UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO PROFESIONAL

FUNCIÓN DE SIMULACIÓN DE SISTEMAS DE PULVERIZACIÓN EN ACE

PREVIO A LA OBTENCIÓN DEL TÍTULO:

INGENIERÍA EN MECATRÓNICA

PRESENTADO POR:

21251067 CARLOS ANDRÉS ESCOBAR PINEDA

ASESOR: ING. JOSE ORDOÑEZ

CAMPUS SAN PEDRO SULA, CORTÉS, HONDURAS, C.A.

ABRIL 2021

DEDICATORIA

La siguiente investigación va dedicada a todas las personas que han avanzado la humanidad. A esas mentes que veían los celulares y el internet cuando apenas existía un radio. Sin ellos y ellas nada de esto sería posible. Pensar en lo que harían esas personas con la tecnología de hoy en día es lo que me da la energía para seguir adelante.

RESUMEN EJECUTIVO

La siguiente investigación brinda un producto en C# que da un valor agregado al programa ACE, de OMRON. Como base se tiene una función en C# que recibe de parámetros el robot y variables globales de ACE y es capaz de dibujar la trayectoria del robot en el espacio 3D, solamente cuando el efector final este encendido. Una aplicación de esto es ver la trayectoria de procesos de soldadura y pintura directamente en ACE.

Finalmente se extendió el código en C# que dibuja la trayectoria en ACE para obtener una simulación de un sistema de pulverización industrial. Para simular el rociado de pintura en el espacio 3D se dibujaron partículas en forma de cono elíptico. Para calcular la altura del cono se derivó una fórmula de posición de las partículas de pinturas, partiendo del efector final del robot. El código final permite configurar todas las variables involucradas como: masa de partículas, diámetro de efector final, presión del robot, numero de iteraciones de dibujo, tiempo entre cada iteración, numero de particiones en el cono elíptico, ángulo phi del cono elíptico y tiempos a evaluar la posición de las partículas.

Palabras Clave: ACE, C#, Simulación, Trayectoria, Rociado

ABSTRACT

The following investigation provides a C# product that adds value to OMRON's program, ACE. As a base, there is a function in C# that receives the robot and ACE's global variables as parameters and has the ability to draw the robot's trajectory in 3D space, only when the end effector is on. One application of this is to view the welding and painting process path directly in ACE.

Finally, the C# code that draws the trajectory in ACE was extended to obtain a simulation of an industrial spraying system. To simulate paint spraying in 3D space, the particles were drawn in the shape of an elliptical cone. To calculate the height of the cone, a formula for the position of the paint particles was derived, starting from the end effector of the robot. The final code allows configuring all the variables involved such as: particle mass, end effector diameter, robot pressure, number of drawing iterations, time between each iteration, number of partitions in the elliptical cone, phi angle of the elliptical cone and times to evaluate the position of the particles.

Keywords: ACE, C #, Simulation, Path, Spray

Índice de Contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE SIGLAS Y GLOSARIO

ACE	Automation Control Environment

I. INTRODUCCIÓN

ACE es un entorno de control de automatización. OMRON desarrollo este software poderoso para el modelado de procesos industriales. Tiene grandes características como la programación offline en V+ de robots industriales. Pero una de las características mas interesantes de ACE es la habilidad de extensión. ACE provee la facilidad de agregar código en C# y extenderlo de la manera que sea posible para el desarrollador. De esta manera es posible crear librerías y funciones que dan valor agregado a ACE. Esta investigación busca crear un producto en C# que da un valor agregado que hasta hoy se creía que no era posible en ACE.

Se consultó la manera de graficar la trayectoria de robots de soldadura y pintura en ACE y la misma gente de ACE dijo que esto no era posible en el programa. La investigación utilizara código en C# junto con las librerías de ACE para hacer esto posible. Después de tener la trayectoria se irá un poco más allá y se simulará el rociado de pintura. Se verá un algoritmo para determinar la trayectoria de partículas de pintura, junto con la aplicación de conos elípticos y densidad de partículas en el espacio tridimensional. El algoritmo termina calculando y posicionando varios puntos en las variables globales de ACE. Suficientes número de iteraciones y puntos dibujados traerán a vida una simulación de rociado de pintura industriales en el entorno de ACE. Además, este rociado se programará para hacerlo solamente cuando el efector final del robot este encendido.

Los algoritmos de trayectoria de partículas y C# no son suficiente para dar vida a esta investigación. También es necesario conocer las librerías de ACE, esto involucra las funciones y objetos utilizados en memoria. Un problema común al querer extender el programa ACE con C# es la poca documentación para guiarse y hacerlo. Esta investigación abordara funciones de librerías de ACE como ser: dibujar puntos, mover puntos en el espacio tridimensional, agregar variables globales, leer la posición de un robot, obtener el estado del efector final de un robot, entre otros. Es la combinación de conocimientos de librerías de ACE, C# y modelados de trayectoria lo que dará vida a la simulación de rociado de pintura en ACE.

II. PLANTEAMIENTO DEL PROBLEMA

La siguiente investigación busca codificar en C# un valor agregado o característica nueva al software ACE, el cual dibuje la trayectoria y simule el rociado de pintura industrial. Dibujar la trayectoria cuando el efector final del robot este encendido en ACE ya es un gran aporte, pero también se extendió y se tiene la opción de simular un sistema de pulverización y controlar todas las variables involucradas.

2.1 PRECEDENTES DEL PROBLEMA

Antes de llegar a la simulación del rociado de pintura industrial en ACE se habló de simulación de trayectoria de pintura y soldadura. En su momento el Ingeniero intercambio palabras con las personas de ACE, consultando si se podía dibujar la trayectoria de un robot cuando el efector final de este estuviera encendido. La gente de ACE respondió que esta característica en particular no estaba disponible en ACE, pero que sería un buen valor agregado.

De esta manera surge la presente investigación de simulación de trayectoria de pintura y soldadura en ACE a través de C#. Lo que luego evoluciono a la simulación de un sistema de pulverización, con variables controladas.

2.2 DEFINICIÓN DEL PROBLEMA

El problema a solucionar es uno que se creía no tener solución. Esto daría un valor agregado al programa ya que el dibujo de la trayectoria de un robot en el espacio tridimensional en ACE solamente cuando el efector final este encendido simplemente no era posible. Si se creía que la trayectoria no era posible, mucho menos la simulación del rociado de pintura industrial. Esta investigación busca crear un producto que brinde esas soluciones o valores agregados.

2.3 JUSTIFICACIÓN

Esta investigación ataca directamente una necesidad que ya existía, pero no se tenía un producto(función/librería) que lo lograra. Ya que desde antes se quería ver las trayectorias de los robots cuando los efectores finales estuvieran encendidos, pero no se podía. Al cumplir con los objetivos se podrá suplir esa necesidad.

2.4 PREGUNTAS DE INVESTIGACIÓN

¿Cómo codificar un producto en ACE en C# que dibuje la trayectoria de cualquier robot solamente cuando el efector final este encendido?

¿Cómo codificar un producto en ACE en C# que simule un sistema de pulverización en cualquier robot?

¿Cómo se puede desarrollar un modelo matemático con variables configurables para la simulación de un sistema de pulverización industrial?

2.5 OBJETIVOS

Si todos los objetivos de esta investigación se cumplen, se tendrá una pequeña pero poderosa extensión y mejora al programa ACE. Se habrá dado una solución a algo que antes se creía que no era posible. Hacer esto es más que el valor agregado, ya que es de esta manera que los programas van madurando y llegando a lo que conocemos hoy en día.

2.5.1 OBJETIVO GENERAL

Codificar un producto (función/librería) en C# que de un valor agregado al software ACE, el cual tenga la capacidad de dibujar trayectoria de robots y simular sistemas de pulverización industriales con variables controladas programables.

2.5.2 OBJETIVOS ESPECÍFICOS

- Codificar un producto (función/librería) en ACE en C# que dibuje la trayectoria de cualquier robot solamente cuando el efector final este encendido.
- Codificar un producto (función/librería) en ACE en C# que simule un sistema de pulverización en cualquier robot.
- Desarrollar un modelo matemático con variables configurables para la simulación de un sistema de pulverización industrial.

III. MARCO TEÓRICO

3.1 INTEGRACIÓN DE SOFTWARE Y ROBÓTICA

En el poco tiempo de 5 décadas el termino software ha avanzado en concepto y definición. Antes se hablaba de software y se entendía como una seria de instrucciones o algoritmos que desarrollaban una tarea específica. Con el paso del tiempo y desarrollo de interfaces gráficas, se llegó al concepto de aplicaciones de escritorio, paginas web y aplicaciones móviles. Hoy en día se puede hablar de Inteligencia Artificial, Internet de las Cosas y Aprendizaje Automático. La evolución del concepto de software afecta el concepto de robótica. Aunque en robótica se habla de partes físicas llevando a cabo tareas, hay que programar estas partes y para programarlas se utilizan diferentes tipos de softwares. Por ende, la manera de interactuar y programar robots también ha ido evolucionando con el tiempo.

Este software evolucionado no solo permite programar un robot de forma más eficiente y fácil, sino que también poco a poco va abriendo brechas hasta llegar a tareas que antes no eran posibles. Hoy en día hay problemas que son analizados, manejados y resueltos todo por un software. Algunos de estos problemas simplemente son imposibles para un ser humano, ya sea por cálculos complicados o porque simplemente son millones de casos y cálculos y llevaría demasiado tiempo.

Un ejemplo claro de esto son algoritmos de planeación de ruta en robótica. La programación de movimiento de robots ha crecido y madurado igual que el concepto de software. Así como al comienzo el software desarrollaba una tarea específica, el código fuente de un robot era igual. Una serie de movimientos predeterminadas y específicos que se repetían una y otra vez y eso era otro. A diferencia de los robots de hoy que cuentan con Inteligencia Artificial y conexión a Internet.

Hay un estudio que utilizó los algoritmos de genética con la programación de movimiento de robots. La meta era encontrar la ruta más rápida y eficiente dado el punto de origen y conociendo el punto destino. Los algoritmos de genética utilizan la selección natural de billones de muestras iterativas y repetitivas. Shao(2021, p.3) lo describe de la siguiente manera:

El robot comienza en el origen, y cuando llega a cierto punto, va a encontrar un nuevo obstáculo. Actualmente, el robot puede sintetizar la información detectada y con el conocimiento previo de obstáculos conocidos, y rediseñar la ruta de ahí en adelante. Luego moverse a un nuevo punto de origen por una nueva ruta, and encontrar un nuevo obstáculo al llegar a otro punto, repetir el proceso, hasta que finalmente el robot llega al punto destino.

Resulta ser que los algoritmos de genética fueron muy eficientes en esta tarea específica de diseño de rutas de robots sin intervención humana. Se observa que este proceso sería no solo tedioso, sino que imposible para un ser humano, dadas las billones de iteraciones y versiones involucradas en los algoritmos de genética.

Así como hay tareas que son imposibles para humanos, hay otras tareas que poco a poco van aumentando el porcentaje de aceptación en los resultados hasta llegar a algo nuevo. Este hecho se puede observar en la integración de software con los carros. Poco a poco se fueron agregando controles remotos, radios, sistemas de conexión de bluetooth, hasta llegar a la Inteligencia Artificial y la era de carros autónomos. Ahora es totalmente posible utilizar el Aprendizaje Automático o Aprendizaje de Máquina para entrenar o enseñarle al software como debe de comportarse. Esto puede aplicarse muy bien a la tarea de manejar un carro y entrenar el código fuente de uno para que lo haga. Y la habilidad de manejar puede alcanzarse con software de otra manera, utilizando métodos de control de lógica difusa entrenados con la habilidad de manejar. Y eso es exactamente lo que se hizo en un estudio en el cual Rafsanjani et al. (2019) concluyeron que "El prototipo del carro autónomo que se realizo va acorde el diseño del prototipo que puede seguir la trayectoria, pero el prototipo varias veces se sale de la pista al no tener la dirección correcta en las vueltas" (p. 8). El algoritmo de este estudio tuvo un valor de error de 0.149712. Aunque no es tan efectivo como los algoritmos de genética con la planeación de trayectoria, este error no es tan mal. Y con el paso del tiempo y mejora del hardware y software este error se va reduciendo, hasta llegar a revoluciones tecnológicas como carros autónomos.

Por otro lado, hay tareas altamente complejas que simplemente no se pueden lograr sin la alta integración y sincronía de la robótica, el software y el usuario. Un claro ejemplo de esto es el manejo de las plantas nucleares. En esencia son operados controlando y monitoreando entornos

hostiles con temperaturas y valores extremos. Un descuido o mal manejo de la energía nuclear y el resultado es un desastre como el de Chernobyl, que hoy en día sigue afectando el área. Esta tarea altamente compleja no se ha podido automatizar del todo, pero si se ha integrado la Inteligencia Artificial para auxiliar a los operadores y aumentar la eficiencia operaria. Un estudio creo robots inteligentes para que los operadores trabajaran en conjunto con ellos. Al final, Fang y Wang (2020, pg.4) sostuvieron que:

> El sistema robot auxiliar en operadores NPP es una herramienta útil para que los operadores NPP puedan mejorar las operaciones NPP. El sistema va a mostrar los resultados mas óptimos de base de datos basado en una Inteligencia Artificial. La decisión final correcta les pertenece a los operadores NPP. El sistema es una herramienta de apoyo para los operadores NPP. No esta hecho para reemplazar a los operadores NPP.

Como se había dicho, el manejo de la energía nuclear es tan compleja que requiere de la integración correcta de robots, software y operarios para funcionar correctamente. Y el más mínimo error en uno de los 3 puede causar un desastre grave e irreparable. Sin embargo, al igual que los carros autónomos, poco a poco el margen de error se va disminuyendo y los avances de hardware y software tarde o temprano nos llevaran al manejo correcto de la energía nuclear.

3.1.1 ROBOTICA Y SIMULACIONES

Otro gran avance del software que ha afectado enormemente el área de la robótica son las simulaciones. Las simulaciones permiten: programar robots sin necesidad de estar con ellos físicamente, arreglar y distribuir equipos industriales, probar procesos industriales, y mucho más. De hecho, tener simulaciones integradas a la robótica permite avances que antes no eran posibles para el humano. Se puede ver un claro ejemplo con una investigación que desarrollo un robot de 6 ejes, el diseño y cinemática del robot fue simulado completamente. Diseñar un robot con tantas piezas y movimientos ya de por si es una tarea compleja, y es claro que un robot con 6 ejes sería difícil de hacer sin simulaciones. Esta investigación proveyó el modelo esqueleto del robot, el diseño del robot, la gráfica de la trayectoria del movimiento del robot, planos con dimensiones del robot, entre otros. Se utilizan parámetros Denavit-Hartenberg para asignar coordenadas. A

continuación, se presenta una imagen de las coordenadas y ejes asignados al robot usando los parámetros DH.

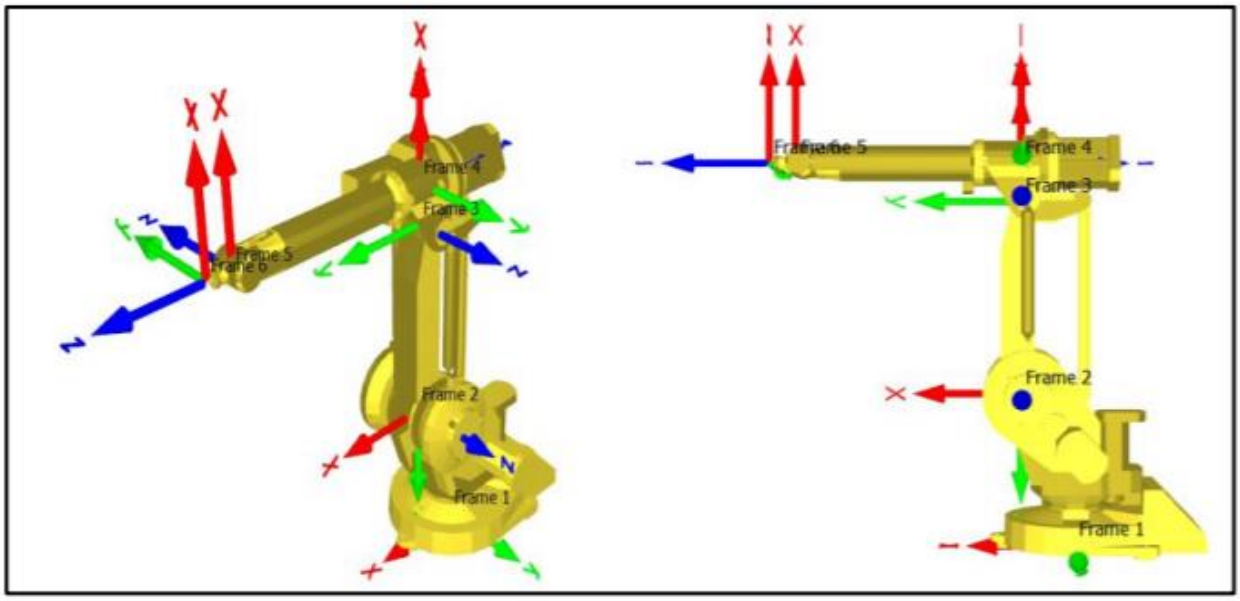

Figura 1 - Coordenadas y Ejes Asignados al Robot Usando los Parámetros DH

Fuente: Talli y Meti

Se aprecia que este diseño y asignación de ejes sería algo complejo para realizar por un ser humano. En la investigación, Talli y Meti (2020) cerraron diciendo "La optimización del diseño, cinemática y dinámica del modelo y movimiento estudios han sido analizados exitosamente y propuestos a los ingenieros en robótica para reducir el esfuerzo y el tiempo de ciclo en la industria automovilística, aeroespacial, electrónica, medica" (pg. 8). De aquí se puede decir que las simulaciones si ayudan en el avance de la robótica.

Una investigación similar a la anterior es una que utiliza MATLAB para crear un robot con 5 grados de libertad. De nuevo, una tarea algo compleja, pero posible, para un ser humano. En esta investigación, Sabri et al. (2021) concluyeron que "Simulacion robotica utilizando MATLAB demostró que la simulación ayuda al robot determinando el movimiento usando torsión en la cinemática de la articulación o el brazo del robot, para que pueda optimizar el torque usado para el movimiento del robot" (pg. 9). De aquí se observa que las simulaciones no solo ayudan en el diseño de los robots, sino que también en la cinematica.

Hay otra investigación que utiliza las simulaciones y lleva la robótica industrial a niveles nunca antes pensados. Esta investigación integra la realidad virtual, de manera que el usuario puede entrar a su espacio virtual y diseñar un proceso industrial por completo. Esto es revolucionario porque el usuario puede realizar todo el proceso industrial sin necesidad de tener todo el equipo, sin mencionar el ahorro monetario para probar el proceso industrial. El usuario puede colocar y mover libremente equipos industriales, bandas, robots, etc. El usuario también puede programar los robots industriales. Y finalmente, el usuario puede simular una entrada continua al proceso y ver todo el proceso industrial en tiempo real. Esto no es solo sorprendente, sino que también da una nueva habilidad al usuario de tener todo el proceso en tiempo real y poder analizar cada parte, esto hace las tareas de encontrar errores y aplicar mejores mucho mas fácil. A continuación, se presenta una imagen de la investigación donde se observa una realidad virtual de un proceso industrial.

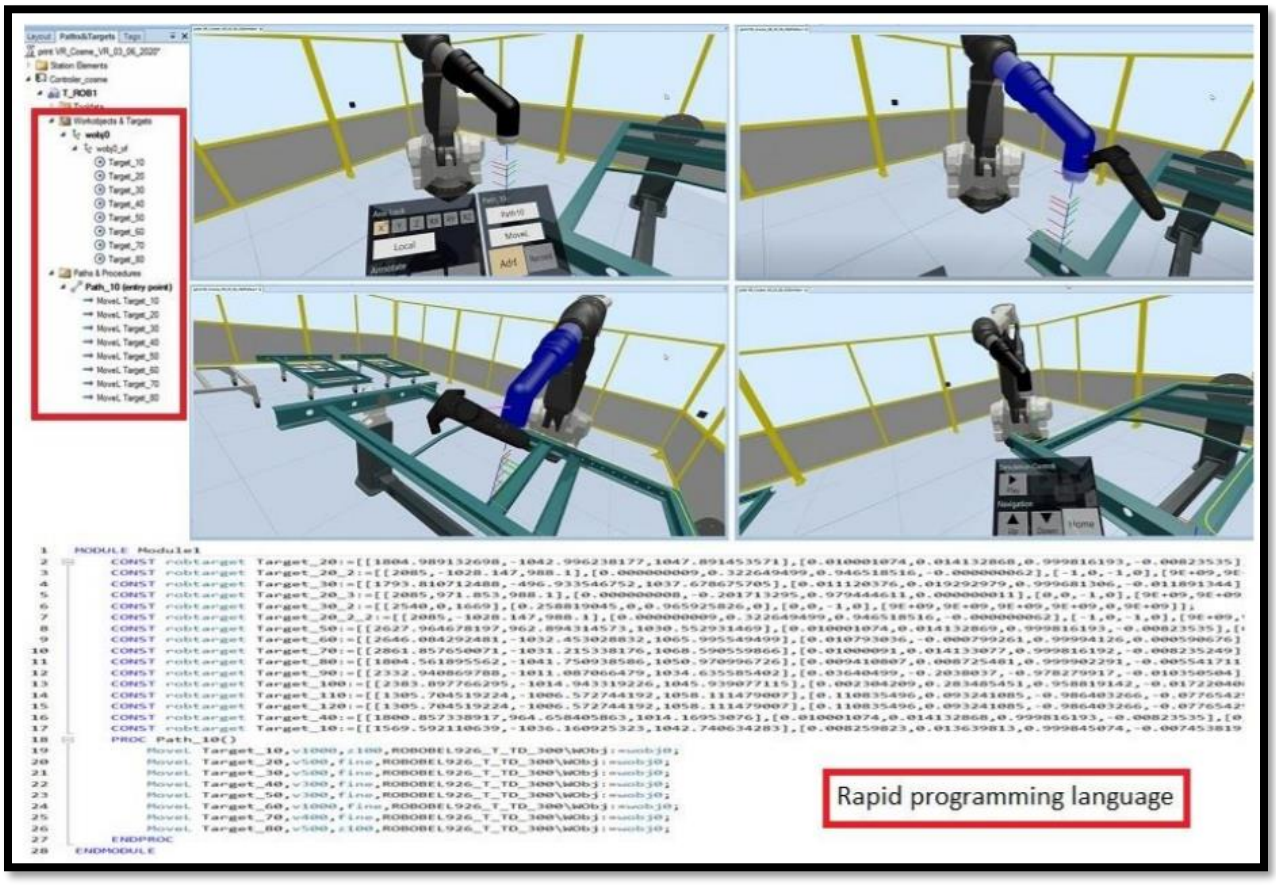

Figura 2 - Realidad Virtual de un Proceso Industrial

Fuente: Holubek et al

Holubek et al. (2021, pg.6) describen las ventajas de esta realidad virtual así:

> Las ventajas de crear una simulación robótica en el entorno de realidad virtual, además de reducir el tiempo de creación de una futura simulación robótica, yace en la habilidad de rápidamente verificar objetivos y trayectorias creados, pero también en la habilidad de modificar esos objetivos, ajuste de zona, ajuste de velocidad y rápidamente calibrar por traducción y reorientación de objetivos creados en un sistema coordenado en un entorno de realidad virtual.

No es necesario saber mucho de robótica para detectar el gran avance que esta realidad virtual trae, sin mencionar el ahorro de tiempo y dinero, sino que una sensación de control y monitoreo total en tiempo real. Aparte de realizar los procesos industriales en menos tiempo, son menos susceptibles a errores, y si hay uno, es fácil recorrer el proceso y ver que esta pasando. Es evidente que las simulaciones traen grandes ventajas al diseño de robots y procesos industriales. Pero hay otro concepto ya introducido en la investigación de realidad virtual que también está avanzando la robótica y es el de programación offline.

3.1.2 ROBÓTICA Y PROGRAMACIÓN OFFLINE

La programación offline, como su nombre lo dice, es una programación que puede realizarse sin tener una conexión en línea, es decir, una conexión a internet. Es lo que ha permitido a los usuarios poder programar los robots sin necesidad de estar con ellos. Esto se puede lograra con entornos simulados, donde el robot se programa con el mismo código que usaría en la vida real y se prueba. La idea es que, si todo sale bien en la simulación, se podrá llevar ese mismo código al robot físico sin necesidad de hacer ningún cambio. Hay varias investigaciones de programación offline con robots, desde simulaciones hasta mecanismos que el usuario programa una vez y se dejan funcionando.

Un ejemplo de una investigación es un sistema de programación offline para un brazo robótico ABB. Esta investigación aprovecha los archivos DXF, generados a partir de los CAD, para realizar integraciones con otros programas y aplicaciones. Al final esta investigación se genera el código de trayectoria de soldadura, el cual puede ser usado en el robot físico y funciona de manera aceptable. Esto es muy similar a la presente investigación. Al final, Xu y Liu (2019)

concluyeron que "A través de la simulación y verificación experimental del software Robot Studio, la factibilidad de este sistema de programación offline es probada. La trayectoria dibujada por el CAD puede ser simulada y aplicada al brazo robótico de soldadura" (p. 7) Aquí se pueden ir viendo la productividad y ventajas de la programación offline.

La misma ventaja de la programación offline puede verse con la investigación que utiliza la fotogrametría para generar modelos 3D de objetos con geometrías irregulares. Hay objetos que son difíciles y costosos de modelar, de aquí la fotogrametría. Thevara y Kumar (2018) describen la fotogrametría de la siguiente manera: "La fotogrametría digital es un sistema de medición sin contacto que utiliza una secuencia de imágenes obtenidas mediante cámaras de alta resolución, con el fin de identificar características, puntos y coordenadas únicos en la superficie de los objetos" (pg.3). A continuación, se muestra una imagen de la investigación de la fotogrametría, el objeto con geometría irregular es analizado de varios ángulos por varios cámaras, el software puede transformar las coordenadas de proyecciones a un modelo que se puede utilizar en el siguiente paso del proceso.

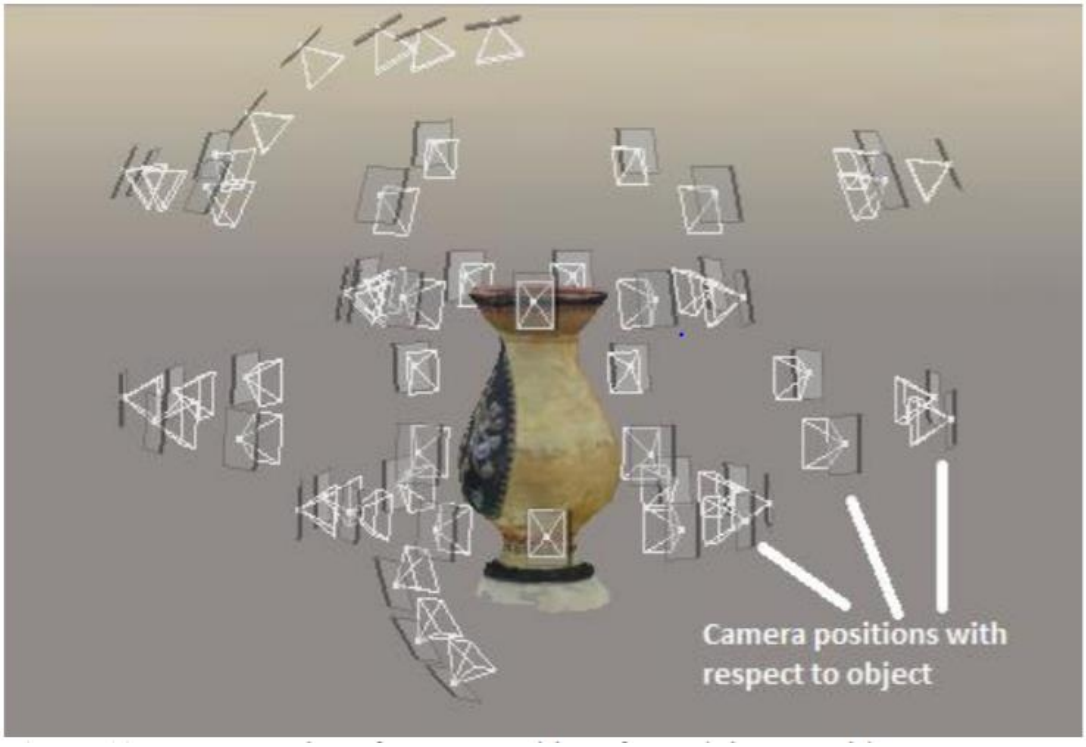

Figura 3 - Ejemplo de Fotogrametría

Fuente: Thevara y Kumar

Pero lo interesante es que luego de generar esos modelos de manera más barata y eficiente, estos son utilizados en programación offline para programar la trayectoria del robot sin estar con el robot físicamente. Thevara y Kumar (2018, pg.13) concluyen diciendo:

> En el futuro, el software se utilizará para obtener el modelo 3D de objetos de prueba fabricados específicamente para el uso del proyecto. Estos modelos 3D luego se exportarán como archivos STL que luego se importarán a RobotStudio para operaciones adicionales, incluyendo simulación del proceso de pulido y cálculo del tiempo de ciclo.

De nuevo se ve como la programación offline, a veces auxiliada de otros métodos, permite programar la trayectoria de los robots sin necesidad de estar con ellos físicamente. En el caso de la investigación de la fotogrametría se puede programar la trayectoria del robot industrial solo analizando el objeto irregular.

Hay un tercer ejemplo de una investigación que se auxilia de la programación offline para generar las trayectorias de los robots industriales. El software realizado en esta investigación analiza un archivo STL, reconstruye la topología, y utiliza un algoritmo de cortes para ir generando y uniendo las trayectorias de superficie. En los resultados, Liu et al. (2020) dicen "El método de planificación de ruta fuera de línea propuesto en este documento es principalmente para la planificación de ruta de la superficie de la pieza de trabajo, y es adecuado para campos de procesamiento como pulverización y limpieza" (pg. 5). Se observa que la programación offline normalmente se auxilia de alguna otra tecnología o entrada, como la fotogrametría o análisis de un STL, para llegar a un modelo sobre el cual se pueda programar la trayectoria. También la aplicación más común es programar la trayectoria de robots industriales.

Hay un último ejemplo de una investigación que utiliza un software de simulación de estaciones de trabajo industriales llamado K-Roset. Se modelan procesos industriales y se programan robots Kawasaki de manera offline. Golda y Kampa (2017) describen el inicio del proceso así "Como se mencionó anteriormente, el primer paso en diseño de una celda de producción robotizada o completamente automatizada. El sistema de fabricación fue la creación de una base de datos de objetos CAD que representan equipos elementales en sistema real" (pg.3). Al comparar todas estas investigaciones con programación offline se ve una base de simular la estación de trabajo, colocar los robots, programarlos de manera offline para tener un

código fuente con garantía de que va a funcionar en el robot físico. Aparte de que cambiar código y volver a probar un proceso simulado es mucho más rápido y eficiente y ahorra tiempo.

Es clara de relación entre la evolución del software y la evolución del hardware. Aunque el fenómeno también funciona a la inversa, evolución de hardware fomenta la evolución del software. Avances en Inteligencia Artificial, simulaciones y programación offline han llevado la robótica industrial a niveles antes no imaginados. Ahora es posible diseñar, implementar e incluso programar un proceso industrial sin necesidad de tener los equipos físicos. Estos avances han llegado a otras dos grandes áreas de la robótica industrial, el pintado y la soldadura.

3.2 PINTURA INDUSTRIAL AUTOMATIZADA

Uno de las partes más importantes de los procesos industriales a la hora de crear piezas es el acabado, esto incluye pulido y pintura. Al igual que la robótica y el software, la pintura también ha ido evolucionando. Al comienzo la pintura era aplicada de forma manual, pero en un mundo con tanta demanda por suplir la mano de obra humana ya no es suficiente. No debería se ser sorpresa saber que la Inteligencia Artificial y programación offline y demás han llegado a la pintura industrial y dado lugar a la pintura industrial automatizada.

La automatización de pintura se ve en muchas áreas aparte de la robótica industrial. Hay una investigación de un robot que pinta estructuras para auxiliar a los ingenieros civiles. La investigación reitera que la construcción es una actividad poco organizada y con trabajo físico intenso. En su investigación, Yan (2019) describe el robot de la siguiente manera "El robot consta principalmente de dos secciones, la sección del robot y la sección de control. La sección Robot pinta la pared y la sección de control controla los movimientos verticales y laterales del robot" (pg.3). Se ve que la robótica auxilia en todos los lugares donde hay tareas repetitivas e intensas. Utilizar robots para pintura en ingeniería civil representa menos costos, mayor productividad, y un ambiente con menos riesgo para el personal de pintura.

Otra área donde la pintura automatizada ha entrado es en la aviación, específicamente en China, donde hay una alta demanda por equipos de vuelo. Se consideran las partes de la aeronave como figuras geométricas, como ser cilindros, esferas, superficies cónicas, entre otras. En esta investigación se logró automatizar la pintada de un jet, pero el trabajo se hizo de una buena

manera modular. De manera que este mismo trabajo serviría para la pintura automatizada de aeronaves y cohetes. El funcionamiento del hardware y software del pintado es descrito por Wang et al. (2018, pg.2) de la siguiente manera:

> El sistema de robot de teleplataforma, que integra la teleplataforma de gran espacio e IR, proporciona una solución de automatización para el pintado de grandes aeronaves civiles. La teleplataforma está impulsada por cable y tiene un gran espacio de trabajo con un costo razonable. Sin embargo, la rigidez y precisión de la teleplataforma es débil. El IR se adjunta a la teleplataforma para realizar una pintura local precisa. Durante la pintura en funcionamiento, la teleplataforma se acopla en las paradas planificadas una por una.

Como las palabras que describen el costo del sistema lo dicen, es razonable. Ahorra trabajo y tiempo y puede ser utilizado en otras estructuras y naves. Se observa la automatización de pintura en diversas áreas como construcción y aeronáutica, pero las mayores aplicaciones están en los robots industriales. En el siguiente ejemplo vemos los sistemas de pulverización.

Figura 4 - Ejemplo de Sistema de Pulverización

Fuente: Zhang et al

La siguiente investigación cuenta con un robot rociador industrial de 6 ejes, el objetivo es diseñar el actuador final del robot rociador. Zhang et al. (2020) concluyeron describiendo el actuador de la siguiente manera "El accesorio de pistola de pulverización mecánica diseñado es

relativamente simple, el diseño es razonable y simple de procesar, puede cumplir con los requisitos de sujeción de la pistola rociadora y instalado en el extremo de la brida del robot" (pg.7). Se puede ir viendo el auge de los robots de pintura automatizados.

Entrando al área automovilística se tiene una investigación que utiliza el método de Taguchi para variar parámetros de entrada y automatizar el pintado automovilístico. En esta investigación en particular se utilizó el robot industrial de pintura Fanuc 250ib. Luego un software utiliza el método de Taguchi para detectar el parámetro de entrada con mas peso y variarlo de manera que los costos se reduzcan y aumente la productividad y la calidad del pintado. Chidhambara et al. (2018) llegaron a que "El proceso robótico de pintura aerosol de componentes plásticos se examinó con éxito utilizando DOE de Taguchi. El objetivo era maximizar la DFT. Se consideraron tres factores a saber; Flujo de pintura, dar forma al aire y a la viscosidad" (pg.7)

Como se había dicho, la pintura automatizada con robots industriales se aprovecha al máximo en la industria automovilística. La investigación anterior era uno de muchos ejemplos. A continuación, se tiene otro ejemplo de una investigación con robots industriales automovilísticos, pero con una diferencia. Lo diferente en este proceso es que se utiliza software no solo para automatizar la trayectoria de los robots del pintado, sino también de la soldadura. Esta investigación va un poco más a fondo en el mercado de robots industriales automovilísticos. Incluso, Wang (2020, pg. 4) termina dando un estado de los robots industriales en el mercado automovilístico, y dice lo siguiente:

> En resumen, la sociedad moderna está experimentando un rápido desarrollo económico, la demanda de la gente de los automóviles están aumentando y la demanda del mercado de automóviles está aumentando. La aplicación efectiva de robots industriales en la línea de producción de fabricación inteligente automotriz puede satisfacer el mercado demanda de la industria automotriz. Los robots de proceso utilizan técnicas de fabricación avanzadas y funciones de control inteligente para facilitar la finalización de varios procesos en fabricación de automóviles.

Esta última investigación abre la otra gran área de robots industriales, y es la soldadura automatizada. La soldadura automatizada también tiene un gran auge en la industria

automovilística. Pero pensándolo bien, todos los procesos tienen que unir piezas y mecanismos complejos en cierto punto, y de forma industrial esto se hace con la soldadura.

3.3 SOLDADURA INDUSTRIAL AUTOMATIZADA

Así como los robots industriales y el software avanzado como Inteligencia Artificial encontraron su camino a la pintura industrial, la otra gran área afectada por la automatización de robots y software es la soldadura. Los conceptos son muy similares a la pintura, generar trayectorias con programación offline, simular las trayectorias, y finalmente usar el mismo código fuente con el robot físico. La única diferencia es que el robot en vez de estar pintando esta soldando.

Como primer ejemplo se tiene una investigación en donde el autor compara el rendimiento y eficiencia de la soldadura manual y la soldadura automatizada. Se hicieron las mismas piezas de forma manual y automática y luego se sometieron las piezas a un par de pruebas. A continuación, se tiene el resultado, donde Yan (2019, pg.7) dice lo siguiente:

> El autor compara el rendimiento de soldadura de la soldadura automática con la soldadura manual. Prueba por tracción, prueba de impacto y análisis de microestructura de placas soldadas, y vale la pena determinar el rendimiento de soldadura de soldadura automática. Sin embargo, si no hay una forma estructural adecuada, maduro proceso de soldadura, calidad de punto estable y excelente calidad del personal, no puede obtener la ideal calidad de soldadura incluso con la tecnología de soldadura avanzada.

Se ve que las piezas producidas con soldadura automatizada son casi igual a las piezas producidas de forma manual. Por ende, la soldadura automatizada ayuda en el ahorro de tiempo y aumenta la productividad. Al igual que la pintura, la soldadura automatizada entra en varias áreas, como por ejemplo la aeronáutica.

Hay otra investigación donde se integró el sistema de coordenadas de un robot industrial soldador con equipo industrial durante la producción de tuberías de aeronaves. De hecho, Trushnikov et al. (2018) dicen lo siguiente: “El documento considera la coincidencia de coordenadas del robot de soldadura industrial con industrial equipo de un conjunto de tubería de aeronave utilizando el sistema de retroalimentación de un rastreador láser con un contacto

investigación" (pg.9). La investigación también introduce las partes de una soldadura automática, las cuales se presentan a continuación:

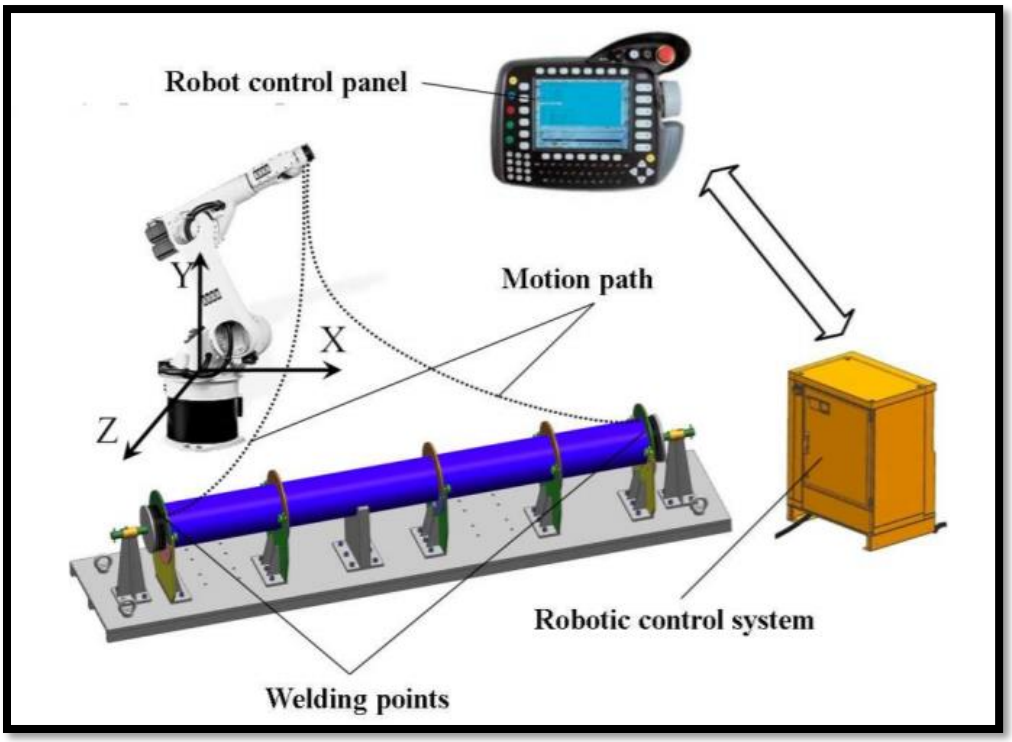

Figura 5 - Partes de una Soldadura Automática

Fuente: Trushnikov et al

De la figura se observa que de hardware se cuenta con el robot industrial de soldadura y los puntos de soldadura. Por el otro lado, el sistema de control robótico es el que toma las decisiones de adónde va a mover al robot. Cuando ya sabe dónde lo quiere mover, solo manda las instrucciones al panel de control del robot para realizar dichos movimientos.

Otra aplicación de la soldadura automatizada se puede ver en una investigación donde se trata con la reparación de culatas de hierro fundido dañadas mediante sistema de soldadura por arco de impresión 3D. Esta investigación aplico la soldadura automatizada por número de capas y se encontró algo importante, que al aumentar el número de capas se reduce la dureza de las piezas. Dentro de las conclusiones de la investigación, cabe destacar lo que Wang et al. (2019, pg.5) resaltaron:

> Aunque la dureza de las piezas de soldadura en diferentes parámetros del proceso de soldadura es diferente, la ley de variación de la dureza a lo largo de la dirección normal

de la interfaz de unión es consistente, y la dureza disminuye gradualmente con el aumento del número de capas.

Definitivamente la soldadura es esencial en muchas áreas de la industria, ya que todo está compuesto por piezas y mecanismos que hay que ensamblar. También es interesante ver como la soldadura puede llegar a afectar las propiedades físicas de las piezas. Con tantas aplicaciones, hay más áreas donde la soldadura automatizada ha jugado un rol importante, y ahora se está viendo en el seguimiento de costura.

Hay una investigación donde se usa una simulación en LABVIEW junto con la técnica de detección de bordes para el seguimiento de trayectoria en costura. La interfaz gráfica del sistema fue desarrollada completamente en LABVIEW. En esa investigación, Parameshwaran et al. (2021, pg.9) describieron el artículo de la siguiente manera:

> Este artículo muestra la simulación para el seguimiento de la costura. Reduce el tiempo de programación en el robot ABB que se utilizará en industrias para la producción en masa. Para cualquier tipo de surco, la técnica de detección de bordes encontrará el coordina, por lo tanto, el sistema de seguimiento de costura es mucho más fácil que en comparación con el método convencional. Los ruidos en la imagen también se reducen debido a la programación VI.

De nuevo, se observa que en el área de la soldadura industrial las simulaciones de trayectoria son comunes, y se varia la entrada o aplicación de la soldadura. Otro ejemplo es de soldadura automatizada se encuentra en una investigación que utiliza 6 robots de soldadura DOF que juntos pueden soldar en todas las posiciones. Se utilizan los parámetros Denavit Hartenberg para asignar coordenadas y el resultado es un robot con 6 ejes de movimiento. A continuación, se muestra una imagen del sistema de coordenadas cinemáticas:

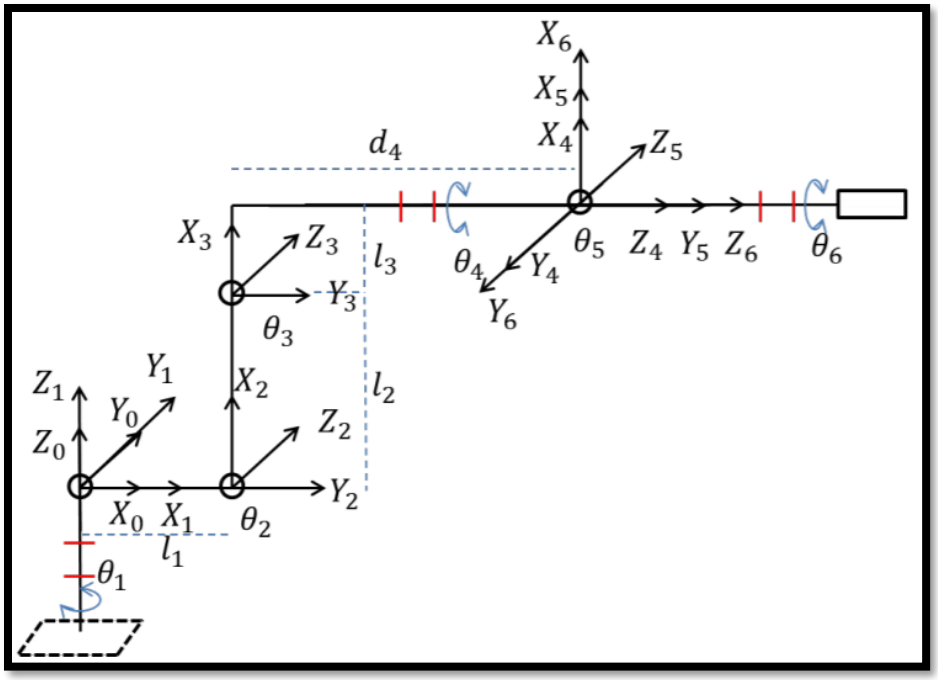

Figura 6 - Sistema de Coordenadas Cinemáticas

Fuente: Muslikhin et al

En la investigacion, Muslikhin et al. (2019, pg.5) describen la planeación de la trayectoria esperada de la siguiente forma:

> La base para la planificación de un manipulador punto a punto (PTP) tiene tres puntos. Punto A (θ1… 6 = 0 90 0 0 0 0), como prefijo luego baja al punto B (θ1… 6 = 0 120 60 0 0 0) así como al punto de inicio del círculo. Finalmente, en punto C (θ1… 6 = 0 52-13 0 0 0), que está a 180 grados del punto A, es decir, estará en el punto después de formar un semicírculo.

En esta descripción de la planeación de la trayectoria se puede apreciar como cada punto en el movimiento esta descrito por los 6 valores de los 6 ejes. Ya se ve que la definición de un semicírculo no es tan intuitiva cuando se cuenta con 6 ejes. Para este punto se puede notar que las aplicaciones de las soldaduras automatizadas son variadas y numerosas. Desde aeronáutica, reparación de culatas, robots de 6 ejes, hasta seguimiento de costura.

Se presentará una última investigación de soldadura automatizada que aborda en el mundo de las industrias de fabricación de láminas de metales. En esa investigación, el software calibrara los parámetros del robot ABB IRB 1520 en base a los valores leídos como ser profundidad

de penetración, ancho del cordón y refuerzo en uniones soldadas en material IS2062 (Grado A). Se utiliza una metodología de respuesta de superficie para encontrar las condiciones óptimas. Kumaran y Raj (2018) afirman que "en este estudio, se encuentra que la Metodología de Superficie de Respuesta es el método que crea un Relación cuantitativa entre la respuesta y los parámetros que deben controlarse para encontrar una condición óptima" (pg.11). Para trabajar con la soldadura se utilizó una simulación en el software Robot Studio. Con esas investigaciones es suficiente para ver el alcance y aplicación que tienen la soldadura y pintura automatizada. Cabe destacar la importancia y peso de las simulaciones en estas investigaciones. De hecho, hay un fenómeno que ha aparecido en un par de las investigaciones vistas anteriormente, y es la simulación de trayectorias de robots.

3.4 SIMULACIÓN DE TRAYECTORIA EN ROBÓTICA

La simulación de trayectoria es la representación gráfica del movimiento cinemático de un objetos u objetos. Estas graficas son extremadamente importantes porque representan el recorrido de un robot industrial. Si se cuenta o se calculan estas trayectorias de antemano se puede predeterminar el camino de un robot. En otras palabras, las trayectorias son los movimientos del robot.

Hay investigaciones que han empujado por mejorar la autenticidad de las simulaciones y hacer que estas se parezcan más a la realidad. Hay una investigación en particular que utiliza el método de Kane para las simulaciones de la dinámica del movimiento de un robot industrial. Utilizar el método de Kane resuelve un problema común en las simulaciones de robot industriales. El problema común es que la estabilidad de un robot industrial no es buena y esto causa que se pierda efecto en la simulación, reduciendo la autenticidad y credibilidad de la simulación. De acuerdo a Luan et al. (2021, pg.6) el método funciona porque:

> El robot giratorio adopta la estructura de marco dividido, la vertical del robot móvil adopta la estructura anidada de dos brazos interior y exterior, y la correa síncrona levanta el brazo exterior para lograr el efecto de dos velocidades. Puede reducir eficazmente los requisitos de altura del espacio de operadores y aumentar la estabilidad.

Nótese que esta investigación ha dado un paso en mejorar las simulaciones de movimiento de robots industriales. Hay otra investigación que busco perfeccionar el algoritmo de movimiento cinemático de un robot industrial. Este algoritmo se encarga de analizar y comparar las múltiples soluciones, o trayectorias, posibles. La simulación se llevo a cabo en el programa MATLAB. He et al. (2020) afirman que "en este artículo se analiza la ecuación cinemática del robot industrial 6-GDL y su cinemática inversa se estudia y mejora el algoritmo" (pg.3). Otra vez se ve como un método o algoritmo es incorporado para mejorar o corregir los problemas de las simulaciones. En este caso si logro mejorarse el movimiento cinemático del robot industrial. Ya que estos problemas son muy conocidos, no es sorpresa que haya mas de una investigación resolviendo o atacando el mismo problema.

En este caso, se tiene otra investigación muy similar, la cual también hace su simulación en MATLAB, y también utiliza un algoritmo de selección inversa en un robot industrial de 6 grados de libertad. Esta investigación también busca optimizar y mejorar la cinemática directa e inversa de un robot industrial. Luyang y Yichao (2020, pg.4) lo describen así:

> En el control de movimiento punto a punto de un robot brazo, se conocen las variables articulares de las seis articulaciones en el punto de partida, y las variables articulares correspondiente al punto final se puede calcular mediante cinemática inversa según la posición final del punto dado. Por lo tanto, la trayectoria de movimiento de cada articulación se puede representar mediante un θ(t) suave función pasando por el punto de inicio y el punto final.

Esta investigación, al igual que la anterior, obtuvo resultados muy positivos que pudieron medirse en la simulación de MATLAB. Cabe mencionar que, con un método adecuado y las herramientas de simulación correcta, se ve que es posible mejorar la simulación de la cinemática de robots industriales.

Ciertas investigaciones hacen integraciones de forma diferente. Una investigación creo un entorno de control virtual, donde se hacen integraciones con modelos 3D de robots industriales. Se busca determinar el espacio de trabajo de un robot industrial. A continuación, se puede apreciar la imagen de la integración del entorno de control con la simulación de movimiento en PLM Siemens NX.

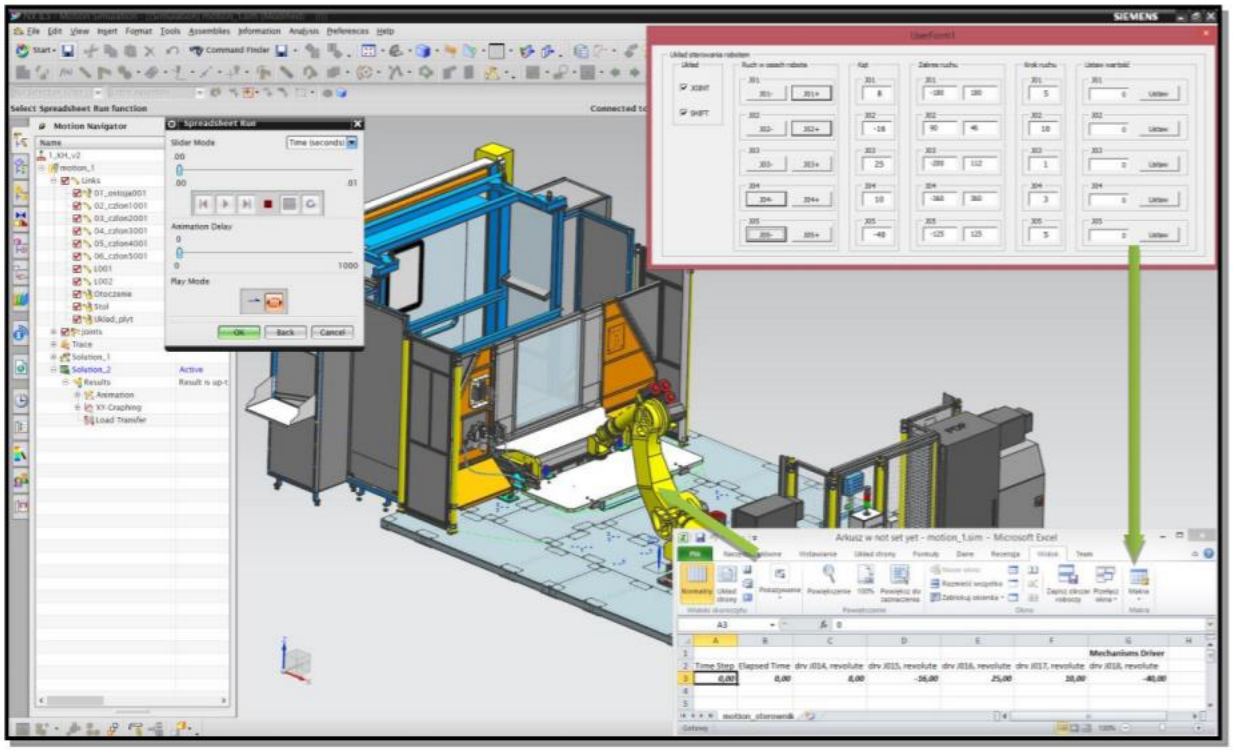

Figura 7 - Integración del Entorno de Control con la Simulación de Movimiento en PLM Siemens NX

Fuente: Herbuś y Ociepka

La imagen muestra el entorno simulado y el control del movimiento del robot industrial. Sin duda muy completa. En esta investigación, Herbuś y Ociepka (2016) concluyeron que "el modelo creado del robot manipulador, preparado para la simulación de movimiento, es el sistema independiente y es totalmente controlable mediante el panel de control virtual y podría utilizarse en cualquier sistema robotizado en producción" (pg.7).

Entre más investigaciones se observan, mas es notable el sin número de avances que las simulaciones de movimiento y trayectoria junto con la programación offline han traído al mundo de la robótica industrial. Tareas que antes hubieran tomado días e incluso semanas por los arreglos físicos de equipo industrial, ahora puede hacerse en segundos con las simulaciones y

realidad virtual. Incluso si se desean hacer cambios, estos pueden hacerse en tiempo real y el entorno industrial puede analizar nuevamente.

IV. METODOLOGÍA

4.1 ENFOQUE

El enfoque de esta investigación es de tipo incremental. Se espera variar un código o librería nueva en C# para que de mejores resultados en la simulación de sistemas de pulverización. La presente investigación tuvo 3 incrementos, los cuales pueden ser descritos de la siguiente manera.

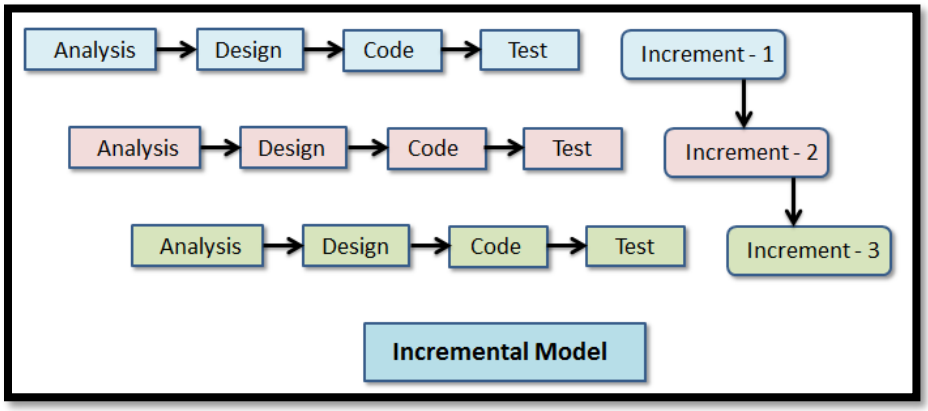

Figura 8 - Modelo Incremental

Fuente: Educba

El Incremento 1 fue un producto para dibujar la trayectoria de un robot solamente cuando el efector final este encendido. En este incremento se empezó a trabajar con las librerías de ACE, donde se leyeron datos del robot y agregaron variables globales visibles. Al final del Incremento 1 se contaba con una función que recibía como parámetro cualquier robot y dibujaba su trayectoria. Esto quiere decir que se podía integrar fácilmente y usar con varios robots al mismo tiempo.

El Incremento 2 fue un producto para simular un sistema de pulverización, dibujando conos elípticos a través de la trayectoria. Para este Incremento ya se contaba con el conocimiento de C# en ACE para dibujar los puntos, pero se ocupaba un modelo matemático para calcular la posición de estos puntos. En el Incremento 2 se derivó una fórmula de posición de las partículas de pintura. Finalmente, se creó una función que recibía como parámetro un punto en el espacio tridimensional y a partir de ahí simular el rociado de pintura. Esto, integrado con el Incremento 1 brinda rociados de pintura a través de la trayectoria solamente cuando el efector final este encendido.

El Incremento 3 fue un producto para simular un sistema de pulverización, tomando en consideración el ángulo del efector final. Se agrego algo que hacia falta a la simulación anterior. El análisis del modelo matemático estaba bien, pero siempre consideraba que el rociado iría hacia abajo. En la vida real, el ángulo del efector final de un robot industrial puede estar en casi todas las direcciones del espacio tridimensional. Para esto se paso los puntos calculados en el Incremento 2 por una matriz de rotación que toma en cuenta el rollo, guiñada y cabeceo del robot. De esta manera el rociado queda en dirección del efector final.

4.2 VARIABLES DE INVESTIGACIÓN

La variable principal a medir en esta investigación es las simulaciones de sistemas de pulverización en Ace. En otras palabras, las posiciones calculadas de las partículas.

La posición de las partículas está afectada por las siguientes variables independientes configurables: número de iteraciones, tiempo entre iteraciones, el número de particiones para el cono elíptico, ángulo phi del cono elíptico, y los tiempos a evaluar la posición de las partículas.

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

ACE es un software de simulación de procesos industriales diseñado por Omron. En ACE es posible agregar bandas, piezas y robots industriales para definir un proceso industrial. Se puede hacer programación offline con los robots industriales. Se programa el movimiento de los robots y finalmente se corre la simulación y se alimenta el proceso con piezas para ver cómo funciona. Es muy fácil detectar y corregir un error, al igual que hacer cambios en alguna parte del proceso. ACE es tan completo, que permite escribir código en C# para extender sus funcionalidades y características.

C# es un lenguaje de programación de propósito general, mayormente utilizado para la programación orientada a objetos. En esta programación se definen clases y objetos que mapean a la realidad y se definen propiedades y métodos sobre tales objetos. El software final funciona con instancias de estos objetos interactuando entre sí. En esta investigación se creará un objeto nuevo capaz de dibujar la trayectoria de pintura y soldadura. Finalmente, ACE va a interactuar con este objeto nuevo para realizar la integración.

4.4 METODOLOGÍA DE ESTUDIO

Tomando en cuenta cómo funciona ACE y C#, se analizará la comunicación y forma de dibujar estas trayectorias de soldadura y pintura. Actualmente se cuenta con ACE, donde ya hay objetos de programación que representan a los robots y tienen su código de movimiento. Se creará un nuevo objeto y producto en C# que se encargará de leer los objetos de ACE y usar las librerías para dibujar la trayectoria y simular el rociado de pintura.

Como primer parte se dibuja la trayectoria del robot solamente cuando el efector final este encendido. A continuación, se describe el algoritmo central para graficar la trayectoria de cualquier robot en ACE. Este funciona a base de dos variables programables, las cuales son el número total de iteraciones y el tiempo entre cada iteración. Se harán iteraciones o ciclos, cada uno separada por un tiempo y el total definido por las variables. Durante cada una de estas iteraciones, el algoritmo realiza lo siguiente:

1. Leer el estado del efector final del robot, si este encendido continuar con el paso 2 sino ir al paso 5.
2. Utilizar la función de ACE para obtener la posición actual del efector final.
3. Utilizar función de ACE para convertir esa posición a un Punto en el espacio tridimensional de ACE.
4. Utilizar funciones de ACE para agregar una variable global visible, el valor de esta variable es el valor obtenido en el paso 3.
5. Esperar el tiempo configurable entre cada iteración.

Con este algoritmo se programó un producto (función/librería) que recibe como parámetro el robot en particular del cual se quiere dibujar la trayectoria. Es decir, esta función podría utilizarse con cualquier tipo de robot.

Para este punto se tiene una función con la habilidad de simular la trayectoria de cualquier robot solamente cuando el efector final este encendido. Para dar más valor agregado al producto de la investigación, se decidió agregar una segunda parte a la simulación de trayectoria. Se agrega la simulación del sistema de pulverización.

Ahora hay que ver como se dibuja un sistema de pulverización, en este caso de pintura. En este algoritmo hay un par de variables configurables: la presión del robot, la masa de las partículas de pintura, el diámetro del efector final, el número de particiones para el cono elíptico, el ángulo phi del cono elíptico y los tiempos a evaluar la posición de las partículas. Dado un punto en el espacio tridimensional, el siguiente algoritmo dibuja la simulación de un sistema de pulverización.

1. Usar el punto actual como el origen
2. Utilizar el modelo matemático para calcular la fórmula de posición de las partículas
3. Evaluar la fórmula de posición en los tiempos predefinidos
4. Rotar los puntos de acuerdo al rollo, guiñada y cabeceo del robot
5. Agregar estos puntos como variables globales en ACE

Con este algoritmo se programó un producto (función/librería) que recibe como parámetro el robot en particular sobre el cual se quiere simular el rociado. La función simplemente dibuja un rociado de pintura sin necesidad de que el robot se esté moviendo o que el efector final este encendido.

Finalmente, para obtener la simulación final del rociado de pintura industrial en ACE, basta con combinar los dos algoritmos anteriores. El primer algoritmo ya nos da la habilidad de ir agarrando cada punto del efector final solo cuando este encendido. Si usamos cada uno de esos puntos como entrada para el segundo algoritmo, tendremos rociados en varios puntos de la trayectoria. Con el producto final (función/librería) se recibe como parámetro el robot y su efector final y simula el rociado de pintura configurable.

Se observa que esta investigación funciona con 3 grandes partes integradas, las cuales representan las iteraciones del enfoque incremental. Las 3 partes son las siguientes:

1. Producto (función/librería) en C# que dibuja la trayectoria de un robot solamente cuando el efector final este encendido
2. Producto (función/librería) en C# que simula un rociado de pintura sobre un punto origen
3. Producto (función/librería) en C# que simula el rociado de pintura sobre la trayectoria de un robot solamente cuando el efector final este encendido

4.5 CRONOGRAMA DE ACTIVIDADES

A continuación, se muestra un Diagrama de Gantt con las actividades para llevar a cabo esta investigación.

Actividad	Abril 2021				Mayo 2021				Junio 2020			
	1	2	3	4	1	2	3	4	1	2	3	4
1. Desarrollar un producto (función/librería) que dibuja la trayectoria de un robot solamente cuando el efector final este encendido												
2. Desarrollar un producto (función/librería) en C# que simula un rociado de pintura sobre un punto origen												
3. Desarrollar un producto (función/librería) en C# que simula el rociado de pintura sobre la trayectoria de un robot solamente cuando el efector final este encendido												

Tabla 1 - Diagrama de Gantt

V. RESULTADOS Y ANÁLISIS

Al final se logró desarrollar un producto más completo del que se había planeado. El producto original era uno que fuera capaz de graficar la trayectoria de cualquier robot en ACE solamente cuando el efector final estuviera encendido. Completar esta parte no fue tan sencilla, ya que requirió entender las librerías y funciones de ACE. Con estas se pudo obtener el punto actual del efector final del robot, el estado del efector final, también se interactuó con las variables globales de ACE (las cuales sirvieron para agregar puntos en el espacio tridimensional).

En cuanto al modelado matemático, se derivó una fórmula de posición de las partículas en el tiempo. Para la simulación del sistema de pulverización, se estudió el rociado de pintura industrial, en donde las partículas salen en trayectoria con forma de cono elíptico en el espacio tridimensional. Además, la densidad de partículas disminuye a medida se alejan del efector final del robot. A continuación, se presenta la derivación de la fórmula utilizada.

Se comienza con la definición de fuerza y sus dos fórmulas:

$$F = m * a = P * A$$

Donde:

m es la masa de las partículas de pintura

a es la aceleración de las partículas de pintura

P es la presión regulable del robot de pintura

A es el área donde se distribuye la presión

El área donde se distribuye la presión es el efector final. Por ende, la formula del área es la siguiente:

$$A = \frac{\pi * d^2}{4}$$

Donde:

d es el diámetro del efector final de donde salen las partículas

Con esto se puede despejar para la aceleración de las partículas de pintura. La fórmula seria la siguiente:

$$a = \frac{P * \pi * d^2}{4 * m}$$

Con esto se obtiene la magnitud del vector de aceleración de las partículas al ser rociadas. El siguiente paso sería descomponer la aceleración de en sus componentes tridimensionales. El análisis se presentará en el caso en el que el rociado va en dirección negativa en el eje Z. A continuación, vemos una imagen de la partícula a analizar y su diagrama de cuerpo libre, el cual incluye la gravedad y aceleración del rociado.

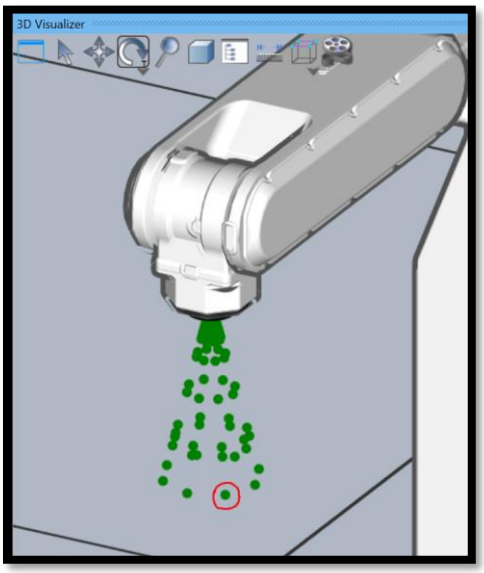

Figura 9 - Caso con Rociado Hacia Abajo y Partícula a Analizar

Fuente: Propia

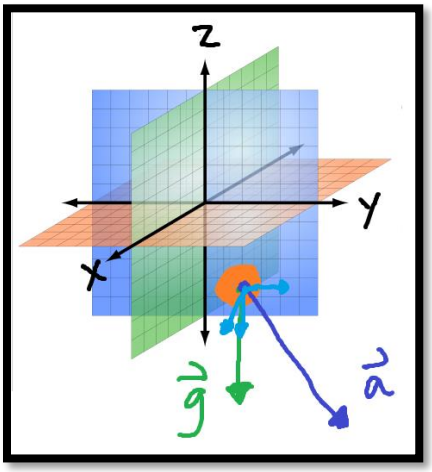

Figura 10 - Diagrma de Cuerpo Libre de Partícula a Analizar

Fuente: Propia

Para descomponer el vector de aceleración del rociado en sus 3 componentes, se utilizaron las fórmulas de conversión de ecuaciones esféricas a cartesianas. Esto es conveniente ya que cuando se dibuja el cono elíptico, se definen particiones, las cuales crean un angulo tetha en el plano XY, y un ángulo phi desde el eje Z. De esta manera, las fórmulas de descomposición de aceleración de rociado son las siguientes:

$$a_x = a * sen(\varphi) * \cos(\theta)$$
$$a_y = a * sen(\varphi) * \text{sen}(\theta)$$
$$a_z = a * cos(\varphi)$$

Ya con el vector de aceleración en componentes, se asume movimiento uniformemente acelerado en los 3 ejes y que las partículas parten del reposo y del origen. De esta manera, la ecuación clásica de posición en movimiento uniformemente acelerado quedaría de la siguiente manera:

$$\mathbf{r} = \frac{1}{2}\mathbf{a}t^2 + \mathbf{v_o}t + \mathbf{r_0}$$
$$\mathbf{r} = \frac{1}{2}\mathbf{a}t^2$$

Finalmente, usamos la ecaucion en cada dimension y a la aceleracion en Z se le resta la gravedad, y con esto, se llega a la formula final de posicion de las particulas de pulverizacion en function del tiempo.

$$\boldsymbol{r_x}(t) = \frac{1}{2}\boldsymbol{a_x}t^2$$
$$\boldsymbol{r_y}(t) = \frac{1}{2}\boldsymbol{a_y}t^2$$
$$\boldsymbol{r_z}(t) = \frac{1}{2}(\boldsymbol{a_z} - \boldsymbol{g})t^2$$

Este analisis fue suficiente para brindar las formulas que sirven en todos los casos, pero todavia falta un ultimo paso. Los puntos del rociado deben de estar en la direccion del efector final. Para lograr este efecto, se utiliza la matriz rotacional en base a los valores de rollo, guiñada y cabeceo del robot. A continuacion, se peresenta la matriz:

$$R = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

Figura 11 - Matriz Rotacional con Rollo, Guiñada y Cabeceo

Fuente: LaValle

Utilizando esta matriz se pueden ajustar los puntos calculados para que el rociado quede siempre en dirección del efector final. A continuación, se presentan dos imágenes donde el robot ha movido el efector final y podemos ver que el rociado siempre va con los ángulos correspondientes.

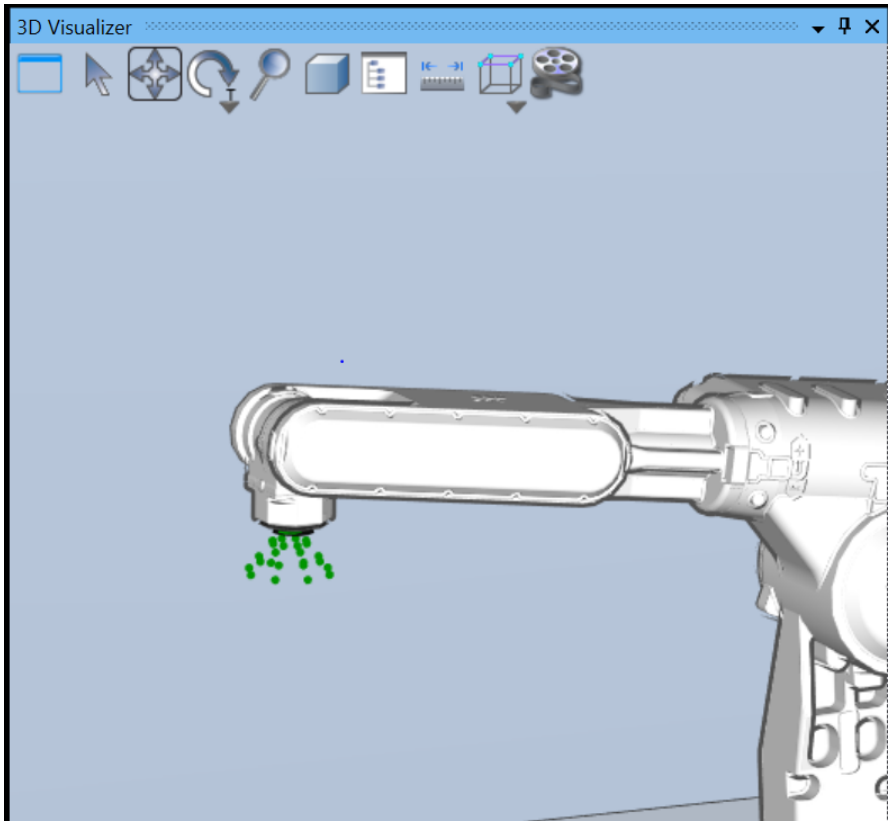

Figura 12 - Ejemplo de Rociado con Efector Final hacia Abajo

Fuente: Propia

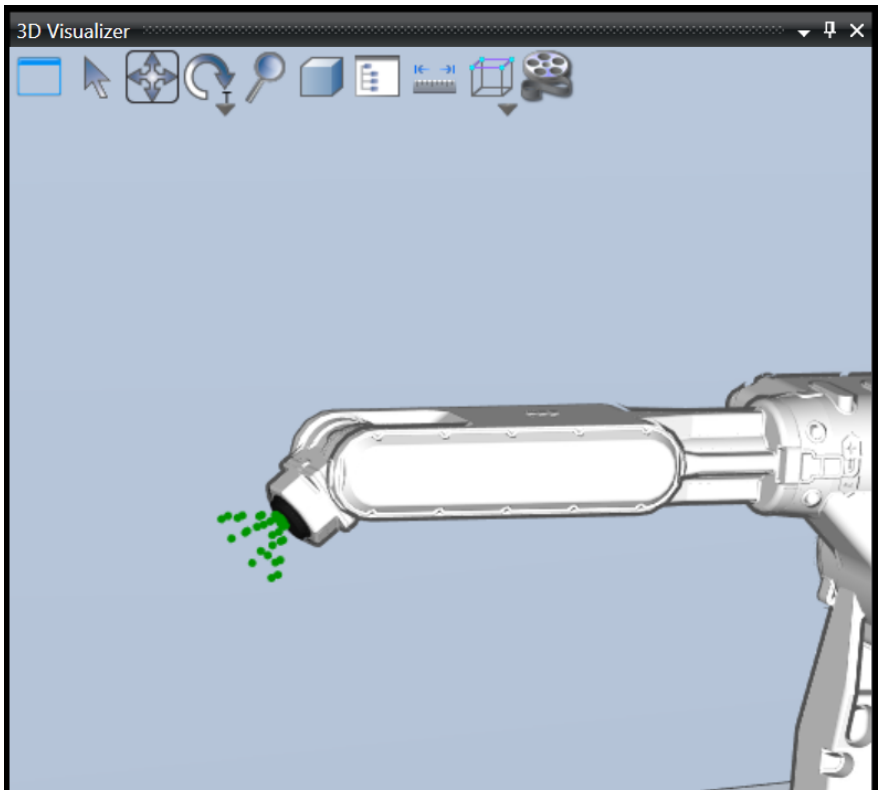

Figura 13 - Ejemplo de Rociado con Efector Final a un Ángulo

Fuente: Propia

Con esta parte se concluye el modelo matemático para la simulación de sistemas de pulverización, pero todavía falta integrar esto con los algoritmos anteriores. A continuación, se describe la combinación y algoritmo final para la simulación de sistemas de pulverización en ACE.

1. Eliminar todas las variables globales anteriores que se hayan utilizado en la simulación de rociado.
2. Correr iteraciones. Dentro de cada iteración:
    1. Se verifica que el efector final del robot este encendido
    2. Se agarra el punto actual del efector final
    3. Se utiliza este punto como el origen en los calculos
    4. Se utiliza el modelo matemático para calcular la formula de posición de las partículas
    5. Se evalua la formula en varios tiempos predefinidos
    6. Rotar los puntos de acuerdo al rollo, guiñada y cabeceo del robot
    7. Se agregan estos puntos como variables globales visibles en ACE

Tambien se presenta un diagrama de flujo del algoritmo.

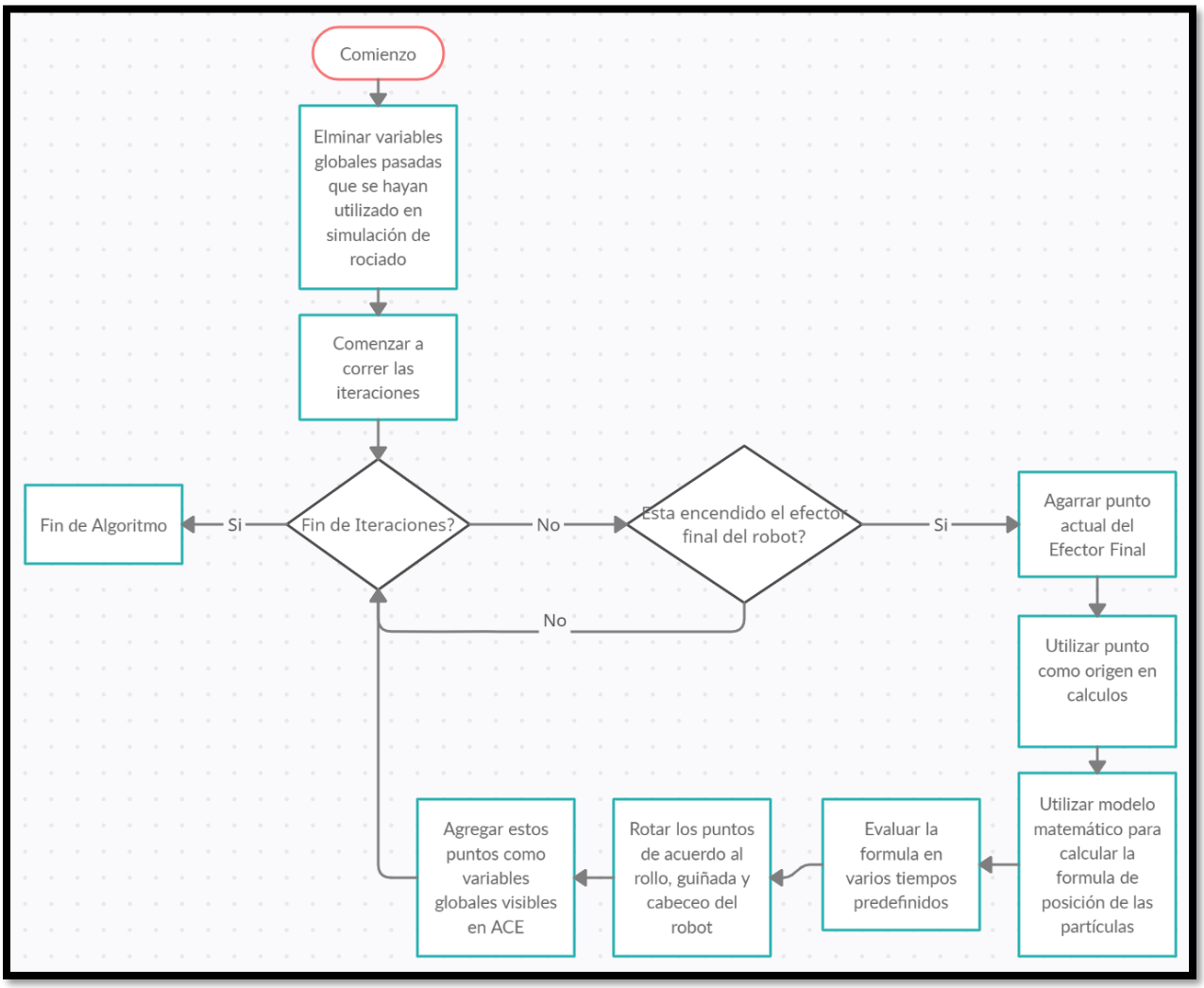

Figura 14 - Diagrama de Flujo del Algoritmo Final

Fuente: Propia

Para probar la primera parte que grafica la trayectoria, se creó un proyecto de prueba en ACE y se agregó un robot i4. Se agregó código en V+ para mover al robot entre 3 puntos y encender el efector final entre los puntos 1 y 2. Finalmente se llamó la función desarrollada brindando el robot y su efector final como parámetro. Para este punto solo se pueden configurar el número de iteraciones y el tiempo entre cada iteración. Esto afecta directamente la forma en que se vera la trayectoria. Por ejemplo, si el tiempo entre iteraciones es muy alto los puntos quedaran dispersos y ya no parecerá trayectoria. A continuación, se observan 2 vistas de la trayectoria del robot i4 en ACE solamente cuando el efector final este encendido.

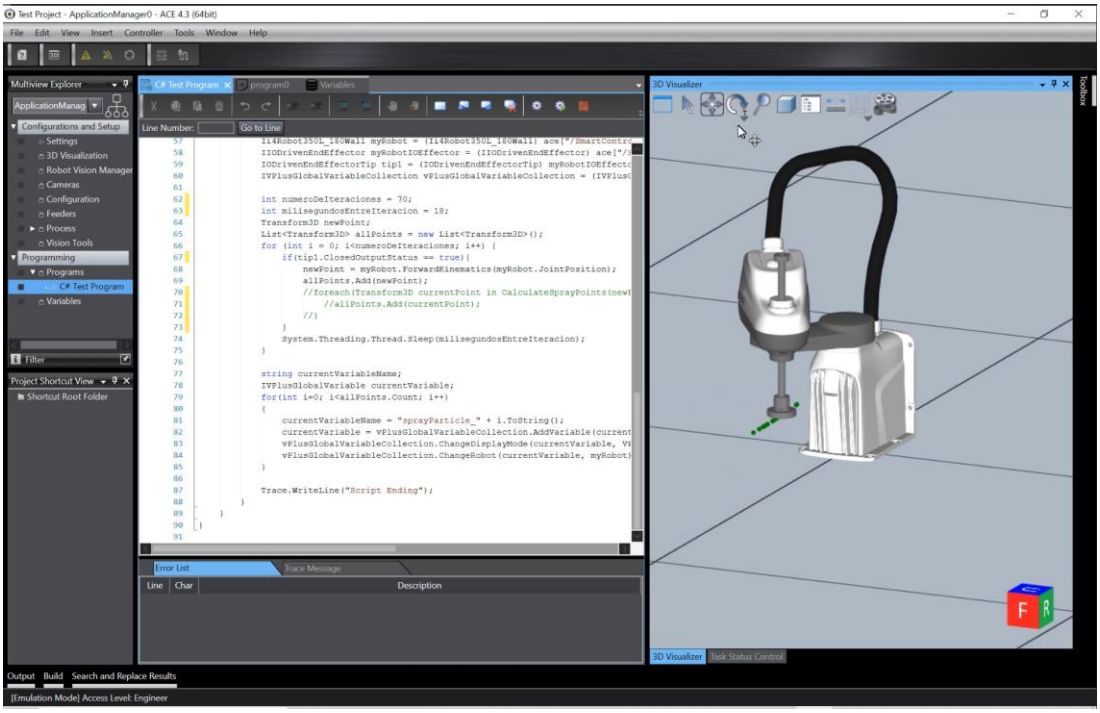

Figura 15 - Vista 1 de Trayectoria de Robot i4 en ACE

Fuente: Propia

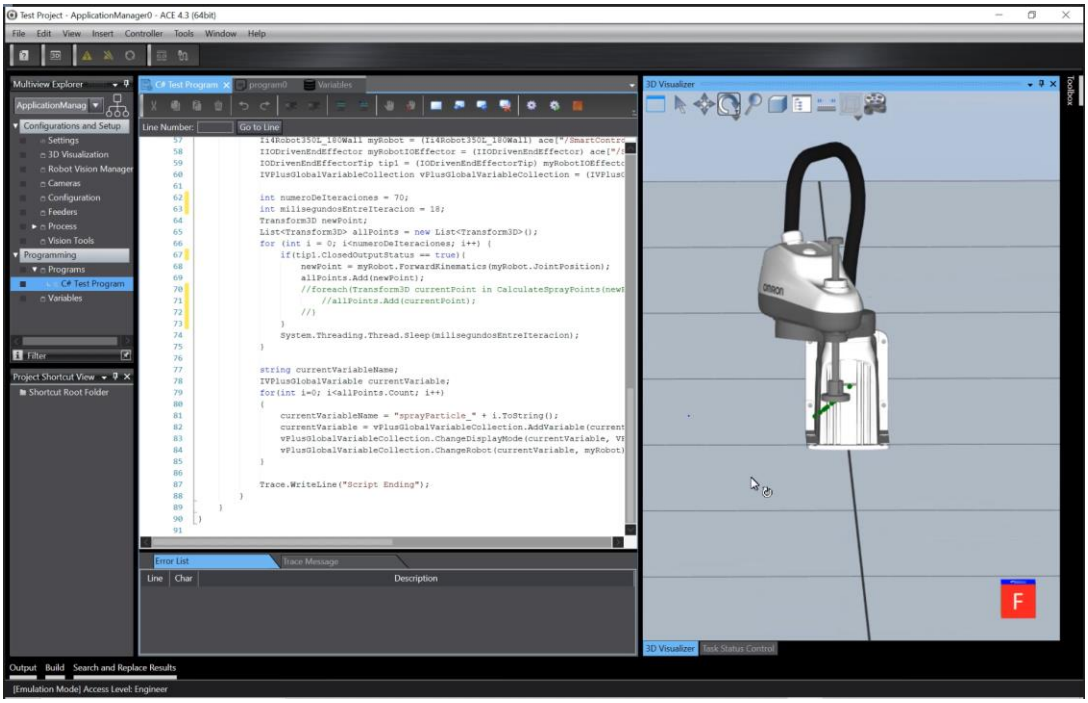

Figura 16 - Vista 2 de Trayectoria de Robot i4 en ACE

Fuente: Propia

Luego para probar la segunda parte del rociado de pintura, se creó un proyecto de prueba en ACE donde se agregaron 3 robots, un i4 y 2 viper. Se mando a llamar la función que dibuja el rociado de pintura sobre los 3 robots. A continuación, se presentan los resultados:

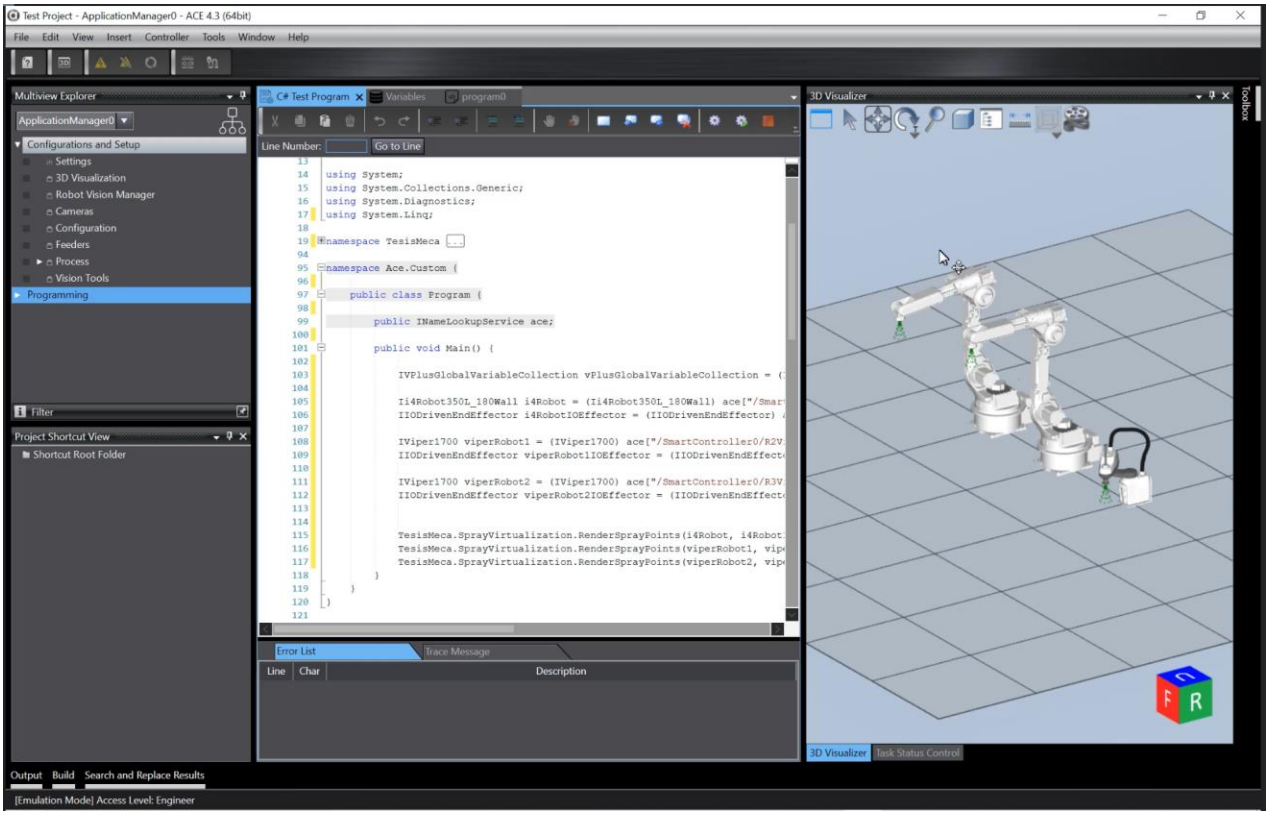

Figura 17 - Vista 1 Simulación de Rociado en 3 Robots en ACE

Fuente: Propia

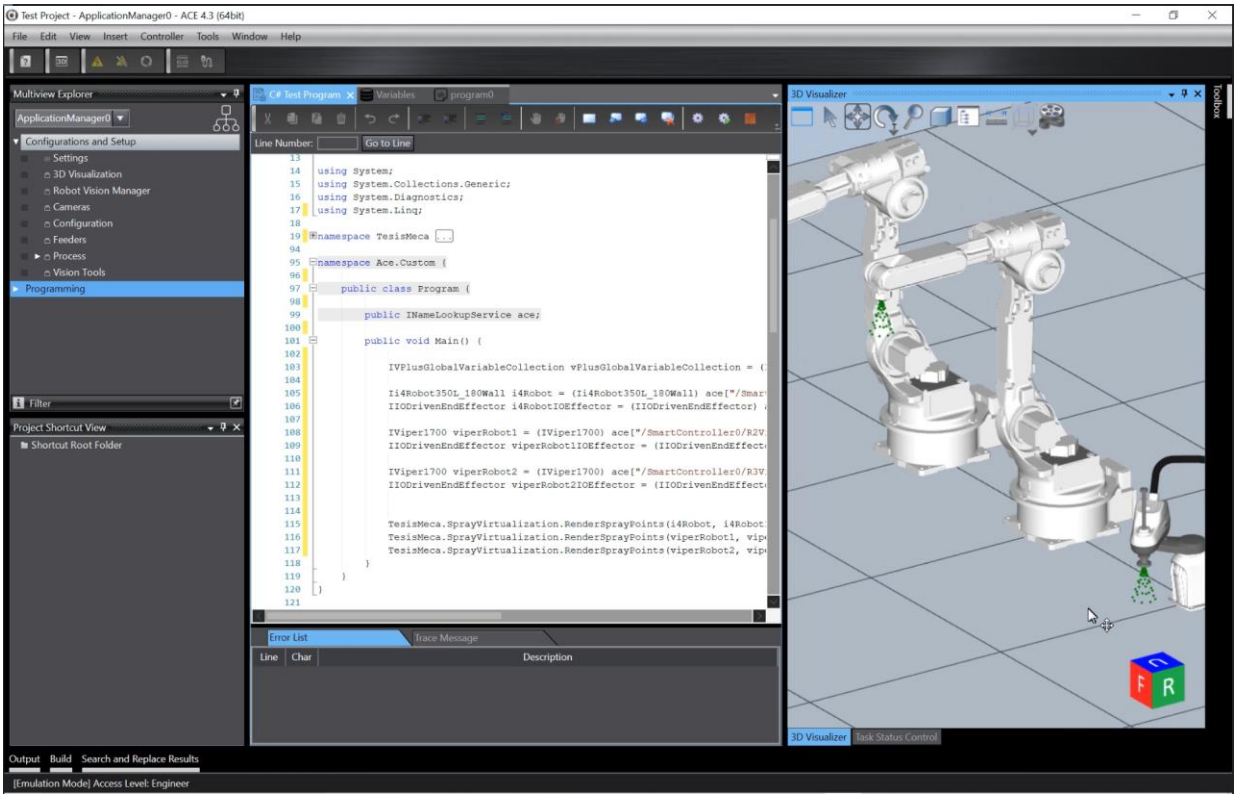

Figura 18 - Vista 2 Simulación de Rociado en 3 Robots en ACE

Fuente: Propia

Para probar el algoritmo final se creó un proyecto en ACE donde solo se agregó un robot viper. Luego se programó el robot en V+ para moverse en entre 3 puntos y encender el efector final entre los puntos 1 y 2. Finalmente se llamó la función codificada con el robot agregado y se puede observar la simulación del rociado de pintura industrial.

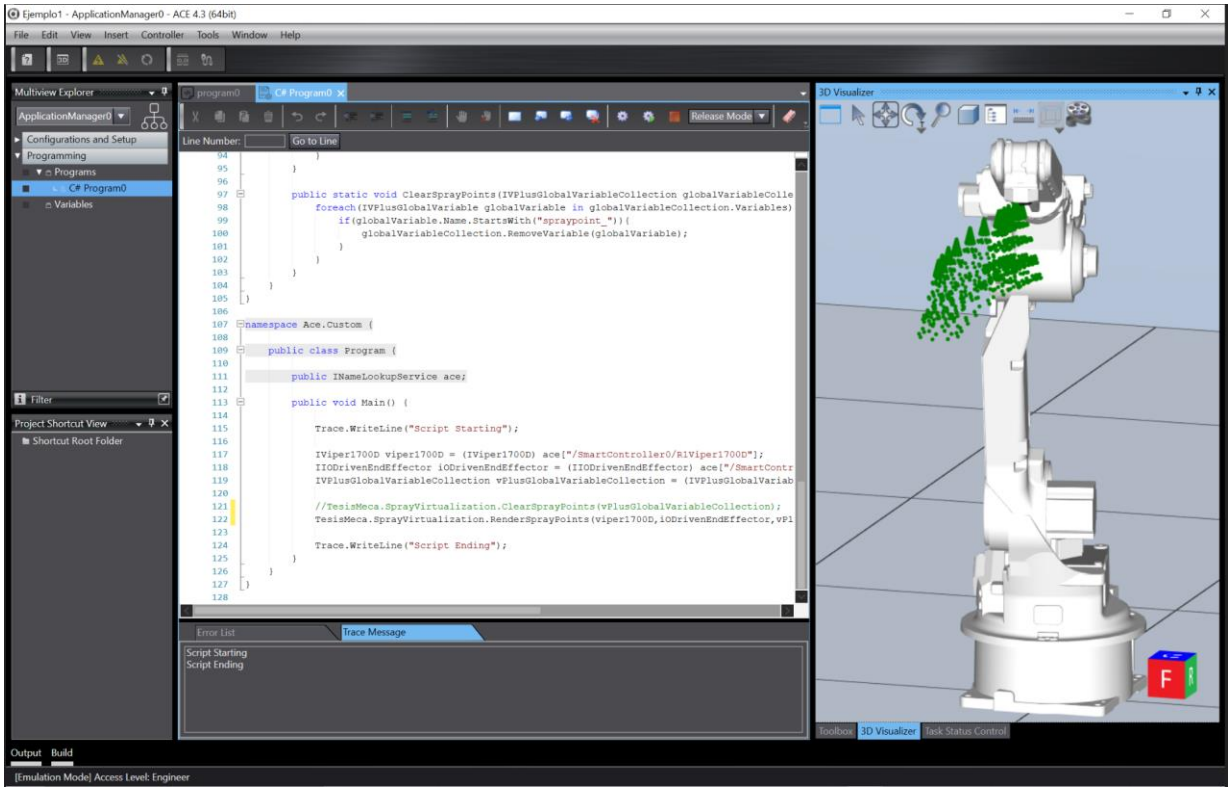

Figura 19 - Simulación de Rociado de Pintura Industrial en ACE

Fuente: Propia

Se puede observar como a través de la trayectoria del robot se van haciendo varios rociados de pintura y el efecto final son varios conos elípticos. Para este punto se pueden configurar todas las variables independientes y apreciar la forma en que cada una afecta la dependiente, o la simulación del rociado de pintura. A continuación, describimos la relación de cada variable independiente con la dependiente.

Variable Independiente	Efecto Causado Sobre Variable Dependiente
Número de Iteraciones	El número de iteraciones afecta la calidad de la imagen del rociado. Si solo hubiera una iteración solo se vería un cono, pero si fueran demasiadas el proceso sería bien pesado y después de un punto agregar más iteraciones no hace diferencia al ojo humano. Un número medio crea el efecto de rociado y no hace que el proceso sea pesado.
Tiempo Entre Iteraciones	Lo mismo sucede con el tiempo entre iteraciones. Si este tiempo fuera muy pequeño, aunque hubiera varias iteraciones, estarían tan cerca entre sí que solo parecería que hay un rociado para el ojo humano. Por el otro lado, si este valor es muy grande los rociados quedarían muy separados y no se vería real. De nuevo, un numero media deja los rociados cerca y sin cargar el proceso.
Número de Particiones del Cono Elíptico	El número de particiones del cono elíptico afecta la forma en que se ve el rociado. 8 es un número que lo hace ver como cono y no es muy cargado a la vez.
Angulo Phi del Cono Elíptico	El ángulo phi del cono elíptico ajusta la apertura del cono elíptico directamente.
Tiempo	Los tiempos en los que se evalua la función de posición de las partículas afectan directamente la imágen. Escoger los valores adecuados crea el efecto de densidad de particulas.

Tabla 2 - Efectos de Variables Independientes Sobre Dependiente

Este es un buen punto de partida en cuanto al modelo matemático y simulación, pero tomando en consideración que esta es una investigación de tipo incremental, siempre es bueno dejar los caminos y mejores para los futuros incrementos.

Primero hay cuatro variables importantes que harían que esta simulación fuera mas realista, las cuales son: resistencia del aire, flujo masico, viscosidad, y espesor de capa. La resistencia del aire y viscosidad tendrían que incluirse en la derivación de las fórmulas de posición de las partículas, ya que afectan la aceleración de las partículas. Esto haría que las fórmulas fuera más exacta y realistas.

El espesor de capas es importante porque es la forma en que se mide y define la calidad de la pintura industrial. Para incorporarla en el algoritmo actual, primero se debe de calcular el área que cubre un rociado de pintura. También se define o determina el espesor que un solo

rociado aportaría, puede depender de la viscosidad. Se incorporarían numero de rociados en el algoritmo, y cada iteración iría sumando un valor de espesor por rociado sobre el área que las partículas tocan.

Finalmente, se obtuvieron los efectos de las variables dependientes, pero no se obtuvo una manera de medir la calidad de la variable dependiente. Es decir, como se puede saber si el rociado generado es uno de calidad o no. Una forma de contestar esta pregunta es con el mismo método de reconocimiento facial, el cual guarda y luego compara valores como: la distancia entre ojos y nariz, o distancia entre nariz y boca, longitud de boca, etc. En el caso del rociado, primero se tendría que definir valores o porcentajes que hagan que un cono elíptico se veo como un sistema de pulverización de calidad. Por ejemplo, la distancia de la última partícula con el origen, o la densidad de partículas cerca del efector final. Luego habría que calcular esos valores sobre los puntos generados y compararlos con los valores esperados de calidad.

VI. CONCLUSIONES

En cuanto al original y primer objetivo, se puede decir que este se cumplió en su totalidad. Vemos que la trayectoria simulada en ACE en C# de cualquier robot cuando el efector final este encendido es afectado por el número de iteraciones y el tiempo entre cada una de estas iteraciones. Ajustar estos valores ajustara la suavidad de la curva. Pero con un tiempo de alrededor de 100 milisegundos entre cada iteración y 30 iteraciones fue suficiente para obtener trayectorias satisfactorias en las pruebas realizadas. Un punto de mejora en el futuro seria variar el número de iteraciones en base a la distancia de la trayectoria. Ya que una distancia pequeña naturalmente requiere menor número de iteraciones y por el otro lado, un numero de iteraciones bajo sobre una distancia grande puede romper la trayectoria y hacer que aparezcan puntos dispersos en el espacio, lo cual es un efecto no deseado.

En cuanto al objetivo que surgió durante la investigación, vemos que se logro simular los sistemas de pulverización de manera exitosa. Los puntos en el espacio del cono elíptico si eran afectados de la manera esperada por las variables independientes. Se encontró que con 30 iteraciones, con 18 milisegundos entre ellas, con 8 particiones del cono elíptico, con un ángulo phi de 30 grados y evaluando la función de posición de las partículas en los tiempos 0.5,1.0,1.5,2.0,2.3 simula la pulverización de manera correcta.

Finalmente, se llegó a un buen punto de partida en el modelo matemático para la simulación de un sistema de pulverización. Se logro descomponer la aceleración en el espacio tridimensional. Y finalmente llegar a ecuaciones de posición del proyectil en las 3 dimensiones, tomando en cuenta la aceleración de la gravedad. Este modelo es un buen punto de partida a una simulación más completa. Variables importantes que se recomienda agregar en futuras iteraciones serian: flujo másico, viscosidad, espesor por rociado y medición de calidad de variable dependiente.

VII. RECOMENDACIONES

Para que el código en C# sirviera en conjunto con ACE, fue necesario utilizar librerías y funciones propias de ACE. Estas se usan por ejemplo para leer la posición de los robots, o agregar las partículas que son puntos en las variables globales de ACE. Aunque el programador tenga todo el conocimiento de C#, la documentación de ACE en si es muy escasa. Es por eso que una recomendación es dedicar una parte del tiempo a estudiar ACE o buscar una capacitación del software con alguien que lo sepa manejar. Saber usar el programa y entender sus conceptos ayudara a la hora de programar una solución personalizada en ACE.

BIBLIOGRAFIA

Shao, J. (2021). Robot Path Planning Method Bases on Genetic Algorithm. *Journal of Physics: Conference Series, 1881*(022046), 3. https://doi.org/10.1088/1742-6596/1881/2/022046

Rafsanjani, Wibawa, I., & Ekaputri, C. (2019). Speed and steering control system for self-driving car prototype. *Journal of Physics: Conference Series, 1367*(012068), 8. https://doi.org/10.1088/1742-6596/1367/1/012068

Fang, X., & Wang, L. (2020). Nuclear Power Plant Operator Auxiliary Robot System. *IOP Conf. Series: Materials Science and Engineering, 768*(022055), 4. https://doi.org/10.1088/1757-899X/768/2/022055

Talli, A., & Meti, V. (2020). Design, simulation, and analysis of a 6-axis robot using robot visualization software. *IOP Conf. Series: Materials Science and Engineering, 872*(012040), 3-8. https://doi.org/10.1088/1757-899X/872/1/012040

Sabri, M., Fauzi, R., Fajar, M., Geubrina, H., & Sabri, F. (2021). Model and simulation of arm robot with 5 degrees of freedom using MATLAB. *IOP Conf. Series: Materials Science and Engineering, 1122*(012032), 9. https://doi.org/10.1088/1757-899X/1122/1/012032

Holubek, R., Ružarovský, R., Velíšek, K., & Janíček, M. (2021). Novel trend in case study of industrial robot programming and production system design using the virtual reality. *IOP Conf. Series: Materials Science and Engineering, 1009*(012023), 5-6. https://doi.org/10.1088/1757-899X/1009/1/012023

Xu, Z., & Liu, Y. (2019). ABB Robotic Arm Offline Programming System. *Journal of Physics: Conference Series, 1267*(012064), 7. https://doi.org/10.1088/1742-6596/1267/1/012064

Thevara, D., & Vasanth, C. (2018). Application of photogrammetry to automated finishing operations. *IOP Conf. Series: Materials Science and Engineering, 402*(012025), 3-13. https://doi.org/10.1088/1757-899X/402/1/012025

Liu, Z., Chen, J., Mei, Z., & Li, C. (2020). ROS-based robot offline planning simulation system. *IOP Conf. Series: Materials Science and Engineering, 711*(012002), 5. https://doi.org/10.1088/1757-899X/711/1/012002

Golda, G., & Kampa, A. (2017). Manipulation and handling processes off-line programming and optimization with use of K-Roset. *IOP Conf. Series: Materials Science and Engineering, 227*(012050), 3. https://doi.org/10.1088/1757-899X/227/1/012050

Jayaraj, A., & Divakar, H. (2018). Robotics in Construction Industry. *IOP Conf. Series: Materials Science and Engineering, 376*(012114), 3. https://doi.org/10.1088/1757-899X/376/1/012114

Wang, L., Chen, L., Shao, Z., Zhang, Z., & Du, L. (2018). Stop Planning of Teleplatform for Large Civil Aircraft Painting with Industrial Robot. *IOP Conf. Series: Materials Science and Engineering, 394*(032064), 2. https://doi.org/10.1088/1757-899X/394/3/032064

Zhang, C., Xu, X., Wang, G., & Men, J. (2020). Design of spraying robot end actuator and experimental study on motion trajectory simulation. *IOP Conf. Series: Materials Science and Engineering, 740*(012044), 7. https://doi.org/10.1088/1757-899X/740/1/012044

Chidhambara, K., Shankar, B., & Vijaykumar. (2018). Optimization of Robotic Spray Painting process Parameters using Taguchi Method. *IOP Conf. Series: Materials Science and Engineering, 310*(012108), 7. https://doi.org/10.1088/1757-899X/310/1/012108

Wang, W. (2020). Applied Research of Industrial Robots in Automotive Intelligent Manufacturing Production Line. *Journal of Physics: Conference Series, 1550*(042061), 4. https://doi.org/10.1088/1742-6596/1550/4/042061

Yan, Z. (2019). Discussion on the welding performance of automatic welding of industrial robot and manual welding. *IOP Conf. Series: Materials Science and Engineering, 637*(012004), 7. https://doi.org/10.1088/1757-899X/637/1/012004

Trushnikov, V., Grishin, M., & Pavlov, P. (2018). Matching a Welding Robot Coordinate System With Technological Equipment During the Assembly of Aircraft Pipes. *IOP Conf. Series: Materials Science and Engineering, 194*(022041), 3-9. https://doi.org/10.1088/1755-1315/194/2/022041

Wang, H., Li, C., Ding, Y., Li, J., & Chen, S. (2019). Experimental study on repairing of damaged cast iron cylinder heads by 3D printing arc welding system. *IOP Conf. Series: Materials Science and Engineering, 474*(012032), 5. https://doi.org/10.1088/1757-899X/474/1/012032

Parameshwaran, R., Maheswari, C., Nithyavathy, N., Govind, R., Selvakumar, N., Dharshan, V. & Vasanth, M. (2021). LABVIEW Based Simulation on Welding Seam Tracking Using Edge Detection Technique. *IOP Conf. Series: Materials Science and Engineering, 1055*(012026), 9. https://doi.org/10.1088/1757-899X/1055/1/012026

Muslikhin, Irmawati, D., Arifin, F., Nasuha, A., & Hasanah, N. (2019). Kinematics Simulation of Welding Manipulator Based on V-REPPRO EDU. *Journal of Physics: Conference Series, 1413*(012002), 2-5. https://doi.org/10.1088/1742-6596/1413/1/012002

Kumaran, K., & Raj, S. (2018). Optimization of parameters involved in robotic MIG welding process based on quality responses. *IOP Conf. Series: Materials Science and Engineering, 402*(012016), 11. https://doi.org/10.1088/1757-899X/402/1/012016

Luan, Y., Guo, J., & Liu, H. (2021). Structural Dynamics Simulation Analysis of Industrial Robot Arm Based on Kane Method. *Journal of Physics: Conference Series, 1871*(012155), 6. https://doi.org/10.1088/1742-6596/1871/1/012155

He, D., Liu, F., & Wang, F. (2020). Optimal Design of Industrial Robot Kinematics Algorithm. *Journal of Physics: Conference Series, 1624*(042029), 3. https://doi.org/10.1088/1742-6596/1624/4/042029

Luyang, S., & Yichao, P. (2020). Study on trajectory optimization algorithm of industrial robot in joint space. *Journal of Physics: Conference Series, 1676*(012206), 4. https://doi.org/10.1088/1742-6596/1676/1/012206

Herbuś, K., & Ociepka, P. (2016). Determining of a robot workspace using the integration of a CAD system with a virtual control system. *IOP Conf. Series: Materials Science and Engineering, 145*(052010), 5-7. https://doi.org/10.1088/1757-899X/145/5/052010