UNIVERSIDAD NACIONAL DE COLOMBIA

Laboratorio 1

Juan Carlos Naranjo Jaramillo, David Santiago Rodriguez Almanza
jnaranjoj@unal.edu.co, drodriguezal@unal.edu.co
Profesor: Pedro Fabian Cárdenas Herrera

I. Descripción del problema

En un contexto empresarial altamente competitivo y en constante evolución, la publicidad y el marketing han surgido como elementos esenciales para destacar en el mercado y consolidar la identidad de marca. Sin embargo, con el avance imparable hacia la robotización de las industrias, surge la pregunta crucial: ¿Es viable llevar a cabo estos procesos de manera eficiente mediante la implementación de robots industriales? Esta interrogante plantea un debate sobre la intersección entre la tecnología, la publicidad y la producción industrial, y cómo estas fuerzas convergen para moldear el panorama empresarial contemporáneo.

En base a lo anterior, como parte de una campaña publicitaria se propone exaltar la identidad de marca haciendo uso de robots industriales los cuales deben dibujar el logotipo de una reconocida marca sobre varias superficies en base a ciertos parámetros de funcionamiento que debe tener el robot.

II. Procedimiento

*II-A. **Diseño de la herramienta***

En primer lugar se requería diseñar un gripper que fuera capaz de sostener un marcador firmemente para seguir de manera correcta las trayectorias programadas dibujando el logotipo de adidas, la marca elegida por nosotros.

El problema de diseñar el gripper era un problema muy abierto, por lo cual se decidió tomar inspiración del software robotStudio, más específicamente de la herramienta MyTool utilizada en varios de los tutoriales que la empresa ABB y posteriormente los profesores de la Universidad Nacional de Colombia pusieron a nuestra disposición, esto debido a que esta herramienta se utiliza como un entrenamiento para seguir trayectorias, justamente el trabajo que debe hacer el robot.

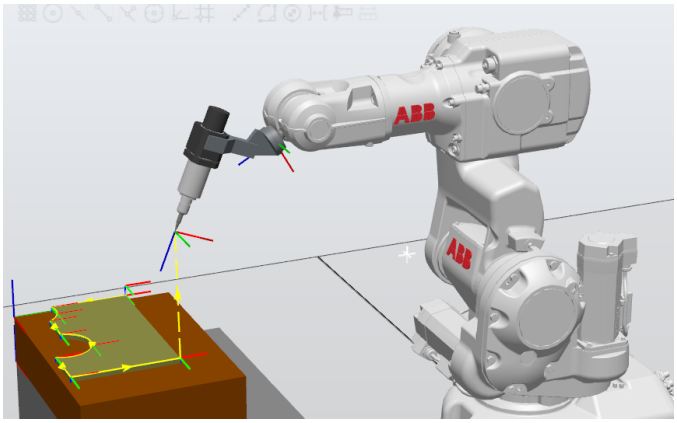

Figura 1: Herramienta de entrenamiento MyTool en el software RobotStudio

Se adoptó la geometría propuesta en el software y se adecuó para nuestros propósitos, un requerimiento que se debía tener era la sujeción del marcador, ya que la posición óptima para escribir es con el marcador completamente recto, es decir, si no se tenía un buen sistema de sujeción, el marcador terminaría por caer. La solución propuesta e implementada para este problema fue utilizar un gripper construido en dos partes, la primer parte que denominaremos el cuerpo, era la pieza que se encargaba de almacenar el marcador por lo cual se restringía el tamaño mínimo que debía tener la pieza y además tenía como función acoplar la herramienta el robot, la segunda pieza la denominaremos la tapa, la cual se introducía en la parte superior del cuerpo y mediante un roscado tanto en la tapa como en cuerpo, se juntaban para formar en conjunto el gripper, cabe destacar que la tapa tenía un pequeño agujero para permitir que la punta del marcador saliera.

Sin embargo las condiciones del laboratorio no son perfectas y existía la posibilidad de tener que lidiar con ciertas irregularidades en el nivel de la superficie, por

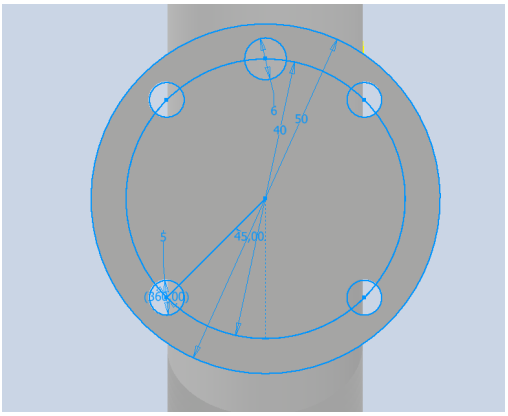

Figura 3: Medidas Base sujetador

lo que además se propuso que la herramienta tuviera un amortiguador para permitir una variación pequeña en la altura de la herramienta, tal variación debería responder a los cambios de nivel que puedan existir en el laboratorio, la solución propuesta fue colocar un resorte dulce tal que aprisionara el marcador contra la tapa y además permitiera un rango elástico en el cuál el marcador pudiera introducirse un poco más hacia el cuerpo del gripper.

A continuación se presenta un modelo 3D del gripper diseñado en función a los criterios de diseño seleccionados (Sujeción del marcador y variación de la longitud):

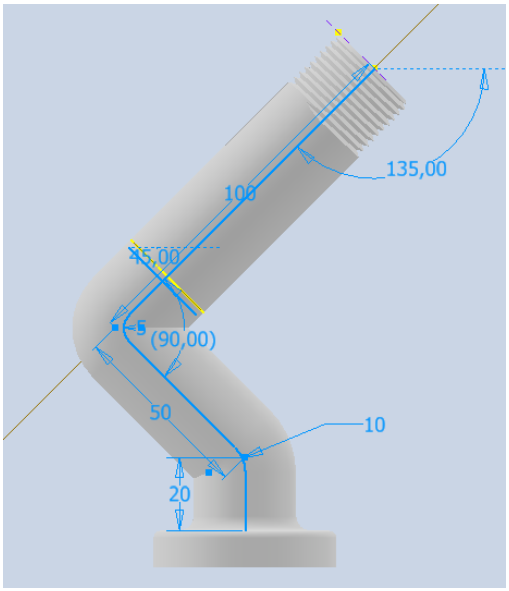

Figura 2: Medidas del sujetador

[H] [H] [H]

II-B. Configuración en RobotStudio

1. **Creación de la mesa de trabajo:** Dado que el robot no iba a dibujar en el piso, se insertó un paralelepípedo que hacía las veces de mesa para colocar sobre el la superficie del logo a recorrer.

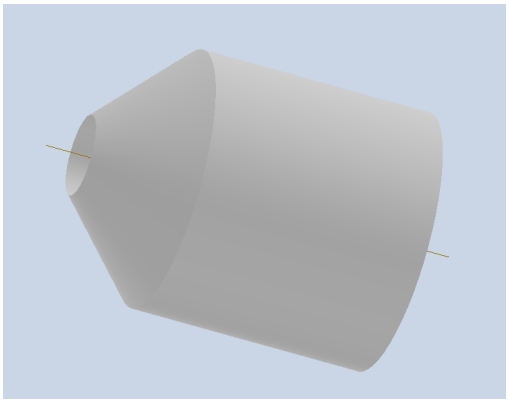

Figura 4: Tapa Sujetador

Figura 5: Sujetador impreso

Procedimiento para la creación de un paralelepípedo:

- Seleccionar la pestaña **Modelling** y usando el menú Solid crear un Tetraedro.
- Insertar las dimensiones del tetraedro.
- Desde la sección **Layout** se puede modificar el nombre del sólido.
- Se puede dar click derecho sobre el tetraedro en la sección **Layout** y en la opción **Position** y después **Set position**, cambiar la posición del tetraedro en el espacio de trabajo.

2. **Inserción del modelado del logotipo RobotStudio:** En primer lugar se debe modelar el logotipo a escribir en un sofware de modelado como inventor.
   - En la sección **Import Library** luego **Browse for**

library insertar el logo de adidas.

- Dar click derecho sobre la sección **Layout** y en la opción **Position** y después **Set position**, cambiar la posición del logo, intuitivamente colocarlo sobre la mesa. Se puede cambiar el nombre si se desea.

3. **Crer un nuevo controlador:** Desde la pestaña **Home**, dar click en **Virtual Controller**, luego en **New Controller** para finalmente crear el controlador del robot, es aquí donde posteriormente se va a cargar el código generado en RAPID.

4. **Creación del workObject:** Es necesario la creación de un sistema coodenado denominado **WorkObject** el cual servirá de referencia a nuestro robot para calcular todos los puntos que debe recorrer respecto a esa base coordenada.

   Procedimiento:

   - En la pestaña **Home**, dar click sobre**Other** y luego **Create WorkObject**.
   - En el menú que se despliega, en la sección **User Frame** dar click sobre la opción **Frame by Points** y luego **Three Points**.
   - Se despliega un menú el cual solicita 3 puntos para definir el origen y la orientación del sistema coordenado, el primer punto hace referencia al origen, el segundo a la dirección del eje **x**, y el tercero a la dirección del eje de referencia **y**, con estos tres puntos se define un plano y a partir de ese plano el eje **z** dado que debe cumplir la mano de la regla derecha en base a los ejes x y y que fueron definidos.
   - Si se desea es posible cambiar el nombre del WorkObject desde la ventana **Name**.

5. **Definición de los target en la trayectoria y también los Paths:** Desde **Home** en la sección **Path Programming** se puede definir un nuevo Path dando click en **Path** y posteriormente **Empty Path**, los Paths se van creando con un número ascendente para la organización de los mismos en el entorno virtual, sin embargo su nombre no tiene relevancia si se trabaja directamente programando sobre RAPID.

Figura 6: Trayectoria definida

   - Verificar que contiene un objeto de trabajo y un objeto de herramienta y que están seleccionados en la sección Parámetros, en caso necesario seleccionarlos, esto para que si en algún futuro debemos redefinir la posición del WorkObject, todos los puntos que ya teníamos definidos se muevan también.
   - Desde la sección **Path Programming** seleccionar la opción **Target** y **Create Target**, esta acción abrirá un menú donde podemos ir seleccionando todos los puntos de manera consecutiva del Path que queremos que siga el robot, y cuando finalmente se tengan todos los puntos seleccionados, dar sobre **Apply**.
   - También existe la opción **Teach Target** para guardar el punto de la posición actual del Robot.
   - Hasta el momento solo se ha fijado la posición de los puntos pero no la orientación, sin embargo robotStudio crea todos los puntos con una orientación por Default aunque se puede forzar desde el editor de Targets a establecer una orientación deseada, este no fué el caso y se requiere modificar la orientación antes de añadir los puntos al Path, para esto damos click derecho en cualquier Target creado y en la opción **Modify** se puede ajustar la orientación deseada, finalizamos con **Apply**, y dado que la superficie es plana, lo normal es que se quira que todos los puntos tengan la misma orientación, por lo que sobre el mismo punto al cual le fijamos una orientación, damos click derecho y sobre **Copy position and orientation**, para posteriormente seleccionar todos los otros puntos y dar sobre **copy orientation**, esta instrucción establece solo la orientación por lo que la posición no cambia.
   - Finalmente seleccionar todos los puntos que se

desee queden en un Path, dar click derecho y en el menú que se despliega dar click sobre **Add to Path** y posteriormente seleccionar el Path en el cual se desean guardar los puntos seleccionados, adicionalmente se recomienda configurar los parámetros de movimiento **(MoveL, MoveJ, Speed, Zone)** para que la instrucción se fijé en lo que deseamos, sin embargo es posible editarlo posteriormente una a una dando click sobre la intrucción y en **Modify Instruction**. En nuestro caso particular utilizamos un Path diferente por cada letra, alternando entre instrucciones MoveL con velocidad 100 para escribir y MoveJ con velocidad 1000 para reposicionar el robot cuando no está escribiendo.

- Se puede crear un Path Main que contenga todos los demás Paths dando click derecho sobre **Paths and Procedures** en la sección **Paths & Targets** para simular toda la trayectoria desde el entorno de simulación, sin embargo esto no es del todo necesario ya que también se puede añadir manualmente los Paths desde RAPID e igualmente simular desde allí o sincronizar con WorkStation.

6. **Configuración de salidas y entradas digitales:** Cómo requerimiento de diseño se requiere contar con dos botones, uno para comenzar la rutina y otro para enviar el robot a zona de mantenimiento, además de dos LED's indicadores de movimiento.
   - Para crear estas señales, desde la pestaña **Controller** se debe expandir la opción de **Configuration** en la barra de tareas para tener acceso a la opción **I/O System**, al dar click ahí se desprende un menú y dese la opción **signal** podemos crear señales de usuario dando click derecho, se despliega un menú donde debemos insertar nombre de la señal, tipo de la señal dado que puede ser entrada o salida análoga o digital o también grupos de entradas y salidas y finalmente después de establecer todos esos valores, se da en **Ok**.
   - Se debe repetir el procedimiento anterior para cada entrada y salida.
7. **Programación en RAPID:** Ya con todas las entradas y salidas digitales creadas, se deben utilizar las entradas para manipular la secuencia de movimiento del robot, por lo que se utilizó secuencias **While** para repetir la secuencia ilimitadas veces, y secuencias **If, Else** para generar las diferentes ramificaciones del programa, tales como comenzar la secuencia de dibujo del logo, o llevar el robot a la pose de mantenimiento. A continuación se muestra el código RAPID empleado para clarificar esta cuestión. PONER CODIGO RAPID

8. **Simulación:** Para simular en RobotStudio hay dos alternativas, mediante el entorno de simulación de RobotStudio o desde RAPID, se debe tener en cuenta que las dos deben ser equivalentes si RAPID y el entorno de simulación están sincronizados, en caso de que se haya trabajado desde RAPID, hace falta sincronizar RAPID a WorkStation y en el caso contrario, se debe sincronizar de WorkStation a RAPID, esto se hace desde **Home** en la opción **Synchronize** del sub menú **Controller**. Sin embargo al tener entradas y salidas digitales, para simular adecuadamente el programa hace falta también simular las señales, por lo que desde la pestaña **Simulation** se puede activar esta opción dando click en **I/O Simulator**, se despliega un menú con algunas señales internas del robot definidas por Default, sin embargo desde la opción **Edit List**, se puede seleccionar las señales que se desean simular seleccionando la opción **User List**, esas señales se organizaran en un tablero de comando con inputs y outputs.

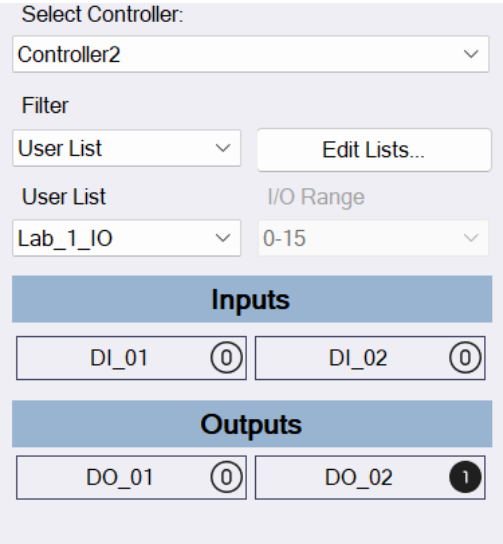

Figura 7: Entradas y salidas digitales en RobotStudio

9. **Modificación del objeto de trabajo:** Si se requiere, se puede modificar el WorkObject y con ello, todos los puntos definidos respecto a él por lo cual no es necesario volver a establecer todos los Targets de una nueva trayectoria, esto generalmente se utiliza cuando se programa las trayectorias de un robot industrial pero las medidas del espacio real de trabajo no coinciden con las del entorno virtual de desarrollo, por lo que se puede ajustar las trayectorias directamente en el espacio de trabajo real solo modificando el WorkObject. Desde la ventana **Home** en la cinta **Paths & Targets, T_ROB1, WorkObjects & Targets**, dar click derecho sobre el WorkObject y en **Modify**, se abrirá el mismo menú que se desplegó cuando se creó en WorkObject y siguiendo el mismo procedimiento, se puede re-editar el WorkObject, al dar **Apply**, todos los puntos se desplazaran respecto a la nueva posición del WorkObjects, y para finalizar, hace falta sincronizar la estación de trabajo con RAPID.

Figura 8: Resultado Final

III. Diagrama de flujo de acciones del robot

La secuencia que debe seguir el robot en ambas trayectorias es:

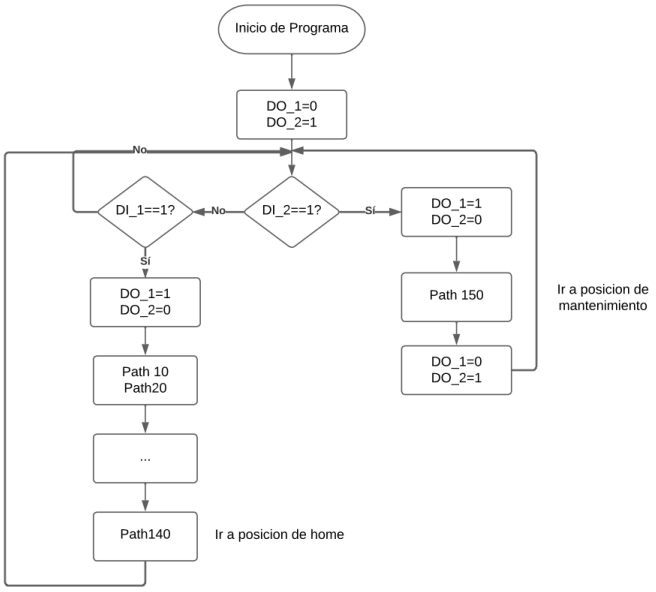

Figura 9: Diagrama de flujo

IV. Plano de planta de la ubicación de cada uno de los elementos

Para el dibujo en plano:

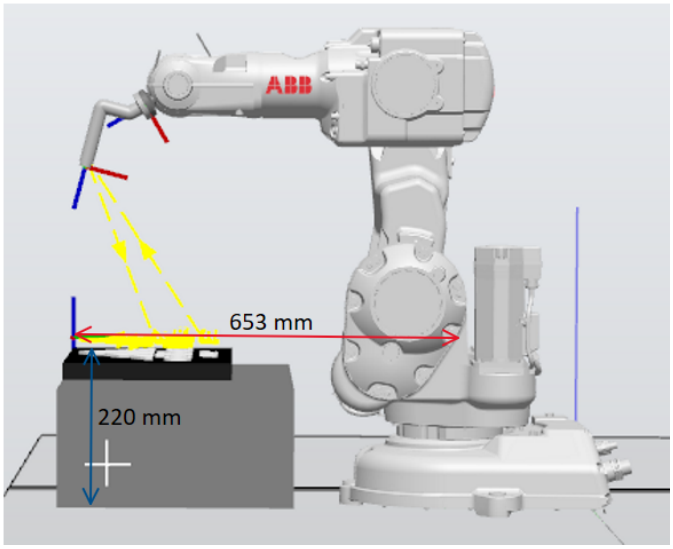

Figura 10: Configuración en el espacio de trabajo en el plano XZ

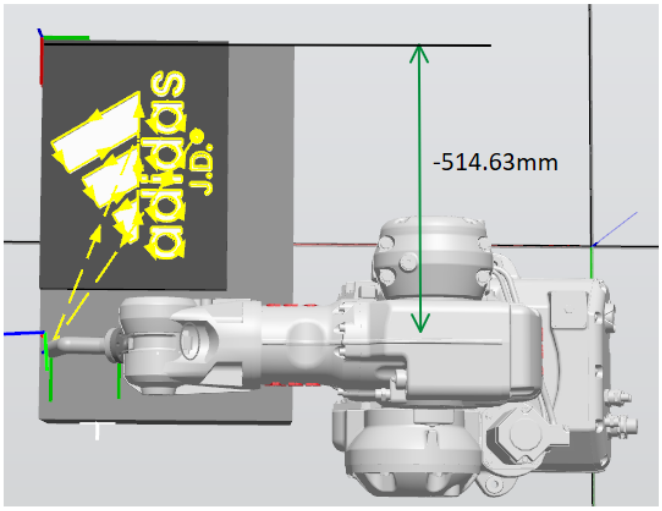

Figura 11: Configuración del espacio de trabajo en Y

Para el plano inclinado

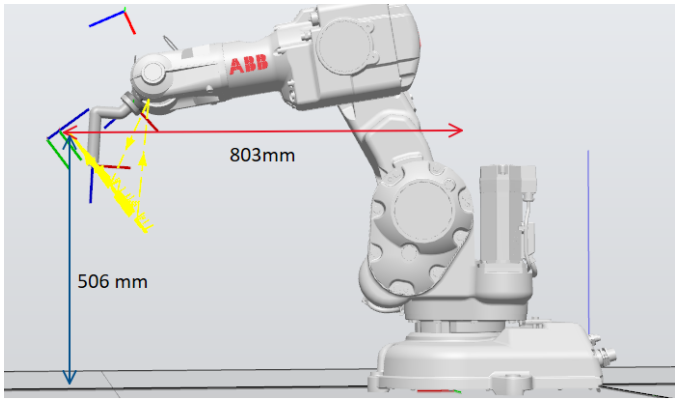

Figura 12: Configuración Plano inclinado en XZ

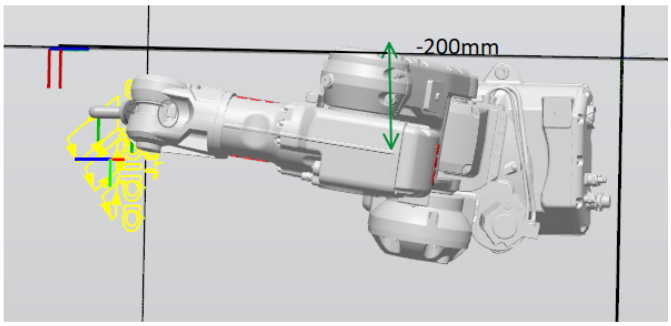

Figura 13: Distancia en Y

V. Descripción de las funciones utilizadas

1. **WHILE TRUE DO :** Ciclo que permite la repetición de una rutina de programa reiteradas ocasiones, en nuestro caso particular infinitas.

2. **IF DI_01 = 1 THEN :** Función que permite discriminar varias rutinas de programación, ejecutando solo una en función de algunos parámetros definidos.

3. **PATH_X() :** Rutina de programación que representa una secuencia de movimiento a través de ciertos puntos con algunos parámetros de movimiento definidos (MoveL, MoveJ, Speed, Zone, WorkObject).

4. **Main:** Función principal del programa donde se coloca la lógica del código y desde donde se llama a todas las demás subrutinas que componen el programa.