

Implementación de un robot colaborativo a procesos poliméricos

Isai Josue Rios Rosas¹, Francisco D. Calvo López², María Judith Percino Zacarías³, Enrique Pérez-Gutiérrez³, David Pinto-Avendaño⁴

¹Faculta de Electrónica, Decanato de Ingeniería Universidad Popular Autónoma del Estado de Puebla, México. Email: isaijosue.rios@upaep.edu.mx

²Faculta de Electrónica, Decanato de Ingeniería, Universidad Popular Autónoma del Estado de Puebla, México. Email: franciscodomingo.calvo@upaep.mx

³Unidad de Polímeros y Electrónica Orgánica, Instituto de Ciencias, Benemérita Universidad Autónoma de Puebla, México. Email: Judith.percino@correo.buap.mx, enrique.pgutierrez@correo.buap.mx

⁴Dirección de Innovación y Transferencia del Conocimiento, Benemérita Universidad Autónoma de Puebla, México. Email: david.pinto@correo.buap.mx

Resumen

En este artículo se describe la primera fase de implementación del robot colaborativo ABB YuMi IRB 14000 en un laboratorio de procesos poliméricos. Se detallan las herramientas seleccionadas, los procedimientos de calibración y la programación del robot para tareas de manipulación y reconocimiento de objetos. Los resultados muestran la importancia de una correcta calibración del espacio de trabajo y las herramientas para asegurar la precisión en la manipulación de objetos. Se discuten las implicaciones de estos hallazgos y se proponen mejoras para futuras aplicaciones del robot en el ámbito de la investigación de polímeros.

Palabras clave: Robot colaborativo, YuMi, Procesos poliméricos, Calibración, Manipulación, Reconocimiento

Abstract

This paper describes the first phase of the implementation of the ABB YuMi IRB 14000 collaborative robot in a polymer processing laboratory. It details the selected tools, calibration procedures, and robot programming for handling and object recognition tasks. The results highlight the importance of correct workspace and tool calibration to ensure precision in object handling. The implications of these findings are discussed, and improvements for future applications of the robot in polymer research are proposed.

Keywords: Collaborative, YuMi, Polymer, Calibration, Handling, Recognition

1. Introducción

La implementación de robots colaborativos en procesos de laboratorios ha cobrado gran relevancia debido a su capacidad para trabajar de manera segura junto a los seres humanos, optimizando así la eficiencia y precisión en tareas repetitivas y delicadas. Este artículo se enfoca en la primera fase de integración del robot colaborativo ABB YuMi IRB 14000 en un entorno de laboratorio especializado en procesos poliméricos. Se describen las herramientas utilizadas, los procedimientos de calibración y la programación necesaria para asegurar una operación segura y precisa. Además, se presentan los resultados obtenidos y se discuten las mejoras propuestas para futuras implementaciones.

2. Robot Colaborativo YuMi IRB 14000

El ABB YuMi IRB 14000 es un robot colaborativo de dos brazos, diseñado específicamente para trabajar de manera segura junto a personas en diferentes entornos. Cada brazo del YuMi tiene 7 grados de libertad, permitiéndole realizar movimientos complejos con alta precisión. Sus brazos están acolchados y su diseño es compacto y ligero, lo que maximiza la seguridad para el personal que se encuentra en contacto directo con el robot.

El YuMi incorpora características como la predicción de colisiones y parada instantánea al detectar alguna colisión. Este robot es ideal para el manejo de piezas

pequeñas, manejo de material delicado, y tareas de medición e inspección.

El ABB YuMi IRB 14000, con sus características avanzadas de seguridad y precisión, se adapta perfectamente a los entornos de laboratorio donde se manejan materiales delicados como los polímeros. En esta sección se detallan las herramientas seleccionadas y su integración en el laboratorio



Figura 1. Robot ABB YuMi IRB 14000 vista general.

2.1. Selección de herramientas para el robot

Gripper: El gripper será utilizado para agarrar y manipular objetos de diversas formas y tamaños. Esta herramienta es esencial para la manipulación segura y precisa del material de laboratorio.



Figura 2. Smart Gripper ABB YuMi vista general

Cámara: La cámara permitirá la identificación de objetos, mediante visión artificial. Esta capacidad de reconocimiento visual es fundamental para reconocer equipo y sustancias, asimismo, asegurar la correcta manipulación del equipo.



Figura 3. Smart Gripper ABB YuMi vista cámara

Tapete Antirreflejante: El tapete antirreflejante mejora la capacidad de la cámara para reconocer objetos, eliminando reflejos no deseados que puedan interferir con el sistema de visión artificial.

2.2. Identificación de material de laboratorio a utilizar

Para el manejo seguro y eficiente de sustancias poliméricas, se empleará una variedad de materiales de laboratorio que incluyen probetas, frascos y contenedores específicos para el almacenamiento y manipulación de polímeros. Para garantizar la integridad del proceso es requerido que cada herramienta y contenedor haya sido sometido a pruebas de resistencia con la herramienta a utilizar.

3. Preparación del Robot, herramientas y espacio de trabajo

Iniciaremos con la calibración del robot y sus herramientas en el software RobotStudio. Primero, importamos el modelo del robot y configuramos el Smart Gripper, seguido por la calibración de la cámara y del espacio de trabajo. Estos pasos son cruciales para asegurar una operación segura y eficiente.

3.1. Calibraciones

Para poder programar el robot de forma correcta es necesario realizar la calibración de cada una de las herramientas a utilizar ya que de lo contrario podría provocar colisiones con el entorno de trabajo o dañar al personal de laboratorio.

3.1.1. Calibración del robot

Posicionamiento Inicial:

Colocar ambos brazos del YuMi en una posición de referencia. Esta posición suele ser una configuración predefinida que alinea los brazos en una postura neutral.

Ejecutar Calibración:

Ejecutar el procedimiento de calibración, que generalmente incluye actualizar el “Cuentarrevoluciones”.

Validación y Ajuste:

Validar que las posiciones alcanzadas por el brazo coincidan con las posiciones predeterminadas. Si es necesario, realizar ajustes finos hasta que el brazo esté correctamente calibrado.

Guardar Configuración:

Una vez calibrados ambos brazos, guardar la configuración en el sistema del robot para asegurar que estas calibraciones se mantengan durante las operaciones.

3.1.2. Calibración de la cámara

Es importante tener un posicionamiento inicial de la cámara para adquirir la imagen.



Figura 3. Posicionamiento para adquisición de imagen.

Se añade una plantilla de calibración en nuestro espacio de trabajo.

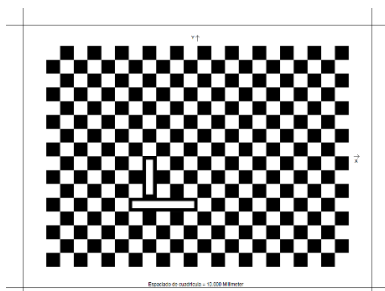


Figura 4. Cuadrícula de calibración.

Realizamos la adquisición de la imagen y procedemos a calibrar los parámetros dentro de la interfaz para poder ajustar y obtener una imagen adecuada.

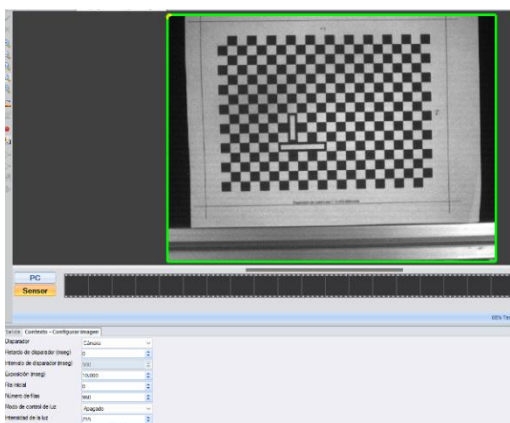


Figura 5. Configuración de adquisición de imagen en software.

Adquisición de nueva imagen y realizar calibración de la cámara como nos indica el fabricante.

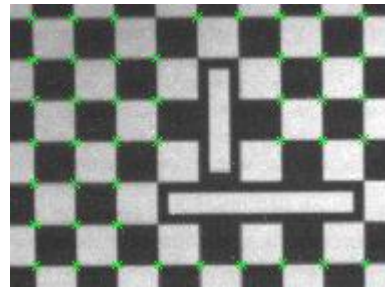


Figura 6. Reconocimiento de la hoja de calibración en software.

Obtenemos los resultados y la fiabilidad del espacio de trabajo.

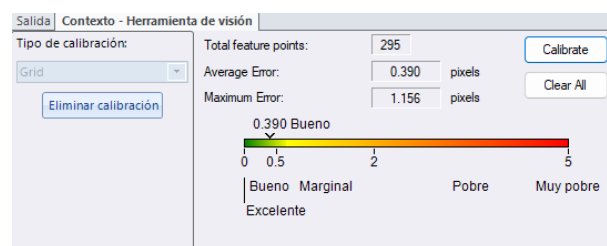


Figura 7. Resultados de la calibración.

Con esto finaliza la calibración de la cámara respecto al espacio de trabajo.

3.1.3. Calibración en el espacio de trabajo

Desde el software creamos un nuevo TCP que este será la punta del gripper, ya que lo utilizaremos como palpador o calibrador de punta, lo establecemos a la distancia en Z de 135.00mm.

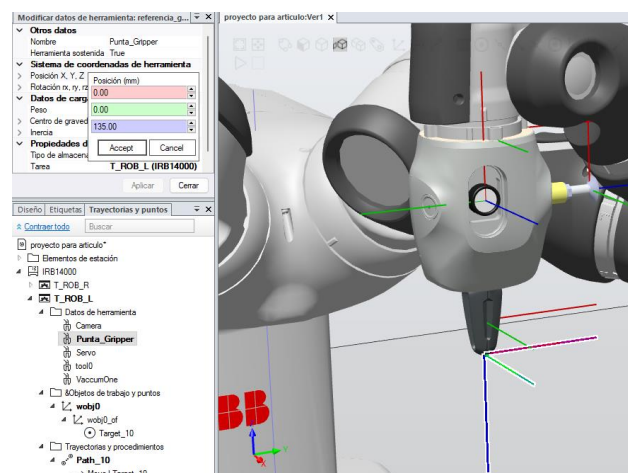


Figura 8. Creación de la herramienta en el espacio de trabajo del software.

Sincronizamos la estación de trabajo con RAPID para obtener los valores de las herramientas que utilizaremos y posteriormente darlos de alta en el flexpendant en el archivo de "Calib Data".

```

1  MODULE CalibData
2  PERS tooldata Servo:=[TRUE],[0,0,114.2],[1,0,0,0]],[0.24,[8.:
3  PERS tooldata VacuumOne:=[TRUE],[63.5,18.5,37.5],[0.70710678:
4  PERS tooldata Camera:=[TRUE],[[-7.3,28.3,35.1],[0.5,-0.5,0.5,
5  TASK PERS tooldata Punta_Gripper:=[TRUE],[[0,0,135],[1,0,0,0]
6  ENDMODULE

```

Figura 9. Códigos obtenidos de las herramientas creadas e importadas.

Dentro de la pestaña de “Jogging”, encontraremos la herramienta cargada, realizamos la selección para usarlo como apoyo en la calibración.

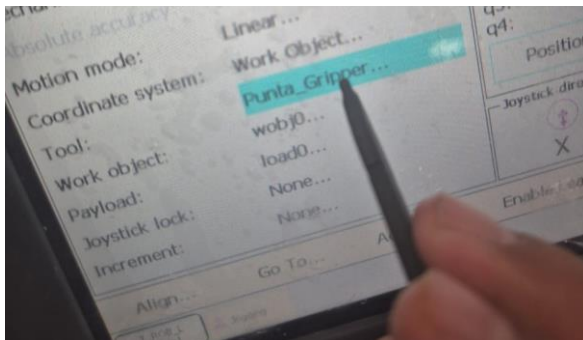


Figura 10. Ventana de “Jogging” en el controlador físico.

Creamos un nuevo espacio de trabajo en el controlador, este será los límites donde el robot buscará el objeto a detectar.

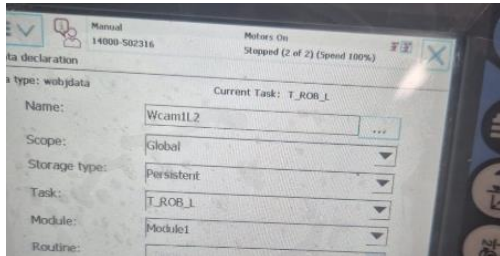


Figura 11. Creación de un espacio de trabajo en el controlador físico.

Declaramos nuestro espacio de trabajo para que se calcule por medio de 3 puntos, Centro, X e Y.



Figura 12. Posicionamiento de la herramienta en las referencias del espacio de trabajo, centro, extremo en X y extremo en Y respectivamente.

Se modifica las posiciones respectivas en el controlador y aceptamos, posteriormente esperamos a que termine de calcular nuestro espacio de trabajo.

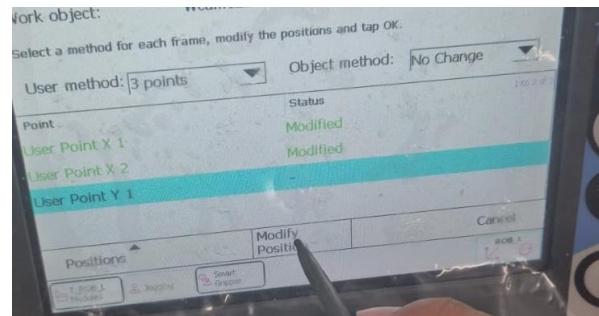


Figura 13. Adquisición de puntos a nuestro espacio de trabajo.

Para corroborar nuestra operación podemos mover el brazo del robot de forma manual y observar como cambian los valores de X,Y y Z dependiendo la posición en que se encuentre nuestra herramienta.

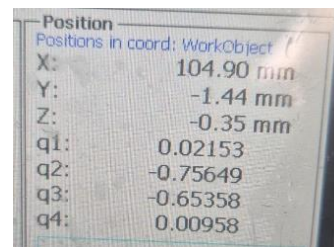


Figura 14. Coordenadas X, Y y Z de nuestro espacio de trabajo.

Regresamos a la posición de Home y retiramos la plantilla de calibración.

3.1.4. Calibración Servomotores

Para la calibración del gripper nos vamos a la pestaña de Smart gripper y configuramos el gripper en posición mínima y realizamos la calibración.

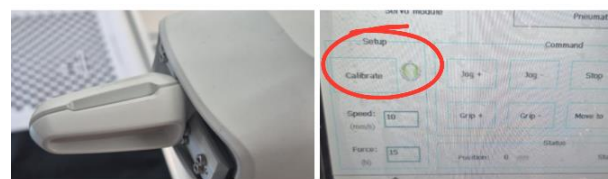


Figura 15. Posición de calibración del gripper y ventana de calibración.

Se realizó un programa de prueba para verificar que funciona correctamente para el agarre de un objeto, para esta prueba se utilizó un frasco de vidrio, inicializaremos el gripper con una fuerza de agarre de 10N y a una velocidad de 10mm/s, posteriormente y le aplicaremos una fuerza de 7N para asegurar la integridad del recipiente.


```

PROC Pick_obj()
  !inicializa el gripper con valores default
  g_Init \maxSpd := 10, \holdForce := 10;
  WaitTime 5; !Tiempo de espera
  !Agarra el objeto con 7N de fuerza
  g_GripIn \holdForce := 7;
  WaitTime 5; !Tiempo de espera
  !Suelta el objeto
  g_GripOut \holdForce := 7;
ENDPROC

```

Figura 16. Código ejemplo del funcionamiento del gripper.



Figura 17. Pruebas físicas del gripper en recolección y liberación del objeto.

4. Programación del robot

Para la programación del robot ABB YuMi IRB 14000 se planea utilizar el software RobotStudio, donde se configurará la cámara y las herramientas necesarias para el manejo de objetos y la identificación visual en el entorno de trabajo. El proceso incluirá la definición de trayectorias precisas para evitar colisiones y asegurar la correcta manipulación de los objetos.

La configuración de la cámara permitirá la adquisición de imágenes y el reconocimiento de patrones para identificar correctamente los frascos de laboratorio que serán manipulados por el robot. Además, se programarán posiciones auxiliares para guiar al robot de manera segura hasta el punto de captura de imagen, asegurando así una operación eficiente y segura.

4.1. Posicionamiento del robot

Se busca llegar a un punto establecido por coordenadas para que capture la imagen de forma correcta. Así mismo se emplearán trayectorias auxiliares para que llegue a la posición deseada y evitar algún tipo de colisión con el entorno o con el mismo robot.

El punto de referencia que crearemos se encontrará en las coordenadas [400, 0, 400] justo en el centro de nuestro plano de trabajo.

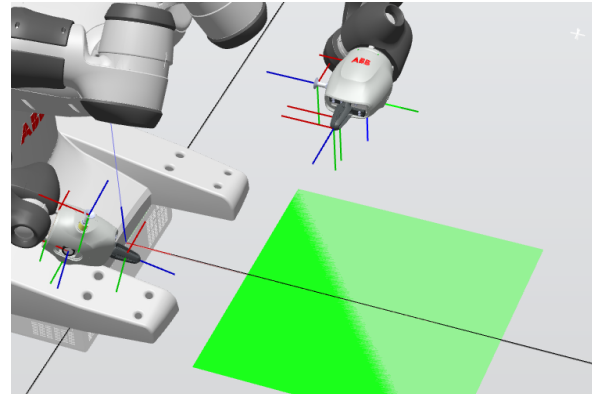


Figura 18. Posición de adquisición de imagen referente al plano de trabajo en simulador.



Figura 19. Posición auxiliar del robot.

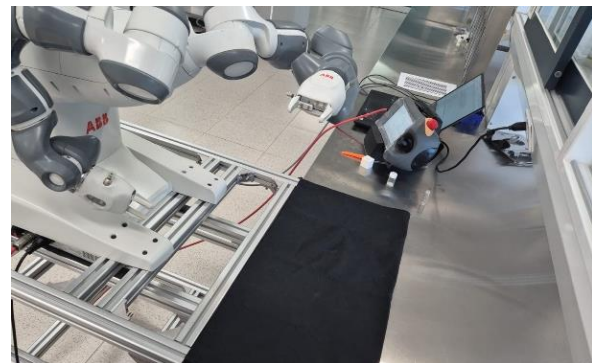


Figura 20. Posición final de captura de imagen.

Una vez llegada a la posición procedemos a realizar la captura de imagen para programar el reconocimiento de objetos.

4.2. Programación general de la cámara

El objeto que identificará el robot es un pequeño frasco de laboratorio. Procedemos a adquirir la imagen de nuestro espacio de trabajo con el objeto a identificar en el campo de visión de la cámara.

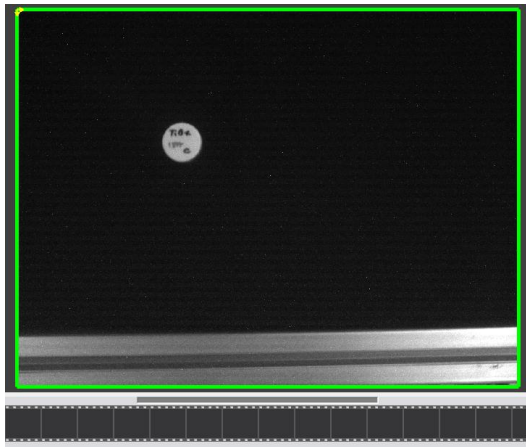


Figura 21. Adquisición de imagen en software.

Con las opciones que nos muestra el software utilizamos la de identificación y patrón.



Figura 22. Herramienta por utilizar en software.

Incluimos dentro de la región de “modelo” nuestro modelo a identificar, en este caso la tapa del frasco.

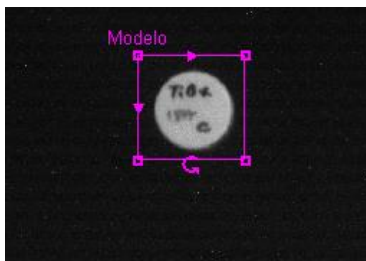


Figura 22. Región de modelo a entrenar.

Aceptamos y tendríamos nuestra herramienta de identificación, la cual nos ayudaría a obtener el centro de nuestro objeto a detectar en coordenadas.

4.3. Código de funcionamiento

¡Se inicia declarando las constantes que nos ayudaran a simplificar el código.

```
PERS wobjdata Wcam1L2:= [FALSE,TRUE,"",[[383.361,-12.2866,-0.704813],[0.71137,0.0153852,0.0151982,0.702485]],[[4.05847,-3.98945,0],[0.967522,0,0,0.252787]]];
```

!asignamos a una variable el nombre de nuestro trabajo de visión artificial

```
CONST string myjob := "calibracion_articulo.job";
VAR cameratarget mycameratarget;
```

Por último, debemos de extraer esas coordenadas de la cámara y enviarlas al controlador para que el robot obtenga las coordenadas a dirigirse y poder llegar al objeto en cuestión

Seleccionamos “salida a RAPID” y enviamos las coordenadas del objeto detectado que son X, Y y el ángulo de inclinación.



Figura 23. Herramienta de envío de datos.

Correlación de datos de la cámara para la pieza 'Tapa_Frasco'		
Componente	Grupo	Resultado
Position x	identificacion_tapa	Fixture.X
Position y	identificacion_tapa	Fixture.Y
Rotation z	identificacion_tapa	Fixture.Angle
Value 1	Constant	0
Value 2	Acquisition	0
Value 3	Circle_1	0
	identificacion_tapa	0
	Inputs	
	Job	

Figura 24. Coordenadas por enviar al controlador.

Con esto terminamos la calibración de la cámara para la identificación de nuestro objeto.

!declaración de detección de objeto

```
CONST robtarget objeto_auxiliar_antes:= [[-5,-14.52,100.44],[0.355457,-0.673058,-0.549159,0.345062],[-1,-1,2,5],[142.013,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget objeto:= [[-28.29,-10.78,25.56],[0.355467,-0.673066,-0.549144,0.345059],[-1,-1,2,5],[142.013,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

!Funcion auxiliar para posición de adquisición de imagen

```
PROC GoFotoL()
```

```
    MoveJ [[125,210,230],[0.0644092,0.843286,-0.105959,0.522965],[0,-1,1,4],[141.536,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z100, tool0;
    MoveJ [[355.00,205,270],[0.469667,-0.443146,0.480795,-0.593186],[-1,-1,2,4],[142.664,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z100, tool0;
    MoveJ [[375,-7,290],[0.0609033,0.78477,0.616734,0.00809459],[-1,0,0,4],[161.611,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z100, Camera;
    MoveJ [[400,0,400],[0.0,0.707106781,0.707106781,0.0],[-1,0,0,4],[161.611,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z100, Camera;
```

```
ENDPROC
```

!Funcion de deteccion del objeto, establece la cámara en modo adecuado, carga el trabajo de visión, inicializa camra, y adquiere datos enviadas

```
PROC DetectedObject()
```

```
    CamSetProgramMode cam_izq;
    CamLoadJob cam_izq, myjob;
    CamSetRunMode cam_izq;

    WaitTime 7;
    CamReqImage cam_izq;
    CamGetResult cam_izq, mycameratarget;
    Wcam1L2.iframe := mycameratarget.cframe;
```

```
ENDPROC
```

!Funcion de que nos ayuda a mover el gripper a las coordenadas del objeto a detectar, transformando las coordenadas de la cámara a las coordenadas del plano de trabajo.

```
PROC MoveToDetectedObject()
```

```
    MoveJ objeto_auxiliar_antes, v100, fine, Servo \WObj:=Wcam1L2;
    MoveJ objeto, v100, fine, Servo \WObj:=Wcam1L2;
    Pick_obj;
    MoveJ objeto_auxiliar_antes, v100, fine, Servo \WObj:=Wcam1L2;
```

```
ENDPROC
```

!Funciones auxiliares**!Posicion de Home**

```
PROC GoHomeL()
```

```
    MoveAbsJ [[0,-130,30,0,40,0],[135,9E9,9E9,9E9,9E9,9E9]]\NoEOffs, v1000, fine, tool0;
ENDPROC
```

!agarrar objeto

```
PROC Pick_obj()  
  g_GripIn \holdForce := 7;  
ENDPROC
```

!inicializar gripper y abrir

```
PROC open_gripp()  
  g_Init \maxSpd := 10, \holdForce := 10;  
  g_GripOut \holdForce := 7;  
ENDPROC
```

!Código Main

```
PROC main()  
  
  GoHomeL;  
  open_gripp;  
  WaitTime 5;  
  GoFotoL;  
  DetectedObject;  
  MoveToDetectedObject;
```

!posicion a soltar el objeto

```
  MoveJ [[400,0,400],[0.0,0.707106781,0.707106781,0.0],[-  
  1,0,0,4],[161.611,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z100, Camera;  
  WaitTime 5;  
  open_gripp;  
  WaitTime 5;  
  
  Waittime 20;  
ENDPROC
```


4.4. Primer pick and place

Una vez terminado el código procedemos a ejecutarlo hasta la función DetectObject() ya que es necesario realizar el posicionamiento de agarre

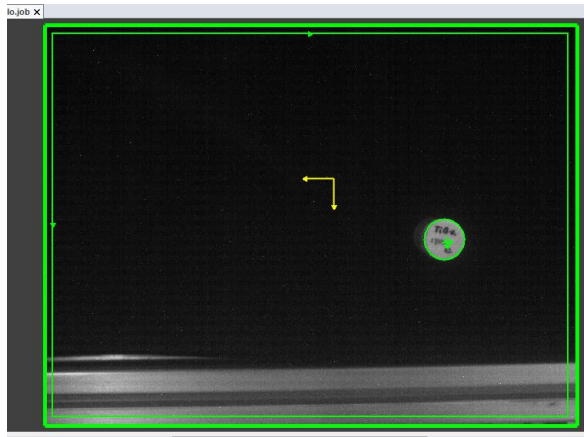


Figura 25. Objeto identificado en software automáticamente.

Una vez el robot haya terminado esa función debemos moverlo de forma manual hasta la posición de agarre que deseemos.



Figura 26. Posicionamiento manual del robot en posición de recolección.

Una vez realizado esto regresamos a home y procedemos a correr el programa completo.

5. Resultados obtenidos

En esta sección se presentan los resultados obtenidos durante la implementación del robot colaborativo ABB YuMi IRB 14000 en el laboratorio de procesos poliméricos.

5.1. Espacio de trabajo y calibración

Uno de los principales hallazgos fue la importancia de tener un espacio de trabajo con medidas específicas y bien calibrado. Durante los experimentos, se observó que la precisión del robot en la recolección de objetos podía verse afectada por variaciones de 2 a 6 mm si el espacio de trabajo y el palpador no estaban calibrados adecuadamente.

Para abordar esta problemática, se propuso realizar las siguientes acciones:

- Establecer un espacio de trabajo específico con dimensiones precisas, garantizando que todos los elementos necesarios para las operaciones del robot estuvieran dentro de este espacio.
- Utilizar un palpador específico para la calibración, lo cual ayudará a minimizar las desviaciones y a asegurar una precisión consistente en las tareas del robot.

5.2. Impacto en la precisión

El análisis de los datos recolectados mostró que, sin un espacio de trabajo bien definido y un palpador específico, es muy complicado corregir las desviaciones de posicionamiento. La precisión del robot en la manipulación de objetos no alcanzó los estándares requeridos para las operaciones de laboratorio.

5.3. Ejemplos prácticos.

Se realizaron pruebas específicas para evaluar el impacto de la falta de calibración adecuada. En una de las pruebas, el robot debía recoger un frasco de vidrio ubicado en el espacio de trabajo. Los resultados mostraron que, debido a las desviaciones de 2 a 6 mm, el robot no fallaba en recoger el frasco, sin embargo, existía la posibilidad de que aumentara el riesgo de daños o errores en el proceso.

Enlace a video demostrativo aquí:

https://drive.google.com/drive/folders/1HRT-eKiJ2Jx32LE9qgdxKoUK9LXLiHB?usp=drive_link

6. Implementación futura

En el futuro, se busca corregir estos errores de desviación con la recolección de objetos para poder expandir la implementación del robot YuMi en otras áreas de la investigación de polímeros. Esto incluye automatización de diversas tareas dentro del laboratorio, tales como la manipulación de muestras, la preparación de ensayos y la realización de pruebas de materiales. La integración se extenderá a procesos repetitivos y precisos, como el reconocimiento de sustancias y la interacción con equipos de medición. Además, se espera la utilización de patrones y la lectura de códigos Data Matrix para garantizar una gestión óptima del inventario y la trazabilidad de los procesos. Los futuros desarrollos también abordarán desafíos potenciales en la escalabilidad y seguridad del sistema, proponiendo soluciones innovadoras para mantener la integridad y eficiencia en el entorno de laboratorio.

7. Conclusiones

La implementación del robot colaborativo ABB YuMi IRB 14000 en el laboratorio de procesos poliméricos ha demostrado ser un avance significativo en la automatización de tareas delicadas y repetitivas. Las principales conclusiones derivadas de esta primera fase de implementación son las siguientes:

Importancia de la Calibración:

La precisión en la manipulación de objetos depende críticamente de una calibración adecuada del espacio de trabajo y de las herramientas utilizadas. Pequeñas desviaciones en la calibración pueden afectar significativamente la eficacia del robot en tareas de recolección y manipulación.

Selección de Herramientas:

La elección correcta de herramientas, como el gripper y la cámara, es fundamental para asegurar una operación eficiente y segura. Estas herramientas permiten al robot manejar objetos de diversas formas y tamaños, y realizar tareas de reconocimiento visual con alta precisión.

Procedimientos de Programación:

La programación detallada del robot, incluyendo la definición de trayectorias y posiciones auxiliares, es esencial para evitar colisiones y asegurar la correcta manipulación de los objetos.

La configuración de la cámara para la adquisición y reconocimiento de imágenes también es crucial para el éxito de las tareas asignadas al robot.

Impacto en la Investigación:

La integración del robot YuMi en el laboratorio abre nuevas oportunidades para automatizar procesos de investigación en polímeros, mejorando la eficiencia y reduciendo el riesgo de errores humanos. Además, la capacidad del robot para trabajar de manera segura junto a los investigadores permite una mayor colaboración y optimización de los recursos del laboratorio.

Mejoras Futuras:

Para futuras fases de implementación, es necesario abordar las desviaciones de posicionamiento identificadas durante esta primera fase. La recolección de datos y la refinación continua de los procedimientos de calibración y programación contribuirán a mejorar la precisión y eficacia del robot en tareas más complejas.

8. Referencias

- [1] ABB Robotics ABB. (s. f.). Manual del operador - RobotStudio [PDF]. Disponible en: https://library.e.abb.com/public/6aeb483836740e11c1257b4b0052375b/3HAC032104-005_revE_es.pdf
- [2] IRB 14000 | ABB. (s. f.). IRB 14000 gripper Manual - IRB 14000 YuMi. ABB AB [PDF]. Disponible: <https://library.e.abb.com/public/6c35d74e5be34fcb93bf6810c449a2bd/3HAC054949%20PM%20IRB%2014000%20Gripper-en.pdf>
- [3] IRB 14000 | ABB. (s. f.). Product Manual - IRB 14000 YuMi. ABB AB. [PDF]. Disponible en: <https://library.e.abb.com/public/fc0d5c31604f48ed998229f626f72036/3HAC052983%20PM%20IRB%2014000-es.pdf>
- [4] IRB 14000 | ABB. (s. f.). Operating Manual - IRB 14000 YuMi. ABB AB. [PDF]. Disponible en: <https://cdn.pennair.com/media/2023/11/3HAC052986-OM-IRB-14000-en.pdf>