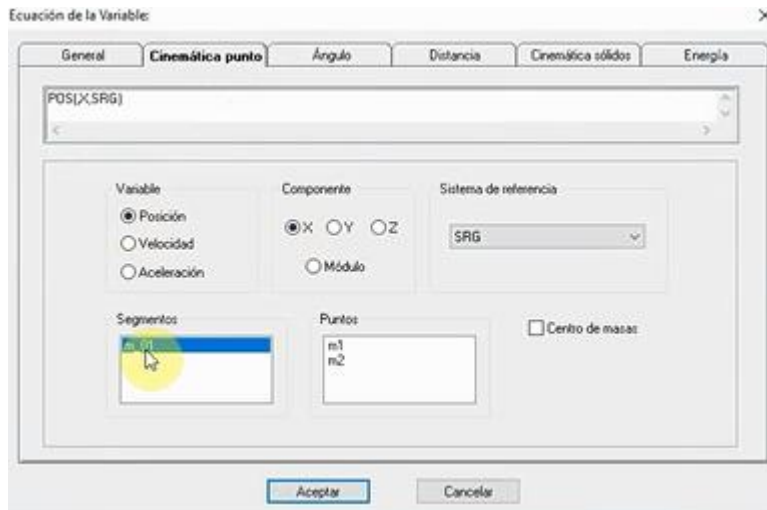


Figura 2.34 Asignación de la cinemática para cada punto.  
(Fuente: Autores)

Se siguen procesos análogos para el resto de los componentes que se quieran analizar, los cuales para el caso de la posición quedan definidos completamente como se aprecia en la Figura 2.35, es importante mencionar que tanto m1 como m2 se refieren al sistema de referencia global y para obtener los desplazamientos locales de m1 (móvil) respecto a m2 (fijo) es necesaria una resta de ambos valores respecto al centro del sistema de referencia global.

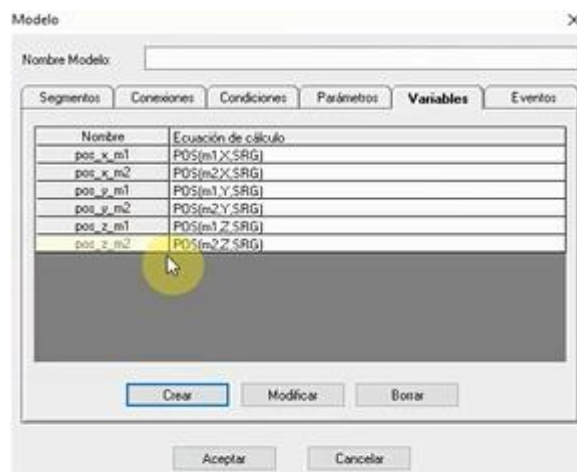


Figura 2.35 Variables de m1 y m2 para definir la posición.  
(Fuente: Autores)

Al completarse el ingreso de variables se procede al análisis, para lo cual se dirige a: Análisis>Escena Análisis>Grabar Escena. Se abre la interfaz de grabación de escena donde se puede distinguir los marcadores como puntos blancos en un espacio digitalizado 3D; previo al inicio se puede modificar la frecuencia y la duración de la grabación, ver Figura 2.36.

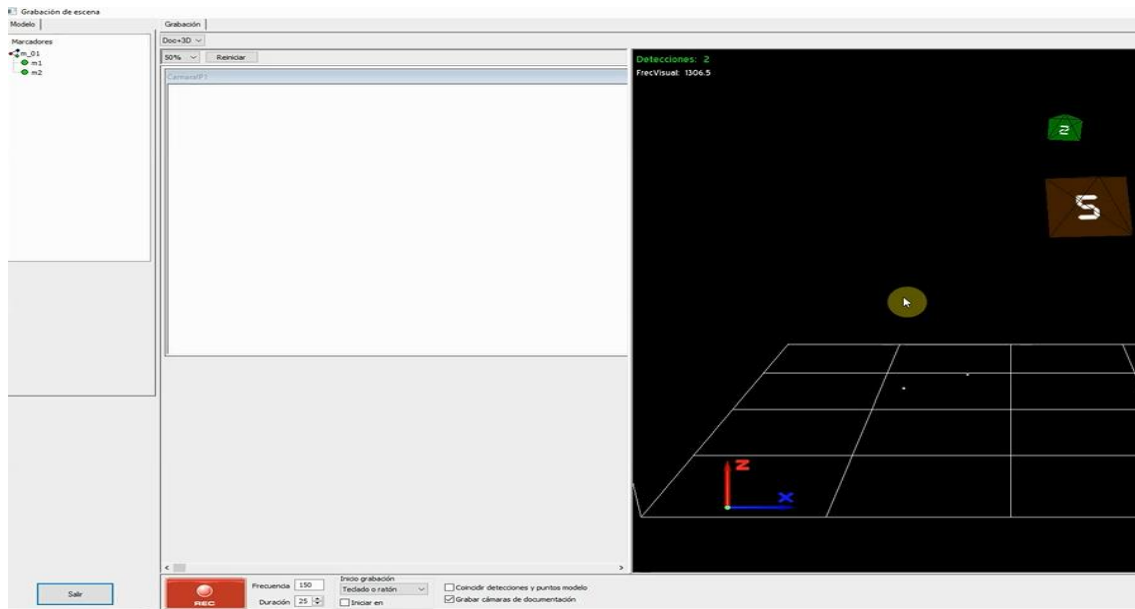


Figura 2.36 Interfaz de grabación de escena.  
(Fuente: Autores)

Para asegurar la confiabilidad de los datos en la grabación se procede a tomar tres pruebas las cuales han sido promediadas para sacar los datos de posición, velocidad y aceleración del efector final y se obtienen curvas como las que se observa en la Figura 2.37.

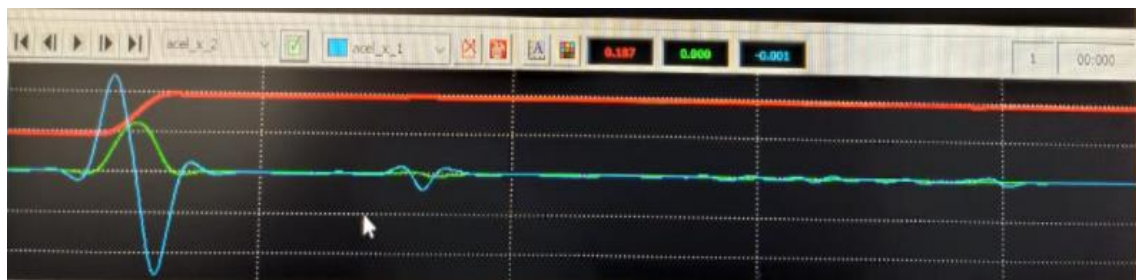


Figura 2.37 Gráficas de las variables en función del tiempo.  
(Fuente: Autores)

Después de realizar la grabación se asegura que los marcadores hayan sido detectados, se presiona aceptar. Ahora para la visualización de las gráficas se presiona Salidas>Animación, en donde al presionar el botón mostrar gráfica se presenta un comparativo entre el espacio 3D y las gráficas de las variables obtenidas en función del tiempo.

Los datos de las gráficas generadas se pueden exportar para su tratamiento en otros programas, para lo cual se ubica el cursor en Salidas>Exportar Variables y se abre una ventana en la cual se puede elegir la frecuencia de representación en puntos/segundo, como se aprecia en la Figura 2.38.



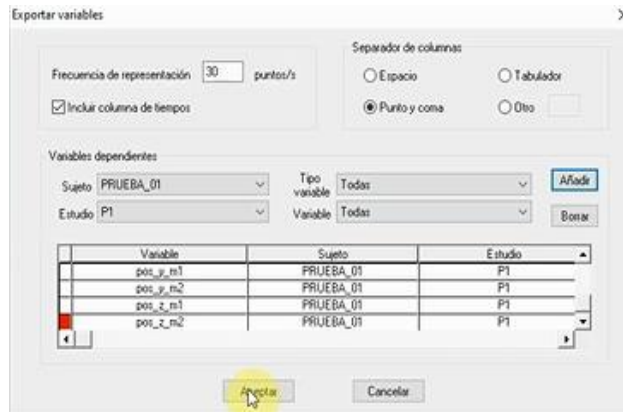


Figura 2.38 Exportación de Variables.  
(Fuente: Autores)

Los datos obtenidos se pueden exportar en un archivo .txt para poder usar en Excel o Matlab posteriormente.

## 2.4 Comparativa de resultados

El análisis de la cinemática directa del efector final del robot cartesiano es solventado en los apartados 2.1, 2.2 y 2.3.

Para validar los resultados se realizan gráficas del error cuadrático medio (ECM), a partir de la obtención de los residuales entre los valores considerados como reales (fotogrametría) y los valores considerados como teóricos (Adams y Matlab), en la Figura 2.39 se observa el código de Matlab que genera las curvas de errores para establecer la respectiva comparativa en el apartado 3.1 y la discusión detallada en el apartado 3.2.

```

%FOTOGRAMETRÍA VS MATLAB
%RESIDUOS DE LA POSICIÓN EN EL EJE X (COORDENADA GENERAL)
eqx_fm=(px_f-dx);
%ERROR CUADRÁTICO MEDIO
ECMqx_fm=sqrt((sum((eqx_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqx_fm=((sum(abs((eqx_fm./px_f)),2)/(500)))*100;
%RESIDUOS DE LA POSICIÓN EN EL EJE Y (COORDENADA GENERAL)
eqy_fm=(py_f-dy);
%ERROR CUADRÁTICO MEDIO
ECMqy_fm=sqrt((sum((eqy_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqy_fm=((sum(abs((eqy_fm./py_f)),2)/(500)))*100;

```

Figura 2.39 Cálculo del error.  
(Fuente: Autores)

### 3 RESULTADOS Y DISCUSIÓN

#### 3.1 Resultados

En esta sección se presentan los resultados gráficos de los tres análisis realizados en el capítulo 2 para las variables: posición, velocidad y aceleración, todas en función del tiempo. Para el cálculo del error porcentual se hacen comparaciones entre los datos del modelo matemático de Matlab y los datos simulados de Adams con los datos reales de fotogrametría obtenidos en el sistema Kinescan/IBV. Se usan en las gráficas subíndices para identificar el análisis con cada metodología, donde:

$q_M$ : representa el análisis en Matlab,

$q_A$ : representa la simulación en Adams y

$q_F$ : representa el estudio fotogramétrico en el sistema Kinescan/IBV.

##### 3.1.1 Desplazamiento del efector final

Las curvas de desplazamiento en los tres ejes coordenados para cada análisis se presentan en la Figura 3.1. Donde  $q$  representa el desplazamiento del efector final:

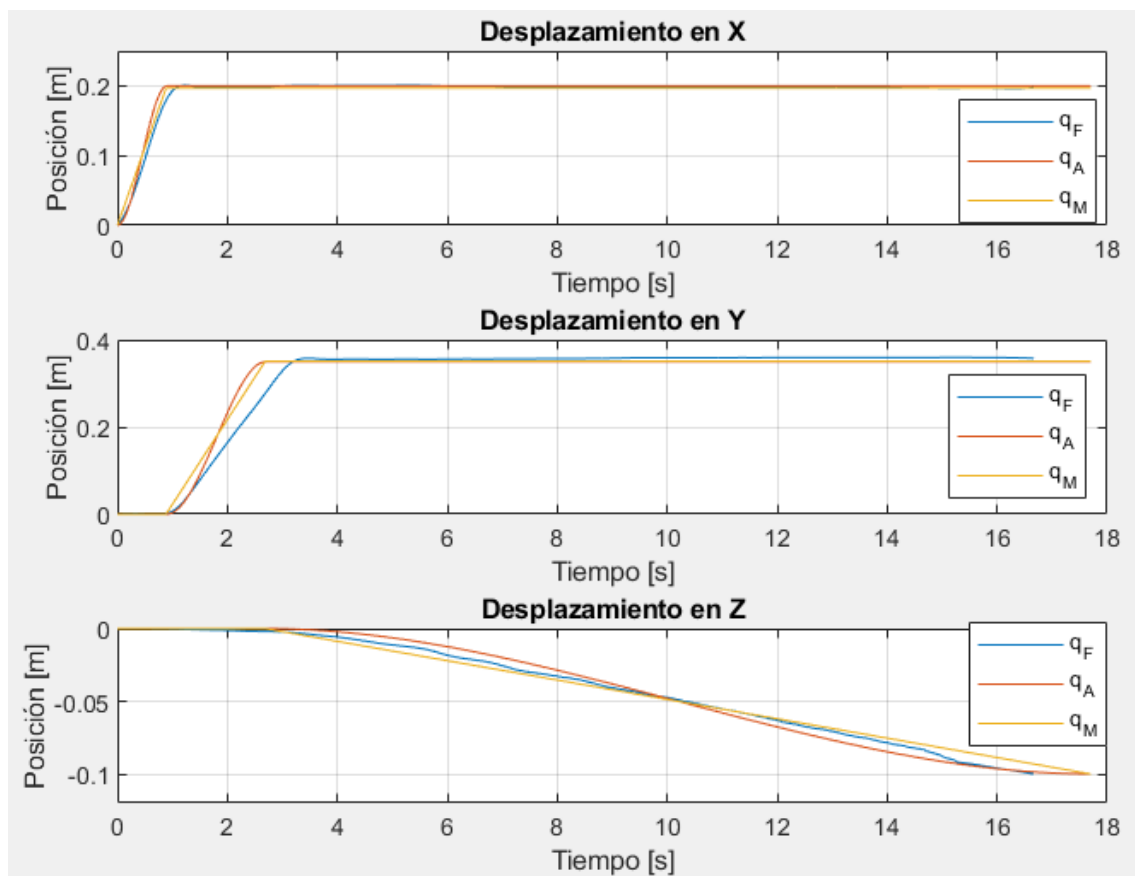


Figura 3.1 Desplazamientos para los tres análisis en los ejes coordenados.  
(Fuente: Autores)

### 3.1.2 Velocidad del efector final

Las curvas de velocidad para los tres ejes coordenados y los tres análisis realizados se presentan a continuación en la Figura 3.2, donde  $q_d$  representa la velocidad del efector final.

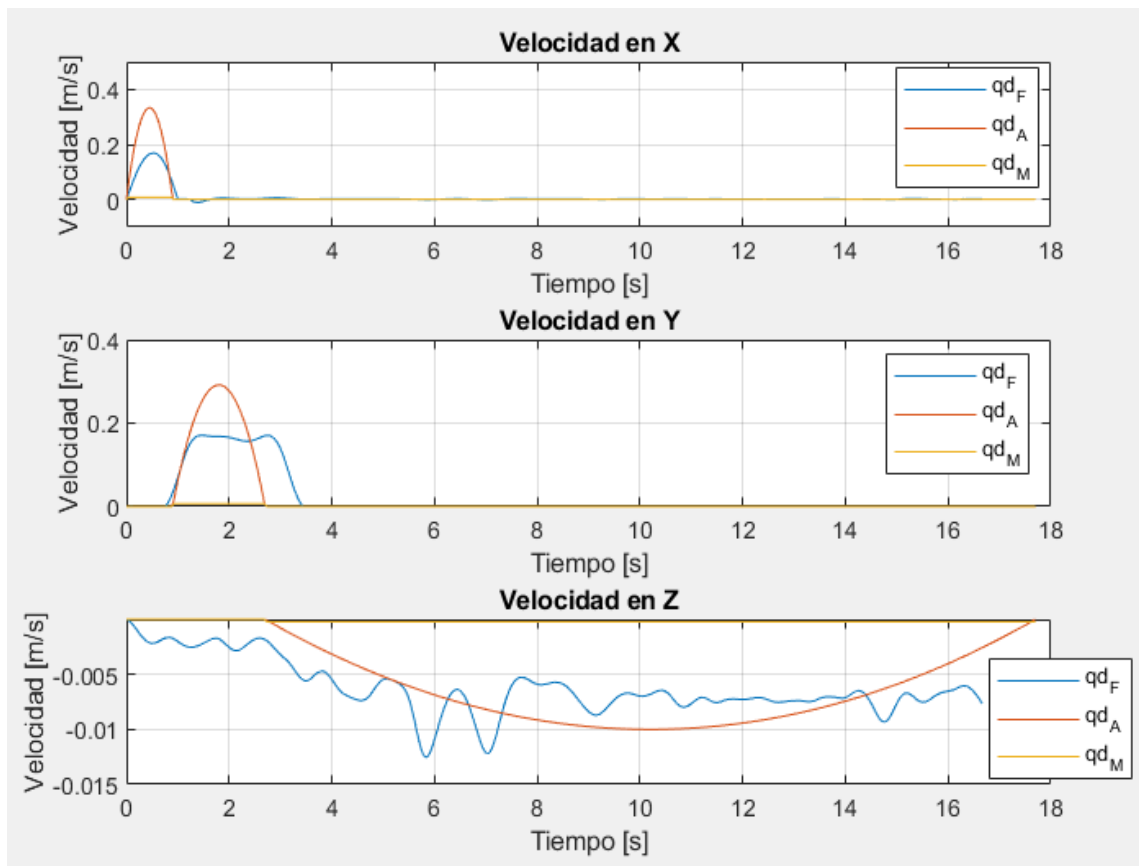


Figura 3.2 Velocidades para los tres análisis en los ejes coordenados.  
(Fuente: Autores)

### 3.1.3 Aceleración del efector final

Las curvas que representan la aceleración en cada eje coordenado y para cada análisis realizado se presentan en la Figura 3.3, donde  $q_{dd}$  representa la aceleración del efector final.

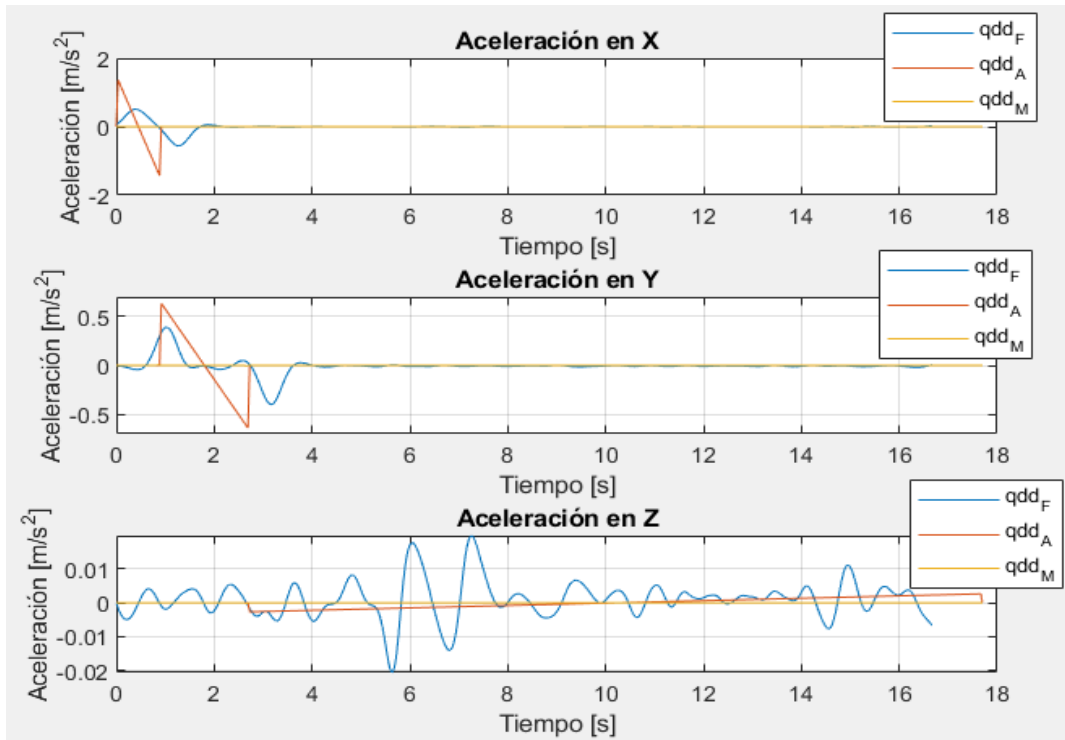


Figura 3.3 Aceleraciones para los tres análisis en los ejes coordenados.  
(Fuente: Autores)

### 3.1.4 Comparación Adams-Kinescan/IBV

Las gráficas comparativas y el error cuadrático medio (ECM), entre la simulación de Adams y la captura de movimiento en Kinescan/IBV se presentan a continuación para cada variable y eje coordenado. El desplazamiento en el eje x se observa en la Figura 3.4.

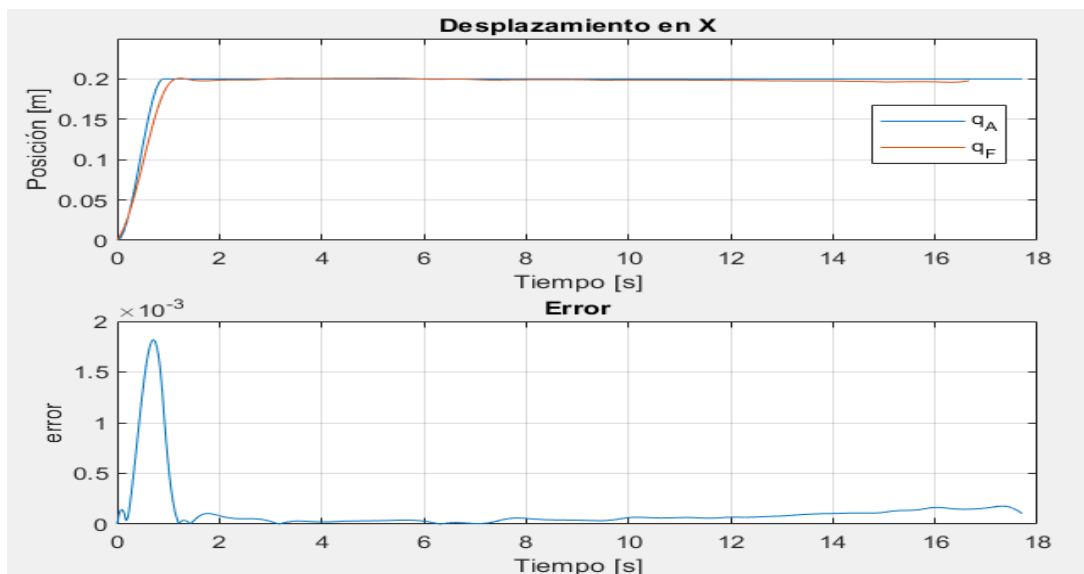


Figura 3.4 Desplazamiento en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

El desplazamiento para el eje y, con su respectivo error cuadrático medio se observa en la Figura 3.5.

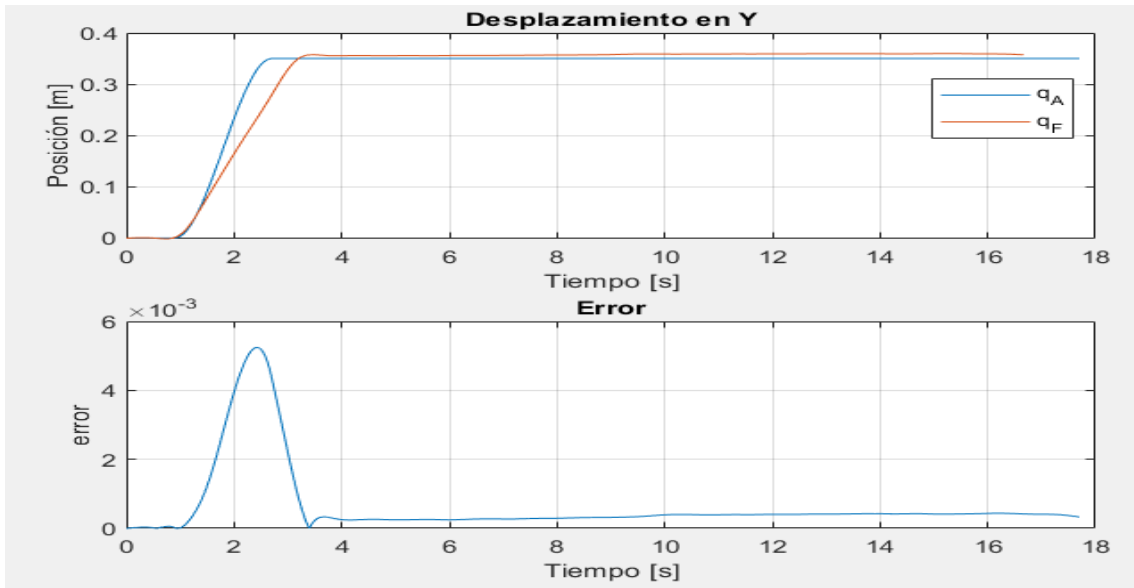


Figura 3.5 Desplazamiento en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

Del mismo modo, se presenta en la Figura 3.6 la curva de desplazamiento para el eje z y su error cuadrático medio.

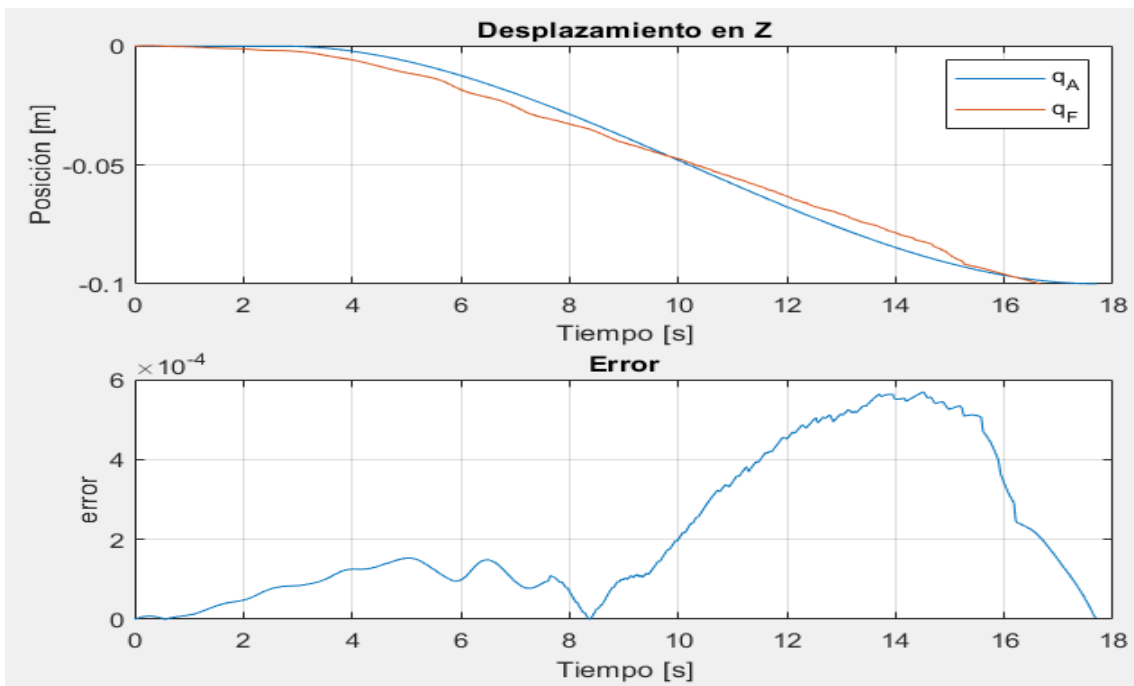


Figura 3.6 Desplazamiento en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

Las curvas para la velocidad en el eje x, y, z con su error cuadrático medio entre Adams y Kinescan/IBV se presentan a continuación en la Figura 3.7, Figura 3.8 y Figura 3.9, respectivamente.

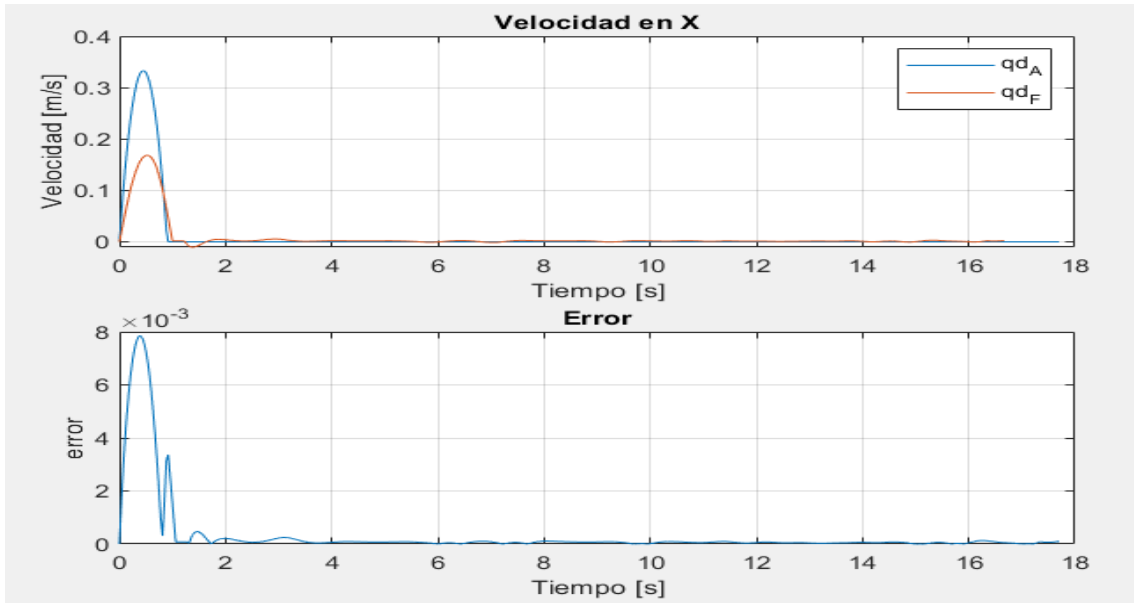


Figura 3.7 Velocidad en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

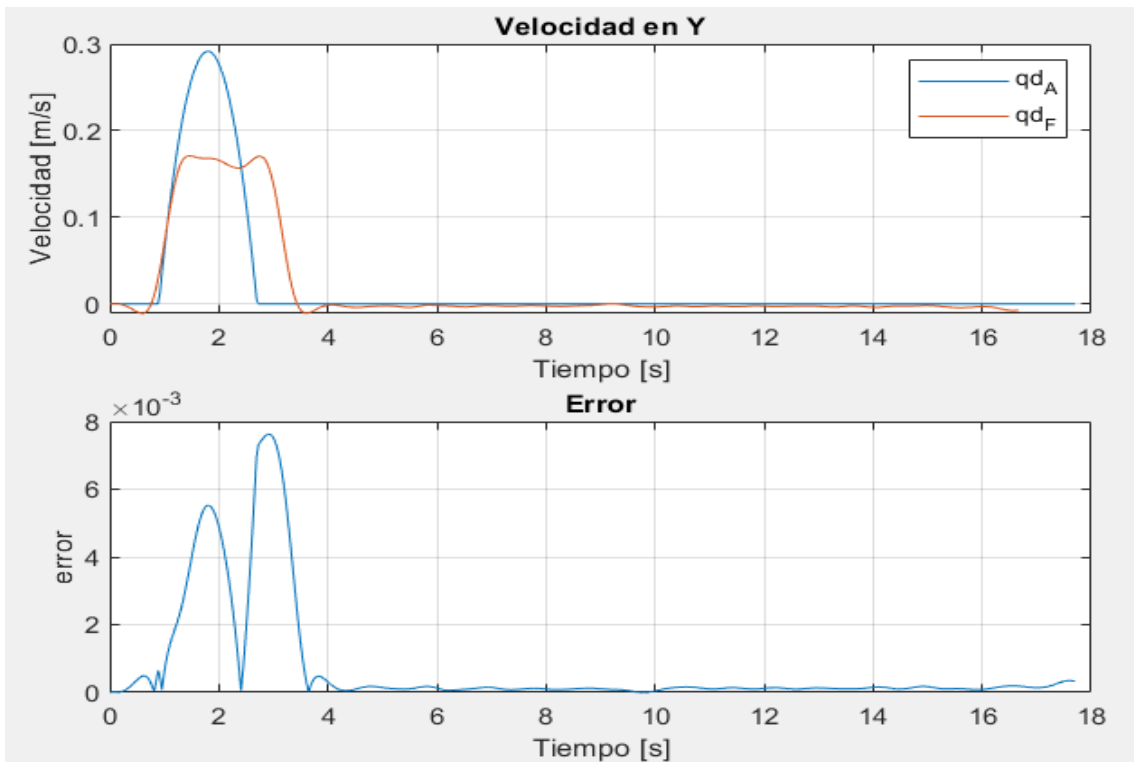


Figura 3.8 Velocidad en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

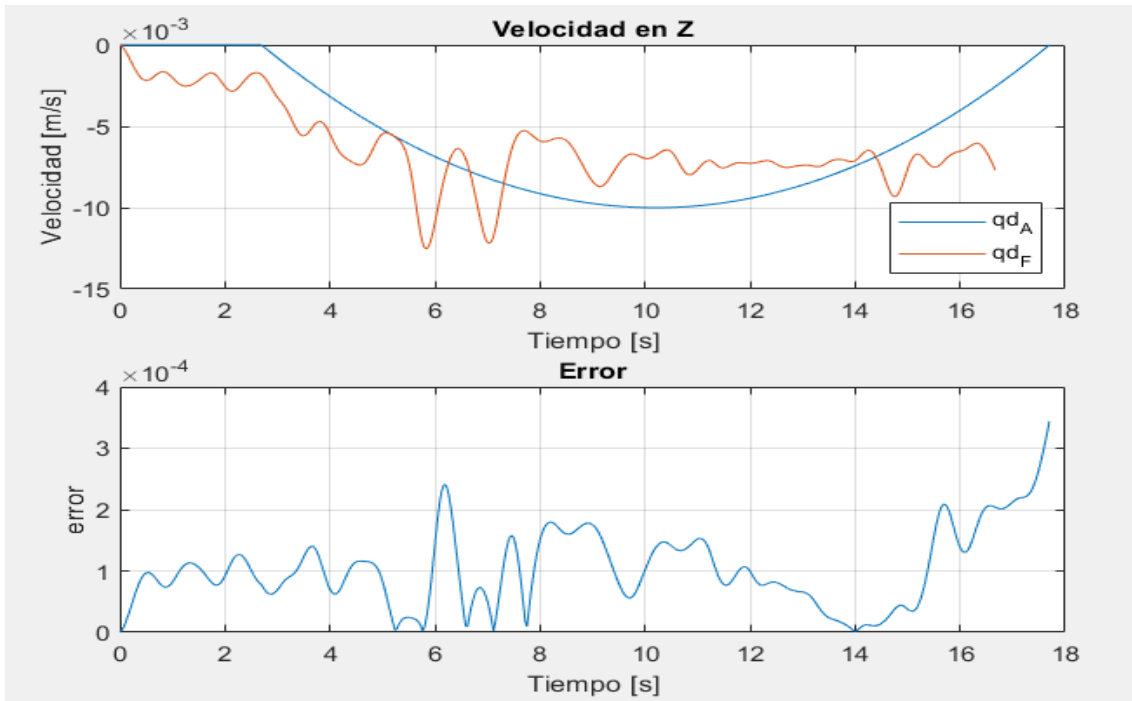


Figura 3.9 Velocidad en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

Las curvas para la aceleración en el eje x, y, z entre Adams y Kinescan/IBV con su error cuadrático medio se observan en la Figura 3.10, Figura 3.11 y Figura 3.12, respectivamente.

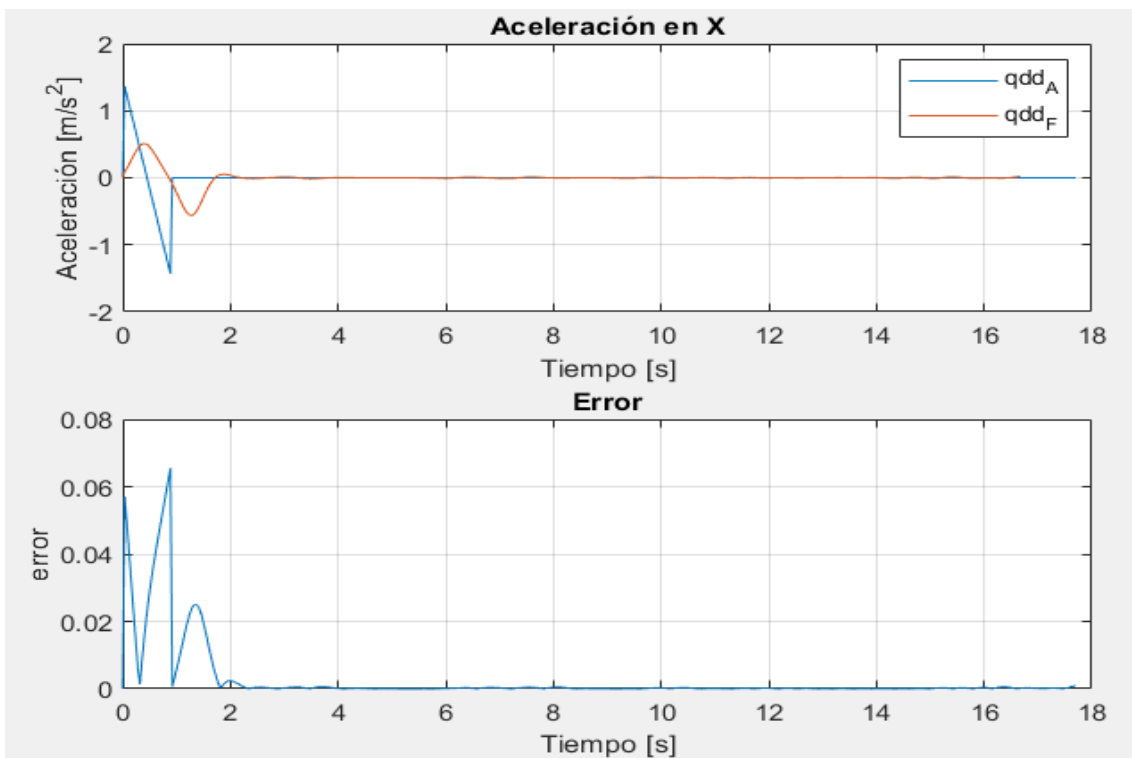


Figura 3.10 Aceleración en el eje x, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

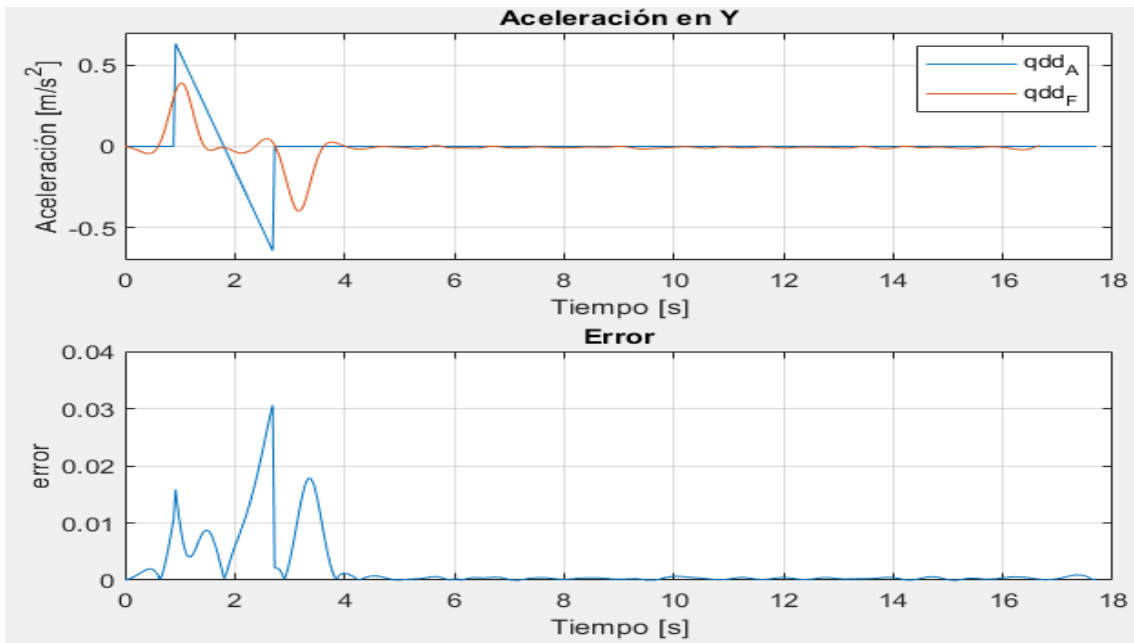


Figura 3.11 Aceleración en el eje y, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

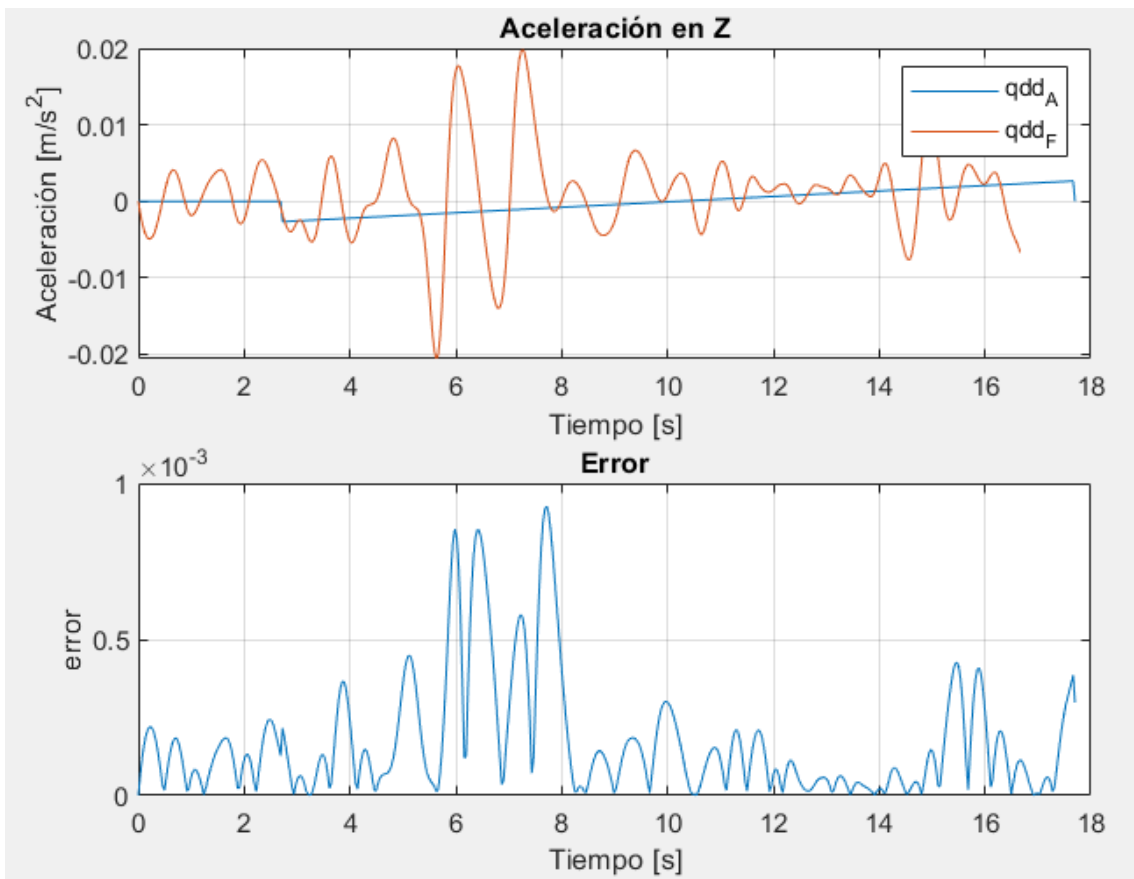


Figura 3.12 Aceleración en el eje z, Adams-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

El error cuadrático medio entre las curvas obtenidas en Adams y fotogrametría se presenta en la Tabla 3.1.



Tabla 3.1 Error cuadrático medio entre Adams y Fotogrametría.

	Eje X		Eje Y		Eje z	
	ECM	%	ECM	%	ECM	%
Posición	0,002	0,140	0,005	0,200	0,001	0,200
Velocidad	0,008	0,400	0,008	0,200	0,001	0,200
Aceleración	0,065	9,100	0,031	65,210	0,001	134,760

(Fuente: Autores)

### 3.1.5 Comparación Matlab-Kinescan/IBV

Las gráficas comparativas y el error cuadrático medio, entre el modelado en Matlab y la captura de movimiento en Kinescan/IBV se observan a continuación para cada variable y eje coordenado.

Las curvas de desplazamiento para el eje x y el eje y se observan en la Figura 3.13 y la Figura 3.14, respectivamente.

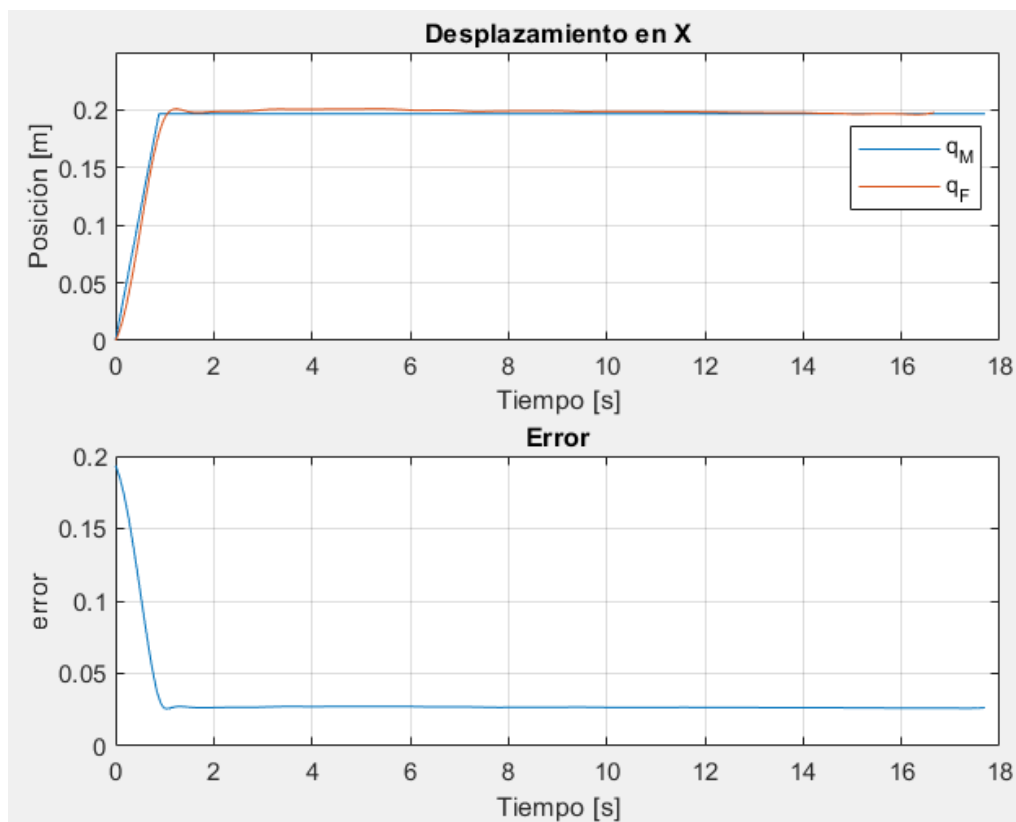


Figura 3.13 Desplazamiento en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

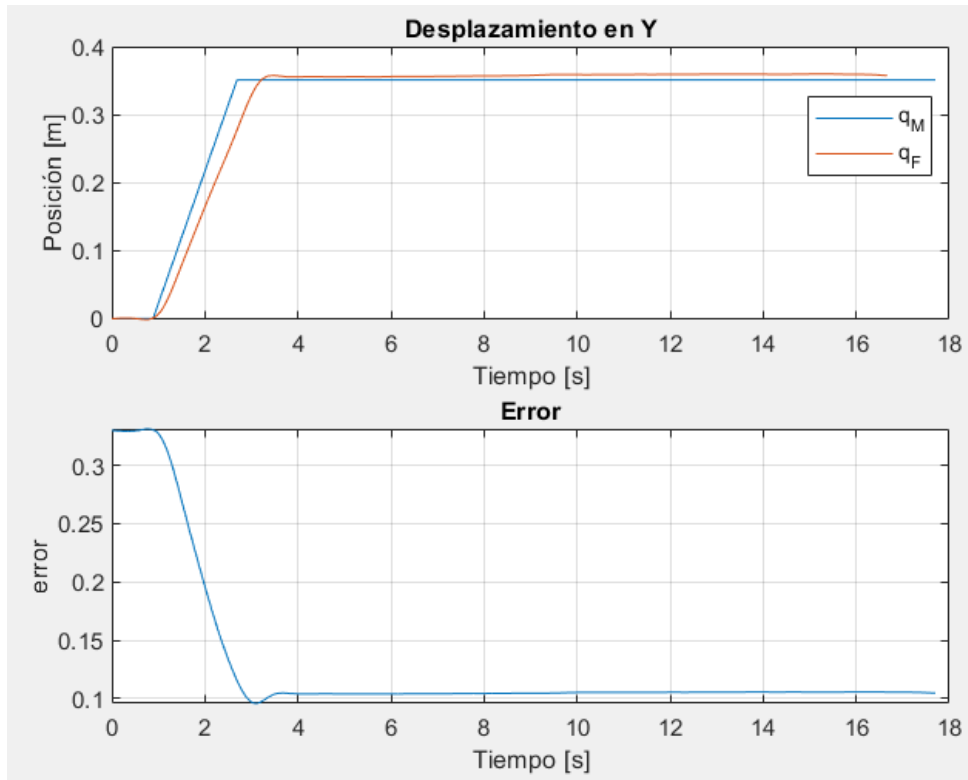


Figura 3.14 Desplazamiento en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

El desplazamiento en el eje z del análisis conjunto Matlab-Kinescan/IBV con su respectivo error cuadrático medio se aprecian en la Figura 3.15.

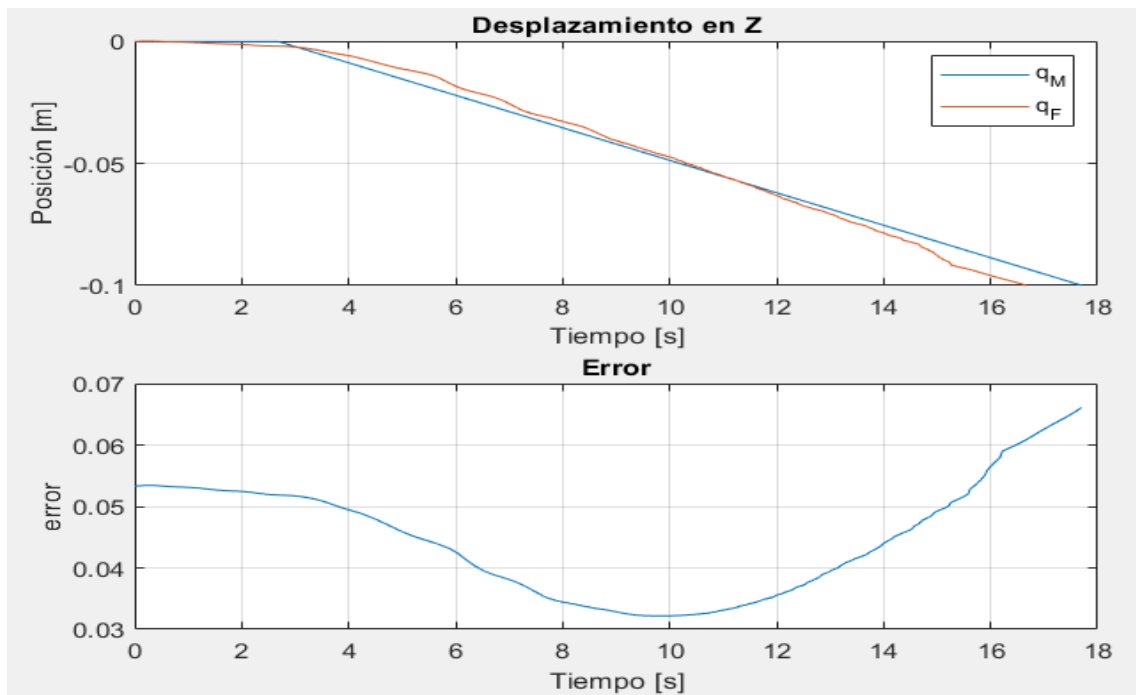


Figura 3.15 Desplazamiento en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

Las curvas de la velocidad para el eje x, y, z con su error cuadrático medio entre Matlab y Kinescan/IBV se aprecian en la Figura 3.16, Figura 3.17 y Figura 3.18, respectivamente.

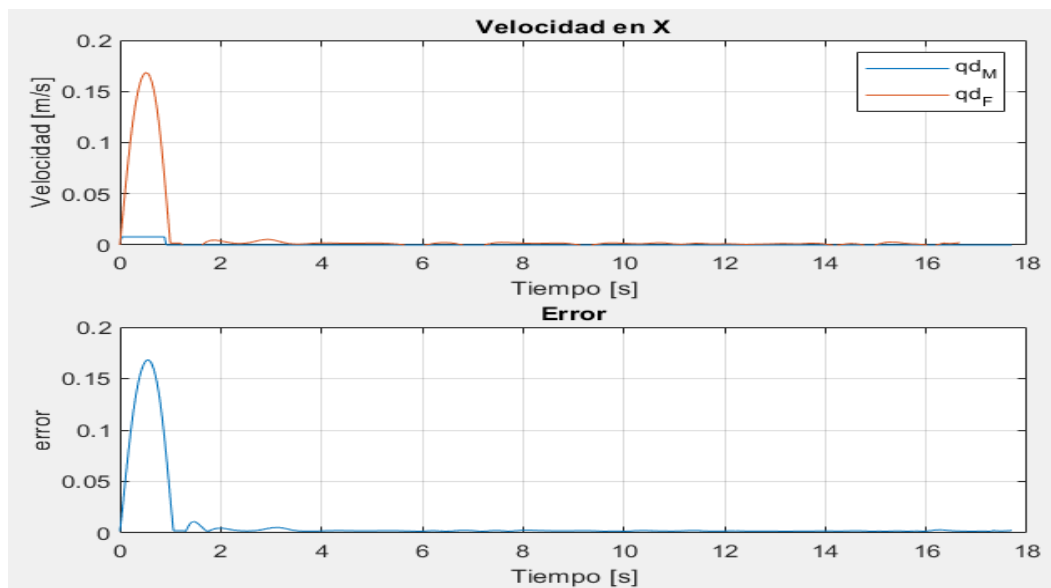


Figura 3.16 Velocidad en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

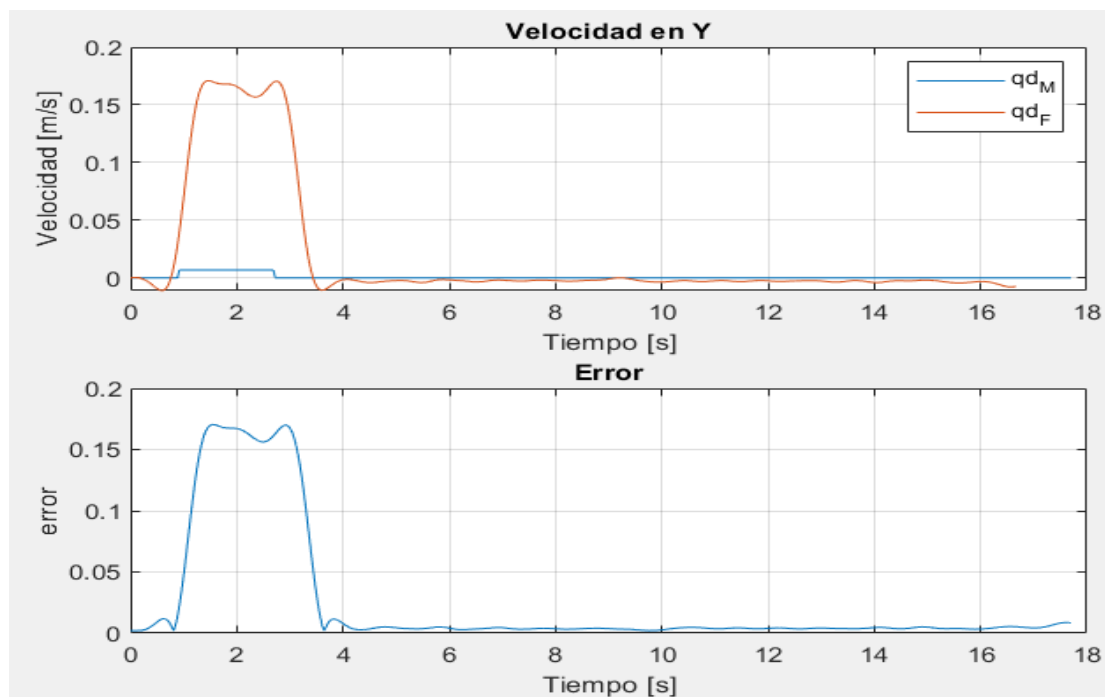


Figura 3.17 Velocidad en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

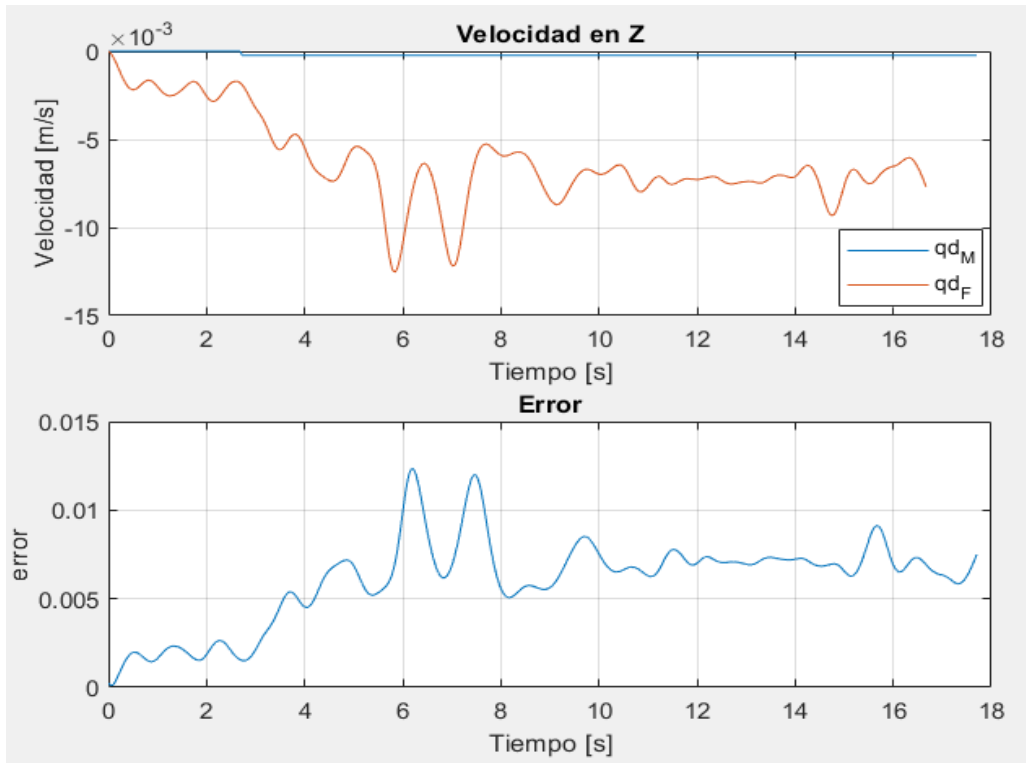


Figura 3.18 Velocidad en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

Las curvas para la aceleración en el eje x, y, z entre Matlab y Kinescan/IBV, con su error cuadrático medio, se observan en la Figura 3.19, Figura 3.20 y Figura 3.21, respectivamente.

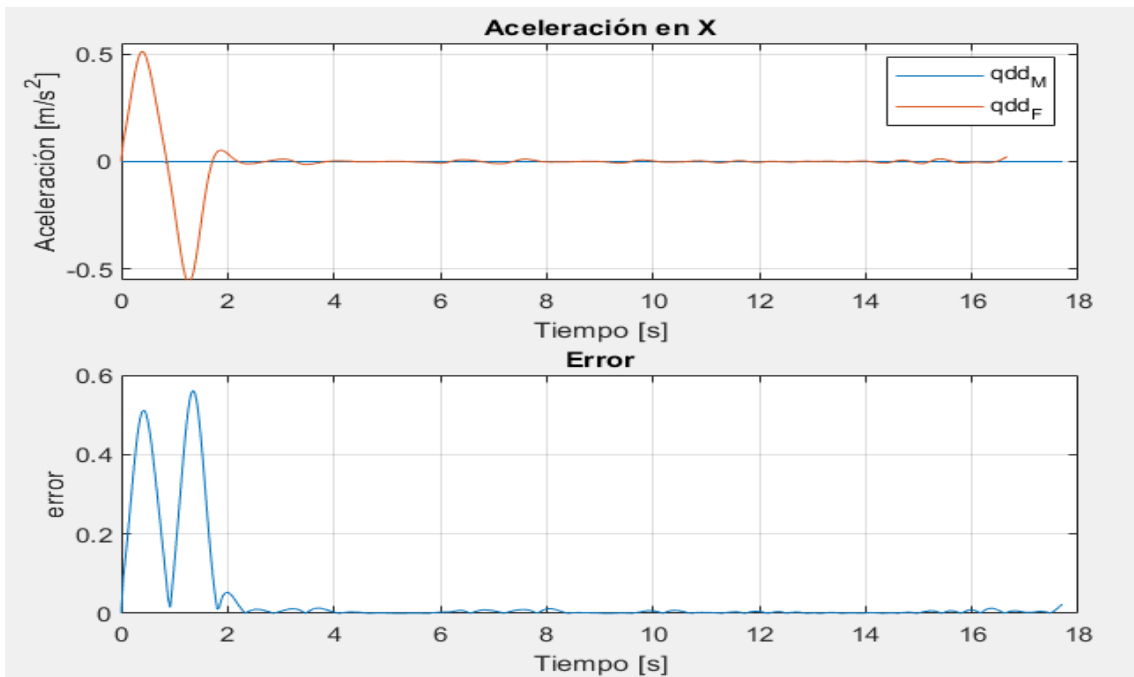


Figura 3.19 Aceleración en el eje x, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

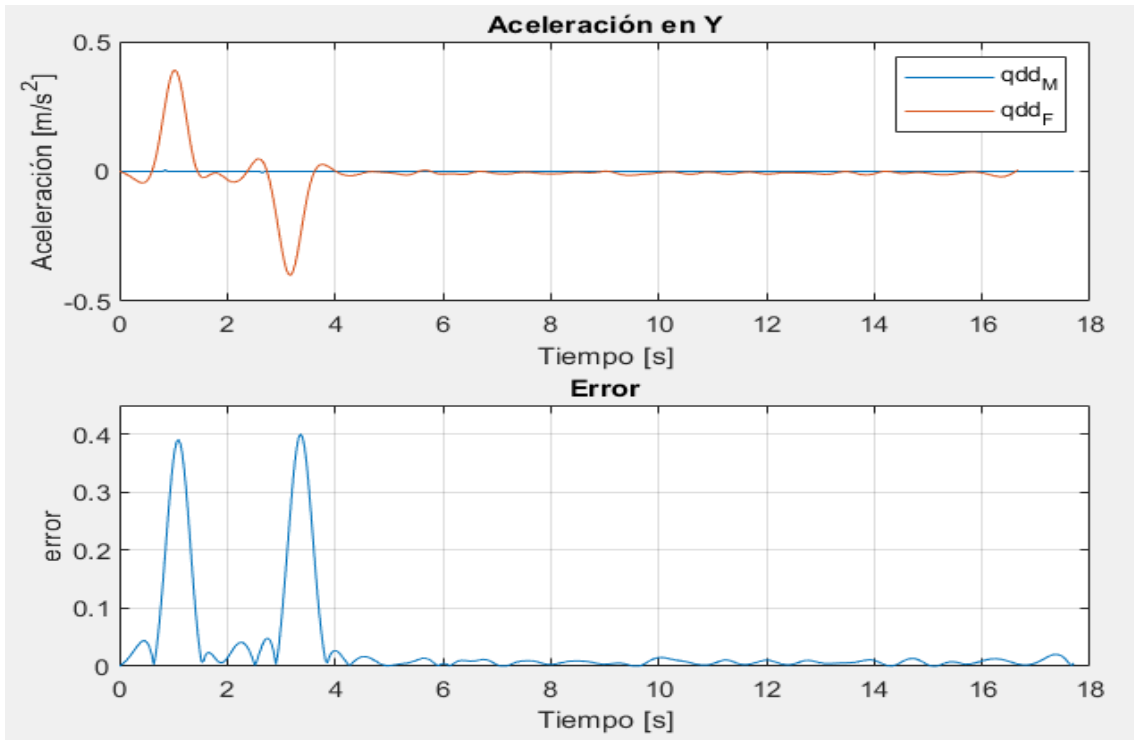


Figura 3.20 Aceleración en el eje y, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

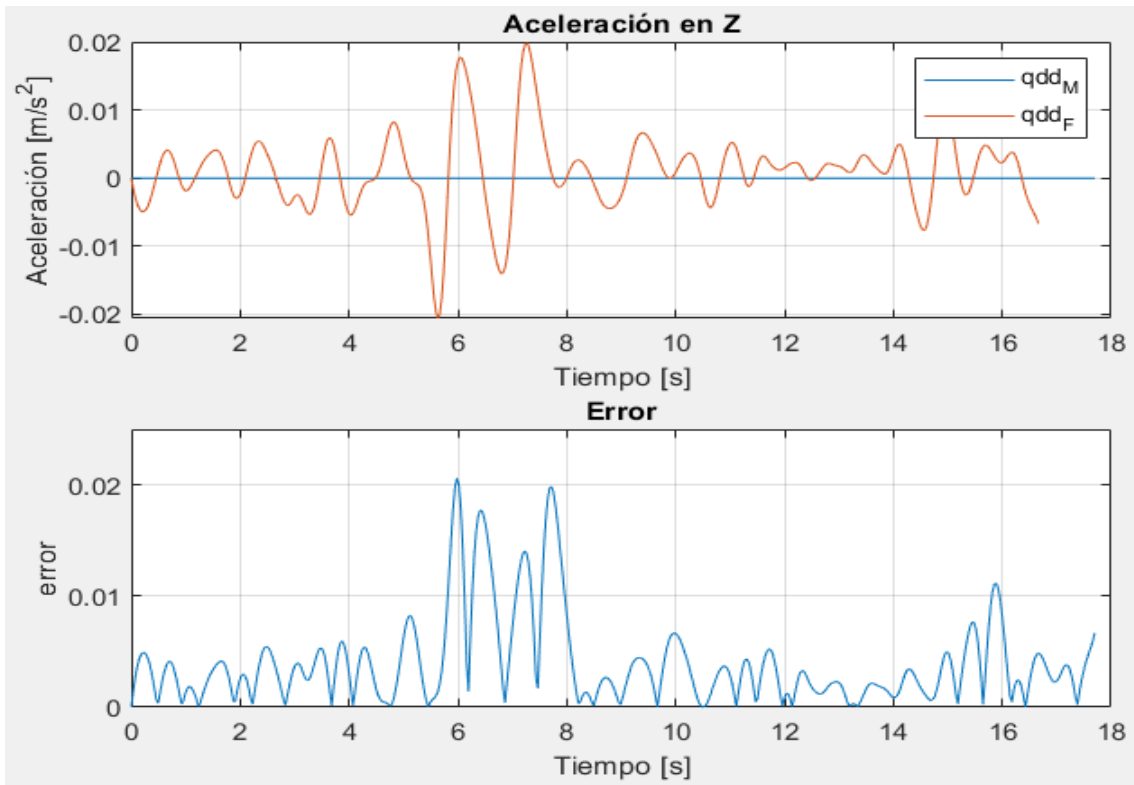


Figura 3.21 Aceleración en el eje z, Matlab-Kinescan/IBV (arriba), ECM (abajo)  
(Fuente: Autores)

El error cuadrático medio entre las curvas de Matlab y las de fotogrametría se presentan en la Tabla 3.2.

Tabla 3.2 Error cuadrático medio entre Matlab y Fotogrametría.

	Eje X		Eje Y		Eje z	
	ECM	%	ECM	%	ECM	%
Posición	0,185	0,200	0,331	0,200	0,066	0,200
Velocidad	0,165	2,150	0,165	0,220	0,012	0,200
Aceleración	0,552	9,050	0,395	61.82	0,021	154,78

(Fuente: Autores)

## 3.2 Discusión

En esta sección se establece la fiabilidad del uso del equipo de fotogrametría para la validación del modelo matemático del robot cartesiano objeto de estudio. Mediante el análisis de los errores máximos mostrados en el apartado 3.1, se realiza un análisis residual como el establecido en (Pulloquina, 2018) y tomando en cuenta los valores referenciales de error cuadrático medio obtenidos en (Misyurin, 2020), con el fin de validar el modelado teórico con la captura de movimientos en tiempo real.

### 3.2.1 Adams-Fotogrametría

En base a la Tabla 3.1, se observa que los valores de posición del efector final en el eje x presentan un error cuadrático medio de 0,002 que representa el 0,14%; para el eje y el error es 0,005 equivalente a 0,2%; y para el eje z el error es de 0,001 igual al 0,2%. Por lo tanto, la posición queda totalmente validada en el comparativo Adams-Fotogrametría, teniendo errores cuadráticos medios menores a 0,8 según lo expuesto en Misyurin.

Por otra parte, la velocidad del efector final para el eje x tiene un error cuadrático medio de 0,008 equivalente al 0,4%; para el eje y el error es de 0,008 equivalente al 0,2%; y para el eje z el error es de 0,0003 equivalente al 0,2%. Entonces, la velocidad queda completamente valida en el comparativo Adams-Fotogrametría.

Finalmente, la aceleración del efector final para el eje x presenta un error de 0,065 igual a 9,1%; para el eje y el error es de 0,031 equivalente a 65,21%; y para el eje z el error es de 0,001 igual a 134,76%. Entonces, se obtienen errores de aceleración excesivos para todos los ejes debido al manejo de cifras significativas del orden  $1e06$  en el sistema Kinescan/IBV lo cual justifica su grado de exactitud, y también a las vibraciones producidas por el motor que acciona el mecanismo gobernado por el tornillo sin fin en el eje z; como se aprecia en la Figura 3.22 y en la Figura 3.23 los errores antes descritos son puntuales y no se reflejan en toda la trayectoria, ya que en ese instante se producen los cambios de dirección del efector final.

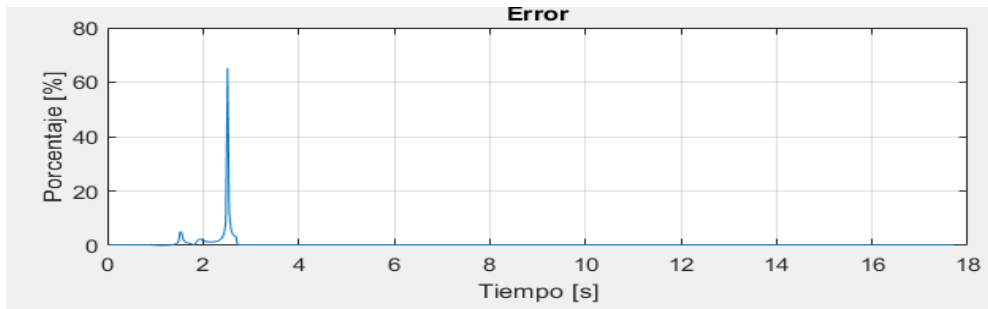


Figura 3.22 Error porcentual en el eje Y para el comparativo Adams-Fotogrametría (Fuente: Autores)

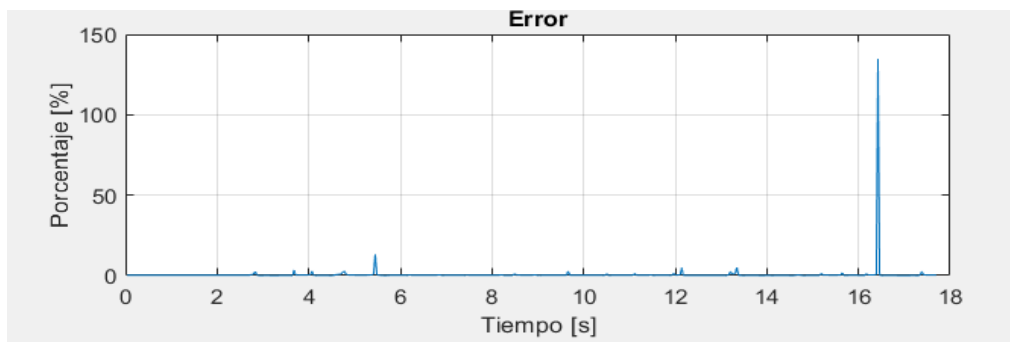


Figura 3.23 Error porcentual en el eje z para el comparativo Adams-Fotogrametría (Fuente: Autores)

Tomando en cuenta que los errores porcentuales de posición y velocidad son menores al 1% y los de aceleración quedan solventados con lo mencionado en el párrafo anterior, además que los valores de error cuadrático medio son menores a 0,1 como se evidencia en la Tabla 3.1, queda validado la simulación de Adams con la captura de movimientos por fotogrametría.

### 3.2.2 Matlab-Fotogrametría

De acuerdo con lo que se observa en la Tabla 3.2, se tiene que los datos de posición del efector final para el eje x el error cuadrático medio es de 0,185 igual al 0,2%; para el eje y el error es de 0,331 equivalente al 0,2%; y para el eje z el error es de 0,066 igual al 0,2%.

Por otra parte, la velocidad del efector final para el eje x presenta un error cuadrático medio de 0,165 igual al 2,15%; para el eje y el error es de 0,165 equivalente al 0,22%; y para el eje z se tiene un error de 0,012 igual al 0,2%.

Finalmente, la aceleración del efector final para el eje x presenta un error de 0,552 equivalente al 9,05%; para el eje y el error es de 0,395 igual al 61,82%; y para el eje z se tiene un error de 0,021 que representa el 154,78%. Por lo tanto, se observa que existen errores de aceleración con valores porcentuales altos respecto a los obtenidos en la posición y velocidad. Se fundamentan estos errores altos debido a la vibración (ruido) excesiva generada en la puesta en marcha del motor gobernante del eje z. Como se observa en la Figura 3.24 y la Figura 3.25, las cuales representan el tiempo de acción de las articulaciones; los errores porcentuales son puntuales y no se evidencian en toda la trayectoria, debido que en ese instante se producen los cambios de dirección del efector final.

El valor máximo de error cuadrático medio en el análisis es de 0,552 menor al límite de 0,8 establecido, es así que la validación del modelo matemático desarrollado en Matlab queda debidamente justificada con la captura de movimientos en tiempo real desarrolladas en el equipo de fotogrametría Kinescan/IBV.

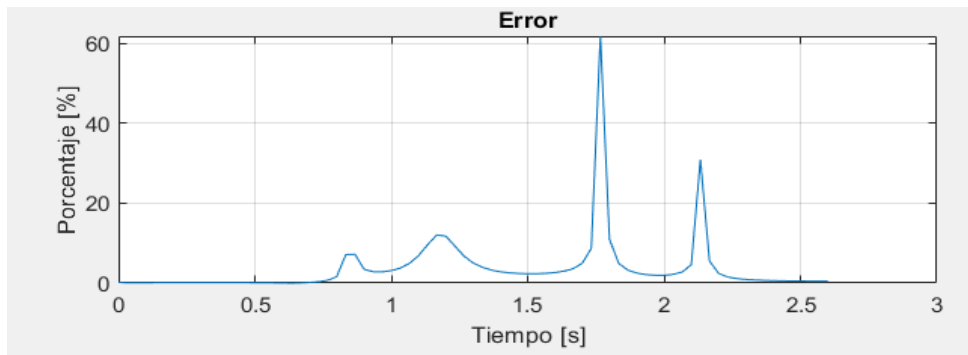


Figura 3.24 Error porcentual en el eje Y para el comparativo Matlab-Fotogrametría (Fuente: Autores)

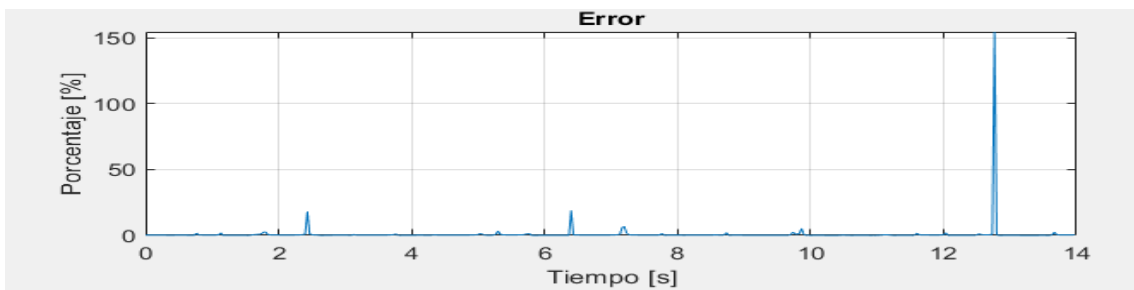


Figura 3.25 Error porcentual en el eje Z para el comparativo Matlab-Fotogrametría (Fuente: Autores)



## 4. CONCLUSIONES

- Después de comparar los resultados modelados y simulados con el equipo de fotogrametría Kinescan/IBV, se comprobó que los datos obtenidos conllevan errores mínimos con los cuales se validó la cinemática directa del robot cartesiano clasificador de objetos ferromagnéticos por peso.
- Se generó un código en Matlab mediante el uso de la librería de robótica de Peter Corke con lo cual se evidenció la simplicidad en la obtención de la matriz de Denavit-Hartenberg, y se generó un modelo matemático fundamentado en el álgebra de cuaterniones duales para la obtención de la cinemática directa del robot cartesiano.
- Se realizó una simulación del funcionamiento del robot cartesiano mediante el uso de Msc.Adams, el cual generó un prototipo virtual aprovechando el diseño CAD del robot con lo cual el programa desarrolló un análisis por elementos finitos de la cinemática del robot.
- Se realizó la captura de movimientos del efector final del robot cartesiano mediante marcadores que plasman los movimientos tridimensionales en representaciones bidimensionales (datos tabulados) que permitieron un análisis fraccionado en ejes.
- Se comprobó que la captura de movimiento a tiempo real mediante el equipo de fotogrametría Kinescan/IBV ofrece una solución integral para la validación de modelos matemáticos de sistemas mecánicos.
- Después de analizar los errores cuadráticos medios obtenidos en la comparativa de las tres metodologías, se evidenció un rango de valores menor al de 0.8 establecido en el trabajo de Misyurin (2020), con lo cual se da por validado el modelo matemático.

## Referencias Bibliográficas

- Spong, M. W., Hutchinson, S., y Vidyasagar, M. (2006). *Robot modeling and control*. (1<sup>era</sup> Ed.). New York: John Wiley & sons, INC.
- Corke, P. I. (2017). *Robotics Toolbox for Matlab*. Disponible online: <http://petercorke.com>.
- Hernández Fraile, A. (2011). *Modelado, simulación y análisis con MSC. ADAMS: aplicación a rodamientos con defectos en una pista de rodadura* (Bachelor's thesis). Recuperado de: <https://e-archivo.uc3m.es/handle/10016/13061>
- Cardoso, E., Fernández, A., Marrero-Osorio, S. A., & Guardado, P. F. (2017). *Modelos cinemático y dinámico de un robot de cuatro grados de libertad*. *Ingeniería Electrónica, Automática y Comunicaciones*, 38(3), 56-75.
- Narváez Herrera, J. F., & Sánchez Bedón, H. S. (2019). *Rediseño del robot cartesiano del laboratorio de automatización industrial de procesos mecánicos* (Bachelor's thesis, Quito, 2019.).
- Riba Romeva, C. (2002). *Diseño concurrente*. Barcelona: Edicions UPC.
- Cazco Silva, P. A., & García Coral, J. A. (2010). *Diseño y construcción de un prototipo de robot cartesiano para tareas de clasificación por peso*. (Bachelor's thesis). Quito
- Gilat, A. (2006). *Matlab: Una introducción con ejemplos prácticos*. (2<sup>da</sup> Ed.). Barcelona: Editorial Reverté.
- Adams, M. S. C., & *Documentation*, C. (2005). Msc. Software Corporation. Disponible en: <https://www.mscsoftware.com/product/adams>
- McConville, J. B. (2015). *Introduction to mechanical system simulation using Adams*. SDC publications. Ann Arbor: SDC Publications
- Martínez Montoya, J. (2018). *Modelado, simulación y análisis con MSC. ADAMS de un sistema de transmisión de par con ruedas libres*. (Trabajo fin de máster). Recuperada de: <http://bibing.us.es/proyectos/abreproy/71222/fichero/TFM-1222-MARTINEZ.pdf>
- Rutty, G. (2019). *Essentials of Autopsy Practice: Reviews, updates and advances*. Leicester: Springer
- Krauss, K. (2007). *Photogrammetry: Geometry from Images and Laser Scans*. Berlín: Walter de Gruyter.
- Gomez, L., Jaramillo, A. Ruiz, M. (2018). *Sistemas de captura y análisis de movimiento cinemático humano: una revisión sistemática*. *Prospectiva*, 16(2), 24-34. Recuperada de: <http://www.scielo.org.co/pdf/prosp/v16n2/1692-8261-prosp-16-02-00024.pdf>
- Luhmann, T., Robson, S., Kyle, S., & Boehm, J. (2013). *Close-range photogrammetry and 3D imaging*. (3<sup>ra</sup> Ed.). Berlín: Walter de Gruyter.
- Kinescan/IBV Manual de Usuario. (2014). Instituto de Biomecánica de Valencia. Versión 5.2.
- Castillo, J. C. M. (2014). *Formación Profesional Básica-Instalaciones de telecomunicaciones*. Editex.

- Quintero, E. B. (2015). *Selección, instalación, configuración y administración de los servidores multimedia*. IFCT0509. IC Editorial.
- Dembowski, K. (2003). *El Gran Libro de Hardware* (Vol. 1342). Marcombo.
- Kitagawa, M., & Windsor, B. (2012). *MoCap for artists: workflow and techniques for motion capture*. Massachusetts: Focal Press.
- Barrientos, A., Peñin, L. F., Balaguer, C., & Aracil, R. (2007). *Fundamentos de robótica* (Vol. 2, pp. 108-122). Madrid: McGraw-Hill.
- Adorno, B. V. (2011). *Two-arm manipulation: from manipulators to enhanced human-robot collaboration [Contribution à la manipulation à deux bras: des manipulateurs à la collaboration homme-robot]*. Montpellier, France.
- Clifford, M. (1871). *Preliminary Sketch of Biquaternions*. Proceedings of the London Mathematical Society, 381-395.
- Hamilton, W. R. (1884). II. On quaternions; or on a new system of. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 10-13.
- Valverde-Salazar, A., & Tsiotras, P. (2018). *Spacecraft Robot Kinematics Using Dual Quaternions*. Robotics MDPI. doi: 10.3390/robotics7040064
- Zenteno-Lara, D., & Valverde-Salazar, A. (s.f.). *Newton - Euler Recursive Algorithm Using Dual Quaternions*. Pending Publication.
- Kunusch, C. (2003). *Identificación de sistemas dinámicos*. Extraído de: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Identificacion/Parte%20III/clase20%20ident/docs/Identificacion.pdf>
- Pulloquina Zapata, J. L. (2018). *Análisis de compatibilidad del modelo dinámico de un robot paralelo tipo 3UPS+ 1RPU para rehabilitación de rodilla, mediante simulación de mecanismos en MSC-ADAMS* (Master's thesis, Quito, 2018).
- Misyurin, S. Y., Arakelian, V., & Avetisyan, A. I. (2020). *Advanced Technologies in Robotics and Intelligent Systems*. Springer International Publishing.
- Aguirre, A. (1994). *Introducción al tratamiento de series temporales*. Madrid: Díaz de Santos

## ANEXO 1

Codificación general de la validación de la cinemática directa del robot cartesiano.

```
%OBTENCIÓN DE LA CINEMÁTICA DIRECTA DEL ROBOT CARTESIANO CLASIFICADO
%DE OBJETOS FERROMAGNÉTICOS
clc;
clear all
    %CREACIÓN DE LA TABLA DE PARÁMETROS D-H
%DEFINICIÓN DE LOS ESLABONES DEL ROBOT CARTESIANO
L1=Link([0 0 0.2 -pi/2 1]);L1.qlim=[-5 5]
L2=Link([0 0 0 pi/2 1]);L2.qlim=[-5 5]
L3=Link([0 0 0 0 1]);L3.qlim=[-5 5]

% VISUALIZACIÓN DE LA TABLA
RCO=SerialLink([L1 L2 L3])

% INGRESO DE LAS VARIABLES qi EN METROS
q1=[0 0.35 -0.1]

% EVALUACIÓN DEL PUNTO FINAL DEL EFECTOR
T=RCO.fkine(q1)

% Variables articulares
q = [0
      0
      0];

% Screw vector
xi = [0 0 0
      0 0 0
      0 0 0
      0 0 0
      0 0 0
      1 0 0
      0 0 0
      0 1 1];

% Creación de la Matriz Denavit - Hartenberg
DH = denavitHartenberg(q, 0);

% POSICIÓN DEL EFECTOR FINAL
%Creación de la trayectoria
q = [signal([0 0.2 0.2], [0 0.9 16.8], 0.0354)
      signal([0 0 0.35 0.35], [0 0.9 1.8 15], 0.0354)
      signal([0 0 -0.1], [0 2.7 15], 0.0354)];
t1=0:0.0354:17.7;
dx=q(1,:);
dy=q(2,:);
dz=q(3,:);

% VELOCIDAD DUAL DEL EFECTOR FINAL
[~, n] = size(q);
W = zeros(8, n);
for i = 2:n
    W(:, i) = endEffectorVelocityDQ(q(:, i), [diff(q(1, i - 1 : i))
diff(q(2, i - 1 : i)) diff(q(3, i - 1 : i))]', denavitHartenberg(q(:,
i), 0), xi, 4);
end
vx=W(6,:);
vy=W(7,:);
vz=W(8,:);
```

```

% ACELERACIÓN DUAL DEL EFECTOR FINAL
acx = diff(q(1, :), 2);
acy = diff(q(2, :), 2);
acz = diff(q(3, :), 2);
[~, n] = size(acx);
A = zeros(8, n+2);
for i = 2 : n
    A(:, i) = endEffectorAccelerationDQ(q(:, i), [diff(q(1, i - 1 :
i)) diff(q(2, i - 1 : i)) diff(q(3, i - 1 : i))]', [acx(i) acy(i)
acz(i)]', denavitHartenberg(q(:, i), 0), xi, 4);
end
ax=A(6,:);
ay=A(7,:);
az=A(8,:);

%GRAFICAS OBTENIDAS EN ADAMS/VIEW
%% Setup the Import Options
opts = spreadsheetImportOptions("NumVariables", 10);

% Specify sheet and range
opts.Sheet = "Hojal";
opts.DataRange = "B3:K503";

% Specify column names and types
opts.VariableNames = ["Time", "Displacement X", "Displacement Y",
"Displacement Z", "Velocity X", "Velocity Y", "Velocity Z",
"Acceleration X", "Acceleration Y", "Acceleration Z"];
opts.SelectedVariableNames = ["Time", "Displacement X",
"Displacement Y", "Displacement Z", "Velocity X", "Velocity Y",
"Velocity Z", "Acceleration X", "Acceleration Y", "Acceleration Z"];
opts.VariableTypes = ["double", "double", "double", "double",
"double", "double", "double", "double", "double", "double"];

% Import the data
tbl = readtable("C:\Users\user\OneDrive - Escuela Politécnica
Nacional\tesis\Matlab\rvctools\Graficas_Adams.xlsx", opts, "UseExcel",
false);

%% Convert to output type
Time = tbl.Time;
Displacement X = tbl.Displacement X;
Displacement Y = tbl.Displacement Y;
Displacement Z = tbl.Displacement Z;
Velocity X = tbl.Velocity X;
Velocity Y = tbl.Velocity Y;
Velocity Z = tbl.Velocity Z;
Acceleration X = tbl.Acceleration X;
Acceleration Y = tbl.Acceleration Y;
Acceleration Z = tbl.Acceleration Z;

%% Clear temporary variables
clear opts tbl

%%GRAFICAS OBTENIDAS EN EL EQUIPO DE FOTOGRAMETRÍA
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%   Workbook:
C:\Users\user\Documents\TESIS\Cinemática\Graficas_Fotogrametria.xlsx
%   Worksheet: Hojal
%

```

```

% Auto-generated by MATLAB on 04-Nov-2020 17:44:17

%% Setup the Import Options
opts = spreadsheetImportOptions("NumVariables", 10);

% Specify sheet and range
opts.Sheet = "Hojal";
opts.DataRange = "A2:J502";

% Specify column names and types
opts.VariableNames = ["Tiempo", "Acel_x", "Acel_y", "Acel_z", "Pos_x",
"Pos_y", "Pos_z", "Vel_x", "Vel_y", "Vel_z"];
opts.SelectedVariableNames = ["Tiempo", "Acel_x", "Acel_y", "Acel_z",
"Pos_x", "Pos_y", "Pos_z", "Vel_x", "Vel_y", "Vel_z"];
opts.VariableTypes = ["double", "double", "double", "double",
"double", "double", "double", "double", "double", "double"];

% Import the data
tbl =
readtable("C:\Users\user\Documents\TESIS\Cinemática\Graficas_Fotogrametria.xlsx", opts, "UseExcel", false);

%% Convert to output type
Tiempo = tbl.Tiempo;
Acel_x = tbl.Acel_x;
Acel_y = tbl.Acel_y;
Acel_z = tbl.Acel_z;
Pos_x = tbl.Pos_x;
Pos_y = tbl.Pos_y;
Pos_z = tbl.Pos_z;
Vel_x = tbl.Vel_x;
Vel_y = tbl.Vel_y;
Vel_z = tbl.Vel_z;

%% Clear temporary variables
clear opts tbl

%GRÁFICAS DE DESPLAZAMIENTO DEL EFECTOR FINAL
%POSICIÓN EN X

figure (1)
subplot(3,1,1)
plot(Tiempo, Pos_x, Time, Displacement_X, t1, dx)
grid on
title('Desplazamiento en X')
legend('q_F', 'q_A', 'q_M')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([0 0.25])
subplot(3,1,2)
%POSICIÓN EN Y
plot(Tiempo, Pos_y, Time, Displacement_Y, t1, dy)
grid on
title('Desplazamiento en Y')
legend('q_F', 'q_A', 'q_M')|
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([0 0.4])
subplot(3,1,3)
%POSICIÓN EN Z
plot(Tiempo, Pos_z, Time, Displacement_Z, t1, dz)
grid on

```

```

title('Desplazamiento en Z')
legend('q_F', 'q_A', 'q_M')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([-0.12 0])

%GRÁFICAS DE VELOCIDAD DEL EFECTOR FINAL
%VELOCIDAD EN X
figure (2)
subplot(3,1,1)
plot(Tiempo, Vel_x, Time, Velocity_X, t1, vx)
grid on
title('Velocidad en X')
legend('qd_F', 'qd_A', 'qd_M')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
ylim([-0.1 0.5])
subplot(3,1,2)
%VELOCIDAD EN Y
plot(Tiempo, Vel_y, Time, Velocity_Y, t1, vy)
grid on
title('Velocidad en Y')
legend('qd_F', 'qd_A', 'qd_M')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
ylim([0 0.4])
subplot(3,1,3)
%VELOCIDAD EN Z
plot(Tiempo, Vel_z, Time, Velocity_Z, t1, vz)
grid on
title('Velocidad en Z')
legend('qd_F', 'qd_A', 'qd_M')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
ylim([-0.015 0])
%GRÁFICAS DE ACELERACIÓN DEL EFECTOR FINAL
%ACELERACIÓN EN X
figure (3)
subplot(3,1,1)
plot(Tiempo, Acel_x, Time, Acceleration_X, t1, ax)
grid on
title('Aceleración en X')
legend('qdd_F', 'qdd_A', 'qdd_M')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
subplot(3,1,2)
%ACELERACIÓN EN Y
plot(Tiempo, Acel_y, Time, Acceleration_Y, t1, ay)
grid on
title('Aceleración en Y')
legend('qdd_F', 'qdd_A', 'qdd_M')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
ylim([-0.7 0.7])
subplot(3,1,3)
%ACELERACIÓN EN Z
plot(Tiempo, Acel_z, Time, Acceleration_Z, t1, az)
grid on
title('Aceleración en Z')
legend('qdd_F', 'qdd_A', 'qdd_M')
xlabel('Tiempo [s]')

```

```

ylabel('Aceleración [m^2/s]')

%CÁLCULO DEL ERROR CUADRÁTICO MEDIO

%FOTOGRAMETRÍA VS ADAMS
%RESIDUOS DE LA POSICIÓN EN EL EJE X (COORDENADA GENERAL)
eqx=(Pos_x-Displacement X);
%ERROR CUADRÁTICO MEDIO
ECMqx=sqrt((sum((eqx.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqx=((sum(abs((eqx./Pos_x)),2)/(500)))*100;
%RESIDUOS DE LA POSICIÓN EN EL EJE Y (COORDENADA GENERAL)
eqy=(Pos_y-Displacement Y);
%ERROR CUADRÁTICO MEDIO
ECMqy=sqrt((sum((eqy.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqy=((sum(abs((eqy./Pos_y)),2)/(500)))*100;
%RESIDUOS DE LA POSICIÓN EN EL EJE Z (COORDENADA GENERAL)
eqz=(Pos_z-Displacement Z);
%ERROR CUADRÁTICO MEDIO
ECMqz=sqrt((sum((eqz.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqz=((sum(abs((eqz./Pos_z)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE X (COORDENADA GENERAL)
eqdx=(Vel_x-Velocity X);
%ERROR CUADRÁTICO MEDIO
ECMqdx=sqrt((sum((eqdx.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdx=((sum(abs((eqdx./Vel_x)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE Y (COORDENADA GENERAL)
eqdy=(Vel_y-Velocity Y);
%ERROR CUADRÁTICO MEDIO
ECMqdy=sqrt((sum((eqdy.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdy=((sum(abs((eqdy./Vel_y)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE Z (COORDENADA GENERAL)
eqdz=(Vel_z-Velocity Z);
%ERROR CUADRÁTICO MEDIO
ECMqdz=sqrt((sum((eqdz.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdz=((sum(abs((eqdz./Vel_z)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE X (COORDENADA GENERAL)
eqddx=(Acel_x-Acceleration X);
%ERROR CUADRÁTICO MEDIO
ECMqddx=sqrt((sum((eqddx.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddx=((sum(abs((eqddx./Acel_x)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE Y (COORDENADA GENERAL)
eqddy=(Acel_y-Acceleration Y);
%ERROR CUADRÁTICO MEDIO
ECMqddy=sqrt((sum((eqddy.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddy=((sum(abs((eqddy./Acel_y)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE Z (COORDENADA GENERAL)
eqddz=(Acel_z-Acceleration Z);
%ERROR CUADRÁTICO MEDIO
ECMqddz=sqrt((sum((eqddz.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddz=((sum(abs((eqddz./Acel_z)),2)/(500)))*100;

%FOTOGRAMETRÍA VS MATLAB

```



```

%RESIDUOS DE LA POSICIÓN EN EL EJE X (COORDENADA GENERAL)
eqx_fm=(Pos_x-dx);
%ERROR CUADRÁTICO MEDIO
ECMax_fm=sqrt((sum((eqx_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqx_fm=((sum(abs((eqx_fm./Pos_x)),2)/(500)))*100;
%RESIDUOS DE LA POSICIÓN EN EL EJE Y (COORDENADA GENERAL)
eqy_fm=(Pos_y-dy);
%ERROR CUADRÁTICO MEDIO
ECMay_fm=sqrt((sum((eqy_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqy_fm=((sum(abs((eqy_fm./Pos_y)),2)/(500)))*100;
%RESIDUOS DE LA POSICIÓN EN EL EJE Z (COORDENADA GENERAL)
eqz_fm=(Pos_z-dz);
%ERROR CUADRÁTICO MEDIO
ECMaz_fm=sqrt((sum((eqz_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqz_fm=((sum(abs((eqz_fm./Pos_z)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE X (COORDENADA GENERAL)
eqdx_fm=(Vel_x-vx);
%ERROR CUADRÁTICO MEDIO
ECMdx_fm=sqrt((sum((eqdx_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdx_fm=((sum(abs((eqdx_fm./Vel_x)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE Y (COORDENADA GENERAL)
eqdy_fm=(Vel_y-vy);
%ERROR CUADRÁTICO MEDIO
ECMdy_fm=sqrt((sum((eqdy_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdy_fm=((sum(abs((eqdy_fm./Vel_y)),2)/(500)))*100;
%RESIDUOS DE LA VELOCIDAD EN EL EJE Z (COORDENADA GENERAL)
eqdz_fm=(Vel_z-vz);
%ERROR CUADRÁTICO MEDIO
ECMdz_fm=sqrt((sum((eqdz_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqdz_fm=((sum(abs((eqdz_fm./Vel_z)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE X (COORDENADA GENERAL)
eqddx_fm=(Acel_x-ax);
%ERROR CUADRÁTICO MEDIO
ECMqddx_fm=sqrt((sum((eqddx_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddx_fm=((sum(abs((eqddx_fm./Acel_x)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE Y (COORDENADA GENERAL)
eqddy_fm=(Acel_y-ay);
%ERROR CUADRÁTICO MEDIO
ECMqddy_fm=sqrt((sum((eqddy_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddy_fm=((sum(abs((eqddy_fm./Acel_y)),2)/(500)))*100;
%RESIDUOS DE LA ACELERACIÓN EN EL EJE Z (COORDENADA GENERAL)
eqddz_fm=(Acel_z-az);
%ERROR CUADRÁTICO MEDIO
ECMqddz_fm=sqrt((sum((eqddz_fm.^2),2))/(500));
%ERROR ABSOLUTO PORCENTUAL
EPqddz_fm=((sum(abs((eqddz_fm./Acel_z)),2)/(500)))*100;

%GRÁFICAS COMPARATIVAS ENTRE FOTOGRAMETRÍA Y ADAMS
%GRÁFICAS DE POSICIÓN
figure(4)
subplot(2,1,1)
plot(Time,Displacement_X,Tiempo,Pos_x)
grid on

```

```

title('Desplazamiento en X')
legend('q_A', 'q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([0 0.25])
subplot(2,1,2)
plot(Time, ECMgx)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
figure(5)
subplot(2,1,1)
plot(Time, Displacement_Y, Tiempo, Pos_y)
grid on
title('Desplazamiento en Y')
legend('q_A', 'q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
subplot(2,1,2)
plot(Time, ECMgy)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

figure(6)
subplot(2,1,1)
plot(Time, Displacement_Z, Tiempo, Pos_z)
grid on
title('Desplazamiento en Z')
legend('q_A', 'q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
subplot(2,1,2)
plot(Time, ECMgz)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

%GRÁFICAS DE VELOCIDAD
figure(7)
subplot(2,1,1)
plot(Time, Velocity_X, Tiempo, Vel_x)
grid on
title('Velocidad en X')
legend('qd_A', 'qd_F')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
subplot(2,1,2)
plot(Time, ECMgdx)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
figure(8)
subplot(2,1,1)
plot(Time, Velocity_Y, Tiempo, Vel_y)
grid on
title('Velocidad en Y')
legend('qd_A', 'qd_F')

```

```

xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
subplot(2,1,2)
plot(Time,ECMgdy)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

figure(9)
subplot(2,1,1)
plot(Time,Velocity_Z,Tiempo,Vel_z)
grid on
title('Velocidad en Z')
legend('qd_A', 'qd_F')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
subplot(2,1,2)
plot(Time,ECMgdz)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

%GRÁFICAS DE ACELERACIÓN
figure(10)
subplot(2,1,1)
plot(Time,Acceleration_X,Tiempo,Acel_x)
grid on
title('Aceleración en X')
legend('qdd_A', 'qdd_F')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
subplot(2,1,2)
plot(Time,ECMgddx)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
figure(11)
subplot(2,1,1)
plot(Time,Acceleration_Y,Tiempo,Acel_y)
grid on
title('Aceleración en Y')
legend('qdd_A', 'qdd_F')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
ylim([-0.7 0.7])
subplot(2,1,2)
plot(Time,ECMgddy)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
ylim([0 0.04])
figure(12)
subplot(2,1,1)
plot(Time,Acceleration_Z,Tiempo,Acel_z)
grid on
title('Aceleración en Z')
legend('qdd_A', 'qdd_F')
xlabel('Tiempo [s]')

```

```

ylabel('Aceleración [m^2/s]')
subplot(2,1,2)
plot(Time,ECMgddz)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

%GRÁFICAS COMPARATIVAS ENTRE FOTOGRAMETRÍA Y MATLAB
%GRÁFICAS DE POSICIÓN
figure(13)
subplot(2,1,1)
plot(tl,dx,Tiempo,Pos_x)
grid on
title('Desplazamiento en X')
legend('q_M','q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
ylim([0 0.25])
subplot(2,1,2)
plot(tl,ECMqx_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
figure(14)
subplot(2,1,1)
plot(tl,dy,Tiempo,Pos_y)
grid on
title('Desplazamiento en Y')
legend('q_M','q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
subplot(2,1,2)
plot(tl,ECMqy_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
%ylim([0 0.4])
figure(15)
subplot(2,1,1)
plot(tl,dz,Tiempo,Pos_z)
grid on
title('Desplazamiento en Z')
legend('q_M','q_F')
xlabel('Tiempo [s]')
ylabel('Posición [m]')
subplot(2,1,2)
plot(tl,ECMqz_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

%GRÁFICAS DE VELOCIDAD
figure(16)
subplot(2,1,1)
plot(tl,vx,Tiempo,Vel_x)
grid on
title('Velocidad en X')
legend('qd_M','qd_F')

```

```

xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
ylim([0 0.2])
subplot(2,1,2)
plot(t1,ECMqdx_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

figure(17)
subplot(2,1,1)
plot(t1,vy,Tiempo,Vel_y)
grid on
title('Velocidad en Y')
legend('qd_M','qd_F')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
subplot(2,1,2)
plot(t1,ECMqdy_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

figure(18)
subplot(2,1,1)
plot(t1,vz,Tiempo,Vel_z)
grid on
title('Velocidad en Z')
legend('qd_M','qd_F')
xlabel('Tiempo [s]')
ylabel('Velocidad [m/s]')
subplot(2,1,2)
plot(t1,ECMqdz_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')

%GRÁFICAS DE ACELERACIÓN
figure(19)
subplot(2,1,1)
plot(t1,ax,Tiempo,Acel_x)
grid on
title('Aceleración en X')
legend('qdd_M','qdd_F')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
ylim([-0.55 0.55])
subplot(2,1,2)
plot(t1,ECMqddx_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
figure(20)
subplot(2,1,1)
plot(t1,ay,Tiempo,Acel_y)
grid on
title('Aceleración en Y')
legend('qdd_M','qdd_F')

```

```

xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
ylim([-0.5 0.5])
subplot(2,1,2)
plot(t1,ECMqddy_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
ylim([0 0.45])
figure(21)
subplot(2,1,1)
plot(t1,az,Tiempo,Acel_z)
grid on
title('Aceleración en Z')
legend('qdd_M','qdd_F')
xlabel('Tiempo [s]')
ylabel('Aceleración [m/s^2]')
subplot(2,1,2)
plot(t1,ECMqddz_fm)
grid on
title('Error')
xlabel('Tiempo [s]')
ylabel('error')
ylim([0 0.025])

```