



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

“MODELACIÓN, CONTROL Y SIMULACIÓN DE UN PROCESO DE PINTADO DE PARTES DE CARROCEÍAS USANDO UN BRAZO ROBÓTICO COLABORATIVO UR3 Y TECNOLOGÍAS INMERSIVAS”

Trabajo de titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES:

BEATRIZ GRACIELA MERCHÁN CUNO

ALEXIS GONZALO CHIMBO CORTEZ

Riobamba – Ecuador

2021



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

“MODELACIÓN, CONTROL Y SIMULACIÓN DE UN PROCESO DE PINTADO DE PARTES DE CARROCEÍAS USANDO UN BRAZO ROBÓTICO COLABORATIVO UR3 Y TECNOLOGÍAS INMERSIVAS”

Trabajo de titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES:

BEATRIZ GRACIELA MERCHÁN CUNO

ALEXIS GONZALO CHIMBO CORTEZ

DIRECTOR: Ph.D. JORGE LUIS HERNÁNDEZ AMBATO

Riobamba – Ecuador

2021

© 2021, **Beatriz Graciela Merchán Cuno, Alexis Gonzalo Chimbo Cortez.**

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

Nosotros, Beatriz Graciela Merchán Cuno y Alexis Gonzalo Chimbo Cortez, declaramos que el presente trabajo de titulación es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 18 de noviembre de 2021



Beatriz Graciela Merchán Cuno
1804855169



Alexis Gonzalo Chimbo Cortez
020205472-2

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo: **PROPUESTA TECNOLÓGICA, “MODELACIÓN, CONTROL Y SIMULACIÓN DE UN PROCESO DE PINTADO DE PARTES DE CARROCERÍAS USANDO UN BRAZO ROBÓTICO COLABORATIVO UR3 Y TECNOLOGÍAS INMERSIVAS”**, realizado por la señorita: **BEATRIZ GRACIELA MERCHÁN CUNO** y el señor: **ALEXIS GONZALO CHIMBO CORTEZ**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Marco Viteri PRESIDENTE DEL TRIBUNAL	_____	18/11/2021
Ing. Jorge Luis Hernández, PhD. DIRECTOR DE TRABAJO DE TITULACIÓN	_____	18/11/2021
Ing. Fausto Cabrera, M.Sc. MIEMBRO DEL TRIBUNAL	_____	18/11/2021

DEDICATORIA

Dedico este trabajo a Dios por haberme dado la vida. A mis padres Lida y Jaime, quienes son mi fuente de apoyo incondicional en todas las etapas de mi vida. A mis hermanos: Alex, Christian, Byron, Ariel y hermanas: Patricia, Verónica, Jenny y Mariela, por su ejemplo de superación diaria y brindarme su ayuda siempre. A mi mejor amigo Christian por su apoyo y por compartir conmigo buenos y malos momentos.

Beatriz

Este trabajo está dedicado a mis padres Gonzalo Chimbo y Magdalena Cortez, hermanas y cuñado por el apoyo y ayuda incondicional en todas las decisiones y logros que he alcanzado en el transcurso de mi vida, y que con su ejemplo de perseverancia y trabajo duro se puede lograr todas las metas propuestas.

Alexis

AGRADECIMIENTO

Agradezco a Dios por permitirme concluir esta etapa con el apoyo incondicional de mis padres, hermanos y hermanas. A mi compañero de tesis, Alexis, por su amistad y apoyo durante este largo camino. A mi mejor amigo Christian por siempre estar presente con su amistad incondicional y constante apoyo.

A la Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia- CEDIA por su contribución en innovación, a través de los proyectos CEPRA, en especial el proyecto CEPRA-XIV-2020-08-RVA "Tecnologías Inmersivas Multi-Usuario Orientadas a Sistemas Sinérgicos de Enseñanza-Aprendizaje"; también a la Escuela Superior Politécnica de Chimborazo - ESPOCH y al grupo de investigación GITEA, por el apoyo para el desarrollo de este trabajo.

Beatriz

En primer lugar, agradezco a Dios por permitirme llegar a esta etapa de mi vida y permitirme compartir este momento con mis padres, hermanas, cuñado los cuales han sido un apoyo fundamental. A mi compañera de tesis, Beatriz, por su amistad y apoyo desde que nos conocimos en este largo camino.

A la Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia- CEDIA por su contribución en innovación, a través de los proyectos CEPRA, en especial el proyecto CEPRA-XIV-2020-08-RVA "Tecnologías Inmersivas Multi-Usuario Orientadas a Sistemas Sinérgicos de Enseñanza-Aprendizaje "; también a la Escuela Superior Politécnica de Chimborazo - ESPOCH y al grupo de investigación GITEA, por el apoyo para el desarrollo de este trabajo.

Alexis

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	z
ÍNDICE DE FIGURAS.....	zk
ÍNDICE DE GRÁFICOS.....	zkk
ÍNDICE DE ANEXOS.....	zkk
RESUMEN.....	zx
CAPÍTULO I	
1 INTRODUCCIÓN.....	3
1.1 Antecedentes.....	3
1.2 Justificación.....	5
1.3 Objetivos.....	6
1.3.1 <i>Objetivo general</i>	6
1.3.2 <i>Objetivos específicos</i>	6
CAPÍTULO II	
2 MARCO TEÓRICO.....	7
2.1 Robótica Colaborativa.....	7
2.1.1 <i>Brazo robótico colaborativo UR3</i>	8
2.1.2 <i>Arquitectura del brazo robótico colaborativo UR3</i>	;
2.2 Tecnologías inmersivas.....	32
2.3 Proceso de pintado de partes plásticas de carrocerías.....	35
2.4 Modelación y Control de sistemas.....	39
2.4.1 <i>Proceso de modelación</i>	39
2.4.2 <i>Análisis Cinemático de los Robots</i>	3:
2.4.3 <i>Análisis Dinámico de los Robots</i>	42
2.4.4 <i>Sistema de control</i>	45
2.5 Recursos para simulación.....	46

2.5.1	<i>MatLab</i>	46
2.5.2	<i>Software de simulación 3D</i>	48
CAPÍTULO III		
3	MARCO METODOLÓGICO	52
3.1	Requerimientos del sistema	52
3.2	Concepción de la arquitectura general del sistema	52
3.3	Diagrama de bloques del sistema	53
3.3.1	<i>Bloque Ingreso de datos</i>	53
3.3.2	<i>Bloque de Procesamiento de la información</i>	53
3.3.3	<i>Bloque de Simulación</i>	53
3.4	Elementos <i>hardware</i> del sistema	54
3.5	Esquema de conexión	55
3.6	Herramientas <i>software</i> del sistema	57
3.7	Modelación del UR3	58
3.8	Modelación de las partes plásticas de carrocerías	82
3.9	Diseño Interfaz del Sistema	84
3.10	Determinación de puntos para la trayectoria en el pintado de partes plásticas	87
3.11	Ejecución de la trayectoria en modo manual	88
3.12	Ejecución de la trayectoria en modo automático	8:
CAPÍTULO IV		
4	PRUEBAS Y RESULTADOS	8;
4.1	Validación de los modelos cinemáticos	8;
4.2	Prueba de error de posicionamiento	97
4.3	Prueba de inmersión y usabilidad	99
CONCLUSIONES		: 5
RECOMENDACIONES		: 6
BIBLIOGRAFÍA		7

ÍNDICE DE TABLAS

Tabla 1-2: Características del brazo robótico colaborativo UR3	9
Tabla 2-2: Movimiento y rendimiento del robot UR3	32
Tabla 3-2: Tabla comparativa de los tipos de tecnologías inmersivas.....	34
Tabla 4-2: Tabla comparativa de los softwares 3D.....	48
Tabla 1-3: Parámetros de DH del robot UR3	5:
Tabla 2-3: Resumen de ganancias (kp, ki, kd).....	82
Tabla 3-3: Características de partes plásticas	83
Tabla 4-3: Cantidad de puntos para pintado en modo automático	88
Tabla 1-4: Parámetros de entrada para la función inversekinematic1(xyzrpw).....	8;
Tabla 2-4: Datos de salida al aplicar la función inversekinematic1(xyzrpw).....	92
Tabla 3-4: Datos de salida al aplicar la función forwardkinematic(q).....	93
Tabla 4-4: Error absoluto calculado	93
Tabla 5-4: Error relativo calculado	94
Tabla 6-4: Parámetros de salida al aplicar el modelo de RoboDK	95
Tabla 7-4: Error absoluto de la comparación de los modelos cinemáticos	95
Tabla 8-4: Estado real del sistema.....	97
Tabla 9-4: Error absoluto calculado	98
Tabla 10-4: Error relativo calculado.....	98
Tabla 11-4: Características del grupo de personas encuestadas	99
Tabla 12-4: Datos para el análisis de confiabilidad.....	: 3

ÍNDICE DE FIGURAS

Figura 1-2. Brazo robótico colaborativo UR3	9
Figura 2-2. Arquitectura del UR3.....;	
Figura 3-2. Patrón de pulverizado	37
Figura 4-2. Posición de la pistola.....	38
Figura 5-2. Distancia de la pistola.....	39
Figura 6-2. Método del pintado de un auto	39
Figura 7-2. Etapas de la modelación matemática.....	3:
Figura 8-2. Sistemas de coordenadas en movimiento.....	43
Figura 9-2. Estructura básica del control PID.....	45
Figura 1-3. Diagrama de bloques del sistema.....	52
Figura 2-3. Diagrama de bloques del sistema.....	53
Figura 3-3. Ordenador Asus pro.....	54
Figura 4-3. Gafas de RV, VR BOX.....	55
Figura 5-3. Conexión de dispositivos para el uso de realidad virtual.....	56
Figura 6-3. Conexión para realidad virtual.....	57
Figura 7-3. Diagrama de flujo del programa	57
Figura 8-3. Implementación de control PID para UR3	59
Figura 9-3. Diagrama de los sistemas para la obtención de los parámetros DH.....	59
Figura 10-3. Diagrama de flujo para la cinemática directa	5;
Figura 11-3. Quinto sistema de referencia respecto a la base.....	62
Figura 12-3. Quinto y sexto sistema de referencia.....	63
Figura 13-3. Vector de posición.....	63
Figura 14-3. Punto general de partida.....	64
Figura 15-3. Relaciones para calcular el quinto ángulo	65
Figura 16-3. Relaciones para calcular los ángulos restantes	66
Figura 17-3. Diagrama de flujo para la cinemática inversa.....	68
Figura 18-3. Propiedades físicas del primer eslabón.....	6:
Figura 19-3. Diagrama de flujo para la dinámica directa.....	6;
Figura 20-3. Diagrama de flujo para la dinámica inversa	73
Figura 21-3. Diagrama de flujo para el control.....	75

Figura 22-3. Bloque de control PID	76
Figura 23-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G1PID.....	77
Figura 24-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G2PID.....	78
Figura 25-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b.G3PID.....	79
Figura 26-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b.G4PID.....	7:
Figura 27-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b.G5PID.....	7;
Figura 28-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b.G6PID.....	82
Figura 29-3: Partes plásticas modeladas: a. Consola central b. Apoya brazos c. Caja para espejo retrovisor	83
Figura 30-3: Interfaz de Usuario en guide de MatLab	84
Figura 31-3: Ventana emergente para desplegar el panel del robot.	85
Figura 32-3: Interfaz de Usuario para pintado en modo manual	85
Figura 33-3: Interfaz del Panel de robot	86
Figura 34-3: Interfaz realidad virtual para el pintado de partes plásticas	87
Figura 35-3: Puntos preestablecidos para el pintado delas partes plásticas a) Apoya brazos, b) consola central, c) caja de espejo retrovisor.....	88
Figura 36-3. Diagrama de secuencia de la ejecución de la trayectoria en modo manual.....	8:
Figura 37-3. Diagrama de secuencia de la ejecución de la trayectoria en modo automático	8:

ÍNDICE DE GRÁFICOS

Gráfico 1-4. Resumen de los datos obtenidos al realizar la encuesta.....	9;
Gráfico 2-4. Porcentaje de Inmersión y Usabilidad.....	2

ÍNDICE DE ANEXOS

ANEXO A: Programa del modelo cinemático directo del UR3

ANEXO B: Programa del modelo cinemático inverso del UR3

ANEXO C: Programa para el control

ANEXO D: Evidencias prueba de simulación

RESUMEN

El presente trabajo de integración curricular se desarrolló con el objetivo de modelar, controlar y simular el proceso de pintado de partes de carrocerías usando un brazo robótico colaborativo UR3 y tecnologías inmersivas. Utilizando el software RoboDK se creó el ambiente virtual para el proceso de pintado mediante el modelo 3D del robot UR3 disponible en el software de desarrollo. Se incluyeron tres modelos de partes plásticas que conforman la carrocería de un auto: Caja para espejo retrovisor, Consola central y Apoya brazo. Los modelos 3D de las partes se diseñaron con SolidWorks. Se empleó MatLab para crear ventanas de interacción para el usuario y la programación y ejecución del algoritmo de control para el posicionamiento del UR3. El usuario puede definir la parte que desea pintar, así como el modo de pintado: automático o manual. En el modo automático, el usuario selecciona la parte que desea pintar y el número de capas. En el modo manual, el usuario determina los puntos para definir la trayectoria de pintado. En referencia a las tecnologías inmersivas, se usó la realidad virtual empleando gafas de realidad virtual con lo que el usuario tiene una experiencia de inmersión en el proceso industrial. Para la validación de los modelos cinemáticos se tomaron los datos de posición y orientación de una escena creada para el pintado en modo automático y se introdujeron en las funciones de los modelos cinemáticos. Con el coeficiente de determinación se demostró la validez de las ecuaciones obtenidas y programadas para el modelo cinemático directo. Además, una prueba de error de posicionamiento permitió validar el controlador PID que controla las 6 articulaciones del robot. Mediante una encuesta realizada a un grupo de 15 usuarios se validó que la simulación es inmersiva, intuitiva y fácil de usar.

Palabras clave: <CONTROL>, <MODELADO MATEMÁTICO>, <REALIDAD VIRTUAL>, <SIMULACIÓN>, <UR3>



Firmado electrónicamente por:
ELIZABETH
FERNANDA AREVALO
MEDINA



1914-DBRAI-UPT-2021

ABSTRACT

The present work of curricular integration was developed with the objective of modeling, controlling and simulating the process of painting body parts using a collaborative UR3 robotic arm and immersive technologies. Using the RoboDK software, the virtual environment was created for the painting process using the 3D model of the UR3 robot available in the development software. Three models of plastic parts that make up the body of a car were included: Box for rear view mirror, center console and armrest. The 3D models of the parts were designed with SolidWorks. MatLab was used to create interaction windows for the user and the programming and execution of the control algorithm for positioning the UR3. The user can define the part to be painted, as well as the painting mode: automatic or manual. At automatic mode, the user selects the part to be painted and the number of layers. At manual mode, the user determines the points to define the painting path. In reference to immersive technologies, virtual reality was used using virtual reality glasses with which the user has an immersive experience in the industrial process. For the validation of kinematic models, they were taken the position and orientation data of a scene created to the painting in automatic mode and the functions of kinematic models were introduced. With the determination coefficient, the validity of the equations obtained and programmed for the direct kinematic model was proven. Also, a positioning error test allowed to validate the PID controller that controls the 6 joints of the robot. Through a survey carried out on a group of 15 users, it was validated that the simulation is immersive, intuitive and easy to use.

Keywords: <AUTOMATIC CONTROL TECHNOLOGY>, <MATHEMATICAL MODELING >, <<VIRTUAL REALITY>, <SIMULATION>, <ROBOT UR3>



Firmado electrónicamente por:
**NELLY MARGARITA
PADILLA PADILLA**

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Antecedentes

Las cuatro revoluciones industriales a lo largo de la historia han marcado precedentes significativos en el mundo. La primera en el siglo XVIII, teniendo su desarrollo durante todo el siglo XIX, con la introducción de la mecánica y el uso de la hidráulica. La segunda tuvo lugar a finales del siglo XIX, usándose la electricidad y la producción en masa. La tercera aproximadamente en 1970, con el uso de la electrónica y las tecnologías de la información. La cuarta dio inicio en 2011, aplicando nuevas tecnologías en los procesos industriales. Ligada a la cuarta revolución industrial nació el concepto de industria 4.0 que se conoció en la feria de Hannover, Alemania, en 2011 con el propósito de iniciar un proyecto de desarrollo de la fábrica inteligente asociada a la cuarta revolución industrial. En 2014, Shorouf, Ordieres, & Miragliotta exponen en su artículo, el recoger y compartir información como uno de los principios de la Industria 4.0 (Shrouf, et al., 2014). Entre las tecnologías básicas en la que se sustenta la Industria 4.0 se encuentran: Internet de las cosas (IoT) y sistemas Ciberfísicos (CPS), Realidad Aumentada, Realidad Virtual, Simulación, Robótica Colaborativa, Fabricación Aditiva, Big Data, Cloud Computing y Ciberseguridad (Román, 2017) (Sáenz, 2016).

Cualquier cosa u objeto, desde máquinas y piezas, hasta líneas de montaje, incluso instalaciones y plantas de producción se pueden trasladar al mundo virtual gracias a la creación de modelos 3D sobre los cuales se puede experimentar a través de la aplicación de la tecnología de la simulación. Respecto a la simulación Europa tiene algunos proyectos de alto impacto que buscan que la tecnología de la simulación llegue a la industria manufacturera, ya sea como métodos de entrenamiento o de apoyo y seguridad. Entre estos proyectos se mencionan: Fortissimo, que busca hacer llegar la simulación y la modelización a las pequeñas y medianas empresas. Beincpps, quienes están centrados en diseñar una plataforma que incluya sistemas de desarrollo tecnológico y aplicaciones TIC. Pathfinder, busca integrar una solución en la que se engloben tecnologías de simulación y predicción (Galicia, 2017). En 2015, se realiza la virtualización del proceso de pintado en manufacturas industriales, con el objetivo de evaluar la actividad antes de ser construida (Lanverde Ocadiz, et al., s.f.). Al siguiente año se realiza el diseño de un entorno de simulación dinámico de robots evolutivos, para poder estudiar la evolución de una población de robots con

variadas geometrías (Humanes, 2016). Ya en 2018, se usa el modelado y la simulación de robots terrestres para la inspección de alcantarillados, en el que se presenta un claro ejemplo del beneficio de incorporar la tecnología robótica, en especial en actividades que involucran un riesgo para los trabajadores u operarios (Rozas, 2018).

Por otra parte, en cuanto a la tecnología de Robótica Colaborativa, el 17 mayo de 2016, Universal robots publicó una infografía que muestra el crecimiento rápido de los robots colaborativos, desde sus inicios en 1920 hasta el 2016 (Universal Robots, 2016). La robótica colaborativa representa una profunda transformación respecto al funcionamiento de los robots en la industria (Sáenz, 2016). En julio de 2019, Paulino San Miguel con su trabajo de fin de Master titulado Modelado y Simulación de célula robótica colaborativa mediante realidad virtual, busca una solución que integre tecnologías de Industria 4.0 (diseño, simulación y robótica colaborativa), como un antecedente para el desarrollo de aplicaciones para la mejora de la productividad y la seguridad en la Industria 4.0. Y así demostrar el potencial de robótica colaborativa a nivel industrial y de la aplicación conjunta de las diversas tecnologías de la industria 4.0. (Herrera, 2019).

La automatización de la producción es una tendencia creciente a nivel mundial. El capital robótico se concentra en pocos países y en empresas de tamaño grande, siendo la industria automotriz la principal adoptante de esta tecnología a nivel mundial. El uso del 75% de los robots industriales se localiza en cinco países: China, Estados Unidos, Corea, Japón y Alemania, los que, al mismo tiempo, resultan los principales productores de la tecnología (Basco, et al., 2018).

En 2017 se desarrolla una aplicación de realidad virtual a través de la creación de entornos que permiten la supervisión, monitoreo y control de procesos industriales. Esta aplicación simula y trabaja con información en tiempo real y se centra en la formación de operadores en plantas industriales, así como en una herramienta en el área de la educación. (Andaluz, et al., 2018). Un año más tarde se desarrolla una aplicación usando la tecnología de realidad virtual para el Control y Monitoreo de Procesos Industriales en la que se permite el intercambio de datos bidireccional entre una red de módulos de procesos industriales y un entorno virtual, es decir, desarrollar una interfaz hombre máquina, HMI. El HMI es un entorno 3D que garantiza una interacción segura y adecuada de los operadores con los procesos industriales y todos sus componentes con el objetivo de controlar los módulos de proceso reales utilizando los elementos dentro de la aplicación virtual (Castillo , et al., 2018).

Actualmente se encuentra en desarrollo el proyecto de investigación "Tecnologías Inmersivas Multi-Usuario Orientadas a Sistemas Sinérgicos de Enseñanza –Aprendizaje"; proyecto que fue adjudicado en la XIV Convocatoria de CEDIA-CEPRA 2020 y es ejecutado por la Universidad de las Fuerzas Armadas ESPE, Universidad Técnica de Ambato, Escuela Superior Politécnica de Chimborazo y Universidad Tecnológica Indoamérica. Con el propósito de generar ambientes virtuales de laboratorios de automatización enlazados con sus respectivos modelos cinemáticos y dinámicos para introducir y evaluar nuevas metodologías inmersivas para la enseñanza-aprendizaje de la ingeniería (CEDIA, 2020).

1.2 Justificación

En la actualidad la simulación de procesos ayuda a las industrias a obtener información valiosa para evitar posibles errores en la línea de fábrica, de esta forma se previene daños a la maquinaria, pérdidas económicas y posibles pérdidas humanas. Además, ayuda a los usuarios a tener una interacción realista como elementos y procesos que se encuentran en el entorno simulado, de esta manera los usuarios tendrán la posibilidad de apreciar y evaluar el funcionamiento de la maquinaria sin tener que adquirirla, las cuales podrían llegar a costar cantidades enormes de dinero.

La tecnología inmersiva relaciona la realidad virtual y la aumentada, y en la actualidad esta tecnología está siendo introducida también en la industria. El objetivo es la creación de realidades virtuales artificiales simuladas donde el usuario sienta diferentes grados de inmersión. Las apariciones de este tipo de tecnologías permiten interactuar en espacios simulados apegados a la realidad en la cual los usuarios pueden interactuar con máquinas de forma intuitiva y natural. De hecho, estos conceptos y aplicaciones van de la mano con la llamada industria 4.0 donde operarios y usuarios finales pueden ser partícipes virtuales de los procesos de producción.

El robot colaborativo UR3 sirve para atornillar, pintar, soldar y otras actividades. Se propone el uso del robot mencionado para realizar el proceso de pintado de piezas plásticas de carrocerías, las cuales no tengan dimensiones que excedan el área de trabajo del robot UR3, 50 cm a partir de la junta de la base. De esta manera se obtendrá un mejor rendimiento en el proceso gracias a la capacidad de seguir una trayectoria de forma constante, sin llegar a extenuar la producción por circunstancias de repetitividad y aumentando la calidad y número de partes pintadas.

1.3 Objetivos

1.3.1 Objetivo general

Modelar, controlar y simular un proceso de pintado de carrocerías usando un brazo robótico colaborativo UR3 y tecnologías inmersivas.

1.3.2 Objetivos específicos

- Investigar estudios y trabajos reportados sobre procesos de modelado, control y simulación de procesos industriales usando brazos robóticos colaborativos y tecnologías inmersivas.
- Seleccionar las técnicas y herramientas adecuadas para el modelado matemático y virtual del robot colaborativo UR3 y del proceso de pintado de carrocerías enfocado en tecnologías inmersivas.
- Realizar acciones de control sobre el modelo del brazo robótico colaborativo UR3 para realizar un proceso industrial de pintado de carrocerías.
- Implementar una simulación virtual del ambiente industrial para el pintado de carrocerías usando el robot colaborativo UR3 y tecnologías inmersivas.
- Evaluar el modelo, control y simulación de un brazo robótico colaborativo UR3 aplicado al proceso de pintado de carrocerías implementado mediante tecnologías inmersivas.

CAPÍTULO II

2 MARCO TEÓRICO

En este capítulo se describe las bases matemáticas y los conocimientos necesarios en el análisis del movimiento del robot, el control del mismo y los recursos software a usar. Es por eso que en este capítulo se presentan conceptos generales de robótica colaborativa, características generales del UR3, el proceso de pintado para partes plásticas de carrocerías, tecnologías inmersivas, modelado y control del sistema y por último el software usado para la simulación.

2.1 Robótica Colaborativa

En la actualidad el intercambio de datos y la automatización son metas deseadas, en la cual el objetivo es que las máquinas y equipos humanos laboren simultáneamente en red. En el documento realizado por Gipuzkoa ETORKIZUNA ORAIN menciona que “la robótica colaborativa es una manera de combinar flexibilidad y velocidad dentro de las líneas de producción actuales es facilitando el trabajo colaborativo humano-robot” (ORAIN). El avance de labores colaborativas ayuda a la obtención de lo mejor de la experiencia e inteligencia del ser humano con la robustez y velocidad de los robots, permitiendo así optimizar tiempo y reducir costos para las empresas en las cuales opten por implementar robots colaborativos.

CFZ Cobots expresa que “la robótica colaborativa está diseñada expresamente para interactuar con personas, evitando accidentes y compartiendo el mismo espacio de trabajo” (cfzcobots). De esta manera los robots industriales se quedan atrás, porque en el pasado los robots se instalaban en jaulas o dentro de barreas de seguridad, además se debía tener medidas de seguridad tremendamente rigurosas para que no existiera accidentes con el personal humano. Los robots colaborativos “Cobots” con fabricados expresamente para realizar labores de trabajo juntos a seres humanos en la misma área sin la necesidad de contemplar medidas de seguridad adicionales. El autor ESIC Business & Marketing School en su documento indica que “la robótica colaborativa permite que la tecnología y, en concreto, la automatización robótica, sea más accesible para las empresas medianas y pequeñas” (School., 2018).

Con lo mencionado la robótica colaborativa es la interacción de robot-humanos, sin tener que instalar medidas de seguridad altamente rigurosas, ya que por medio de Cobots permite que la automatización industrial instale robots en áreas y procesos industriales, donde antes no era posible, y así permitiendo al trabajador ejecute actividades en las que tenga mejor desempeño evadiendo trabajos peligrosos, impuros o escandalosos. Además, los Cobots son dotados con sensores los cuales evitan que el robot se golpee o cause daño al trabajador con el que trabaja en conjunto.

2.1.1 Brazo robótico colaborativo UR3

Es un robot colaborativo creado por Universal Robots, lanzado al mercado en el año 2015 en Chicago. Es un robot de forma compacta y programación sencilla, lo cual permite que el mudarlo entre tareas sea rápido y simple, lo que permite satisfacer los requerimientos de producciones ágiles. Por lo mencionado el costo de adquisición es bajo y el tiempo de recuperación de la inversión será rápida. El UR3 brinda una alta precisión para áreas de producción pequeñas. Es el asistente perfecto en diversos trabajos, destacando en montaje de elementos de tamaño reducido, atornillado, manejo de herramientas, soldadura, pintura, en donde es indispensable conseguir una calidad uniforme.

En operaciones de colaboración presenta “15 funciones avanzadas de seguridad regulables. Funciones de seguridad con certificado TUV NORD probado de acuerdo con las normas ISO 13849-2008 PL d” (UNIVERSAL ROBOTS).



Figura 1-2. Brazo robótico colaborativo UR3

Fuente: CFZCOBOTS,s.f.

La empresa Creating Revolutions se dedica a la creación de sistemas de localización de servicio al cliente para la industria hotelera. El localizador es un disco de comunicación de un tamaño similar a un disco de hockey, conecta al usuario con el camarero por medio de un Smartphone. El montaje de dicho dispositivo es una tarea compleja la cual se requiere de precisión y repetibilidad. El director ejecutivo de Creating Revolutions, Einar Rosenberg tras una ardua investigación concluyó que el UR3 era el mejor, mencionando lo siguiente “El UR3 era elegante, rápido y sin problemas. Sabíamos que tenía la precisión exacta que realmente queríamos. Nuestra tasa de rechazo de proyecto ha pasado de dos dígitos a menos del 1%, nuestra sobrecarga también se ha reducido significativamente, mientras que el aumento de nuestra producción se multiplica” (Universal Robots, 2016). Es un robot poli-articulado, cuenta con características que le diferencian de los otros cobots, las cuales se presentan en la tabla 1-2, donde se detalla el rendimiento, especificaciones, movimientos y funciones. Además, se presenta las características de la caja de control en donde se muestra las funciones que tiene la caja de control.

Tabla 1-2: Características del brazo robótico colaborativo UR3

UR3
Rendimiento

Rango de temperatura ambiente	0 - 50 °	
Consumo de energía	Min 90W, estándar 125W, Max. 250W	
Operación de colaboración	15 funciones avanzadas de seguridad regulables. Función de seguridad con certificado TUV NORD. Probado de acuerdo con las normas: EN ISO 13849:2008 PL d	
Especificación		
Carga útil	3kg / 6.6lb	
Alcance	500mm / 19.7in	
Grados de libertad	6 articulaciones giratorias	
Programación	Interfaz gráfica del usuario PolyScope con pantalla táctil de 12" con soporte	
Funciones		
Clasificación IP	IP64	
Clase ISO Sala limpia	5	
Ruido	70dB	
Puertos de E/S en herramienta	Entrada Digital	2
	Salida digital	2
	Entrada analógica	2
	Salida analógica	0
E/S de fuente de aliment. en herramienta	12v / 24v 600mA en herramienta	

Caja de control		
Funciones		
Clasificación IP	IP20	
Ruido	<65 dB (A)	
Puertos de E/S	Entrada digital	16
	Salida digital	16

:

	Entrada analógica	2
	Salida analógica	2
E/S de fuente de alimentación	24v 2A	
Comunicación	TCP/IP 100 Mbit, Modbus TCP, Profinet, EthernetIP	
Fuente de alimentación	100-240vca, 50-60Hz	
Rango de temperatura ambiente	0-50°	

Fuente: UNIVERSAL ROBOTS, 2020.

2.1.2 Arquitectura del brazo robótico colaborativo UR3

La arquitectura de los robots es una réplica de la anatomía de las extremidades superiores o inferiores del cuerpo humano, entonces, para indicar los distintos elementos que componen al robot se usan términos como codo, hombro, brazo, muñeca, entre otros. La figura 2-2 muestra la arquitectura del brazo robótico UR3.

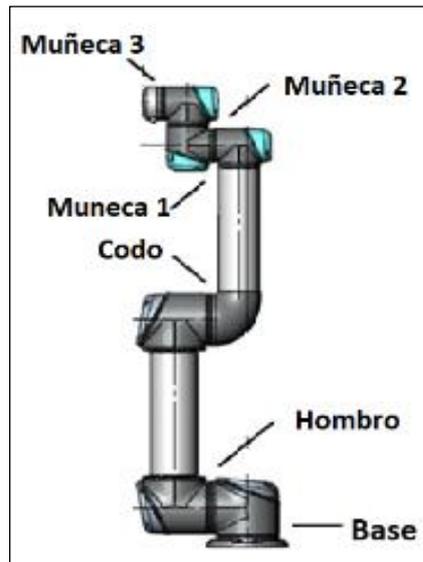


Figura 2-2. Arquitectura del UR3

Fuente: Gier,2015.

Realizado por: Merchán, B. & Chimbo, A. 2020

El radio de acción, la velocidad con la que se puede mover cada articulación y la repetitividad son datos importantes. Dichos datos nos permiten identificar que tan confiable puede ser los resultados. En la tabla 2-2 se da a conocer los movimientos que puede ejecutar cada articulación y también se muestra el valor de la repetitividad. Se observa que las articulaciones tienen un radio de acción de 360° en las 5 primeras articulaciones y en la última articulación el radio de acción

es infinita, lo cual lo hace muy útil para desarrollar trabajos de atornillado, pegado, entre otras. Además, se aprecia que la velocidad de todas las articulaciones es de 180°/s. También se muestra la repetibilidad que tiene el robot es de $\pm 0,1$ mm.

Tabla 2-2: Movimiento y rendimiento del robot UR3

Rendimiento		
Repetibilidad	± 0.1 mm / ± 0.1 mm / ± 0.0039 in (4 mil.)	
Movimiento		
Movim. del eje del brazo robot.	Radio de acción	Velocidad máxima
Base	$\pm 360^\circ$	$\pm 180^\circ/s$
Hombro	$\pm 360^\circ$	$\pm 180^\circ/s$
Codo	$\pm 360^\circ$	$\pm 180^\circ/s$
Muñeca 1	$\pm 360^\circ$	$\pm 360^\circ/s$
Muñeca 2	$\pm 360^\circ$	$\pm 360^\circ/s$
Muñeca 3	Infinita	$\pm 360^\circ/s$
Herramienta típica		1m/s / 39.4in/s

Fuente: UNIVERSAL ROBOTS, 2020.

2.2 Tecnologías inmersivas

“Borran la línea entre el mundo físico y el mundo digital o simulado, es decir, crean o extienden la realidad” (Microsoft, 2020). Permitiendo que el usuario viva una experiencia única y apreciar el vivir dentro de un mundo virtual, además se puede implementar en diferentes áreas entre las cuales tenemos el diseño, educación, medicina, construcción, publicidad, adiestramiento de personal. Existen diferentes tipos de tecnologías inmersivas, que a continuación se da las características de cada tecnología:

Tecnología Háptica

Este tipo de tecnología permite sentir los elementos que se tiene en el entorno virtual. El termino háptico es un adjetivo utilizado para describir algo basado o con sentido del tacto. Esta tecnología utiliza dispositivos capaces de aparentar con bastante realismo sensaciones hápticas. Esta tecnología utiliza sistemas propioceptivos el cual “es el sentido que comunica al organismo de la

posición de los músculos, sistema táctil el cual permite interactuar con las terminaciones nerviosas de la piel, y el sistema vestibular relativo a la percepción de la posición de la cabeza, su aceleración y desaceleración ” (González, s.f.).

Realidad Aumentada

“La Realidad Aumentada (RA) es la unión del mundo real y el virtual para crear nuevas condiciones para la imagen, donde los objetos físicos y digitales coexisten e interactúan en tiempo real” (Korowajczenko, 2012). Lo que significa que existe una proyección de datos digitales multimedia en la realidad.

Realidad Inmersiva

En el documento escrito por José Manuel Menéndez y David Jiménez Bermejo mencionan “es aquella en la que se simula un mundo artificial tridimensional, y en el que el nivel de inmersión del usuario es tal que lo distingue como su entorno real” (Menéndez, et al., 2018). Estos entornos cambian según los movimientos que realiza el usuario o de la forma de interactuar con los elementos del entorno. La realidad inmersiva está fundamentada en la recreación de áreas reales en las que se puede interactuar con el entorno.

Realidad virtual (RV)

“La realidad virtual es una simulación de un ambiente tridimensional generado por computadora, en el que el usuario es capaz tanto de ver como de manipular los contenidos de ese ambiente” (Escartín, 2000). Con lo mencionada se puede decir que la realidad virtual es la simulación tridimensional elaborada en computadora la que suministra información como visión, sonido, entre otros, con la finalidad que el usuario conciba que se halla en el lugar simulado. Esta tecnología usa un casco-visor HMD, el cual aíslan a los usuarios del mundo real de forma que sólo experimenten las sensaciones de la RV a través de la visión del mundo virtual que se les proyecta y los recursos acústicos proporcionados (Luna, s.f.).

Realidad disminuida

“Realidad disminuida es la acción de retirar objetos reales de la imagen, es una variante dentro del campo de la RA” (Álvarez, 2010). La mencionada tecnología es poco conocida, es lo opuesto a la realidad aumentada, permite eliminar porciones de la realidad para dejar lo fundamental.

Holograma

Un holograma “es una foto realizada con el empleo de un láser y que la principal característica que posee es el carácter tridimensional de la imagen” (Toledo, et al., 2009). La tecnología mencionada consigue conjeturar un nuevo perfil de interaccionar con los objetos y personas, por medio de dos haces de luz que brinda la tercera dimensión en un entorno real.

Teleinmersión

La teleinmersión es la cual “permite crear espacios virtuales de colaboración entre profesionales, ofreciendo entornos muy cercanos a la realidad” (Méndez, et al., 2010). Esta tecnología no está muy perfeccionada por motivo que depende de la conexión a internet y del ancho de banda, esos requerimientos son necesarios para tener una experiencia placentera.

En la tabla 3-2 se muestra las principales características que tiene los diferentes tipos de tecnologías. De las tecnologías presentadas se selecciona la realidad virtual por su característica de ambiente tridimensional generado en un computador, por motivo que el presente trabajo tiene como objetivo “Implementar una simulación virtual del ambiente industrial para el pintado de carrocerías usando el robot colaborativo UR3 y tecnologías inmersivas”. Uno de los puntos importantes cuando se aborda un proyecto con este tipo de tecnología es necesario los dispositivos de interacción: guantes, mandos de consola, joysticks, teclado y ratón, etc; ya que son a través de estos dispositivos con los que los usuarios se mueven e interactúan por el mundo virtual. Además, se requiere un dispositivo para una experiencia de inmersión en el entorno virtual, un casco visor.

Tabla 3-2: Tabla comparativa de los tipos de tecnologías inmersivas

Tecnología	Característica
Háptica	Sentir los elementos que se encuentran en el entorno virtual
Realidad aumentada	Proyección de datos digitales multimedia en la realidad
Realidad inmersiva	Simulación de un mundo artificial tridimensional en el que el usuario percibe como su entorno real
Realidad virtual	Ambiente tridimensional generado en computador, el usuario puede ver como manipular el contenido
Realidad disminuida	Retira objetos reales de las imágenes.
Holograma	Foto que posee carácter tridimensional

Teleinmersión	Espacios virtuales muy cercanos a la realidad, para colaboración entre profesionales
---------------	--

Realizado por: Merchán, B. & Chimbo, A. 2020

2.3 Proceso de pintado de partes plásticas de carrocerías

El pintado de partes metálicas y plásticas en automóviles son iguales y se divide en pintura de preparación y pintura de acabado. Cada tipo de pintura tiene una función específica, la pintura de preparación (fondo) tiene como objetivo fortificar la adherencia, nivelar y preparar la superficie para la pintura de acabado.

El proceso de pintado que se debe seguir para piezas nuevas sin imprimir es el siguiente (Espí, 2015):

1. Limpieza, soplado y desengrasado: Elimina el polvo y suciedad que se hayan acumulado en las piezas durante el almacenamiento
2. Matizado del plástico: En la superficie del plástico a pintar se realiza un matizado con abrasivo tridimensional o esponjas abrasivas para favorecer la adherencia.
3. Limpieza, soplado y desengrasado: Con disolvente específico para plásticos y papel o trapos exentos de deshilados.
4. Enmascarado de las zonas precisas
5. Aplicación de imprimación de anclaje: Se aplica la imprimación de anclaje o promotor de adherencia sobre la superficie de plástico que deba ser pintada.
6. Aplicación del aparejo: dependiendo de la flexibilidad de la pieza se utiliza diferentes aparejos.
7. Aplicación de la pintura de acabado.

Método de pintado

Existen diferentes colores de carrocerías y pueden ser de distinta naturaleza según sea el tipo de pigmento que los componen. Los acabados que se pueden dar a la pintura se clasifican dependiendo el tipo de aplicación que necesita cada color, entre los acabados tenemos (PINTARMICOHE, 2015) :

- Acabado monocapa
 - Se aplica en una sola capa.
 - El color y el brillo aparecen simultáneamente.

- Este tipo de acabado solo se ofrece para colores sólidos (no presenta ninguna partícula que destaque).
- Acabado bicapa
 - Se aplica en dos capas.
 - La primera capa es el color (color base), de aspecto mate al secarse.
 - La segunda para es laca (o barniz), proporciona protección y brillo al color
 - Este acabado es aplicable a colores metalizados (incorporan partículas de aluminio en su composición), perlados (incorporan partículas de mica, nácar o similares, que proporcionan reflejos de diferentes tonalidades) y sólidos.
- Acabado tricapa
 - Se aplica en tres capas.
 - La primera capa es un color de fondo (capa de base).
 - La segunda capa es un efecto (capa de efecto). Suele ser un color perlado muy transparente.
 - La tercera capa es el barniz
 - Este acabado se aplica a colores perlados. Es lo más habitual en blancos perlados. (PINTARMICOCHE, 2015)

La mayoría de los colores se realizan en acabado bicapa, los colores solidos pueden realizarse en acabado monocapa, y algunos colores especiales perlados requieren un acabado tricapa.

Manejo de la pistola aerográfica

La aplicación de la pintura con la pistola es un trabajo fácil, pero el cual necesita bastante practica hasta llegar a conocer y dominar el método.

Regulación de la pistola

Las pistolas más implementadas son las de gravedad, por lo tanto, se hacer referencia a dicha pistola. Esta pistola tiene el depósito de la pintura en la parte superior, por lo que el producto ingresa en el canal de aplicación por el efecto de gravedad. Existen tres parámetros para regular la pistola aerográfica (PINTARMICOCHE, 2015):

- Longitud del abanico:

Las pistolas aerográficas atomizan la pintura en forma de abanico cuando salen del cabezal. Dicho abanico genera una marca de forma elíptica si se aplica la pintura sin mover la pistola, a esto se

denomina patrón de pulverizado figura 3-2. La regulación de longitud de abanico actúa sobre la altura de este patrón. La regulación normal para trabajar es con la longitud máxima del mismo, por lo que este regulador debería estar abierto al máximo.

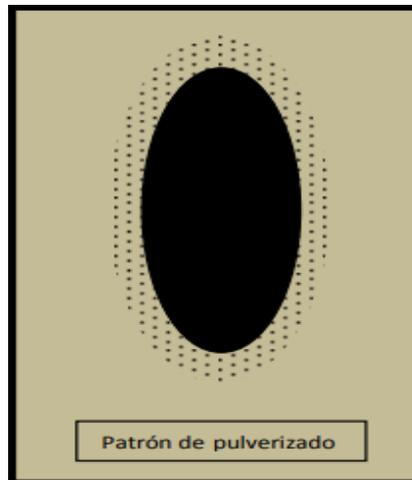


Figura 3-2. Patrón de pulverizado

Fuente: PINTARMICOCHE,2015.

- Caudal

La regulación de caudal actúa directamente sobre la aguja que permite o cierra el paso del producto a través del pico de fluido. Al reducir el caudal, se limita el recorrido de la aguja hacia atrás, impidiendo el paso del producto. La regulación depende, principalmente, de la velocidad a la que se desplaza la pistola a lo largo del objeto; a más velocidad se debe aumentar el caudal y a menos velocidad menos caudal.

- Presión de entrada

La entrada de aire comprimido debe ser regulada con la presión adecuada, esta presión viene determinada por el tipo de producto a aplicar. Lo común al aplicar la capa base es entre 1.8 y 2 bar, el barniz y el esmalte monocapa es entre 2 y 2.5 bar

Posición de la pistola

Se debe tener firme la pistola, a continuación, se acerca la pistola a la parte que se va a pintar, se debe ubicar de manera que el eje longitudinal del cabezal de pulverización de la pistola completamente perpendicular a la parte a pintar (PINTARMICOCHE, 2015).

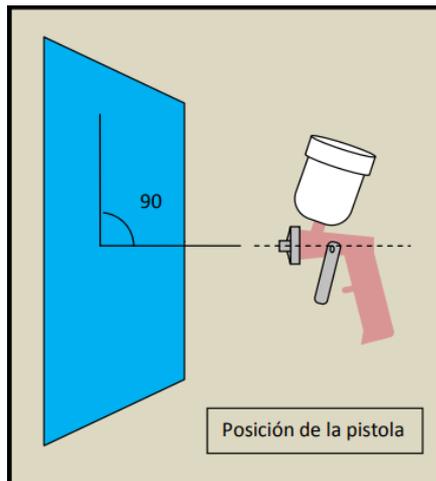


Figura 4-2. Posición de la pistola

Fuente: PINTARMICOHE,2015.

Distancia de aplicación

La pistola se debe mantener a una distancia invariable de la parte a pintar durante todo el desplazamiento. La distancia debe ser de aproximadamente 15cm o 6plg de la parte a pintar (PINTARMICOHE, 2015).

Movimiento de la pistola

Siempre debe estar en movimiento la pistola cuando se esté aplicando la pintura, por motivo que si detiene se provocara una sobre carga del producto. Se debe trazar ráfagas paralelas entre sí en ambos sentidos (de izquierda a derecha, y de derecha a izquierda), se inicia por la parte superior del objeto y culminando en la inferior (se puede hacerlo inversamente, es decir desde la parte inferior hasta la superior). La longitud del recorrido de la ráfaga debe de ser la justa para no tener que moverse de la ubicación en la que se encuentre (PINTARMICOHE, 2015).

Solapado de las ráfagas

Con el objetivo de repartir uniformemente la cantidad de pintura en toda la pieza, se debe solapar las ráfagas. Es decir, cada ráfaga tiene que montarse parcialmente en la anterior. Una regla sencilla que garantiza el solapado correcto es que el ancho de la ráfaga se lo divide mentalmente en 4 partes. La siguiente ráfaga debe cubrir 3 de las 4 partes anteriores, y así sucesivamente hasta terminar la pieza (PINTARMICOHE, 2015).

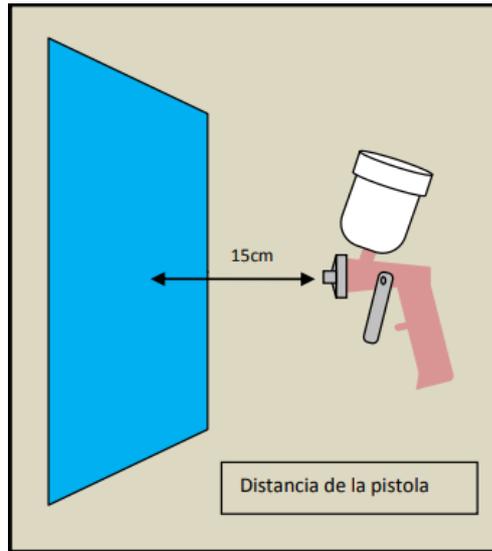


Figura 5-2. Distancia de la pistola

Fuente: PINTARMICOCHE,2015.

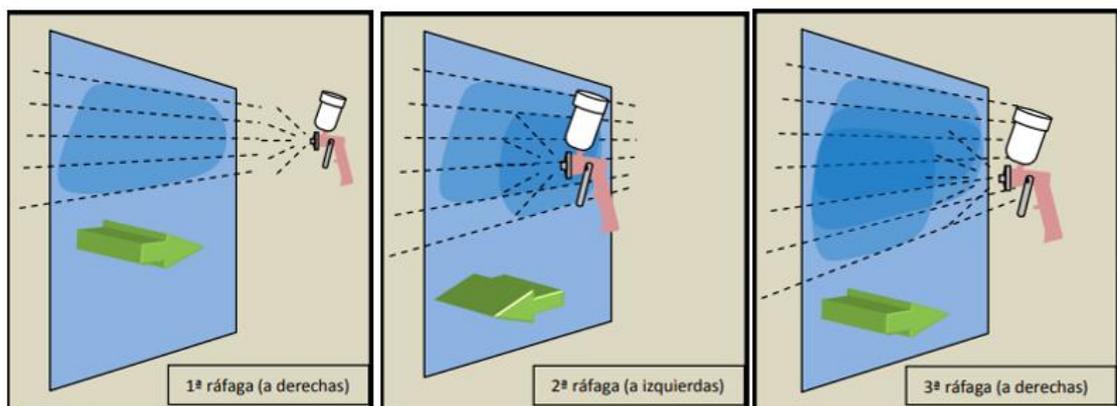


Figura 6-2. Método del pintado de un auto

Fuente: wikiHow,2020.

2.4 Modelación y Control de sistemas

2.4.1 Proceso de modelación

Se define a la modelación como una representación simplificada de un sistema desarrollado para comprender, predecir y controlar el comportamiento de dicho sistema (relación entre señales de entrada y salida) (García, 2018). El modelo de un sistema puede ser una réplica exacta de este, con la diferencia del material que lo compone o de su escala, inclusive puede ser una abstracción de

las propiedades física dominantes del sistema (Herrera, s.f.). Bajo esta concepción se denomina modelo matemático a la descripción matemática de las características dinámicas de un sistema.

Para construir un modelo se tienen tres etapas cíclicas mostradas en la figura 7-2, la etapa de conceptualización es la obtención de un modelo mental del proceso a partir del conocimiento humano de la estructura real del mismo. En esta etapa se conoce como funciona el modelo además cuáles son sus características más relevantes a considerar para iniciar y limitar el proceso de modelación. La siguiente etapa es la de formalización en la que se aplican principios generales físicos, matemáticos, entre otros al modelo mental obteniendo como resultado un modelo formal en el cual se representa el mundo real en función de un conjunto de ecuaciones. Por una última etapa, la evaluación que consiste en el análisis del modelo mediante simulación de las hipótesis sobre las cuales se asienta el modelo y su consistencia.

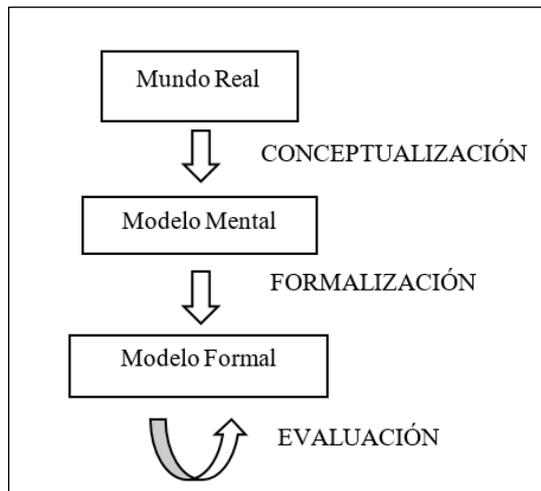


Figura 7-2. Etapas de la modelación matemática

Fuente: Steenmeijer,2014.

Realizado por: Merchán, B. & Chimbo, A. 2020

2.4.2 Análisis Cinemático de los Robots

En la etapa de formalización, para obtener el modelo del robot UR3 es necesario resolver el problema cinemático para determinar el movimiento espacial (posición y orientación) del mismo con respecto de un sistema de referencia ubicado en la base. A continuación, se describirá de manera general, el cálculo de la cinemática directa e inversa:

Cinemática Directa

Se resuelve para determinar la posición y orientación del extremo conociendo los valores de los ángulos entre los eslabones denotadas por q_n donde n es el número de articulaciones del brazo. Se calcula a través de matrices de transformación homogénea haciendo uso de la representación Denavit-Hartenberg (DH) (Barrientos, et al.), la cual se trata de un algoritmo mediante el cual se sitúan los sistemas de coordenadas de cada articulación. Obteniendo de esta forma los parámetros θ_i , d_i , a_i y α_i (el ángulo, offset, longitud y ángulo respectivamente) para sustituirlos en la matriz de transformación homogénea:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \theta_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1-2)$$

donde ${}^{i-1}A_i$ es la transformación de coordenadas del sistema de referencia i al sistema de referencia $i - 1$. Así, al multiplicar cada transformación homogénea de la forma:

$$T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n \quad (2-2)$$

Se obtiene T , que es la matriz que da posición y la orientación del sistema de coordenadas del efector final en función de las variables articulares y expresado en coordenadas del sistema base (Vilchis, 2020).

Cinemática Inversa

Al contrario de la cinemática directa, esta determina el valor de los ángulos en función de la posición y orientación del efector final, en general se establece como sigue: dada una transformación homogénea de 4x4

$$H = \begin{bmatrix} R & p \\ 0^T & 1 \end{bmatrix} \in E(3) \quad (3-2)$$

Con $R \in SO(3)$ (conjunto de todas las matrices de rotación de 3x3), encontrar una o todas las soluciones que satisfagan la ecuación:

$${}^0T_n(q_1, q_2, \dots, q_n) = H \quad (4-2)$$

Donde

$${}^0A_1 {}^1A_2 \dots {}^{n-1}A_n = H \quad (5-2)$$

Sin embargo, al desarrollar la ecuación (4-2) se obtienen ecuaciones no lineales con n variables desconocidas. Es por ello que se opta por usar un método geométrico para relacionar dichas variables.

2.4.3 *Análisis Dinámico de los Robots*

Para obtener completamente el modelo del UR3 también es necesario para la etapa de formalización conocer la dinámica (directa e inversa) para establecer la relación entre fuerzas y pares aplicadas en los actuadores y el movimiento del brazo robótico. Los modelos dinámicos usados en este proyecto se basan en el algoritmo computacional de Newton-Euler (N-E), desarrollado por Luh (Luh, 1982) debido a la dificultad de un análisis clásico en el que se obtienen expresiones fuertemente no-lineales (Ingeniería Automatización , 2017). A continuación, se describirá de manera general, el cálculo de la dinámica:

Dinámica Inversa

Newton-Euler permite obtener un conjunto de ecuaciones recursivas hacia delante de velocidad y aceleración lineal y angular las cuales están referidas a cada sistema de referencia articular. Las velocidades y aceleraciones de cada elemento se propagan hacia adelante desde el sistema de referencia de la base hasta el efector final. Las ecuaciones recursivas hacia atrás calculan los pares y fuerzas necesarios para cada articulación desde la mano (incluyendo en ella efectos de fuerzas externas), hasta el sistema de referencia de la base (Ingeniería Automatización , 2017).

La formulación de N-E se basa en los sistemas de coordenadas en movimiento (Ingeniería Automatización , 2017):

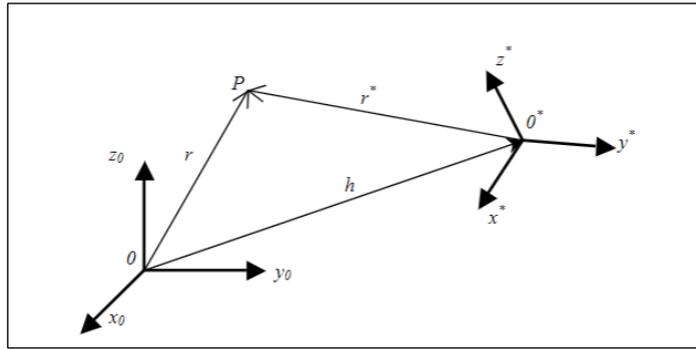


Figura 8-2. Sistemas de coordenadas en movimiento

Fuente: Ingeniería Automatización, 2017.

Realizado por: Merchán, B. & Chimbo, A. 2020

Conforme la figura 8-2 se tiene que el sistema de coordenadas O^* se desplaza y gira en el espacio respecto del sistema de referencia de la base O , el vector que describe el origen del sistema en movimiento es h y el punto P se describe respecto del sistema O^* a través del vector r^* , de acuerdo a esto, la descripción del punto P respecto del sistema de la base es:

$$r = r^* + h \quad (6-2)$$

$$\frac{dr}{dt} = \frac{dr^*}{dt} + \frac{dh}{dt} = v^* + vh \quad (7-2)$$

donde v^* es la velocidad del punto P respecto del origen del sistema O^* en movimiento y vh es la velocidad del origen del sistema O^* respecto de la base. Si el punto P se desplaza y gira respecto del sistema O^* la ecuación (7-2) debe escribirse como:

$$v = \frac{dr^*}{dt} + \frac{dh}{dt} = \left(\frac{d^*r^*}{dt} + wxr^* \right) + \frac{dh}{dt} \quad (8-2)$$

donde $\frac{d^*r^*}{dt}$ es la velocidad lineal del punto P respecto del origen O^* y wxr^* es la velocidad angular del punto P respecto del origen O^* .

De manera similar la aceleración general del sistema se puede describir como:

$$a = \frac{dv}{dt} = \frac{d^2r^*}{dt^2} + \frac{d^2h}{dt^2} = a^* + ah \quad (9-2)$$

$$a = \frac{d^2 r^*}{dt^2} + 2wx \frac{d^* r^*}{dt} + wx(wxr) + \frac{dw}{dt} xr^* + \frac{d^2 h}{dt^2} \quad (10-2)$$

Dinámica Directa

Las ecuaciones de movimiento planteadas en el apartado anterior permiten resolver el problema de la dinámica directa. En el artículo (Efficiente dynamic computer simulation of robotic manipulators, 1982), se presenta cuatro métodos para la resolución del problema dinámico directo de una cadena cinemática abierta utilizando la formulación de N-E por medio de una comparación de la eficiencia computacional de los cuatro métodos presentados, los autores concluyen que el tercero de los presentados es el más eficiente frente al tiempo de cómputo (Ingeniería Automatización , 2017).

Walker y Orín resuelven la ecuación general del robot (Ingeniería Automatización , 2017):

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + K(q)^T k = \tau \quad (11-2)$$

donde:

$H(q)\ddot{q}$: Matriz no singular NxN de los momentos de inercia.

$C(q, \dot{q})$: Matriz NxN que contabiliza los efectos de las aceleraciones centrífugas y de Coriolis.

$G(q)$: Vector Nx1 que contabiliza los efectos de la gravedad.

$K(q)$: Matriz Jacobiana 6xN que especifica los pares (fuerzas) creados en cada articulación debido a las fuerzas y momentos externos aplicados sobre el elemento N.

k : Vector 6x1 de los momentos y fuerzas externas ejercidos sobre el elemento N.

τ : Vector Nx1 de los pares (fuerzas) de cada articulación.

q : Vector Nx1 de las variables articulares.

De la ecuación (11-2) se observa que las fuerzas y pares en las articulaciones son funciones lineales de las aceleraciones articulares. Se define b como un vector equivalente a los efectos de la gravedad, las aceleraciones centrífugas y de Coriolis y las fuerzas externas aplicadas sobre el elemento N:

$$b = C(q, \dot{q})\dot{q} + G(q) + K(q)^T k \quad (12-2)$$

Por lo tanto, se reescribe la ecuación (11-2) como:

$$H(q)\ddot{q} = (\tau - b) \quad (13-2)$$

2.4.4 Sistema de control

Una vez obtenido el modelo del sistema se procede a realizar el control, tiene como objetivo obtener, mediante la manipulación de las variables de control, un dominio sobre las variables de salida, de modo que estas alcancen unos valores prefijados (Gómez Valentín, et al., 2004). El mecanismo de control a usarse es un PID (Proporcional-Integral-Derivativo) este usa la realimentación para rechazar las perturbaciones, elimina el error estacionario con la acción integral y puede anticipar el futuro con la acción derivativa (Vilchis, 2020). En la figura 9-2 se muestra la estructura básica de control PID, donde: $r(t)$ se denomina variable de referencia, $u(t)$ es la señal de control, $y(t)$ es la variable controlada y $e(t)$ es el error y resulta de la diferencia entre $r(t)$ y $y(t)$, si el error es grande significa que el estado del sistema se encuentra lejos del estado deseado, por el contrario, si es pequeño el sistema ha alcanzado el estado deseado.

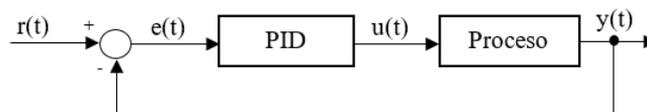


Figura 9-2. Estructura básica del control PID

Fuente: Rojas,2010.

Realizado por: Merchán, B. & Chimbo, A. 2020

En cuestiones de robótica, este algoritmo se puede describir de la siguiente manera:

$$\tau = K_p \tilde{q} + K_I \int_0^t \tilde{q} dt + K_D \dot{\tilde{q}} \quad (14-2)$$

Donde τ es la variable de control y \tilde{q} es el error de posición. Dicha variable de control está compuesta por tres términos: proporcional, integral y derivativo, calculados con respecto \tilde{q} . K_p , K_I y K_D son las ganancias correspondientes a cada término (Vilchis, 2020).

2.5 Recursos para simulación

Existen varios motivos para emplear software de simulación en el área de ingeniería y la ciencia (MathWorks, 2017):

- Resulta más fácil, más económico o más seguro crear y simular un modelo matemático de un sistema real que crear y probar un prototipo físico.
- Si no se tiene disponibilidad del sistema físico, es viable modelizarlo con mayor o menor fidelidad como un sistema dinámico, el cual se podrá simular para examinar diferentes opciones de diseño.
- Una vez se tenga disponible el dispositivo físico, las pruebas de control se reducen puesto que ya se realizó un trabajo previo con los modelos matemáticos diseñados en software.

2.5.1 MatLab

Es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Su nombre es la abreviación de “MATriz LABoratory”. Matlab es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir herramientas reutilizables, funciones y programas especiales (conocidos como M-archivos) en código Matlab, además de agrupar en Toolbox (también llamadas librerías). Matlab, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos (Casado, 2007).

Esta aplicación se usa para cálculo simbólico y numérico para modelos matemáticos en diferentes contextos de aplicación. Para el desarrollo de un sistema de control se usa desde la modelización de la planta hasta el diseño y ajuste de los algoritmos de control y la lógica de supervisión, finalizando con la implementación, MatLab ofrece (MathWorks, 2017):

- Un entorno de diagramas de bloques multidominio para modelizar la dinámica de la planta, diseñar algoritmos de control y ejecutar simulaciones de lazo cerrado.

- Funciones prediseñadas y herramientas interactivas para analizar el sobreimpulso, el tiempo de subida, el margen de fase, el margen de ganancia y otras características de rendimiento y estabilidad en los dominios de la frecuencia y el tiempo.
- Lugar de raíces, diagramas de Bode, LQR, LQG, control robusto, control predictivo de modelos y otras técnicas de diseño y análisis.
- Ajuste automático de sistemas de control PID, de ganancia programada y SISO/MIMO arbitrarios.
- Modelización, diseño y simulación de la lógica de supervisión para llevar a cabo la planificación, el cambio de modo y la detección, aislamiento y recuperación de errores (FDIR).

Requerimientos del sistema

A continuación, se detallan los requerimientos para el uso de la versión 2017a de MatLab (MathWorks, 2019):

Sistemas operativos

- Windows 10
- Windows 8.1
- Windows 7
- Windows Server 2012

Procesadores

Mínimo: cualquier procesador Intel o AMD x86-64

Recomendado: cualquier procesador Intel o AMD x86-64 con cuatro núcleos lógicos y compatibilidad con el conjunto de instrucciones AVX2

Disco

Mínimo: 2 GB de espacio en disco duro solo para MATLAB, 4-6 GB para una instalación típica

RAM

Mínimo: 2 GB

Recomendado: 4 GB

Gráficos

No se requiere una tarjeta gráfica específica.

Se recomienda una tarjeta gráfica acelerada por hardware compatible con OpenGL 3.3 con 1 GB de memoria GPU.

2.5.2 *Software de simulación 3D*

Actualmente existe una amplia variedad de programas de simulación 3D que permiten generar objetos tridimensionales e imágenes fotorrealistas, sin embargo, pocos reúnen tres características claves que como usuarios interesan mucho; utilizar poco espacio en disco al instalarse, bajo consumo de recursos del computador y que sea software libre (Virtual Visit with 3D Modeling of the academic spaces of the UTSV, 2018).

Entornos universales de simulación de robots

La tabla 4-2 presenta una lista de programas universales para modelado, simulado y análisis del comportamiento de robots. Lo que tienen en común todos estos programas es que el usuario tiene la posibilidad de construir y probar modelos con robots de diferentes fabricantes en un mismo entorno. El modelo virtual es controlado por medio de un script en diferentes lenguajes de programación: C/C++, LUA, Matlab, Octave, Python, dependiendo del entorno. Después de analizar los diferentes simuladores y sus características, se ha optado por escoger RoboDK para desarrollar la simulación 3D. Debido a su capacidad para simular entornos 3D complejos, la compatibilidad con tecnología de realidad virtual y además permite importar la mayoría de los formatos 3D estándar como los formatos STL, STEP (o STP) e IGES (o IGS).

Tabla 4-2: Tabla comparativa de los *softwares* 3D

Desarrolladores	Software	Lenguaje de programación	Características
Coppelia Robotics	V-Rep	LUA, C/C++, Python, Java, Matlab, Octave.	Permite la programación en diversos lenguajes. Permite la simulación en entornos 3D complejos y en varios medios, como aire, agua. Tiene un foro soporte atendido por profesionales.
Cyberbotics	Webots	C/C++, Java, Python, Matlab	Fácil de programar y de transferir a campo real, es utilizado a nivel educativo por su simplicidad y sus archivos son muy

			compactos y ligeros. Una de sus principales ventajas es que permite al usuario interactuar con el modelo durante la simulación.
MathWorks/ Peter Corke	Robotic Systems Toolbox/ Robotics Toolbox	Octave, Matlab	<p>Proporciona una gran cantidad de funciones para el estudio y simulación de robots más comunes.</p> <p>Las rutinas se escriben generalmente de manera sencilla, para permitir una fácil comprensión.</p> <p>Transforma datos de varios tipos como: vectores, cuaternios, transformaciones homogéneas y más.</p> <p>Software libre.</p>
Open Source Robotics Foundation	Gazebo	C++	<p>Permite importar robots mediante el formato SDF. Este formato basado en XML describe objetos y entornos para simulación de robots, visualización y control.</p> <p>Permite el desarrollo de prototipos, el sistema dispone de modelos propios.</p>
RoboDK	RoboDK	Python, C#, C++, Visual Basic, Matlab	<p>Simulador enfocado en aplicaciones de robots industriales. Los programas de robot pueden ser creados, simulados y generados fuera de línea para un brazo robot específico y un controlador de robot.</p> <p>Admite más de 300 modelos de robots de más de 30 fabricantes.</p> <p>Permite mostrar los espacios de trabajo de los robots para evaluar la alcanzabilidad de un robot para asegurarse de que la ubicación de alguna pieza es razonable dado el tamaño del robot.</p> <p>Soporta la mayoría de los formatos 3D estándar como los formatos STL, STEP (o STP) e IGES (o IGS).</p> <p>Si trabaja en un entorno de fabricación, uno de los mayores beneficios es que el software le permite programar el robot sin interrumpir la producción.</p>

			<p>Si trabaja en diseño o arquitectura de productos, el mayor beneficio puede ser la facilidad de programación y la enorme flexibilidad que proporciona el uso de un robot para construir sus prototipos.</p> <p>En la última versión de RoboDK se ha introducido compatibilidad con auriculares de realidad virtual.</p>
--	--	--	---

Fuente: Jakubiec, 2018.

Realizado por: Merchán, B. & Chimbo, A. 2020

Interfaz de Aplicación del Programa (API) de RoboDK

Es un conjunto de rutinas y comandos que RoboDK presenta mediante un lenguaje de programación que permite la programación de diversos robots utilizando un lenguaje de programación único. En comparación con la programación de robots específica del proveedor, con la API del programa es posible simular y programar cualquier robot usando un lenguaje de programación único/universal. La API de RoboDK de Matlab está disponible después de instalar el programa. Además, es compatible con la realidad virtual (VR) para tener una experiencia simulada. (RoboDK, 2017).

Las siguientes tareas pueden ser ejecutadas con la API de RoboDK (RoboDK, 2017):

- Automatizar la simulación: Crear macros para automatizar tareas específicas dentro del simulador de RoboDK, tales como mover objetos, sistemas de coordenadas o robots.
- Programación fuera de línea: Programación de robots fuera de línea desde un lenguaje de programación universal.
- Programación en línea: Es posible mover los robots y recuperar su posición actual desde la API de RoboDK. RoboDK moverá los robots usando controladores del robot.
- El mismo programa que se utiliza para la simulación se puede utilizar para generar programas de robot y para mover el robot en tiempo real.

Requerimientos del sistema

A continuación, se detallan los requerimientos para el uso de RoboDK 5.1 (RoboDK, 2020):

Sistema operativo:

- Windows Vista, Windows 7
- Windows 8 y Windows 10 (versión de 32 o 64 bits)
- Mac OS (64 bits), Ubuntu 16 o Ubuntu 18 (64 bits)

Memoria: 2 GB mínimo. Recomendado 4 GB o más.

Configuración de gráficos: una pantalla con una resolución mínima de 1024x768 píxeles. El controlador de gráficos debe ser compatible con OpenGL 3.0 o posterior. Se recomiendan tarjetas gráficas con OpenGL acelerado por hardware, incluida la memoria interna.

Espacio en disco duro: 40 GB, 1 GB libre.

Mouse: se recomiendan 2 botones (mínimo), 3 botones o 2 botones con la rueda central del ratón.

Realidad virtual: es compatible con cualquier auricular que funcione con SteamVR / OpenVR. Esto se debe a que usa la biblioteca OpenVR para desarrollarlo, que es compatible con una variedad de auriculares diferentes (aunque es difícil encontrar una lista definitiva). La nueva función ha sido probada con Oculus Rift S y HTC Vive, pero también debería funcionar con otros auriculares. Para opciones más económicas, incluido Google Cardboard, se necesita usar un programa como RiftCat que permite que la PC transmita la pantalla al teléfono inteligente.

CAPÍTULO III

3 MARCO METODOLÓGICO

Luego de realizada la revisión bibliográfica en el capítulo anterior, este capítulo describe el diseño del *software* y *hardware* del sistema, detallando cada una de las etapas que lo conforman.

3.1 Requerimientos del sistema

- Permitir la interacción del usuario con el modelo virtual del robot UR3 por medio de mouse y teclado.
- Permitir la inmersión en el entorno 3D con el uso de gafas de realidad virtual.
- Cumplir trayectorias establecidas de acuerdo al objeto a pintar.
- Permitir el ingreso de n capas de pintura.
- Permitir mover el robot y guardar los movimientos.

3.2 Concepción de la arquitectura general del sistema

En la figura 1-3 se aprecian los elementos constitutivos del sistema como son un ordenador en el cual se desarrollará el entorno virtual en el que el usuario va a interactuar por medio del teclado y mouse. La experiencia de inmersión se obtiene mediante el uso de gafas de RV.

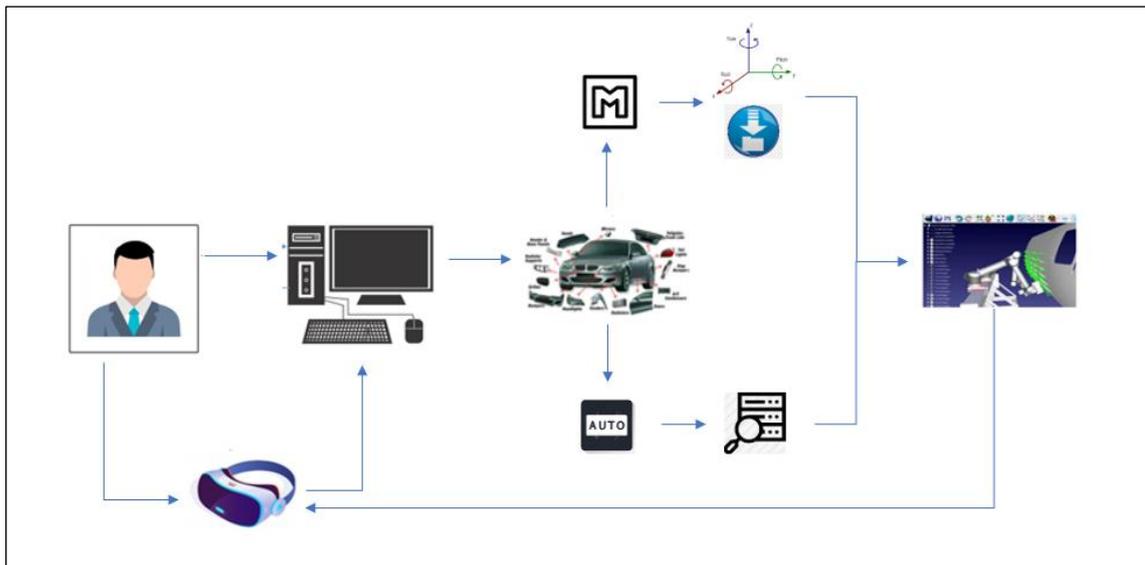


Figura 1-3. Diagrama de bloques del sistema

Realizado por: Merchán, B. & Chimbo, A. 2020

3.3 Diagrama de bloques del sistema

En la figura 2-3 se puede observar 3 bloques: ingreso de datos, procesamiento de la información y simulación. A continuación, se presenta una breve descripción de cada uno.

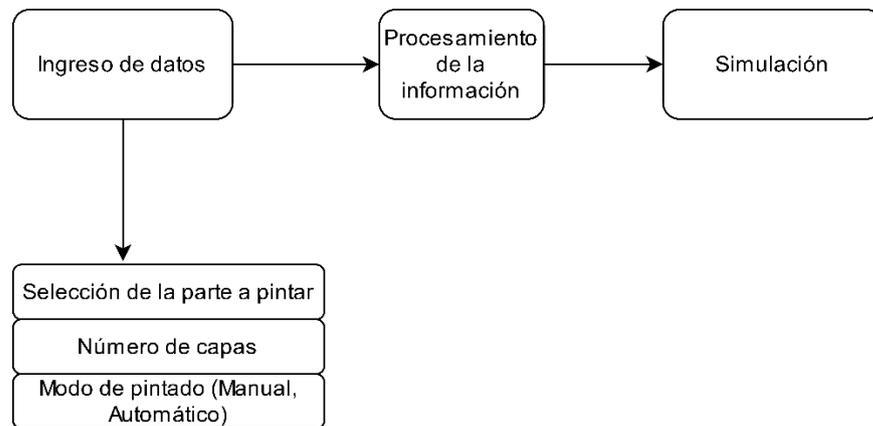


Figura 2-3. Diagrama de bloques del sistema

Realizado por: Merchán, B. & Chimbo, A. 2020

3.3.1 *Bloque Ingreso de datos*

En esta etapa se da la posibilidad que el usuario seleccione la pieza a pintar, ingresar un n número de capas y elegir el modo de pintado. Esta información será procesada en la siguiente etapa.

3.3.2 *Bloque de Procesamiento de la información*

Al seleccionar el modo manual de pintado el usuario crea la trayectoria que debe realizar el robot de acuerdo con la parte de la carrocería seleccionada anteriormente. Se aplica un controlador PID para comparar la posición deseada con la posición actual del efector final y reducir el error de posicionamiento. Al seleccionar el modo automático de pintado se ejecutará la trayectoria preestablecida que debe realizar el robot de acuerdo con la parte de la carrocería seleccionada anteriormente. El proceso de pintado se realizará n veces.

3.3.3 *Bloque de Simulación*

En este bloque se apreciará los movimientos que realizará el brazo robótico al alcanzar los puntos establecidos en las trayectorias del bloque anterior.

3.4 Elementos *hardware* del sistema

A continuación, se detallan las principales características de los elementos *hardware* a usar:

Ordenador

El principal elemento hardware para el sistema a implementar es un ordenador el cuál se selecciona de acuerdo a los requerimientos de los softwares a usar. En la figura 3-3 se muestra que se usó un ordenador portátil marca Asus pro de séptima generación.

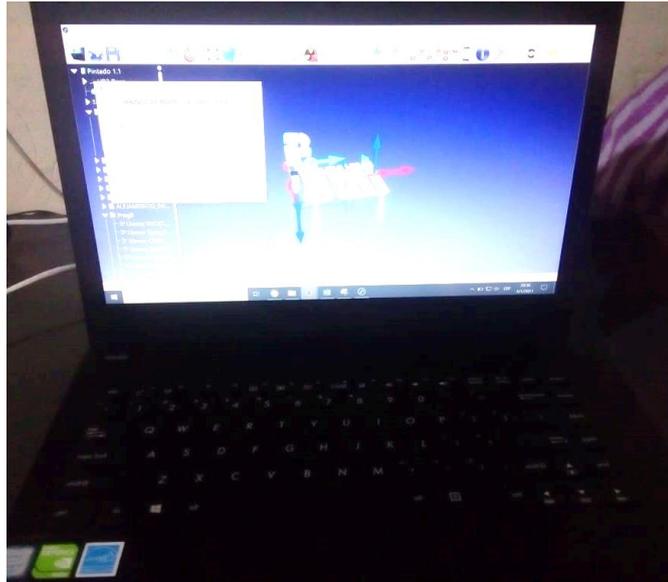


Figura 3-3. Ordenador Asus pro

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se detallan las características:

- Sistema operativo: Windows 10 (64 bits).
- Procesador: Intel CORE i7
- Disco: 512 ssd
- RAM: 12GB.
- Tarjeta gráfica: Nvidia gforce 2GB.
- Mouse: 2 botones (mínimo), 3 botones o 2 botones con la rueda central del ratón.

Gafas de RV

La selección de las gafas de RV se realiza de acuerdo con el requerimiento del software RoboDK para la aplicación de RV, por lo cual se puede usar cualquier visor de realidad virtual compatible con SteamVR.



Figura 4-3. Gafas de RV, VR BOX

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se detallan las características de las gafas seleccionadas (VR BOX, 2020):

- Compatible con pantallas de 3.5" a 6".
- Ángulo de visión entre 85° y 95°.
- Lentes de gran tamaño, equivale a ver una pantalla de 100" desde 3m.
- Ópticas fabricadas en resina de alta calidad.
- Libertad total de movimiento.

Las funciones que se pueden ejecutar con este tipo de gafas son las descritas a continuación (VR BOX, 2020):

- 360°+3D: permite experimentar una inmersión total con la experiencia en vídeo, mirar hacia dónde quiera y no se pierde detalle de su alrededor, debido a que forma parte de la acción.
- JUEGO: Con el mando ergonómico Bluetooth se puede disfrutar de nuevas experiencias en multitud de juegos 3D especialmente desarrollados para este tipo de gafas.

3.5 Esquema de conexión

En la figura 5-3 se aprecia la conexión que se realizó entre el PC y las gafas de realidad virtual, en la cual se observa que se debe conecta mediante USB el smartphone a la PC, para transmitir el

entorno virtual desde el PC hacia el Smartphone. La pantalla del smartphone se dividirá en dos partes y en cada parte se presentará imágenes similares. Se inserta el smartphone dentro de las gafas de realidad virtual (VR BOX) para que el usuario pueda usarlas y tener la inmersión en el entorno simulado.

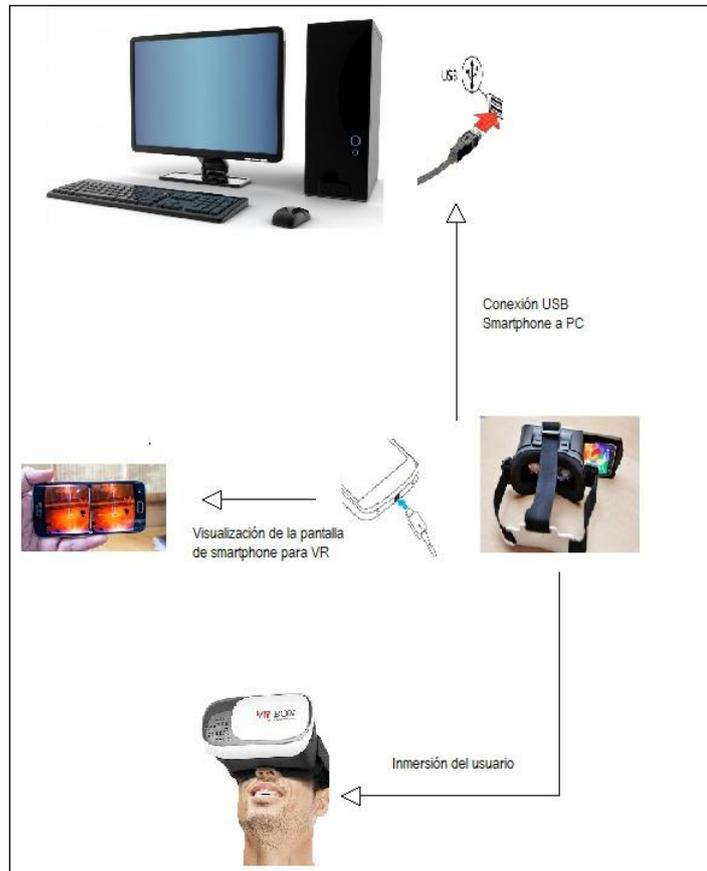


Figura 5-3. Conexión de dispositivos para el uso de realidad virtual

Realizado por: Merchán, B. & Chimbo, A. 2020

RoboDK es compatible con la realidad virtual (VR) para tener una experiencia simulada de las simulaciones RoboDK. Se debe seleccionar Conectar→Conectar auriculares de RV (Mayús+V) para iniciar la experiencia de realidad virtual, como se muestra en la figura 6-3.

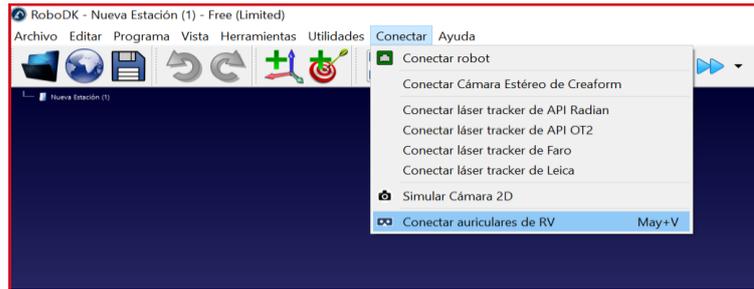


Figura 6-3. Conexión para realidad virtual

Realizado por: Merchán, B. & Chimbo, A. 2020

3.6 Herramientas *software* del sistema

En la figura 7-3 se presenta el diagrama de flujo que describe el funcionamiento del programa.

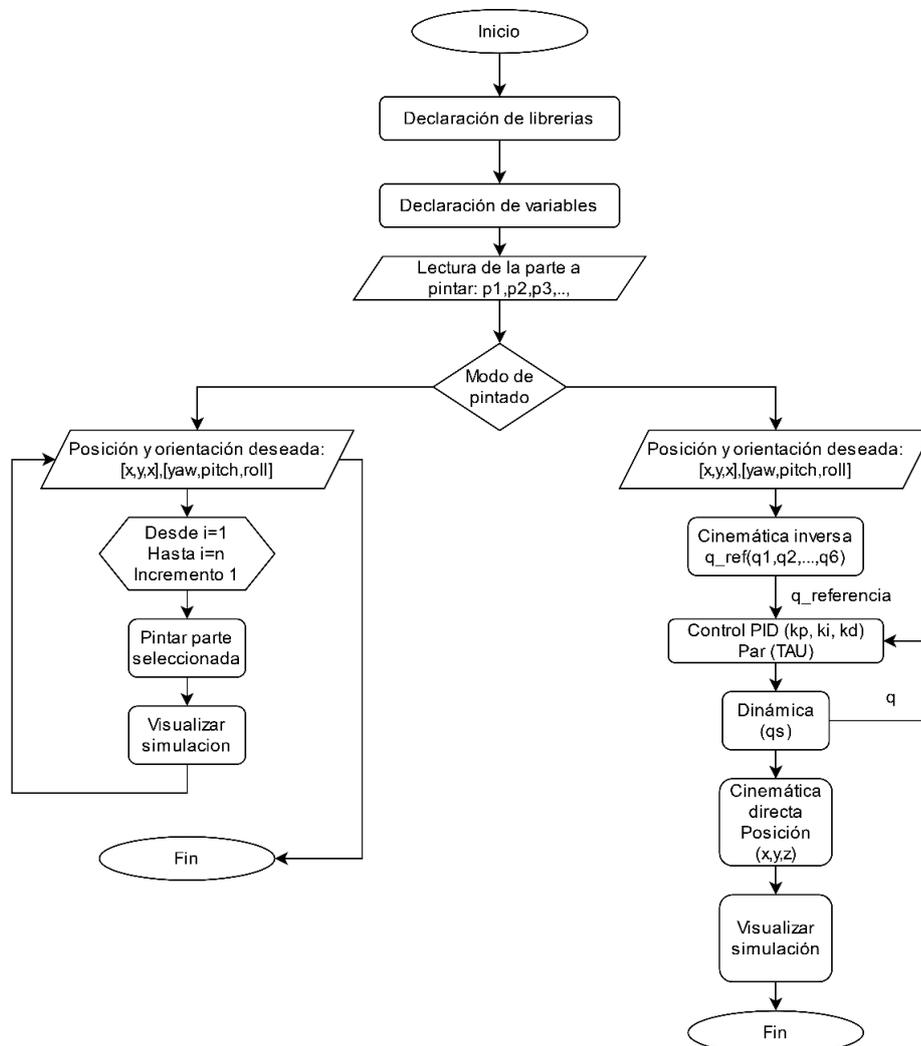


Figura 7-3. Diagrama de flujo del programa

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Para usar la API de RoboDK para MatLab se requiere declarar la clase Robolink, que es el enlace a RoboDK y permite crear macros para RoboDK, simular aplicaciones y generar programas sin conexión (RoboDK, 2017).
- Se declara variables globales para poder usarlas en el algoritmo.
- Luego de declaradas las variables es necesario una lectura de la parte a pintar.
- Conociendo la parte a pintar se debe seleccionar el modo que se desea pintar, de esta manera se puede seleccionar el modo manual o el modo automático.
- Si se selecciona el modo automático se requiere ingresar el número de capas que se pintará, luego se ejecuta automáticamente un programa preestablecido para el pintado de la parte seleccionada.
- Si se selecciona el modo manual el usuario tiene la posibilidad de establecer la posición y orientación del brazo robótico e ir estableciendo la trayectoria deseada para pintar la parte seleccionada. En esta opción se ejecutarán las funciones:
 - `inversekinematic1()`: calcula la cinemática inversa del brazo robótico.
 - `CONTROL ()`: ejecuta el control de posicionamiento del brazo robótico.
 - `dinámica ()`: calcula la dinámica del brazo robótico, para establecerlo en la posición y orientación deseada.
 - `forwardkinematic()`: calcula la cinemática directa del brazo robótico para conocer si se posicionó correctamente.
- Con cualquier selección del modo de pintado se visualizará la simulación del pintado en RoboDK por medio de gafas de RV.

3.7 Modelación del UR3

La figura 8-3 presenta la implementación en Simulink del control PID para el robot UR3, se toma los valores de posición (x, y, z) y orientación (yaw, pitch, roll) ingresadas por el usuario. Estos datos ingresan al bloque de cinemática inversa para obtener como resultado las posiciones articulares ($q_1, q_2, q_3, q_4, q_5, q_6$) siendo estas las posiciones de referencia. El bloque de control tiene como entrada la posición de referencia y la posición leída para al aplicar (14-2) obtener el par que ingresa al bloque dinámica, este bloque representa el modelo dinámico del robot. Al aplicar este modelo se obtiene las aceleraciones necesarias para mover el robot. Para conocer si estas son correctas se aplica dos integradores y se obtienen las posiciones articulares que al ser

ingresadas al bloque cinemática directa se puede tener la posición y orientación del efector final del UR3.

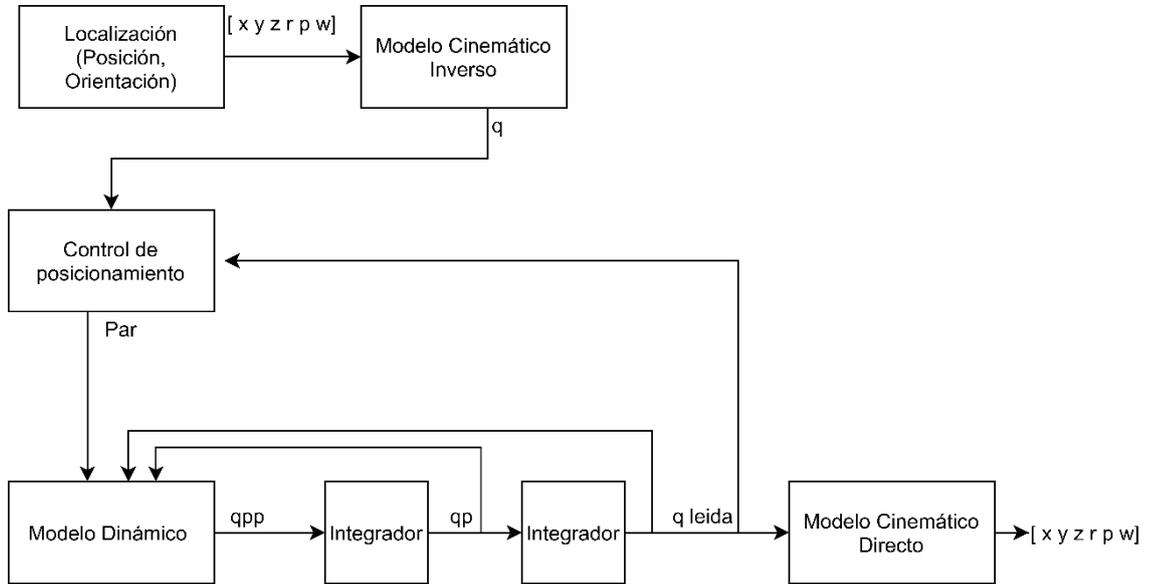


Figura 8-3. Implementación de control PID para UR3

Realizado por: Merchán, B. & Chimbo, A. 2020

Universal Robots proporciona los parámetros para cálculos de cinemática y dinámica de los robots de la serie UR. Los parámetros DH se toman en función a la figura 9-3.

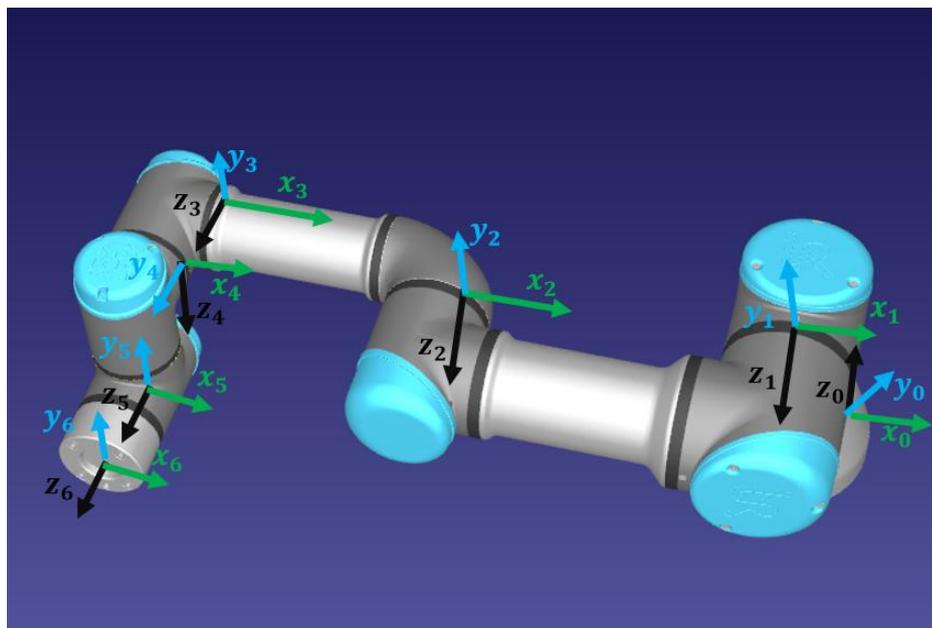


Figura 9-3. Diagrama de los sistemas para la obtención de los parámetros DH

Realizado por: Merchán, B. & Chimbo, A. 2020

Los cuatro parámetros DH (theta, a, d, alfa) dependen exclusivamente de las características geométricas de cada eslabón y de las articulaciones que le unen con el anterior y siguiente. En la tabla 1-3 se muestran los parámetros DH para el robot UR3. Además se muestran la masa y centro de masa de cada eslabón.

Tabla 1-3: Parámetros de DH del robot UR3

Cinemática	theta [rad]	a [m]	d [m]	alfa [rad]	Dinámica	Masa [kg]	Centro de masa [m]
Articulación 1	0	0	0.1519	$\pi / 2$	Eslabón 1	2	[0, -0.02, 0]
Articulación 2	0	-0.24365	0	0	Eslabón 2	3.42	[0.13, 0, 0.1157]
Articulación 3	0	-0.21325	0	0	Eslabón 3	1.26	[0.05, 0, 0.0238]
Articulación 4	0	0	0.11235	$\pi / 2$	Eslabón 4	0.8	[0, 0, 0.01]
Articulación 5	0	0	0.08535	$-\pi / 2$	Eslabón 5	0.8	[0, 0, 0.01]
Articulación 6	0	0	0.0819	0	Eslabón 6	0.35	[0, 0, -0.02]

Fuente: UNIVERSAL ROBOTS, 2020.

Cinemática directa

Tomando los parámetros DH descritos en la tabla 1-3 para la cinemática directa se creó un algoritmo en MatLab para resolver la cinemática directa del brazo robótico UR3. En la figura 10-3 se presenta el diagrama de flujo que describe el funcionamiento del programa para resolver la cinemática directa.

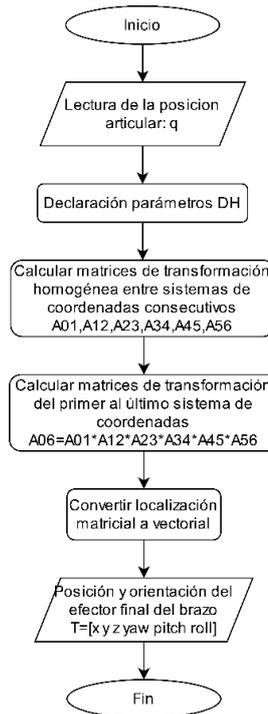


Figura 10-3. Diagrama de flujo para la cinemática directa

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Se hace la lectura de las coordenadas articulares: $q_1, q_2, q_3, q_4, q_5, q_6$.
- Se declara los parámetros DH del brazo robótico
- Se calcula las matrices de transformación homogénea entre sistemas de coordenadas consecutivos, a partir de los parámetros DH con la ecuación (1-2).
- Se calcula la matriz de transformación del primer al último sistema de coordenadas, a través del producto de las matrices de transformación homogénea entre sistemas de coordenadas consecutivos, como se muestra en la ecuación (2-2).
- Se establece la matriz de transformación del primer sistema de coordenadas al último en función del vector q de variables articulares.
- Se convierte la expresión matricial a una vectorial (T).
- Se establece el vector de localización (posición y orientación) del efector final (T) correspondiente a la solución de la cinemática directa.

Cinemática inversa

Para resolver la cinemática inversa se hace uso del método geométrico, debido a la complejidad del UR3 ya que no permite desacoplar el problema de posicionamiento y orientación (Keating, 2017):

Articulación 1

Del quinto sistema de referencia con respecto a la base, se toma el vector posición 0P_5 , como se observa en la figura 11-3.

0P_5 se puede calcular con el uso de la matriz de transformación homogénea 0T_6 . En la figura 12-3 se observa que al desplazar el sexto sistema de referencia $-d_6$ unidades en dirección de z_6 separa poder obtener el quinto sistema de referencia, todo esto expresado en la ecuación (1-3). En la figura 12-3 se observa la proyección de 0P_5 en el plano xy del sistema de referencia base para poder descomponerlo, en base a lo mostrado en las figuras 13-3 y 14-3 se extrae las relaciones matemáticas para calcular θ_1 , que viene dado por la ecuación (4-3).

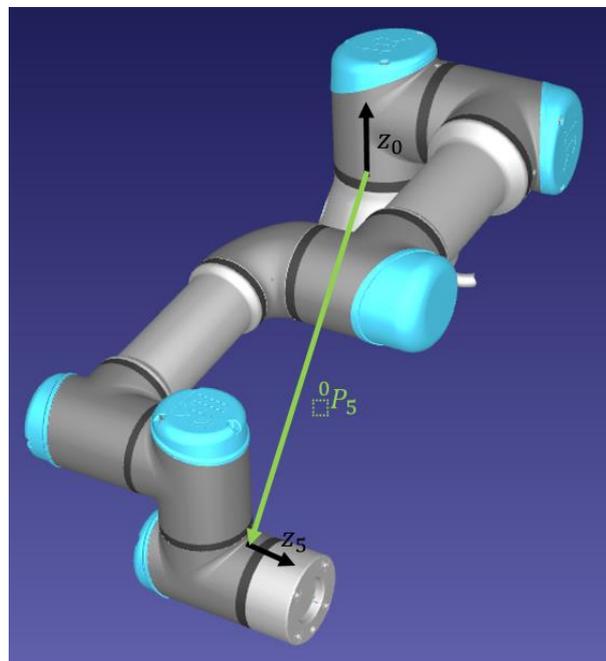


Figura 11-3. Quinto sistema de referencia respecto a la base

Realizado por: Merchán, B. & Chimbo, A. 2020

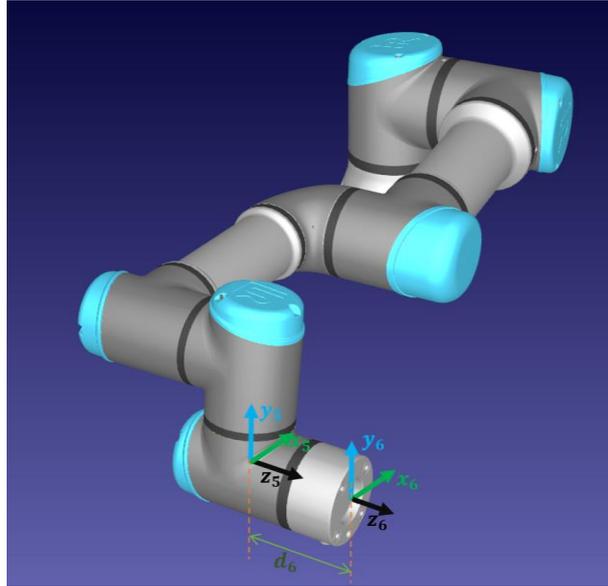


Figura 12-3. Quinto y sexto sistema de referencia

Realizado por: Merchán, B. & Chimbo, A. 2020

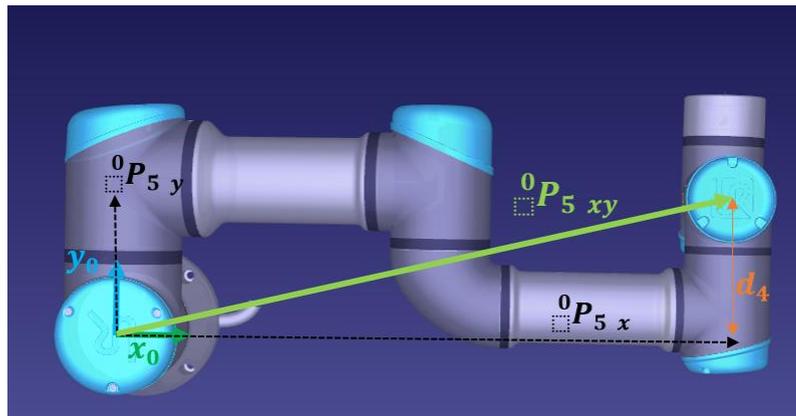


Figura 13-3. Vector de posición

Realizado por: Merchán, B. & Chimbo, A. 2020

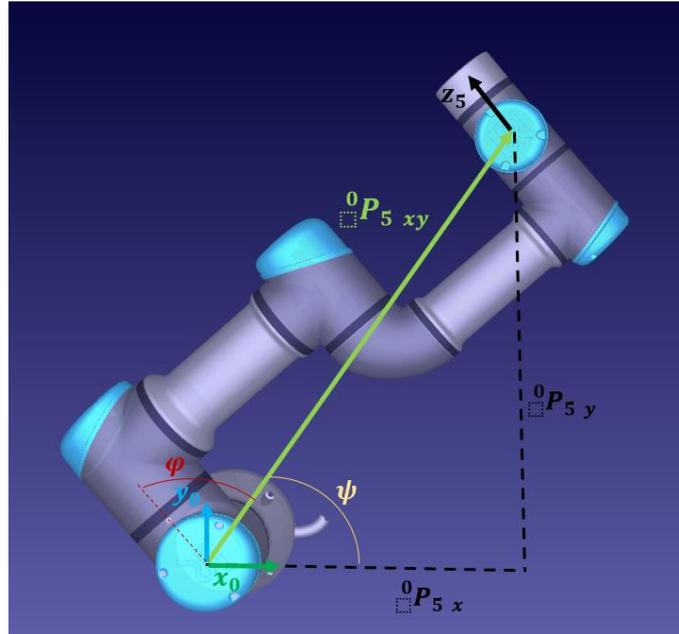


Figura 14-3. Punto general de partida

Realizado por: Merchán, B. & Chimbo, A. 2020

$$\varphi = \pm \arccos \left(\frac{d_4}{\sqrt{(({}^5P_0)_x)^2 + (({}^5P_0)_y)^2}} \right) \quad (2-3)$$

$$\psi = \text{atan2}(({}^5P_0)_y, ({}^5P_0)_x) \quad (3-3)$$

$$\theta_1 = \psi + \varphi + \frac{\pi}{2} \quad (4-3)$$

Articulación 5

Se trabaja con la vista superior del brazo robótico por mayor facilidad para calcular θ_5 , como se observa en la figura 15-3. Se refleja la relación entre el sexto sistema de referencia con el primero, de la cual es posible extraer la ecuación (5-3). Al considerar los valores determinados para el cálculo de θ_1 se obtiene la ecuación (6-3) y por consiguiente se obtiene θ_5 con la ecuación (7-3).

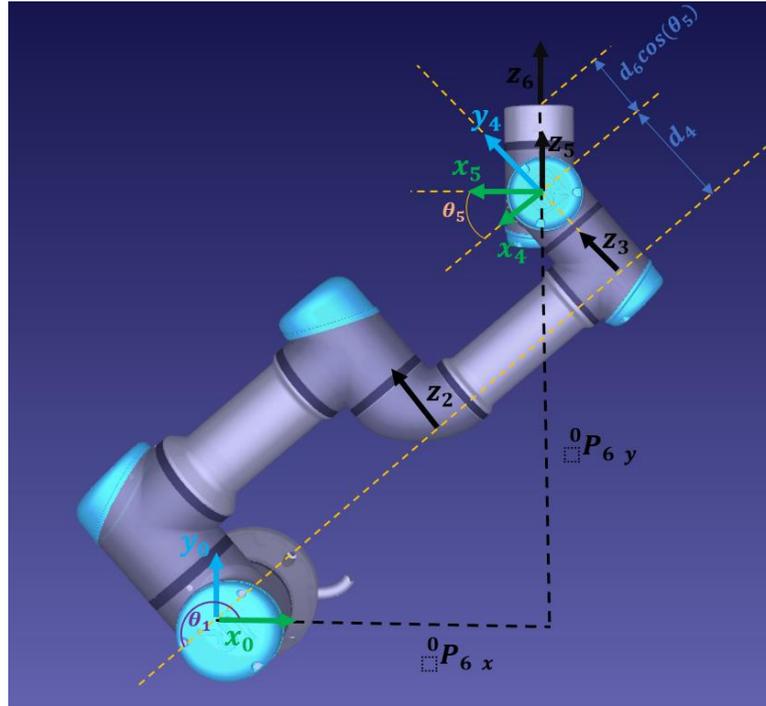


Figura 15-3. Relaciones para calcular el quinto ángulo

Realizado por: Merchán, B. & Chimbo, A. 2020

$$({}^1P_6)_z = d_6 \cos(\theta_5) + d_4 \quad (5-3)$$

$$({}^1P_6)_z = ({}^0P_6)_z \sin(\theta_1) - ({}^0P_6)_y \cos(\theta_1) \quad (6-3)$$

$$\theta_5 = \pm \arccos\left(\frac{({}^1P_6)_z - d_4}{d_6}\right) \quad (7-3)$$

Articulación 6

Para calcular θ_6 primero se debe encontrar 1T_6 , que es la matriz de transformación entre el sexto sistema de referencia y el primero como se expresa en la ecuación (8-3), tomando en cuenta la estructura de una matriz de transformación se deduce las ecuaciones (9-3) y (10-3) y que al despejar θ_6 se obtiene su valor expresado en la ecuación (11-3).

$$({}^1T_6) = (({}^1T_6)^{-1}({}^0T_6))^{-1} \quad (8-3)$$

$$-\sin(\theta_6) \sin(\theta_5) = z_y \quad (9-3)$$

$$\cos(\theta_6) \sin(\theta_5) = z_x \quad (10-3)$$

$$\theta_6 = \text{atan2}\left(\frac{-z_y}{\sin(\theta_5)}, \frac{z_x}{\sin(\theta_5)}\right) \quad (11-3)$$

Para el cálculo de θ_3 , θ_2 y θ_4 , se usan las relaciones que se observan en la figura 16-3.

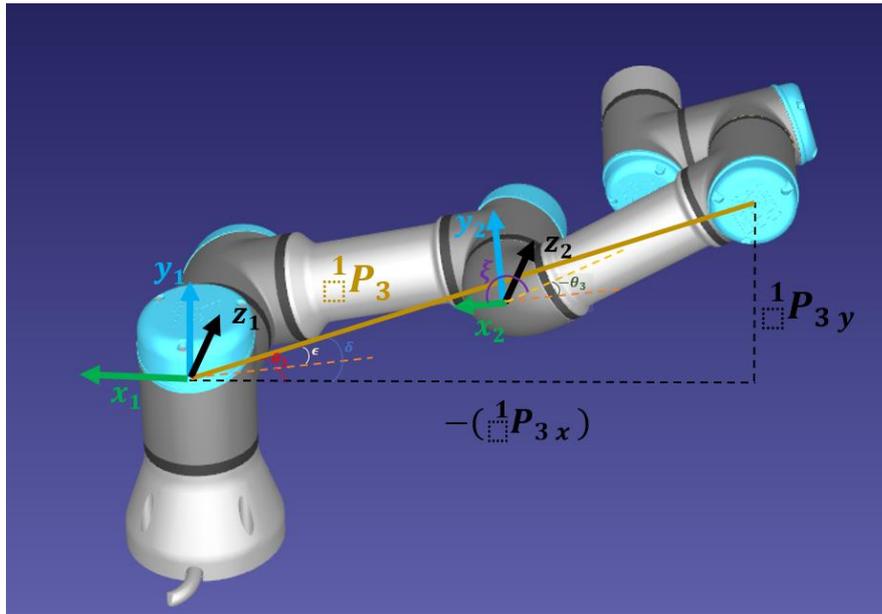


Figura 16-3. Relaciones para calcular los ángulos restantes

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 3

Es necesario primero conocer el tercer sistema de referencia con respecto al primero expresado matemáticamente en la ecuación (13-3), de lo cual se puede plantear las ecuaciones (14-3) y (15-3), con lo que se puede tener el valor de θ_3 , expresado en la ecuación (16-3).

$$({}^1_4T) = {}^1_6T {}^6_4T = {}^1_6T ({}^4_5T {}^5_6T)^{-1} \quad (12-3)$$

$${}^1P_3 = {}^1T_4 \begin{bmatrix} 0 \\ -d_4 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (13-3)$$

$$\cos(\xi) = \frac{\|{}^1P_3\| - a_2^2 - a_3^2}{2 a_2 a_3} \quad (14-3)$$

$$\cos(\xi) = -\cos(\pi - \xi) = -\cos(-\theta_3) = \cos(\theta_3) \quad (15-3)$$

$$\theta_3 = \pm \arccos\left(\frac{\|{}^1P_3\| - a_2^2 - a_3^2}{2 a_2 a_3}\right) \quad (16-3)$$

Articulación 2

Se extrae la relación expresada en la ecuación (17-3), donde los valores de δ y ϵ vienen dados por la ecuación (18-3) que al aplicarle la ley de senos se obtiene la ecuación (19-3), con lo que se puede calcular θ_2 , con la ecuación (19-3).

$$\theta_2 = -(\delta - \epsilon) \quad (17-3)$$

$$\delta = \text{atan2}(({}^1P_3)_{y'} - ({}^1P_3)_x) \quad (18-3)$$

$$\theta_2 = -\text{atan2}(({}^1P_3)_{y'} - ({}^1P_3)_{\hat{x}}) + \arcsin\left(\frac{a_3 \sin(\theta_3)}{\|{}^1P_3\|}\right) \quad (19-3)$$

Articulación 4

Es necesario primero conocer la matriz de transformación que relacione al tercer sistema de referencia con respecto al cuarto expresado matemáticamente en la ecuación (20-3), de esta se deduce que θ_4 se calcula con la expresión de la ecuación (21-3).

$${}^3T_4 = ({}^1T_2 {}^2T_3)^{-1} {}^1T_4 \quad (20-3)$$

$$\theta_4 = \text{atan2}(x_y, x_x) \quad (21-3)$$

En la figura 17-3 se presenta el diagrama de flujo que describe el funcionamiento del programa para resolver la cinemática inversa.

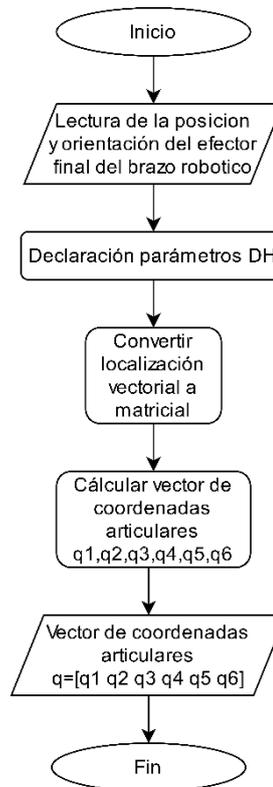


Figura 17-3. Diagrama de flujo para la cinemática inversa

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Se hace la lectura de la localización (posición y orientación) del efector final y se lo almacena en un vector (xyzrpw).
- Se declara los parámetros DH del brazo robótico
- Se convierte la expresión vectorial a una matricial (T).
- Se calcula las coordenadas articulares $q_1, q_2, q_3, q_4, q_5, q_6$ a partir de las ecuaciones (4-3), (19-3), (16-3), (21-3), (7-3) y (11-3) respectivamente.
- Se establece el vector de coordenadas articulares (q) correspondiente a la solución de la cinemática inversa del efector final en la posición y orientación expresadas en la matriz T.

Dinámica

Para el modelo dinámico directo se utiliza el tercer método de Walker y Orin y para el modelo dinámico inverso se emplea el algoritmo computacional Newton Euler, descritos en el capítulo 2, para un robot rotacional de seis grados de libertad. Las ecuaciones NE usadas para el cálculo de la dinámica inversa son las que se describen a continuación:

Vector i de localización:

$${}^{i-1}\rho_i = \begin{pmatrix} a \\ d \sin \alpha \\ d \cos \alpha \end{pmatrix} \quad (22-3)$$

Velocidad angular de las articulaciones:

$${}^i\omega_i = {}^iR_{i-1}({}^{i-1}\omega_{i-1} + z\dot{q}_i) \quad (24-3)$$

Aceleración angular de las articulaciones:

$${}^i\dot{\omega}_i = {}^iR_{i-1}({}^{i-1}\dot{\omega}_{i-1} + z\ddot{q}_i + ({}^{i-1}\omega_{i-1})x(z\dot{q}_i)) \quad (25-3)$$

Aceleración lineal articular:

$${}^i\dot{v}_i = ({}^i\dot{\omega}_i x {}^{i-1}\rho_i) + [{}^i\omega_i x ({}^i\omega_i x {}^{i-1}\rho_i)] + ({}^iR_{i-1})({}^{i-1}\dot{v}_i) \quad (26-3)$$

Aceleración del centro de masa de cada elemento:

$${}^i a_i = ({}^i\dot{\omega}_i x {}^i S_i) + {}^i\omega_i x [{}^i\omega_i x {}^i S_i] + {}^i\dot{v}_i \quad (27-3)$$

Fuerza en el centro de masa de cada elemento:

$${}^i F_i = {}^i a_i m_i \quad (28-3)$$

Par en el centro de masa de cada elemento:

$${}^i N_i = I_i {}^i\dot{\omega}_i + {}^i\omega_i x (I_i {}^i\omega_i) \quad (29-3)$$

Fuerzas articulares:

$${}^i f_i = ({}^iR_{i+1} {}^{i+1}f_{i+1}) + {}^i F_i \quad (30-3)$$

Pares articulares:

$${}^i\eta_i = {}^iR_{i+1} [{}^{i+1}\eta_{i+1}({}^{i+1}\rho_i x {}^{i+1}f_{i+1})] + [({}^{i-1}\rho_i + {}^iS_i) x {}^iF_i] + {}^iN_i \quad (31-3)$$

Fuerzas y pares de accionamientos:

$$\tau_i = ({}^i\eta_i)' ({}^iR_{i-1} z) + b_i \dot{q}_i \quad (32-3)$$

Para aplicar el algoritmo NE se usó los datos de las masas y centro de masas de los eslabones descritos en la tabla 1-3. Además, es necesario las matrices de inercia, las que se obtuvieron con la ayuda del software SOLIDWORKS. Para lo cual se selecciona la articulación para ver sus propiedades físicas, los momentos de inercia se muestran en una ventana como la de la figura 18-3.

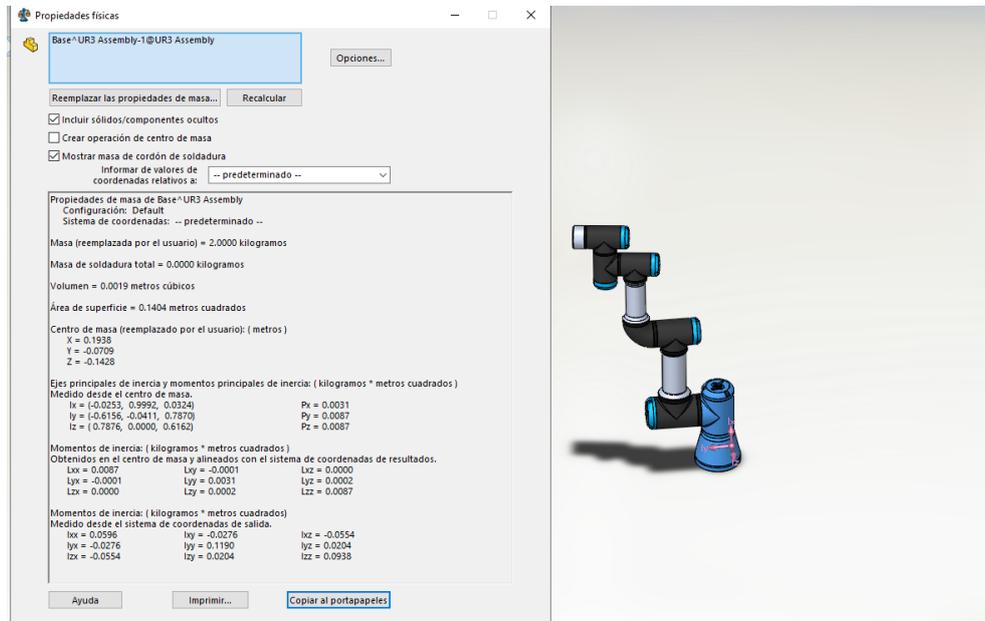


Figura 18-3. Propiedades físicas del primer eslabón

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se presenta las matrices de inercia de los eslabones:

$$I1 = \begin{bmatrix} 0.0087 & -0.0001 & 0 \\ -0.0001 & 0.0031 & 0.0002 \\ 0 & 0.0002 & 0.0087 \end{bmatrix}$$

$$I2 = \begin{bmatrix} 0.0296 & -0.0024 & -0.0004 \\ -0.0024 & 0.0040 & -0.0014 \\ -0.0004 & -0.0014 & 0.0295 \end{bmatrix}$$

$$I3 = \begin{bmatrix} 0.0071 & -0.0009 & 0 \\ -0.0009 & 0.0009 & -0.0003 \\ 0 & -0.0003 & 0.0072 \end{bmatrix}$$

$$I4 = \begin{bmatrix} 0.0007 & 0 & 0 \\ 0 & 0.0005 & 0 \\ 0 & 0 & 0.0007 \end{bmatrix}$$

$$I5 = \begin{bmatrix} 0.0007 & 0 & -0.0002 \\ 0 & 0.0008 & 0 \\ -0.0002 & 0 & 0.0007 \end{bmatrix}$$

$$I6 = \begin{bmatrix} 0.0002 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0002 \end{bmatrix}$$

En la figura 19-3 se presenta el diagrama de flujo que describe el funcionamiento del programa para resolver la dinámica.

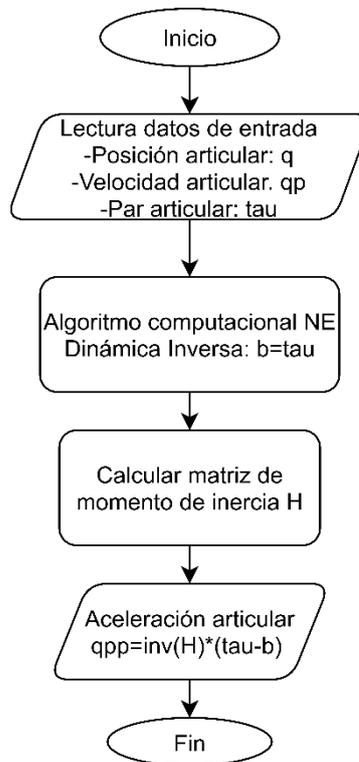


Figura 19-3. Diagrama de flujo para la dinámica directa

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Se hace la lectura de la posición, velocidad y pares articulares.

- Se ejecuta el algoritmo computacional NE, que resuelve la dinámica inversa, para obtener el vector de fuerzas de entrada a las articulaciones.
- Se calcula la matriz de momentos de inercia H .
- Se calcula el vector de aceleración de cada articulación, con la ecuación (13-2).
- Se establece el vector q_{pp} que representa la aceleración de cada articulación utilizando el tercer método de Walker y Orin, para resolver la dinámica directa.

En la figura 20-3 se presenta el diagrama de flujo que describe el funcionamiento del algoritmo computacional NE para resolver la dinámica inversa.

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Se hace la lectura de la posición, velocidad y aceleración de cada articulación.
- Se declaran los parámetros DH, masas de los elementos y matrices de inercia.
- Se calcula los vectores de localización, las matrices de transformación, la velocidad angular de las articulaciones, la aceleración angular de las articulaciones, la aceleración lineal articular, la aceleración del centro de masa de cada elemento, la fuerza en el centro de masa de cada elemento, el par en el centro de masa de cada elemento, las fuerzas articulares, los pares articulares, las fuerzas y pares de accionamientos. Con las ecuaciones (22-3) a (32-3).
- Se establece el vector τ que representa las fuerzas de entrada a las articulaciones

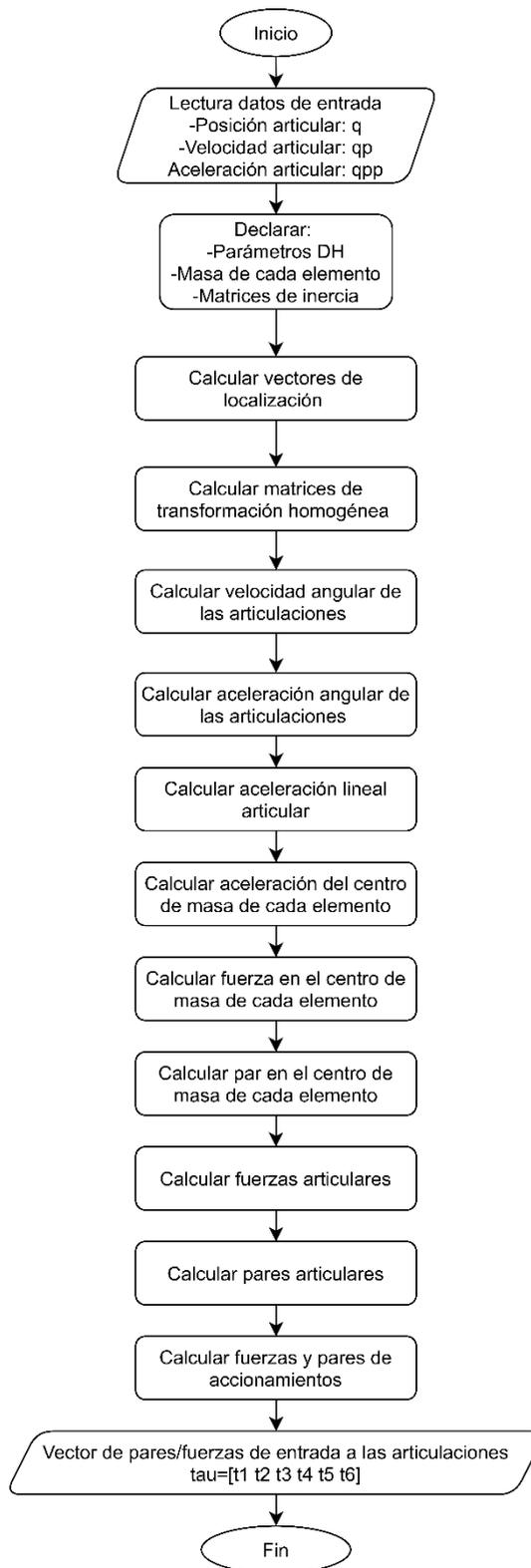


Figura 20-3. Diagrama de flujo para la dinámica inversa

Realizado por: Merchán, B. & Chimbo, A. 2020

Sistema de control

Generación de trayectoria

La planificación de trayectorias tiene como objetivo el encontrar una serie de posiciones para un robot, que permitirán trasladarlo desde un punto inicial a un punto final. Existen dos maneras principales para especificar el movimiento (Universidad de Santiago de Chile):

- Suministrando puntos consecutivos e ignorando la trayectoria espacial que describe el robot entre cada dos puntos.
- Especificando el camino que debe unir los puntos mediante una determinada trayectoria, tal como una línea recta o un círculo, que debe describir el robot en el espacio de trabajo.

Para el desarrollo se ha considerado suministrar puntos consecutivos e ignorar la trayectoria espacial. Tomando en consideración que los puntos no deben estar muy separados, si lo estuvieran el riesgo de que se generen movimientos imprevisibles o no controlados, es grande. Posteriormente el control de posicionamiento realiza el control para que el robot se mueva y pase por los puntos consecutivos dados por el usuario.

Control de posicionamiento

Se desarrolló un sistema de control, para posicionar al brazo robótico en una posición deseada. En la figura 21-3 se presenta el diagrama de flujo que describe el funcionamiento del algoritmo de control.

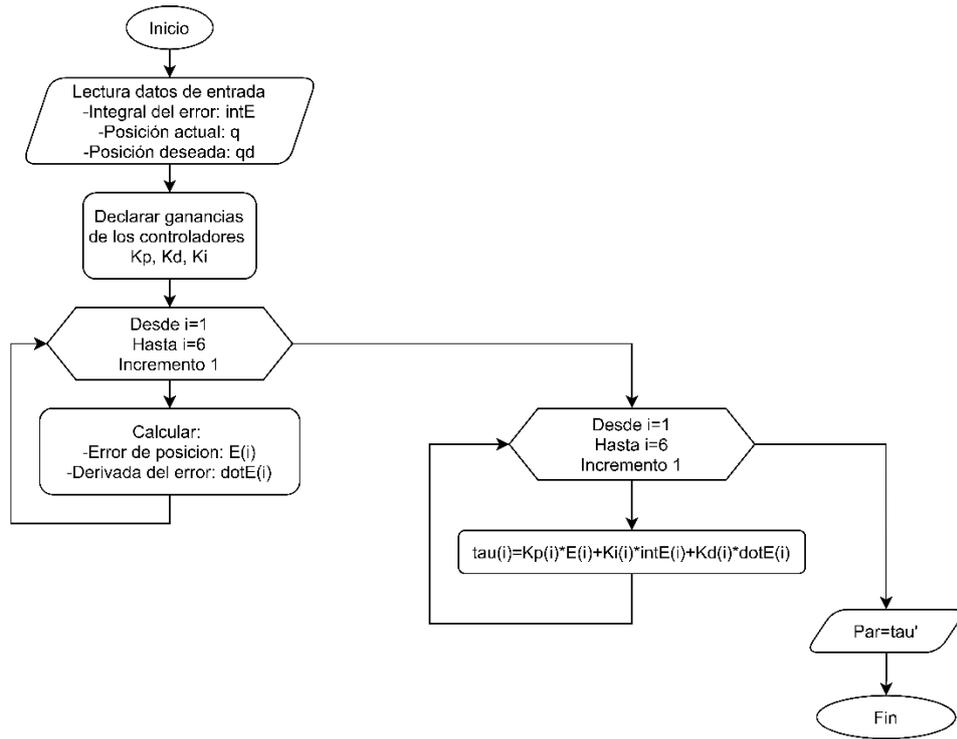


Figura 21-3. Diagrama de flujo para el control

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe el algoritmo del diagrama de flujo presentado:

- Se hace la lectura de la posición real, la posición deseada y de la integral del error.
- Se declaran las ganancias de los controladores (K_p , K_i , K_d).
- Se calcula el error de posición y su derivada.
- Se calcula el par con la ecuación (15-2).
- Se establece el vector par que representa los pares articulares.

La figura 22-3 muestra el bloque de control donde se encuentra el algoritmo para ejecutar el controlador PID. Este bloque tiene como entrada la posición articular deseada y la leída. Además, para la aplicación de la ecuación (15-2). Se usa un integrador para obtener la integral del error. A la salida del bloque de control se tiene el par del brazo robótico.

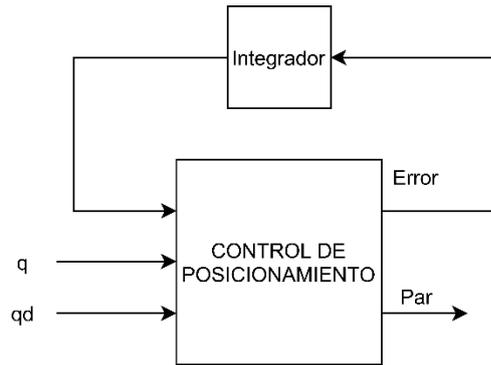


Figura 22-3. Bloque de control PID

Realizado por: Merchán, B. & Chimbo, A. 2020

El controlador se sintonizó por prueba y error basado en el siguiente proceso:

- Se lleva el valor de K_i y K_d a un valor mínimo o igual a 0, para comenzar a ajustar el K_p .
- Se estudia el comportamiento de la planta ante una excitación en la entrada para diferentes valores de K_p hasta lograr que la respuesta se adecúe al comportamiento esperado.
- Una vez obtenido los valores de K_p , se ajusta la acción derivativa modificando la ganancia K_d , de tal manera que la respuesta disminuya su error.
- Finalmente se ajusta la acción integral.

El bloque de control PID representa al controlador con la siguiente función de transferencia:

$$G_{PID} = K_p + \frac{K_i}{s} + K_d s \quad (33-3)$$

Articulación 1:

En la figura 23-3 se observa como varía el error de posicionamiento de la articulación 1, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la primera articulación es la siguiente:

$$G1_{PID} = 25.01 + \frac{1.09}{s} + 5.5001 s \quad (34-3)$$

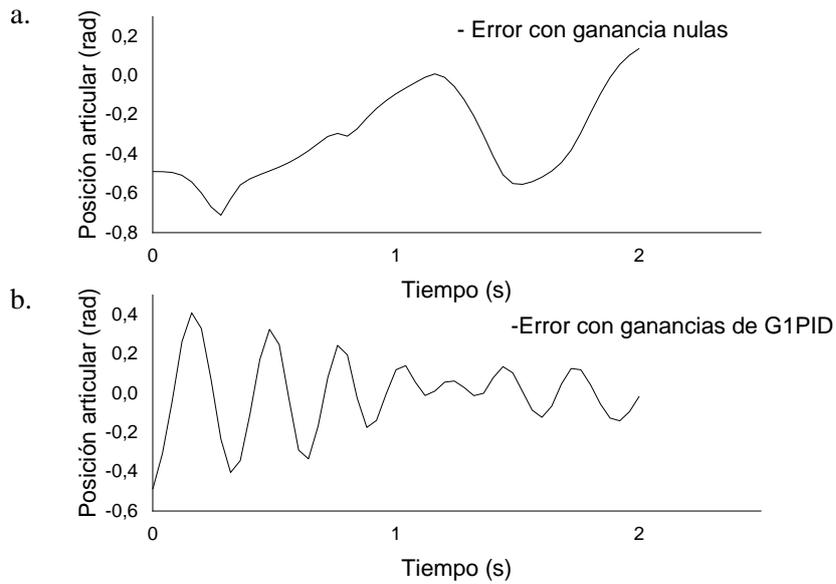


Figura 23-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G_{1PID}

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 2:

En la figura 24-3 se observa como varía el error de posicionamiento de la articulación 2, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la segunda articulación es la siguiente:

$$G_{2PID} = 1.0001 + \frac{1.1001}{s} + 3.9001 s \quad (35-3)$$

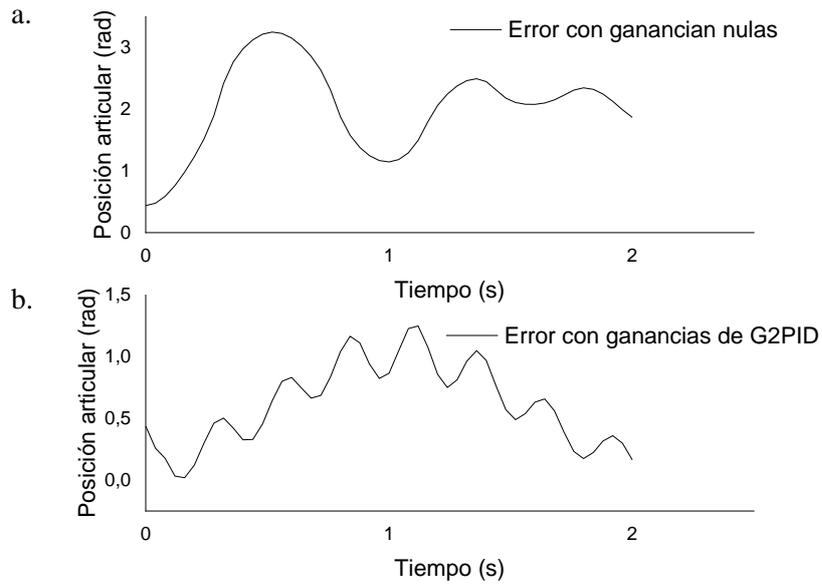


Figura 24-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G_{2PID}

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 3:

En la figura 25-3 se observa como varía el error de posicionamiento de la articulación 3, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la tercera articulación es la siguiente:

$$G_{3PID} = 49.4 + \frac{1.1997}{s} + 80.091 s \quad (36-3)$$

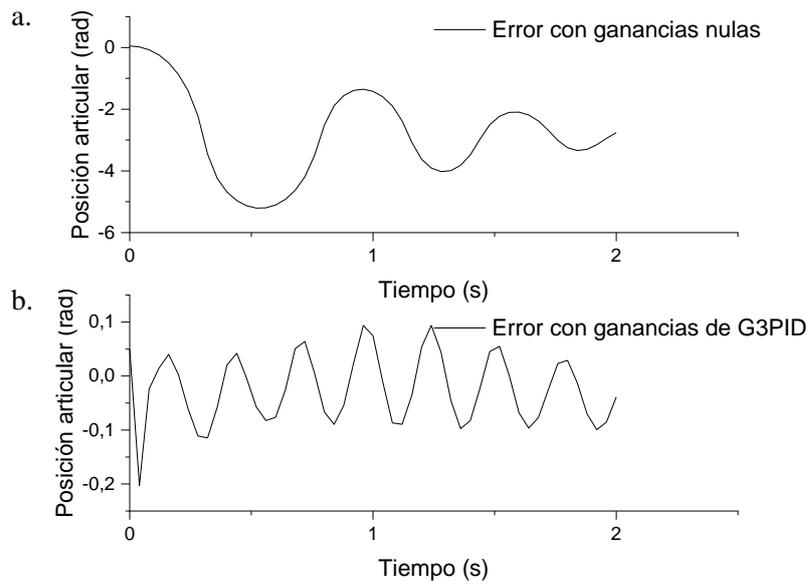


Figura 25-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G_{3PID}

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 4:

En la figura 26-3 se observa como varía el error de posicionamiento de la articulación 4, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la cuarta articulación es la siguiente:

$$G_{4PID} = 47.408 + \frac{0.101}{s} + 18.3710 s \quad (38-3)$$

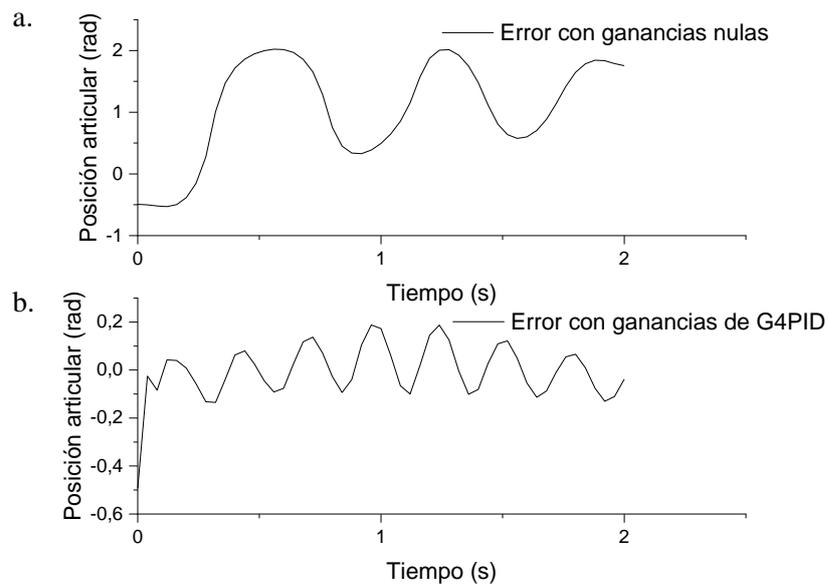


Figura 26-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G_{4PID}

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 5:

En la figura 27-3 se observa como varía el error de posicionamiento de la articulación 5, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la quinta articulación es la siguiente:

$$G_{5PID} = 2 + \frac{0.1}{s} + 8.5555 s \quad (39-3)$$

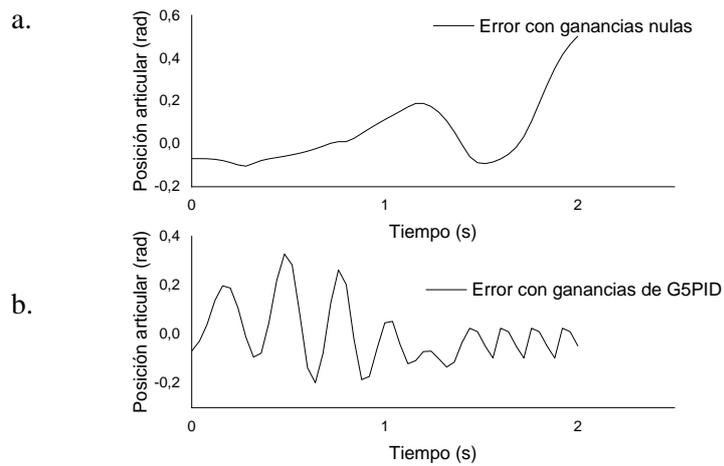


Figura 27-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. $G5_{PID}$

Realizado por: Merchán, B. & Chimbo, A. 2020

Articulación 6:

En la figura 28-3 se observa como varía el error de posicionamiento de la articulación 6, ante distintos valores de las constantes K_p , K_i , K_d .

La función de transferencia del controlador de la sexta articulación es la siguiente:

$$G6_{PID} = 1.3 + \frac{0.1001}{s} + 0.5s \quad (40-3)$$

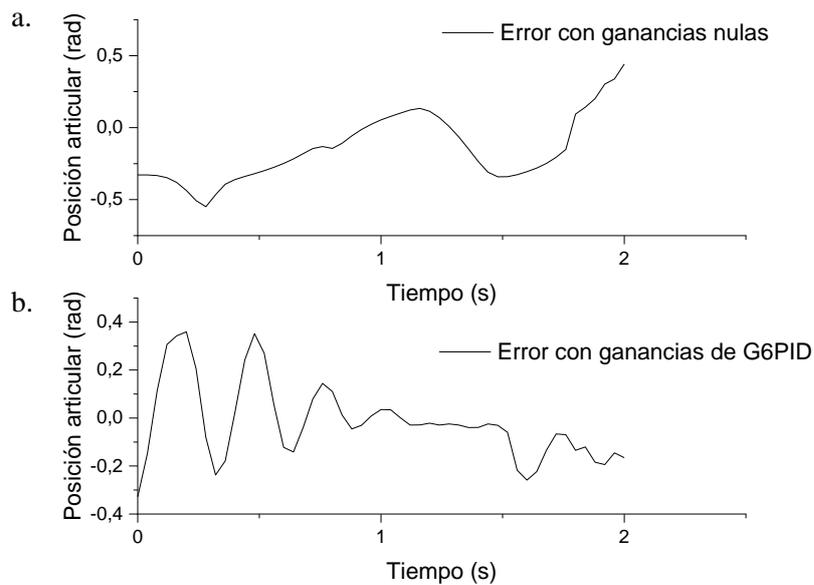


Figura 28-3: Señal de error de posicionamiento con diferentes ganancias (k_p , k_i y k_d) para el controlador PID: a. Nulas y b. G_{6PID}

Realizado por: Merchán, B. & Chimbo, A. 2020

En la tabla 2-3 se muestra las ganancias (k_p , k_i , k_d) para los 6 controladores PID del brazo robótico UR3.

Tabla 2-3: Resumen de ganancias (k_p , k_i , k_d)

	k_p	k_i	k_d
Articulación 1	25.01	1.09	55.001
Articulación 2	10.001	11.001	39.001
Articulación 3	49.4	11.997	80.091
Articulación 4	47.408	0.101	18.371
Articulación 5	2	0.1	9
Articulación 6	1.3	0.1001	0.5

Realizado por: Merchán, B. & Chimbo, A. 2020

3.8 Modelación de las partes plásticas de carrocerías

Para el correcto proceso de modelación de las partes plásticas de carrocerías se seleccionó los objetos que no exceden las dimensiones establecidas del área de trabajo del robot UR3. Por lo cual se eligió las siguientes partes para el proceso de pintado:

- Caja para espejo retrovisor
- Consola central
- Apoya brazos

En la figura 29-3 se aprecia las partes modeladas que cumplen con las dimensiones establecidas para el área de trabajo, las mismas que se diseñaron en el software SolidWorks 2017. El mencionado software permite guardar los archivos en diferentes formatos (igs, stl, iges, step, amf, etc), de entre los cuales se eligió el formato STL, debido que es un formato admitido por el software RoboDK. En la tabla 2-3 se detalla las características de las partes seleccionas para el proceso de pintado.

Tabla 3-3:Características de partes plásticas

Parte plástica	Dimensiones (mm)	Espesor (mm)	Área de la superficie (mm ²)
Consola central	400x250	3	235038.07
Apoya brazos	100x400	3	88908.02
Caja para espejo retrovisor	246x168	3	95388.82

Realizado por: Merchán, B. & Chimbo, A. 2020

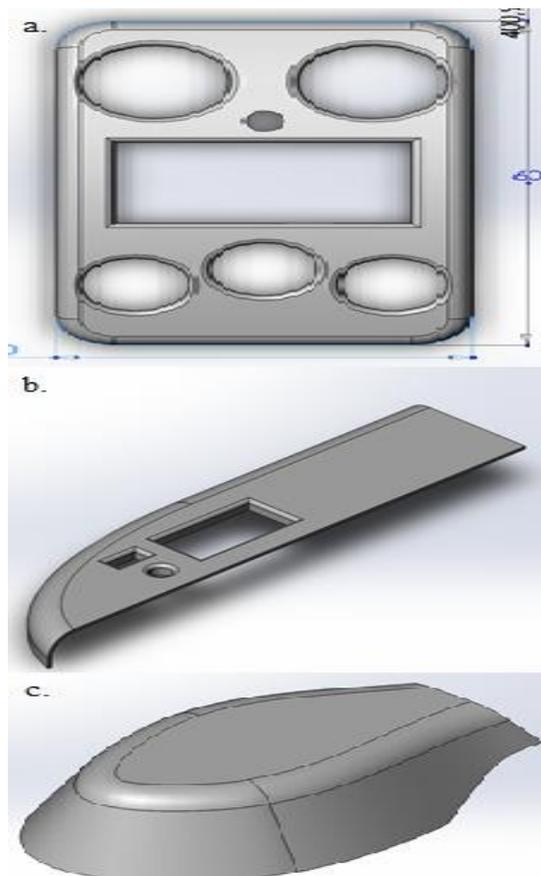


Figura 29-3: Partes plásticas modeladas: a. Consola central b. Apoya brazos
c. Caja para espejo retrovisor

Realizado por: Merchán, B. & Chimbo, A. 2020

3.9 Diseño Interfaz del Sistema.

En esta sección se describe las ventanas realizadas en Guide de Matlab con las cuales el usuario interactuará. La figura 30-3 presenta la interfaz de usuario realizada denominada TEST, la que contiene una lista desplegable para seleccionar la parte a pintar, un espacio para ingresar el número de capas a pintar (necesario para el modo automático). Existen tres botones de los cuales dos son destinados para seleccionar el modo de pintado y el tercer botón destinado para salir del programa.

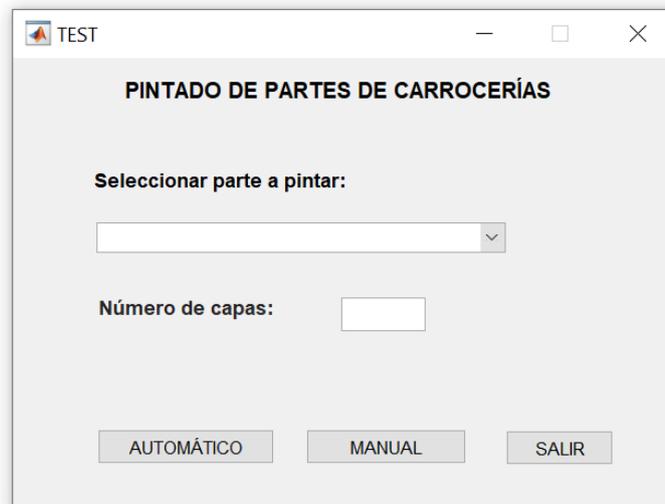


Figura 30-3: Interfaz de Usuario en guide de MatLab

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe cada acción que realiza los elementos de la interfaz:

- Lista desplegable: contiene las partes plásticas de la carrocería.
- Número de capas: se ingresa el número de capas en el cuadro de texto.
- Botón automático: ejecuta el software RoboDK para visualizar la simulación con las trayectorias preestablecidas para cada parte plástica.
- Botón Manual: ejecuta el Guide y el software RoboDK, con los que el usuario interactuará y establecerá los puntos que desea para realizar una trayectoria.

Seleccionado el modo manual se abrirá la ventana denominada PANEL presentada en la figura 31-3. En la que se solicita al usuario dar doble clic en el robot UR3 para desplegar el panel del robot en el software de RoboDK. Además, la ventana posee un botón OK el cual se debe presionar para continuar con la presentación de la siguiente ventana denominada MANUAL.

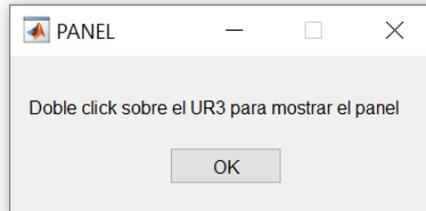


Figura 31-3: Ventana emergente para desplegar el panel del robot.

Realizado por: Merchán, B. & Chimbo, A. 2020

Se creó una ventana que consta de 4 botones como se observa en la figura 32-3, que contiene cuadros de texto para la presentación de los datos de POSICIÓN y ORIENTACIÓN, conjuntamente posee botones denominados actualizar, pintar, transportar y salir.

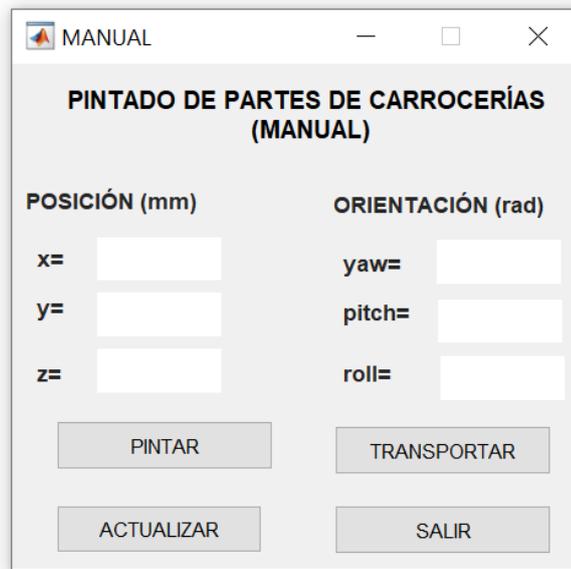


Figura 32-3: Interfaz de Usuario para pintado en modo manual

Realizado por: Merchán, B. & Chimbo, A. 2020

A continuación, se describe cada acción que realiza los elementos de la interfaz:

- Cuadros de texto: presenta los datos de POSICIÓN y ORIENTACIÓN al pulsar el botón actualizar.
- Botón actualizar: actualiza y guarda las n localizaciones del UR3 definidas por el usuario en RoboDK.
- Botón pintar: ejecuta la trayectoria que debe seguir el robot por medio de los puntos definidos por el usuario.
- Botón transportar: mueve los elementos que se encuentran en la banda transportadora hacia el horno.
- Botón salir: regresa al usuario a la ventana mostrada en la figura 30-3.

En la figura 33-3 se presenta el Panel de Robot el que se despliega al hacer doble clic en el robot. En este panel es posible mover los ejes del robot desde la sección Mover los ejes articulados e ingresar valores específicos de ejes de articulaciones en los cuadros de texto. (RoboDK, 2021)

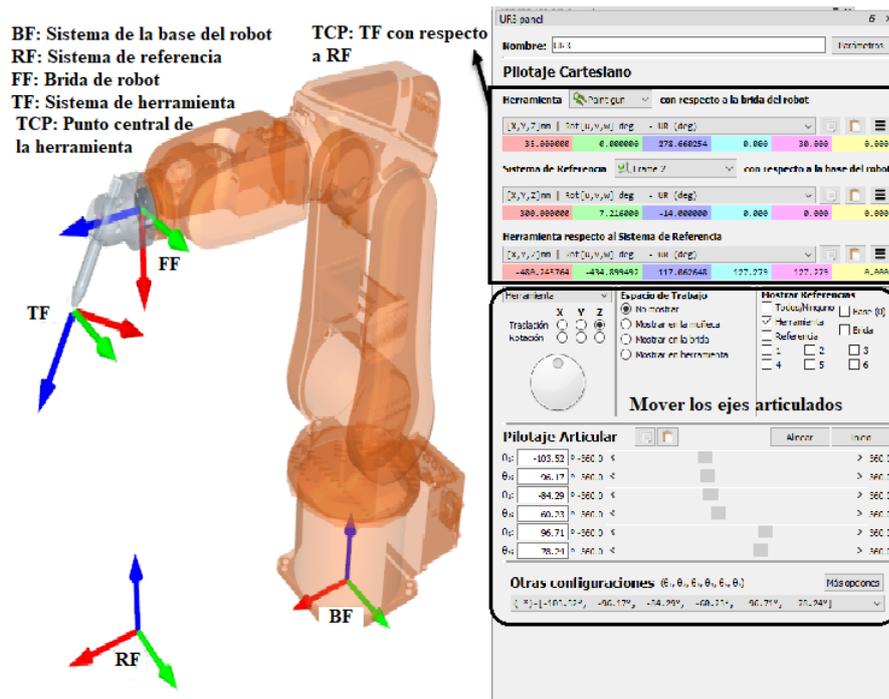


Figura 33-3: Interfaz del Panel de robot

Realizado por: Merchán, B. & Chimbo, A. 2020

En la figura 34-3 se aprecia el entorno desarrollado para la simulación del pintado de partes plásticas. El que está conformado por los siguientes elementos:

- Brazo Robótico UR3
- Cabina De Pintado
- Banda Transportadora
- Mesas
- Partes Plásticas
- Horno

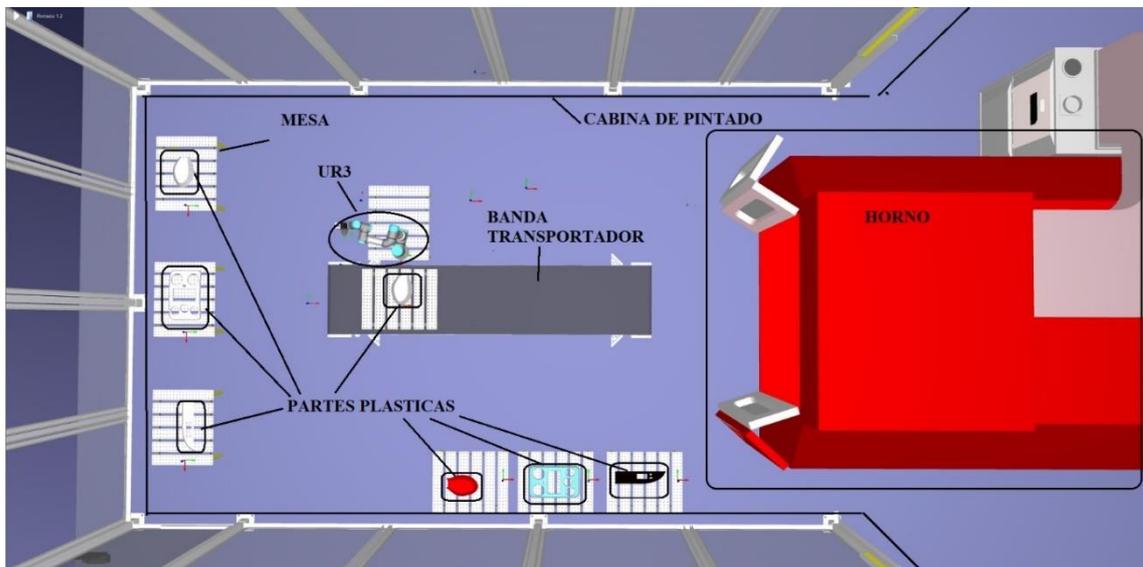


Figura 34-3: Interfaz realidad virtual para el pintado de partes plásticas

Realizado por: Merchán, B. & Chimbo, A. 2020

3.10 Determinación de puntos para la trayectoria en el pintado de partes plásticas

Para la determinación de los puntos se utiliza el método heurístico, este método se fundamenta en la experiencia del individuo, y en la de otros para encontrar la solución más viable al problema. Los puntos determinados en esta sección son utilizados específicamente en el modo automático. En la determinación se considera las recomendaciones que se encuentran en el capítulo II, sección 2.3 Proceso de pintado de partes plásticas de carrocerías, haciendo énfasis en lo descrito a continuación: “cada ráfaga tiene que montarse parcialmente en la anterior. El ancho de la ráfaga se lo divide mentalmente en 4 partes, cada ráfaga debe cubrir 3/4 partes de la ráfaga anterior”. Se determinó la cantidad de puntos con la que se pinta completamente la superficie de cada parte

como se muestra en figura 36-3 y la cantidad de puntos para cada parte plástica se resumen en la tabla 3-3.

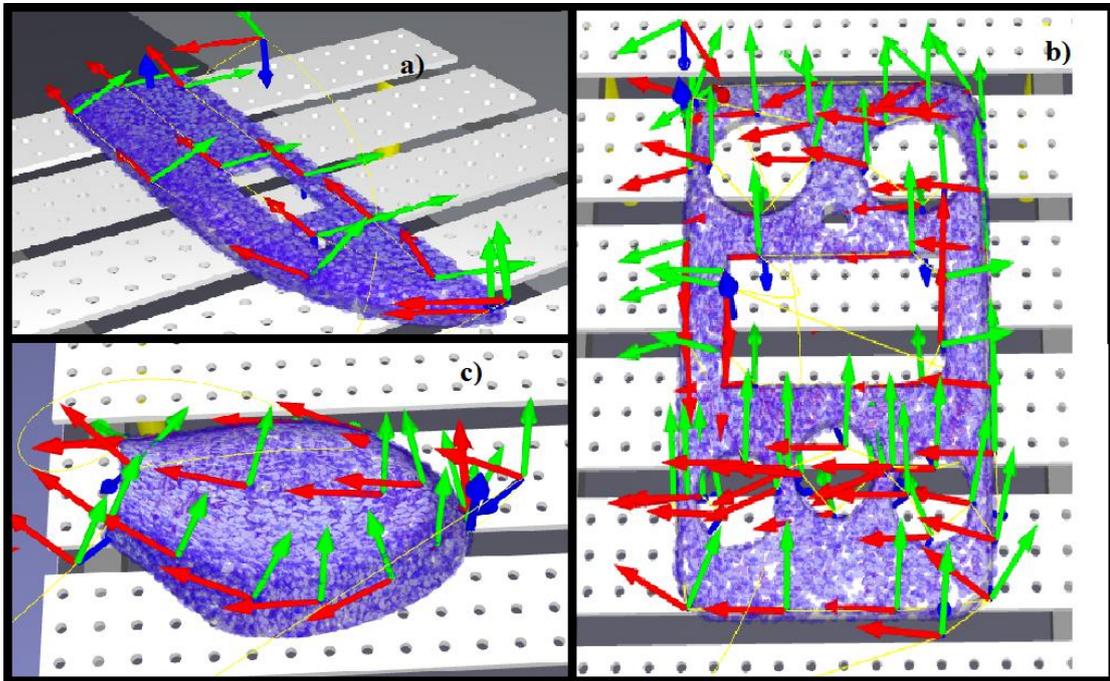


Figura 35-3: Puntos preestablecidos para el pintado de las partes plásticas a) Apoya brazos, b) consola central, c) caja de espejo retrovisor

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 4-3: Cantidad de puntos para pintado en modo automático

Parte plástica	Cantidad de puntos
Consola central	62
Apoya brazos	14
Caja para espejo retrovisor	30

Realizado por: Merchán, B. & Chimbo, A. 2020

3.11 Ejecución de la trayectoria en modo manual

Al seleccionar modo manual se deberá interactuar entre la interfaz de RoboDK y los paneles de guía de Matlab. Se abrirá el software RoboDK con el escenario de la parte a pintar seleccionada. A continuación, se muestra la ventana emergente que solicita dar doble click sobre el brazo

robótico UR3 para mostrar el UR3 panel. Posteriormente se abrirá un guide denominado MANUAL. En la sección Pilotaje Articular del UR3 panel se modifica los valores de los ángulos para alcanzar la localización (posición y orientación) deseada por el usuario. Una vez obtenido dicha localización se procede a guardar esta información con el botón ACTUALIZAR del guide MANUAL. Los valores de posición (x, y, z) y orientación (yaw, pitch, roll) se almacenan en un archivo de Excel. En el cual se tiene 3 hojas, una por cada parte diseñada (consola central, apoya brazos, caja para espejo retrovisor).

Para ejecutar la trayectoria con los puntos seleccionados por el usuario se debe dar click en el botón pintar. Con un script de Matlab se extrae los datos almacenados en el archivo de Excel para su procesamiento. Se ejecuta el modelo de cinemática inversa para obtener las posiciones articulares en función de las localizaciones seleccionadas por el usuario. Al igual que las localizaciones dichas posiciones articulares se almacenan en un archivo de excel. Con un script de MatLab se extrae en forma vectorial los datos almacenados. Se toma la primera posición articular como vector q_1 siendo este la posición inicial del brazo UR3 y la segunda posición articular en el vector q_2 como posición final la cual deberá alcanzar el brazo UR3. El vector q_2 es la posición de referencia que se ingresa para ejecutar el control de posicionamiento. La salida del bloque de control ingresa al bloque del modelo dinámico para obtener una nueva posición que se retroalimenta al bloque de control para ser comparada con la posición de referencia. Esta nueva posición ingresa también en el bloque de cinemática directa para obtener la localización y ser enviada por medio de la función ShowRoboDK() al brazo UR3 en el software RoboDK y así visualizar el movimiento. Esta acción se realizará hasta que el brazo UR3 alcance la posición deseada. Una vez establecido la posición articular deseada, q_2 será la nueva posición inicial y la posición final el siguiente dato de posición almacenado en el archivo Excel y así sucesivamente hasta llegar a la posición q_m , siendo m el número total de posiciones articulares almacenadas. A continuación, en la figura 36-3 se muestra el diagrama de secuencia de la ejecución de la trayectoria en modo manual.

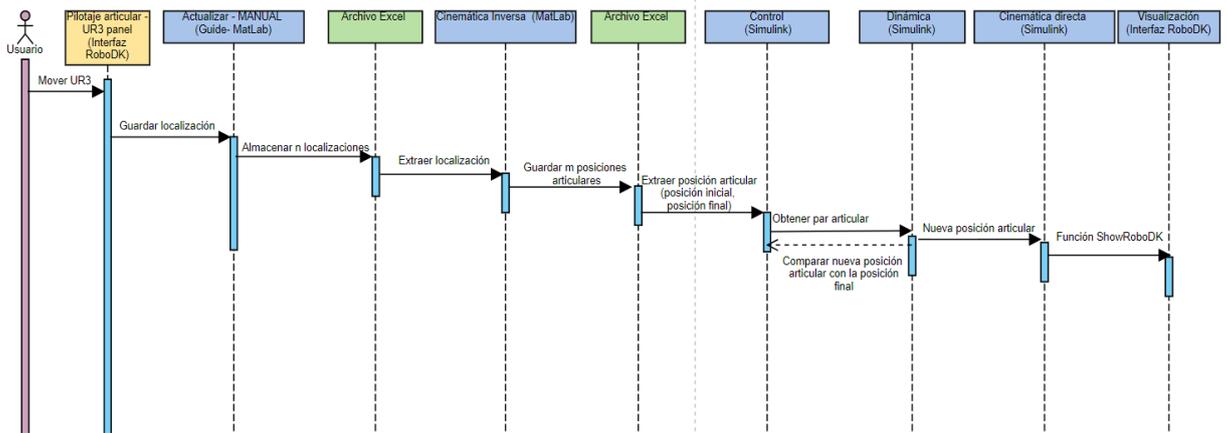


Figura 36-3. Diagrama de secuencia de la ejecución de la trayectoria en modo manual

Realizado por: Merchán, B. & Chimbo, A. 2020

3.12 Ejecución de la trayectoria en modo automático

Al seleccionar el modo automático de pintado a través del guide de Matlab se abre el escenario en RoboDK con la parte seleccionada. El brazo robótico inicia el proceso de pintado con las trayectorias predefinidas en función a los puntos de la tabla 3-3. Con los puntos predefinidos el robot logra pintar en su totalidad la superficie de la parte plástica seleccionada como se observa en la figura 35-3. Este proceso se ejecuta n veces, siendo n el número de capas indicadas por el usuario. A continuación, en la figura 37-3 se muestra el diagrama de secuencia de la ejecución de la trayectoria en modo automático.

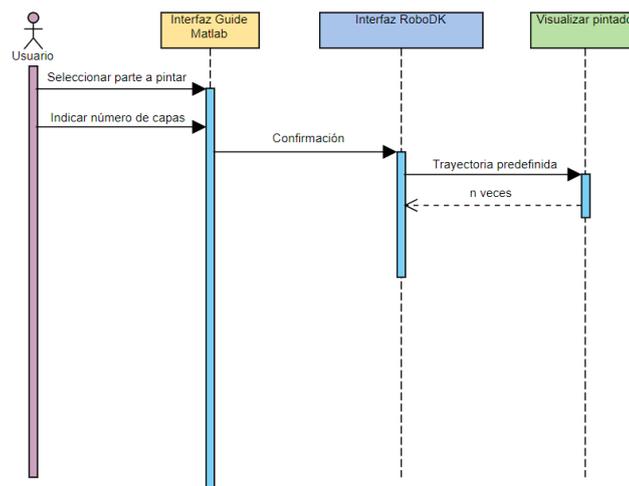


Figura 37-3. Diagrama de secuencia de la ejecución de la trayectoria en modo automático

Realizado por: Merchán, B. & Chimbo, A. 202

CAPÍTULO IV

4 PRUEBAS Y RESULTADOS

En este capítulo se analizan los resultados obtenidos mediante las pruebas realizadas para garantizar el cumplimiento de los requerimientos planteados. Se realizaron las siguientes pruebas: Validación de los modelos cinemáticos, prueba de error de posicionamiento, y prueba de inmersión y usabilidad.

4.1 Validación de los modelos cinemáticos

El objetivo de esta prueba es validar los modelos cinemáticos (directo e inverso) desarrollados en Matlab. Para representar los modelos se creó la función $q = \text{inversekinematic1}(xyzrpw)$ que resuelve la cinemática inversa, tiene como parámetro de entrada el vector de localización del efector final del brazo robótico y como salida el vector de los ángulos de las articulaciones necesarios para lograr dicha localización. La función $T = \text{forwardkinematic}(q)$ para resolver la cinemática directa, tiene como parámetro de entrada el vector de los ángulos de las articulaciones y como salida la localización del efector final.

Modelo cinemático inverso

La tabla 1-4 contiene los parámetros de entrada para la función creada para resolver la cinemática inversa. En esta tabla se tiene 6 columnas: tres correspondientes a la posición (x, y, z) y tres para la orientación (yaw, pitch, roll) del efector final del brazo robótico. Los 13 datos de posición y orientación se tomaron de la escena creada para el pintado del apoya brazo en modo automático.

Tabla 1-4: Parámetros de entrada para la función $\text{inversekinematic1}(xyzrpw)$

x	y	z	yaw	pitch	roll
133.957633	-297.70079	32.6064739	-2.2585429	0.38555346	-2.8274039
63.3747949	-214.33222	37.7130093	-2.9532959	0.49379276	-1.9735165
40.8760937	-53.917648	-58.220823	-2.8418729	-0.7785921	-1.4045313
41.9391147	93.8454987	11.9560819	1.60741805	0.85995647	-0.3641981
88.8803884	94.8189769	46.3307535	3.11016481	0.02486453	-1.6075089
84.8704733	-65.430463	30.1326554	-3.1297585	-0.1529921	-1.6039495
87.4309648	-180.27674	46.3607198	3.11524892	-0.0188561	-1.5459323

119.74199	-259.64478	46.1533874	3.10964363	0.02575986	-1.2643148
129.381867	-181.34009	46.851749	-3.1064686	0.02906711	-1.5224411
124.937941	-104.96836	45.3515827	-3.1064686	0.02906706	-1.5224411
119.953792	72.4028333	44.662004	-2.8020129	0.03566558	-1.554249
154.650771	-9.012029	144.425029	-3.1064686	0.02906709	-2.9137502
154.65077	-9.0120291	144.425026	-3.1064686	0.02906708	-2.9137502

Realizado por: Merchán, B. & Chimbo, A. 2020

La tabla 2-4 contiene los datos de salida de la función creada para resolver la cinemática inversa. En esta tabla se tiene 6 columnas correspondientes a las posiciones articulares (q1, q2, q3, q4, q5, q6) del brazo robótico UR3 para alcanzar las posiciones y orientaciones de la tabla 1-4.

Tabla 2-4: Datos de salida al aplicar la función `inversekinematic1(xyzrpw)`

q1	q2	q3	q4	q5	q6
1.54489873	-0.7647005	1.71507776	-1.6316679	-1.2053069	0.29658742
1.16574625	-0.9641162	2.62264272	-2.8565428	-1.1910715	0.73816643
1.33364922	0.78175629	2.46398126	1.52439923	-2.3920535	1.40133665
3.9910861	0.43599484	2.37001641	1.48209072	-3.1153761	-2.3316526
2.9469453	1.86170759	3.09189439	-0.2716964	-1.5968102	2.9094454
1.27856835	1.45296319	2.88285658	0.34348906	-1.7206611	1.24202448
1.43451227	-0.7127421	2.799836	2.59663439	-1.585899	1.45940824
1.59125407	-0.8680175	2.36091034	-3.0961583	-1.5456953	1.8977316
1.6483378	-0.8779431	2.67304258	2.95005701	-1.5390962	1.69668413
1.66095565	-0.0604306	3.02912755	1.77605707	-1.5386854	1.70930172
2.39978017	2.70965425	3.14022227	-0.9297526	-1.2979174	2.39382117
2.23714377	-2.349588	2.82844655	-2.0400164	-1.5262384	0.89448569
2.23714377	-2.349588	2.82844657	-2.0400164	-1.5262384	0.89448569

Realizado por: Merchán, B. & Chimbo, A. 2020

Modelo cinemático directo

Para resolver la cinemática directa se usa los datos de la tabla 2-4 como parámetros de entrada de la función creada para resolver la cinemática directa. La tabla 3-4 contiene los datos de salida que se obtienen al aplicar la función creada para resolver la cinemática directa. En esta tabla se

tiene 6 columnas: tres correspondientes a la posición (x, y, z) y tres para la orientación (yaw, pitch, roll) del efector final del brazo robótico.

Tabla 3-4: Datos de salida al aplicar la función forwardkinematic(q)

x	y	z	yaw	pitch	roll
133.957633	-297.70079	32.6064739	-2.2585429	0.38555346	-2.8274039
63.3747949	-214.33222	37.7130093	-2.9532959	0.49379276	-1.9735165
40.876092	-53.917648	-58.220821	-2.8418729	-0.7785921	-1.4045313
41.9391149	93.8454989	11.9560815	1.60741806	0.85995647	-0.3641981
88.8803863	94.8189773	46.3307521	3.11016481	0.02486451	-1.6075089
84.8704719	-65.430462	30.1326553	-3.1297585	-0.1529921	-1.6039495
87.430963	-180.27674	46.3607198	3.11524892	-0.0188561	-1.5459323
119.74199	-259.64478	46.1533874	3.10964363	0.02575986	-1.2643148
129.381868	-181.34009	46.8517498	-3.1064686	0.02906711	-1.5224411
124.937941	-104.96836	45.3515802	-3.1064686	0.02906705	-1.5224411
119.95379	72.4028366	44.6620003	-2.8020129	0.03566556	-1.554249
154.650771	-9.012029	144.425029	-3.1064686	0.02906709	-2.9137502
154.650769	-9.0120262	144.425024	-3.1064686	0.02906707	-2.9137502

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 4-4: Error absoluto calculado

Nº	x	y	z	yaw	pitch	roll
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	1.7E-06	0	2E-06	0	0	0
4	2E-07	2E-07	4E-07	1E-08	0	0
5	2.1E-06	4E-07	1.4E-06	0	2E-08	0
6	1.4E-06	1E-06	1E-07	0	0	0
7	1.8E-06	0	0	0	0	0
8	0	0	0	0	0	0
9	1E-06	0	8E-07	0	0	0
10	0	0	2.5E-06	0	1E-08	0

11	2E-06	3.3E-06	3.7E-06	0	2E-08	0
12	0	0	0	0	0	0
13	1E-06	2.9E-06	2E-06	0	1E-08	0
MAX	2.1E-06	3.3E-06	3.7E-06	1E-08	2E-08	0
MIN	0	0	0	0	0	0

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 5-4: Error relativo calculado

Nº	x	y	z	yaw	pitch	roll
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	4.15891E-08	0	3.4352E-08	0	0	0
4	4.76882E-09	2.1312E-09	3.3456E-08	6.2212E-09	0	0
5	2.36273E-08	4.2186E-09	3.0218E-08	0	8.0436E-07	0
6	1.64957E-08	1.5283E-08	3.3187E-09	0	0	0
7	2.05877E-08	0	0	0	0	0
8	0	0	0	0	0	0
9	7.72906E-09	0	1.7075E-08	0	0	0
10	0	0	5.5125E-08	0	3.4403E-07	0
11	1.66731E-08	4.5578E-08	8.2844E-08	0	5.6076E-07	0
12	0	0	0	0	0	0
13	6.46618E-09	3.2179E-07	1.3848E-08	0	3.4403E-07	0
MAX	4.15891E-08	3.2179E-07	8.2844E-08	6.2212E-09	8.0436E-07	0
MIN	0	0	0	0	0	0

Realizado por: Merchán, B. & Chimbo, A. 2020

Al usar los datos obtenidos por las funciones como entrada de la otra, se puede comprobar cómo coinciden los valores. Si se introducen un vector de localización [x y z yaw pitch roll] en la función de cálculo para el modelo cinemático inverso se obtiene un vector de coordenadas articulares [q1, q2, q3, q4, q5, q6]. Cuando se introduzca dicho vector de coordenadas articulares en la función de cálculo del modelo cinemático directo, el resultado será el vector de localización [x y z yaw pitch roll] iguales al inicial, como se puede observar en los ejemplos de las tablas 1-4 y 3-4. Se calculó los errores relativos de cada posición y se presentan en la tabla 5-4. Dando

como error máximo relativo 4.15891E-08 para x, 3.2179E-07 para y, 8.2844E-08 para z, 6.2212E-09 para yaw, 8.0436E-07 para pitch y 0 para roll. Y un valor de error mínimo de 0 para las seis variables. Lo cual demuestra la validez de las ecuaciones obtenidos para el modelo cinemático directo e inverso.

Para validar que el modelo cinemático programado y usado en esta simulación sea similar al modelo del UR3 se lo compara con el modelo que proporciona el software RoboDK. Para ello se usan los datos de la tabla 2-4, como parámetros de entrada para los dos modelos. La tabla 3-4 refleja los resultados del uso del modelo programado en Matlab y la tabla 6-4 los obtenidos con el modelo de RoboDK. En la tabla 7-4 se tiene los errores absolutos al comparar los datos entre las tablas 3-4 y 6-4.

Tabla 6-4: Parámetros de salida al aplicar el modelo de RoboDK

x	y	z	yaw	pitch	roll
132.231888	-295.919327	34.030467	-2.258543	0.385553	-2.827404
62.526586	-211.832395	38.239761	-2.953296	0.493793	-1.973517
39.028466	-51.703860	-58.474348	-2.841873	-0.778592	-1.404531
42.764718	91.074533	11.223708	1.607418	0.859956	-0.364198
86.544768	93.278641	45.528958	3.110165	0.024865	-1.607509
83.461922	-63.084574	29.341553	-3.129758	-0.152992	-1.603949
85.737672	-178.216122	46.022515	3.115249	-0.018856	-1.545932
117.765010	-257.732620	45.839746	3.109644	0.025760	-1.264315
127.305593	-179.585936	46.659892	-3.106469	0.029067	-1.522441
122.857085	-103.430588	44.958209	-3.106469	0.029067	-1.522441
117.080152	72.574089	44.488060	-2.802013	0.035666	-1.554249
151.790013	-8.492599	144.059535	-3.106469	0.029067	-2.913750
151.790011	-8.492596	144.059529	-3.106469	0.029067	-2.913750

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 7-4: Error absoluto de la comparación de los modelos cinemáticos

E_x	E_y	E_z	E_yaw	E_pitch	E_roll
1.725745	1.781463	1.4239931	1E-07	4.6E-07	1E-07
0.8482089	2.499825	0.5267517	1E-07	2.4E-07	5E-07

1.847626	2.213788	0.253527	1E-07	1E-07	3E-07
0.8256031	2.7709659	0.7323735	6E-08	4.7E-07	1E-07
2.3356183	1.5403363	0.8017941	1.9E-07	4.9E-07	1E-07
1.4085499	2.345888	0.7911023	5E-07	1E-07	5E-07
1.693291	2.060618	0.3382048	8E-08	1E-07	3E-07
1.97698	1.91216	0.3136414	3.7E-07	1.4E-07	2E-07
2.076275	1.754154	0.1918578	4E-07	1.1E-07	1E-07
2.080856	1.537772	0.3933712	4E-07	5E-08	1E-07
2.873638	0.1712524	0.1739403	1E-07	4.4E-07	0
2.860758	0.51943	0.365494	4E-07	9E-08	2E-07
2.860758	0.5194302	0.365495	4E-07	7E-08	2E-07

Realizado por: Merchán, B. & Chimbo, A. 2020

Para evaluar el modelo cinemático directo se usa el coeficiente de determinación (R cuadrado). El resultado del coeficiente de determinación oscila entre 0 y 1. Cuanto más cerca de 1 se encuentre su valor, mayor será el ajuste del modelo. De forma inversa, cuanto más cerca de cero, menos ajustado estará el modelo y, por tanto, menos fiable será. Este coeficiente se lo calcula con la ecuación (1-4) (Economiopedia, 2019):

$$R^2 = 1 - \left(\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \right) \quad (1-4)$$

donde:

\hat{y}_i : valor medido

y_i : valor requerido

\bar{y} : promedio del valor requerido

n : número de muestras

Al aplicarse la ecuación (1-4) se obtienen los siguientes valores para el coeficiente de determinación:

R_x=0.99680509

R_y=0.99979077

R_z=0.99984673

R_yaw=1
 R_pitch=1
 R_roll=1

De los datos numéricos obtenidos, se puede apreciar que, para x, y, z los valores son cercanos a 1 y los valores para yaw, pitch, roll son 1. Esto demuestra la validez de las ecuaciones obtenidas y programadas para el modelo cinemático directo.

4.2 Prueba de error de posicionamiento

El objetivo de esta prueba es conocer el error de posicionamiento que presenta el brazo robótico usando un control PID. El bloque que representa el control usado en la simulación tiene la función CONTROL, en la que se ha elaborado un programa que contiene el controlador PID. Esta prueba consiste en ingresar los ángulos de la tabla 2-4 que son el resultado de la validación del modelo cinemático directo, como referencia. El estado real del sistema se muestra en la tabla 8-4. En este bloque se calcula el error, que indica al controlador la diferencia que existe entre la referencia y el estado real del sistema, el error de los datos seleccionados para esta prueba se muestra en la tabla 9-4.

Tabla 8-4: Estado real del sistema

q1	q2	q3	q4	q5	q6
1.54516562	-0.6869607	1.71422479	-1.7476774	-1.209005	0.31060075
1.15873695	-0.8558118	2.62079417	-2.8860952	-1.1563699	0.70262006
1.30106531	0.82982544	2.34974773	1.51599555	-2.3079944	1.32613478
3.97708015	0.41381777	2.26232496	1.48780378	-3.0804585	-2.3562327
2.95789329	1.81746989	3.10995003	-0.2657847	-1.6157572	2.93007663
1.24902609	1.43294992	2.8080371	0.33450497	-1.6505339	1.13189223
1.43285855	-0.6883371	2.77662428	2.59531432	-1.5798004	1.45897825
1.58747293	-0.7641685	2.35690975	-3.1787823	-1.5220298	1.89333104
1.64361982	-0.7644994	2.66454972	2.93742357	-1.5166602	1.68804078
1.65587196	-0.0700078	2.96857674	1.72E+00	-1.5246255	1.70E+00
2.40929833	2.58078174	3.18658316	-0.8995383	-1.2912227	2.3656917
2.19697136	-2.4156423	2.90299904	-1.9833136	-1.4365012	0.84871986
2.19697136	-2.4156423	2.90299906	-1.9833136	-1.4365012	0.84871986

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 9-4: Error absoluto calculado

N°	q1	q2	q3	q4	q5	q6
1	0.0002669	0.0777398	0.0008530	0.1160095	0.0036981	0.0140133
2	0.0070093	0.1083044	0.0018486	0.0295524	0.0347016	0.0355464
3	0.0325839	0.0480692	0.1142335	0.0084037	0.0840591	0.0752019
4	0.0140060	0.0221771	0.1076915	0.0057131	0.0349176	0.0245801
5	0.0109480	0.0442377	0.0180556	0.0059117	0.0189470	0.0206312
6	0.0295423	0.0200133	0.0748195	0.0089841	0.0701272	0.1101323
7	0.0016537	0.0244050	0.0232117	0.0013201	0.0060986	0.0004300
8	0.0037811	0.1038490	0.0040006	0.0826240	0.0236655	0.0044006
9	0.0047180	0.1134437	0.0084929	0.0126334	0.0224360	0.0086434
10	0.0050837	0.0095772	0.0605508	0.0560571	0.0140599	0.0093017
11	0.0095182	0.1288725	0.0463609	0.0302143	0.0066947	0.0281295
12	0.0401724	0.0660543	0.0745525	0.0567028	0.0897372	0.0457658
13	0.0401724	0.0660543	0.0745525	0.0567028	0.0897372	0.0457658
MAX	0.0401724	0.1288725	0.1142335	0.1160095	0.0897372	0.1101323
MIN	0.0002669	0.0095772	0.0008530	0.0013201	0.0036981	0.0004300

Realizado por: Merchán, B. & Chimbo, A. 2020

Tabla 10-4: Error relativo calculado

N°	q1	q2	q3	q4	q5	q6
1	0.0001728	0.1016605	0.0004973	0.0710987	0.0030682	0.0472486
2	0.0060127	0.1123354	0.0007048	0.0103455	0.0291348	0.0481550
3	0.0244321	0.0614887	0.0463614	0.0055128	0.0351410	0.0536644
4	0.0035093	0.0508654	0.0454391	0.0038547	0.0112081	0.0105419
5	0.0037150	0.0237619	0.0058397	0.0217583	0.0118655	0.0070911
6	0.0231057	0.0137741	0.0259532	0.0261554	0.0407560	0.0886716
7	0.0011528	0.0342410	0.0082904	0.0005084	0.0038455	0.0002946
8	0.0023762	0.1196393	0.0016945	0.0266860	0.0153106	0.0023189
9	0.0028623	0.1292153	0.0031772	0.0042824	0.0145774	0.0050943
10	0.0030607	0.1584821	0.0199895	0.0315627	0.0091376	0.0054418

11	0.0039663	0.0475605	0.0147636	0.0324971	0.0051580	0.0117509
12	0.0179570	0.0281131	0.0263581	0.0277953	0.0587963	0.0511644
13	0.0179570	0.0281131	0.0263581	0.0277953	0.0587963	0.0511644
MAX	0.0244321	0.1584821	0.0463614	0.0710987	0.0587963	0.0886716
MIN	0.0001728	0.0137741	0.0004973	0.0005084	0.0030682	0.0002946

Realizado por: Merchán, B. & Chimbo, A. 2020

Se calcula los errores relativos máximos y mínimos de cada ángulo ya que existe un controlador por cada articulación. Lo ideal es tener un error de 0, con la aplicación del control se observa de la tabla 8-4 para la articulación 1 un máximo de 0.0244321 y un mínimo de 0.0001728, para la articulación 2 un máximo de 0.1584821 y un mínimo de 0.0137741, para la articulación 3 un máximo de 0.0463614 y un mínimo de 0.0004973, para la articulación 4 un máximo de 0.0710987 y un mínimo de 0.0005084, para la articulación 5 un máximo de 0.0587963 y un mínimo de -0.0030682 y para la articulación 6 un máximo de 0.0886716 y un mínimo de 0.0002946. Estos valores máximos y mínimos no se encuentran muy lejanos al valor ideal por lo tanto se concluye que el PID controla el sistema para alcanzar el estado de salida deseado.

4.3 Prueba de inmersión y usabilidad

Esta prueba tiene como fin determinar si la inmersión de los usuarios en el ambiente simulado es aceptable y que se asemeje a la realidad. Además, se determinará la usabilidad de la simulación. Se seleccionó un grupo de 15 personas con intereses en el área de estudio. Sus características se presentan en la tabla 11-4. De la cual se aprecia que de los 15 encuestados, 9 son de sexo masculino y 6 de sexo femenino. Además, se menciona el nivel de instrucción del total, 8 tienen hasta un nivel secundario y 7 tienen un nivel de instrucción superior. Esta prueba se realizó en base a la observación directa del usuario, interacción verbal con el usuario y una encuesta en línea.

Tabla 11-4: Características del grupo de personas encuestadas

N	Masculino	Femenino	Nivel de instrucción	
			Secundario	Medio Superior
1	X		X	
2		X		X
3		X		X
4	X		X	

5	X		X	
6	X			X
7	X			X
8		X	X	
9	X		X	
10		X	X	
11		X		X
12		X		X
13	X		X	
14	X			X
15	X		X	
Total	9	6	8	7

Realizado por: Merchán, B. & Chimbo, A. 2020

La encuesta está conformada por ocho preguntas dicotómicas presentadas a continuación:

1. ¿Los escenarios donde se desarrolla la simulación son realistas?

- Si
- No

2. ¿La organización de la información mostrada en el escenario de la simulación es adecuada?

- Si
- No

3. ¿Conoce modelos de realidad virtual que se han utilizados en el proceso de pintado implementando un brazo robótico?

- Si
- No

4. ¿Cree que estar en un espacio virtual es favorable para adquirir conocimientos sobre el proceso simulado?

- Si
- No

5. ¿En general, la experiencia de inmersión con simulación ha sido satisfactoria?

- Si
- No

6. ¿Los colores implementados en el escenario de la simulación son adecuados?

- Si
 - No
7. ¿La cantidad de ventadas o pestañas es adecuada?
- Si
 - No
8. ¿El uso de la simulación es fácil e intuitiva?
- Si
 - No

Aplicada la encuesta a los usuarios se obtuvieron los datos presentados en el gráfico 1-4 en la cual se aprecia el porcentaje que se obtuvo en las opciones. De la encuesta aplicada a los usuarios las preguntas 1,3,5,6 se enfocaron para inmersión en el entorno desarrollado y las preguntas 2,4,7,8 se enfocaron para usabilidad, se realizó el promedio de las 4 preguntas dando como resultado los datos presentados en el gráfico 2-4.

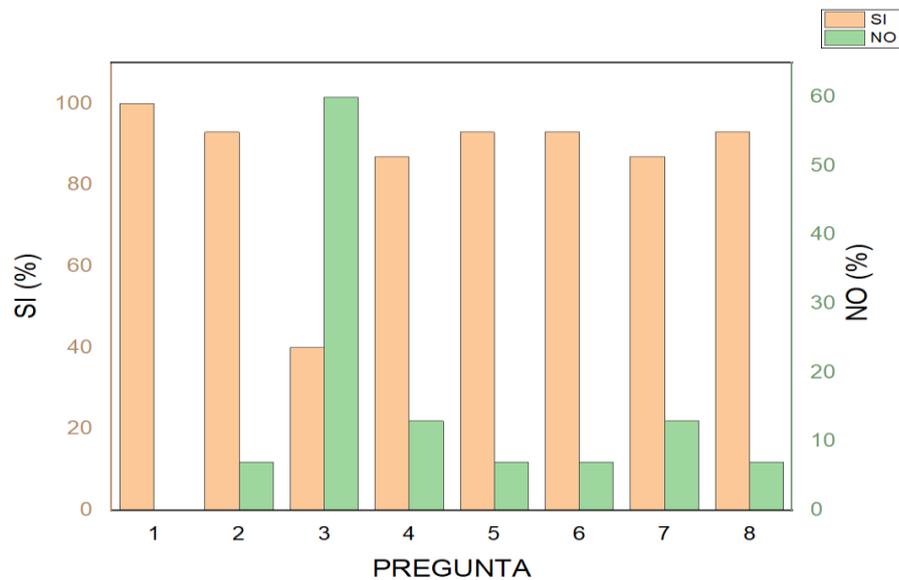


Gráfico 1-4. Resumen de los datos obtenidos al realizar la encuesta

Realizado por: Merchán, B. & Chimbo, A. 2020

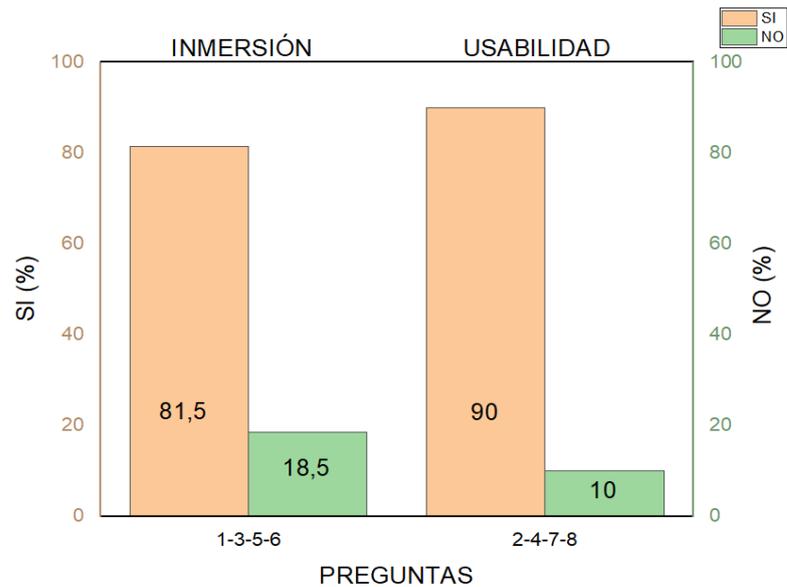


Gráfico 2-4. Porcentaje de Inmersión y Usabilidad

Realizado por: Merchán, B. & Chimbo, A. 2020

Se empleó el coeficiente de Alfa de Cronbach como indicador de confiabilidad para validar la encuesta de la prueba de inmersión y usabilidad de los usuarios en el uso de la simulación. Este indicador se lo calcula con la ecuación (2-4) (Ledesma, et al.):

$$\alpha = \frac{K}{K - 1} \left(\frac{\sum_{i=1}^K \sigma^2 Y_i}{\sigma^2 X} \right) \quad (2-4)$$

Donde:

K = Número de ítems en la escala.

$\sigma^2 Y_i$ = Varianza del ítem i.

$\sigma^2 X$ = Varianza total

El valor mínimo aceptable para el coeficiente alfa de Cronbach es 0.7; por debajo de ese valor la consistencia interna de la escala utilizada es baja (Oviedo, et al., s.f.) .La tabla 12-4 contiene los datos para el análisis de confiabilidad, los cuales están dispuestos en una matriz, las filas representan las personas encuestadas y las columnas las preguntas usadas en la encuesta. Para las respuestas

se encuentra una representación de 1 cuando la respuesta es “si” y 2 cuando sea “no”. Además, se tiene los valores de las varianzas de cada pregunta y la varianza total, necesarios para el cálculo del coeficiente de Cronbach.

Tabla 12-4: Datos para el análisis de confiabilidad

	Preguntas								
N	1	2	3	4	5	6	7	8	Suma
1	1	1	1	1	1	1	1	1	8
2	1	1	2	1	1	2	2	2	12
3	1	1	1	1	1	1	1	1	8
4	1	1	1	1	1	1	1	1	8
5	1	1	1	1	1	1	1	1	8
6	1	1	2	1	1	1	1	1	9
7	1	1	1	1	1	1	1	1	8
8	1	1	2	1	1	1	1	1	9
9	1	1	2	1	1	1	1	1	9
10	1	2	2	2	2	1	2	1	13
11	1	1	2	2	1	1	1	1	10
12	1	1	2	1	1	1	1	1	9
13	1	1	1	1	1	1	1	1	8
14	1	1	2	1	1	1	1	1	9
15	1	1	2	1	1	1	1	1	9
Varianza	0	0.0622	0.2400	0.1156	0.0622	0.0622	0.1156	0.0622	
Suma de varianzas									0.72
Varianza total									2.1156

Realizado por: Merchán, B. & Chimbo, A. 2020

Al aplicar la ecuación (2-4), se obtiene un valor del alfa de Cronbach de 0.75, siendo este un valor aceptable para validar la confiabilidad del cuestionario usado en la prueba. De la prueba realizada se obtuvo un valor de aceptabilidad en la inmersión del 81.5% y de rechazo de un 18.5%, de esta manera se muestra que los usuarios encuestados experimentan una verdadera inmersión en los

entornos de realidad virtual desarrollados. Además, se obtuvo un valor de aceptabilidad en la usabilidad del 90% y un rechazo del 10%, de esta manera se muestra que los usuarios encuestados determinan que la simulación es usable. Concluyendo que la simulación es inmersiva y fácil de usar.

CONCLUSIONES

Se modeló, controló y simuló un proceso de pintado de carrocerías usando un brazo robótico colaborativo UR3 y tecnologías inmersivas. Se tomó tres partes plásticas que conforman la carrocería de un auto: Caja para espejo retrovisor, Consola central y Apoya brazos, de las cuales se simula el proceso de pintado con el uso de un brazo robótico colaborativo UR3 y puede ser visualizado mediante gafas de RV.

El modelo cinemático directo se calculó a través de matrices de transformación homogénea, mientras que el modelo cinemático inverso por método geométrico, debido a la configuración antropomórfica del brazo robótico y al no poseer una articulación esférica no es factible la aplicación del método de desacoplamiento cinemático. Se calculó el coeficiente de determinación (R cuadrado) demostrando una muy buena complementariedad y correspondencia entre los modelos cinemáticos directo e inverso obtenidos.

Para el modelo dinámico inverso se aplicó el algoritmo computacional de Newton Euler debido a la eficiencia en los tiempos de cálculo, mientras que para el modelo dinámico directo se aplicó el tercer método de Walker y Orín que está basado en la formulación de Newton Euler.

Se realizó un control de posicionamiento usando un controlador PID. Este controlador se validó con la prueba de error de posicionamiento. Se calcularon los errores relativos máximos y mínimos para cada ángulo ya que existe un controlador por cada articulación. Los valores de error relativo máximo de posicionamiento articular obtenidos para cada articulación son: 0.0244321 para la articulación 1, 0.158482 para la articulación 2, 0.0463614 para la articulación 3, 0.0710987 para la articulación 4, 0.0587963 para la articulación 5 y 0.0886716 para la articulación 6. Concluyendo que el PID controla el sistema para el estado de salida deseado.

Se implementó una simulación virtual del ambiente industrial la cual se validó con una encuesta a los usuarios que la probaron. Se obtuvo un 81.5% de aceptabilidad en la inmersión, indicando que la simulación es inmersiva para los usuarios. Y para la usabilidad se obtuvo un 90% de aceptabilidad, indicando que la simulación es intuitiva, fácil de usar y la cantidad de ventanas es adecuada para el usuario.

RECOMENDACIONES

Se recomienda realizar una prueba conjuntamente con el modelo real del brazo colaborativo UR3, para corroborar el modelo dinámico usado. Debido a la situación sanitaria que se produjo a nivel mundial (COVID-19), no se tuvo acceso al robot colaborativo UR3 y no se realizó esta prueba.

Con el fin de mejorar la experiencia de usuario, se recomienda agregar el uso de controles de realidad virtual para PC.

Aplicar visión artificial para determinar el área a pintar y así evitar movimientos innecesarios del brazo robótico.

Para utilización de la realidad virtual se debe abrir previamente el programa RoboDK y conectar las gafas RV ya que si se lo hace después la vista de usuario se posicionará de forma incorrecta.

Para mejorar la experiencia al utilizar las gafas RV se recomienda ejecutar el programa Rifcat 2.0 como administrador. Si no se lo hace pueden surgir problemas de comunicación entre el ordenador y el smartphone.

BIBLIOGRAFÍA

ÁLVAREZ, Ángel José Fernández. 2010. *De las arquitecturas virtuales a la realidad aumentada: un nuevo paradigma de visualización arquitectónica.* Universidad de A Coruña, Coruña : 2010.

ANDALUZ, Victor, et al. 2018. *Control and monitoring of industrial processes.* Universidad de las Fuerzas Armadas ESPE, Sangolquí : 2018.

Application of simulation models programming of robots. **Jakubiec, Beata. 2018.** 25 de Mayo de 2018, Proceedings of the International Scientific Conference, Vol. V, págs. 283-292.

BARRIENTOS, Antonio; et al. *Fundamentos de Robótica.* Madrid : McGRAW-HILL.

BASCO, Ana Inés; et al. 2018. *Industria 4.0: Fabricando el futuro.* Buenos Aires : 2018.

CASADO, Cristina. 2007. *Manual Básico de MatLab.* 2007.

CASTILLO, Daniel; et al. 2018. *Control and Monitoring of Industrial Processes.* Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador : 2018.

CEDIA. 2020. CEPRA XIV-2020-08. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <https://www.cedia.edu.ec/es/proyectos-ganadores/cepra-xiv/cepra-xiv-2020-08-rva-para-la-ensenanza>.

cfzcobots. CFZCOBOTS. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <https://cfzcobots.com/robotica-colaborativa-para-la-pequena-empresa-industria-4-0/#>.

—. CFZCOBOTS. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <https://cfzcobots.com/robotica-colaborativa-para-la-pequena-empresa-industria-4-0/#>.

DORF, Richard ;& BISHOP, Robert. 2005. *Sistemas de control moderno.* Madrid : PEARSON EDUCACUÓN S.A., 2005. Vol. 10.

Economiopedia. 2019. Coeficiente de determinación (R cuadrado). [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <https://economipedia.com/definiciones/r-cuadrado-coeficiente-determinacion.html>.

Efficiente dynamic computer simulation of robotic manipulators. **Walker, M. W. y Orin, D. E. 1982.** 1982, ASME Journal Dynamic Systems, Measurement, and, Control, Vol. 104.

ESCARTÍN, Emilio R. 2000. idus. [En línea]. [Consulta: 22 octubre 2020]. Disponible en: Recuperado de: <http://hdl.handle.net/11441/45510>.

ESPI, Pilar Santos. 2015. centro-zaragoza. [En línea]. [Consulta: 22 octubre 2020]. Disponible en: http://www.centro-zaragoza.com:8080/web/sala_prensa/revista_tecnica/hemeroteca/articulos/R66_A3.pdf.

- GALICIA, Xunta de. 2017.** *Estado del arte de modelización, simulación y virtualización de procesos.* IGAPE, Galicia : 2017.
- GARCÍA, Juan Martín. 2018.** *Teoría y ejercicios prácticos de Dinámica de Sistemas.* Zaragoza : s.n., 2018.
- GIER, Michelle. 2015.** semanticscholar.org. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: Recuperado de: <https://www.semanticscholar.org/paper/Control-of-a-robotic-arm%3A-A-Application-to-on-surface-Gier/13814931335d7048ed1ef2a6c35c7a7c70f865d6#extracted>.
- GÓMEZ VALENTÍN, Manuel ;& RODELLAR BENEDEÉ, José. 2004.** *Control predictivo de canales de riego utilizando modelos de predicción de tipo Muskingum (primer orden) y de tipo Hayami (segundo orden).* Universidad Politécnica de Ctaluña, s.l. : 2004.
- GONZÁLEZ, Laura Raya. s.f..** acta. [En línea] [Consulta: 19 octubre 2020]. Disponible en: https://www.acta.es/medios/articulos/ciencias_y_tecnologia/061115.pdf.
- HERRERA, Javier José Saballos. s.f..** *Simulación y Modelación de sistemas.* Universidad del Norte de Nicaragua UNN-Jinotega, s.l. : s.f.
- HERRERA, Paulino San Miguel. 2019.** *Modelado y simulación de célula robótica colaborativa mediante realidad virtual.* Universidad de Oviedo, Oviedo : 2019.
- HUMANES, Gonzalo Roa. 2016.** *Diseño y desarrollo de un entorno de simulación dinámico para robots evolutivos.* . Escuela Técnica Superior de Ingenieros Industriales, Madrid : 2016.
- Ingeniería Automatización . 2017.** *Prácticas de Robótica utilizando Matlab.* 2017.
- KEATING, Rayan. 2017.** *Kinematics of a UR5.*
- KOROWAJCZENKO, Karinne Terán. 2012.** researchgate. [En línea]. [Consulta: 01 noviembre 2020]. Disponible en: www.researchgate.net/publication/278406306-REALIDAD_AUMENTADA_SUS_DESAFIOS_Y_APLICACIONES_PARA_EL_E-LEARNING.
- LANVERDE OCÁDIZ, Oscar ;& PÉREZ, Cuauhtémoc. s.f..** *Virtualización del proceso de pintado en Manufacturas Industriales Landaverde.* Centro de Tecnología Avanzada, CIATEQ., Hidalgo : s.f.
- LEDESMA, Rubén; MOLINA, Gabriel ;& VALERO, Pedro.** *Análisis de consistencia interna mediante Alfa de Cronbach: un programa basado en gráficos dinámicos.* Psico-USF, s.l. : s.n.
- LUH, J. Y. S. 1982.** *Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator.*

LUNA, María. s.f. *Aplicación de nuevas tecnologías inmersivas para la innovación en la maquinaria de construcción obra pública y minería*. Asociación española de fabricantes de maquinaria de construcción obras públicas y minería, Zaragoza : s.f.

MathWorks. 2017. Sistemas de control. [En línea]. [Consulta: 01 noviembre 2020]. Disponible en: <https://la.mathworks.com/solutions/control-systems.html>.

—. **2017.** Sistemas Dinámicos. [En línea]. [Consulta: 01 noviembre 2020]. Disponible en: <https://www.mathworks.com/discovery/dynamic-systems.html>.

—. **2019.** *System Requirements- Release 2020a- Windows*.

MÉNDEZ, Juanes ;& ANTONIO , Juan. 2010. redalyc.org. [En línea]. [Consulta: 25 octubre 2020]. Disponible en: <https://www.redalyc.org/articulo.oa?id=201014893011>.

MENÉNDEZ, Manuel José ;& BERMEJO, Jiménez David . 2018. accioncultural. [En línea] . [Consulta: 25 octubre 2020]. Disponible en: <https://www.accioncultural.es/media/2018/ebook/Anuario/2JMMene%CC%81ndezDJBermejo.pdf>.

Microsoft. 2020. BEST PRACTICES. [En línea]. [Consulta: 15 noviembre 2020]. Disponible en: www.bestpractices.com.mx/microsoft/tecnologias-inmersivas/.

Modelos cinemático y dinámico de un robot de cuatro grados de libertad. **CARDOSO, Eileen; et al. 2017.** La Habana : s.n., Diciembre de 2017, Ingeniería Electrónica, Automática y Comunicaciones, Vol. 38.

ORAIN, Gipuzkoa ETORKIZUNA. Robótica Colaborativa & Visual Computing. *gipuzkoa.eus*. [En línea]. [Consulta: 13 noviembre 2020]. Disponible en: https://www.gipuzkoa.eus/documents/20933/1883176/DFG-Industria4-0-Caso-RC_VC-IK4-Tecnalia-Esp.pdf.

Oviedo, Heidi Celina ;& Campo Arias, Adalberto. s.f. *Aproximación al uso del coeficiente alfa de Cronbach*. Revista Colombiana, Bogotá : s.f.

PINTARMICOCHE. 2015. pintarmicoche. [En línea]. [Consulta: 20 noviembre 2020]. Disponible en: <https://www.pintarmicoche.com/wp-content/uploads/2015/03/APLICACION-PINTURA-DE-ACABADO-FASE-2-APLICACION-DE-LA-PINTURA.pdf>.

RoboDK. 2017. API de RoboDK. [En línea]. [Consulta: 12 noviembre 2020]. Disponible en: <https://robodk.com/doc/es/RoboDK-API.html#RoboDKAPI>.

—. **2020.** Foro de RoboDK. [En línea]. [Consulta: 15 noviembre 2020]. Disponible en: <https://robodk.com/forum/Thread-Which-are-the-requerimets-of-RoboDK>.

—. 2021. robodk.com. [En línea]. [Consulta: 20 noviembre 2020]. Disponible en: <https://robodk.com/doc/es/Interface.html>.

ROJAS, David Alejandro Jaramillo. 2010. *Modelo simulación y control de un brazo robótico mediante MatLab y Simulink para soldadura de arco*. Escuela Politécnica Nacional, Quito : 2010.

ROMÁN, José Luis. 2017. *Industria 4.0: la transformación digital de la industria*. Facultad de Ingeniería de la Universidad de Deusto, s.l. : 2017.

ROZAS, Simón Ernesto Martínez. 2018. *Modelado y simulación de robots terrestres para la inspección del alcantarillado*. Universidad de Sevilla, Sevilla : 2018.

SÁENZ, Carolina Castresana. 2016. *Industria 4.0*. Universidad de la Rioja, La Rioja : 2016.

School., ESIC Business and Marketing. 2018. ESIC. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <https://www.esic.edu/rethink/tecnologia/robotica-colaborativa-papel-tienen-las-personas>.

SHROUF, Fadi; ORDIERES, Joaquin ;& MIRAGLIOTTA, G. 2014. *Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm*. IEEE International Conference on Industrial Engineering and Engineering Management, s.l. : 2014.

STEENMEIJER, Maarten. 2014. *Introducción al modelado de sistemas*.

TOLEDO, Serra Rolando; et al. 2009. scielo. [En línea]. [Consulta: 21 noviembre 2020]. Disponible en: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172009000100007&lng=es&tlng=es.

—. 2009. scielo. [En línea]. [Consulta: 12 noviembre 2020]. Disponible en: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172009000100007&lng=es&tlng=es.

Universal Robots. 2016. Universal Robots. [En línea]. [Consulta: 08 noviembre 2020]. Disponible en: Recuperado de : <https://www.universal-robots.com/case-stories/>.

UNIVERSAL ROBOTS. UNIVERSAL ROBOTS. [En línea]. [Consulta: 08 diciembre 2020]. Disponible en: https://www.universal-robots.com/media/1801283/esp_semea_199912_ur3_tech_spec_web_a4.pdf.

Universidad de Santiago de Chile. s.f. UdeSantiagoVirtual. [En línea]. [Consulta: 20 octubre 2020]. Disponible en: <http://www.udesantiagovirtual.cl/moodle2/mod/book/view.php?id=24819>.

Utilización del alfa de Cronbach para validar la confiabilidad de un instrumento de medición de satisfacción del estudiante en el uso del software Minitab. **BOÓRQUEZ, José; et al. 2013.** 2013.

Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology.
pág. 9.

VILCHIS, Topiltzin. 2020. *Diseño y construcción de un robot háptico de tres grados de libertad.*
Universidad Nacional Autónoma de México, México : 2020.

Virtual Visit with 3D Modeling of the academic spaces of the UTSV. **GOMEZ, Manuel; et al.**
2018. 6 1-7, Septiembre de 2018, Tecnología Informática, Vol. 2.

VR BOX. 2020. VR BOX VIRTUAL REALITY GLASSES. [En línea]. [Consulta: 15 diciembre
2020]. Disponible en: [http://www.vr-
box.es/#:~:text=Convierte%20tu%20m%C3%B3vil%20Android%20o,disfrutar%20de%20una
%20experiencia%20%C3%BAnica](http://www.vr-box.es/#:~:text=Convierte%20tu%20m%C3%B3vil%20Android%20o,disfrutar%20de%20una%20experiencia%20%C3%BAnica).

wikiHow. 2020. Automóviles. Reparación y mantenimiento de automóviles. [En línea].
[Consulta: 25 octubre 2020]. Disponible en: <https://es.wikihow.com/pintar-un-auto#aiinfo>.

ANEXOS

ANEXO A: Programa del modelo cinemático directo del UR3

```
% Forward Kinematic.
% Devuelve la matriz de transformación del
% primer sistema de coordenadas al último en función del vector q
% de variables articulares.

function T = forwardkinematic(q)
%% Parámetros Denavit-Hartenberg del robot
teta = q;
d = [151.9 0 0 112.35 85.35 81.9];
a = [0 -243.65 -213.25 0 0 0];
alfa = [pi/2 0 0 pi/2 -pi/2 0];
%% matriz de transformación homogénea 4 x 4 a partir de los parámetros de D-H
function dh=denavit(teta, d, a, alfa)
dh=[cos(teta) -cos(alfa)*sin(teta) sin(alfa)*sin(teta) a*cos(teta);
    sin(teta) cos(alfa)*cos(teta) -sin(alfa)*cos(teta) a*sin(teta);
    0 sin(alfa) cos(alfa) d;
    0 0 0 1];
end
%% Matrices de transformación homogénea entre sistemas de coordenadas consecutivos
A01 = denavit(teta(1), d(1), a(1), alfa(1));
A12 = denavit(teta(2), d(2), a(2), alfa(2));
A23 = denavit(teta(3), d(3), a(3), alfa(3));
A34 = denavit(teta(4), d(4), a(4), alfa(4));
A45 = denavit(teta(5), d(5), a(5), alfa(5));
A56 = denavit(teta(6), d(6), a(6), alfa(6));

%% Matriz de transformación del primer al último sistema de coordenadas
A06 = A01 * A12 * A23 * A34 * A45 * A56;
T=Pose_2_XYZRPW(A06)
% xlsxwrite('Pruebas.xlsx',T,'Forward Kinematic');
end
```

ANEXO B: Programa del modelo cinemático inverso del UR3

```
% Cinemática Inversa

function q = inversekinematic1(xyzrpw)
%% Parámetros Denavit-Hartenberg del robot
d = [151.9 0 0 112.35 85.35 81.9];
a = [0 -243.65 -213.25 0 0 0];
alfa = [pi/2 0 0 pi/2 -pi/2 0];

T= XYZRPW_2_Pose(xyzrpw);

%% Posición de la mano del manipulador
p = T*[0,0,-d(6),1]';

%% Solución de la primera articulación: q1
psi= atan2(p(2),p(1));
phi=acos(d(4)/sqrt(p(1)^2+p(2)^2));
q1 = psi+abs(phi)+pi/2;

%% Solución de la quinta articulación: q5
p06 = T*[0 0 0 1]';
p16_z = p06(1)*sin(q1)-p06(2)*cos(q1);
q5=-acos((p16_z-d(4))/d(6));

%% Solución de la sexta articulación: q6
T06=T;
T01 = denavit(q1, d(1), a(1), alfa(1));
T61 = inv(inv(T01)*T06);
q6 = atan2((-T61(2,3))/sin(q5),T61(1,3)/sin(q5));

%% Solución de la tercera articulación: q3
T16 = inv(T61);
T45 = denavit(q5, d(5), a(5), alfa(5));
T56 = denavit(q6, d(6), a(6), alfa(6));
T64 = inv(T45*T56);
T14 = T16*T64;
p13=T14*[0 -d(4) 0 1]' - [0 0 0 1]';
q3=abs(acos((norm(p13)^2-a(2)^2-a(3)^2)/(2*a(2)*a(3))));
%% Solución de la tercera articulación: q2
q2 = -atan2(p13(2), -p13(1)) + asin((a(3)*sin(q3))/norm(p13(1:3)));
%% Solución de la tercera articulación: q4
T12 = denavit(q2, d(2), a(2), alfa(2));
T23 = denavit(q3, d(3), a(3), alfa(3));
T31 = inv(T12*T23);
T34 = T31*T14;
q4 = atan2(T34(2,1), T34(1,1));
%% Vector de coordenadas articulares
q = [q1 q2 q3 q4 q5 q6]';
% xlswrite('Pruebas.xlsx',q,'Inverse Kinematic');
end
```

ANEXO C: Programa para el control

```
function [ERROR, PAR ]= CONTROL(intE,q,qd)

tau=[0 0 0 0 0 0];
T=0.5;
n=6;

%% Ganancias del control PID

kp_PID = [25.01    1.0001    49.4    47.408    2    1.3];%ganancia✓
proporcional
ki_PID = [1.09    1.1001    1.1997    0.101    0.1    0.1001]; %ganancia✓
integral
kd_PID = [5.5001    3.9001    80.091    18.3710    8.5555    0.5];%ganancia✓
derivativa

%% Control
E=[0 0 0 0 0 0]; %error
dotE=[0 0 0 0 0 0]; %derivada del error
Eant=[0 0 0 0 0 0];%almacena el valor del error
for i = 1:6
    E(i) = qd(i)- q(i); % Error de posición
    dotE(i) = (E(i) - Eant(i))/T; % Derivada del error
    Eant(i) = E(i); % Guarda el valor anterior
end

for i = 1:n
    tau(i) = kp_PID(i)*E(i) + ki_PID(i)*intE(i) + kd_PID(i)*dotE(i); % Par
end

PAR=tau';
ERROR=E;
```

ANEXO D: Evidencias prueba de simulación





ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 10 / 11 / 2021

INFORMACIÓN DEL AUTOR	
Nombres – Apellidos: BEATRIZ GRACIELA MERCHÁN CUNO ALEXIS GONZALO CHIMBO CORTEZ	
INFORMACIÓN INSTITUCIONAL	
Facultad: INFORMÁTICA Y ELECTRÓNICA	
Carrera: INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN	
Título a optar: INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN	
f. Analista de Biblioteca responsable:	 Firmado electrónicamente por: ELIZABETH FERNANDA AREVALO MEDINA

