



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA TÈCNICA
SUPERIOR ENGINYERS
INDUSTRIALS VALÈNCIA



DEPARTAMENTO DE INGENIERÍA DE
SISTEMAS Y AUTOMÁTICA

PRÁCTICA 2:

PROGRAMACIÓN Y SIMULACIÓN ESTACIÓN AUTOMÁTICA DE EMPAQUETADO

ROBÓTICA INDUSTRIAL

MÁSTER UNIVERSITARIO INGENIERÍA INDUSTRIAL

ÁNGEL VALERA FERNÁNDEZ

CATEDRÁTICO DE UNIVERSIDAD

DPTO. DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

INSTITUTO AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

OCTUBRE, 2023

OBJETIVOS

- Utilizar un paquete de programación y simulación de robots industriales
 - Presentar problemática asociada a la simulación de células robotizadas para la automatización de procesos
 - Desarrollar la configuración de las distintas señales y dispositivos relacionados con la célula robótica
 - Programar rutinas RAPID que permitan que el robot realice las operaciones necesarias para llevar a cabo la automatización de aplicaciones industriales
 - Realizar la simulación de la aplicación de empaquetado en el entorno de RobotStudio
 - Ejecutar el programa RAPID en un robot real y verificar que éste es capaz de realizar la aplicación industrial indicada
-

1. Introducción

Como se sabe, el robot es una de las piedras angulares dentro de la automatización de procesos industriales ya que éstos evitan que las personas se tengan que encargar de realizar tareas que son pesadas, monótonas, peligrosas y/o nocivas.

En este caso, una vez programado el robot, éste será el encargado de estar realizando de forma automática y sin descanso las distintas tareas relacionadas con la aplicación industrial que se necesite.

Cuando se aborda el desarrollo de aplicaciones industriales basadas en robots es necesario

- analizar bien la aplicación industrial que se necesita resolver
- realizar un estudio previo del proceso
- obtener las distintas tareas que se deben realizar para resolver la aplicación
- proponer una solución técnica al problema de automatización
- programar las distintas tareas necesarias para el robot
- realizar la simulación de la respuesta del sistema robotizado y verificar que se cumplen con las especificaciones de diseño requeridas
- transferir el programa a la célula robotizada real y verificar que la respuesta del sistema real es la misma que la del sistema virtual

En esta práctica se propone la simulación y la programación de un proceso industrial de control de calidad y de empaquetado, de forma que se debe trabajar con la célula robotizada disponible en el Dpto. de Ingeniería de Sistemas y Automática de la Universitat Politècnica de Valencia. La célula está compuesta por un robot industrial ABB, varias cintas transportadoras controladas con variadores de frecuencia y un sistema de visión artificial.

2. Descripción de la célula de inspección y empaquetado

En la segunda práctica de Robótica Industrial se va a trabajar con la programación y la simulación de una célula de inspección y empaquetado automático de productos. La célula está compuesta por:

- a) Un robot industrial ABB IRB140
- b) Una unidad de control ABB IRC5
- c) Una cinta transportadora
- d) Un sistema de inspección basado en visión artificial

El planteamiento del problema es el siguiente:

1. La cinta transporta los productos (latas de conserva o bricks de leche) que se acaban de elaborar y etiquetar. La cinta está equipada con un variador de frecuencia para controlar un motor que pone en marcha o para la cinta. Además, cuenta también con un sensor de presencia que puede detectar que una lata acaba de llegar al extremo de la cinta.

Para poder interactuar con la cinta, la IRC5 dispone de 3 señales digitales:

- a) *CONVEYOR_FWD*: salida digital que cuando se activa a nivel alto hace que la cinta se ponga en marcha avanzando a una velocidad constante. Si dicha señal pasa a nivel bajo la cinta se parará.
- b) *CONVEYOR_BWD*: salida digital que hace que cuando la señal se activa a nivel alto la cinta se pone en marcha retrocediendo a una velocidad constante. Si dicha señal pasa a nivel bajo la cinta se para.
- c) *CONVEYOR_OBJ_SEN*: entrada digital que se pone a nivel alto cuando el sensor de presencia detecta un objeto al final de la cinta.

Sensor de presencia
(e.d. *CONVEYOR_OBJ_SEN*=1
si objeto en cinta)

Avanza la cinta si s.d.
CONVEYOR_FWD=1

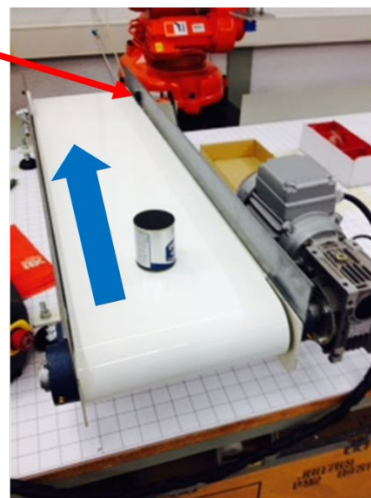


Fig. 1: Cinta transportadora

2. Cuando se detecte que un objeto ha llegado al final de la cinta hay que pararla inmediatamente (desactivando la señal *CONVEYOR_FWD* poniéndola a 0) ya que en esa posición están definidas las posiciones de agarre de los objetos. Estas posiciones están definidas mediante las variable RAPID de posición robtarget *Pos_pick_Lata* y *Pos_pick_Brick*. Por encima de ellas hay otras posiciones: *Pos_prepick_Lata* y *Pos_prepick_Brick*, que son las posiciones de aproximación para poder realizar el agarre.
3. Para poder agarrar los productos el robot cuenta con una ventosa de vacío que se controla con la salida digital *GRIPPER_CLOSE*. Para activar el sistema de vacío la señal se debe poner a nivel bajo (0). Para desactivar la ventosa y que suelte el objeto hay que poner la señal a nivel alto (1).
4. El sistema de inspección por visión artificial es el encargado de realizar el control de calidad y la detección de la clase de objeto que está en la cinta. Si el producto es una lata se activará (se pondrá a un nivel alto) una entrada digital del IRC5: *objeto_Lata*. Si por el contrario se trata de un brick de leche se activará la señal *objeto_Brick*.
5. Si el robot detecta que es un brick deberá cogerlo de la posición *Pos_pick_Brick* y llevarlo a la zona de empaquetado de bricks de leche. Si por el contrario el producto es una lata de conserva se tendrá que coger de la posición definida por *Pos_pick_Lata* y posteriormente se deberá llevar a la zona de empaquetado de esta clase de objetos. Para acceder a dichas zonas se han definido 2 posiciones: *Pos_place_Brick* y *Pos_place_Lata*.



Fig. 2: Robot y posiciones de la aplicación de empaquetado

6. Como en la posición de agarre, se han definido unas posiciones de aproximación. Para la zona de empaquetado de bricks se ha definido la posición *Pos_preplace_Brick*. Para la zona de empaquetado se ha definido la posición *Pos_preplace_Lata*. Además, se ha definido una posición adicional: *Pos_origen*. Sería conveniente que el robot se desplazara desde la cinta a *Pos_origen* y posteriormente se mueva a la zona de empaquetado de las 2 clases de objetos.

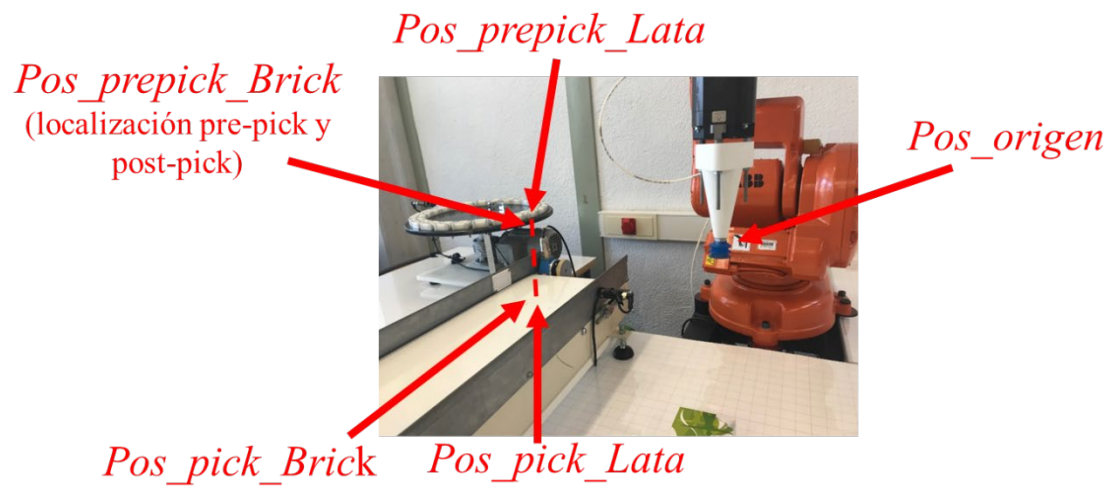


Fig. 3: Posiciones de la aplicación de pick

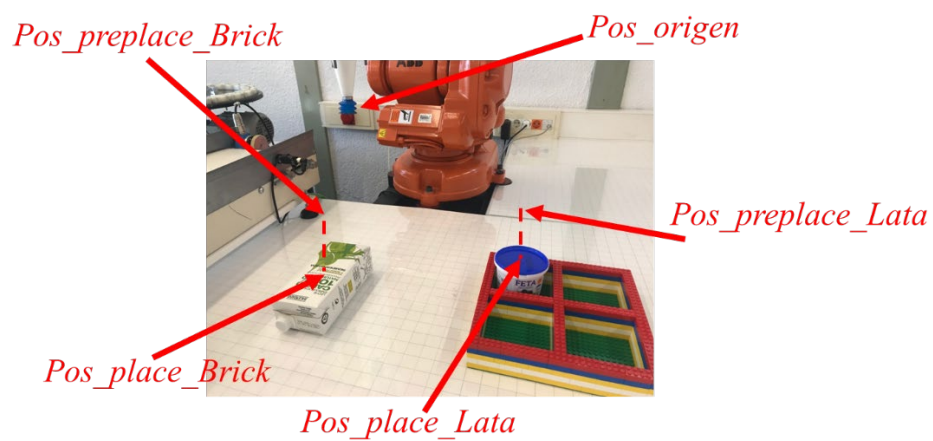


Fig. 4: Posiciones de la aplicación de place

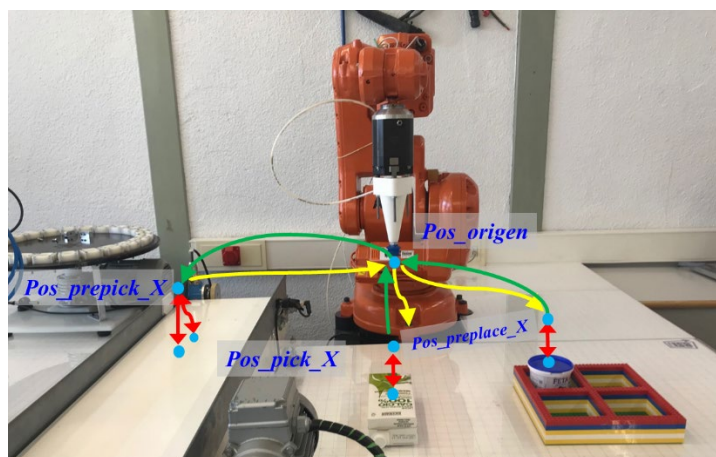


Fig. 5: Planteamiento general de los movimientos de la aplicación

3. Resolución del problema

Para resolver el problema, el procedimiento que hay que seguir es el siguiente:

1. Hay que desplazar al robot hasta la posición de origen (*Pos_origen*) con un movimiento rápido punto a punto. Como además el robot no transporta nada, su velocidad debe ser alta (300mm/s)
2. La cinta debe poner en marcha de avance para que los objetos lleguen a la etapa de inspección. Para activar una señal digital RAPID tiene varias opciones. La primera forma de hacerlo es mediante el comando “**Set** *saldigital*”. Sería igual utilizar “**SetDO** *saldigital, 1*”.
3. Ahora hay que esperar a que tengamos un objeto en el final de la cinta. Esto vendrá indicado porque el sensor de objetos pasará a nivel alto. Para verificar la llegada del objeto se puede utilizar el comando “**WaitDI** *entdigital, 1*”.
4. Cuando el sensor de la cinta detecte que ha llegado un objeto, hay que parar la cinta para que el producto debe de avanzar. Para lograr parar la cinta hay que poner a nivel bajo la señal de la cinta. Esto se puede hacer otra vez con el comando “**SetDO** *saldigital, 0*”, o con el comando “**Reset** *saldigital*”.
5. La entrada digital *objeto_Lata* estará activa a nivel alto cuando el objeto es una lata. De la misma forma, la entrada digital *objeto_Brick* estará activa cuando el objeto es un brick de leche.
6. El robot debe ir a la posición de agarre libre (*Pos_prepick_Lata* o *Pos_prepick_Brick*, en función del tipo de objeto) con un movimiento punto a punto, una velocidad alta (300mm/s) y una zona de aproximación alta (100mm).
7. Para coger el objeto el robot debe desplazarse con un movimiento lineal hasta la posición de agarre, a una velocidad baja (20mm/s) y una aproximación precisa (*fine*). Cuando esté en esa posición el robot deberá activar el vacío de la ventosa (poniendo la señal GRIPPER_CLOSE a nivel bajo), y esperar 0.3s para asegurarse que el objeto esté bien cogido. Para realizar la espera de tiempo en RAPID se tiene la instrucción “**WaitTime**(*tiempo*)”, donde *tiempo* debe estar en segundos.
8. Una vez cogido el objeto, el robot deberá volver a la posición de pre_pick asociada al tipo de objeto con un movimiento lineal y una velocidad baja.

9. Si el producto es un brick, el robot deberá ir primero a la a la posición *Pos_origen*, con una velocidad media de 100mm/s y un movimiento punto a punto. A continuación, el robot deberá ir a la posición *Pos_preplace_Brick* con la misma velocidad y precisión.
10. Una vez alcanzada esta posición, se deberá ir a la posición de empaquetado de bricks con un movimiento lineal y una velocidad baja. El robot volverá a la posición de pre-place con un movimiento lineal a baja velocidad y, por último, deberá volver a la posición de origen.
11. Si el producto es una lata el robot deberá ir primero a la posición de aproximación de empaquetado *Pos_origen* y luego a la posición pre_place de empaquetado de latas (*Pos_preplace_Lata*).
12. Por último, con una instrucción de movimiento lineal y velocidad baja se deberá llevar el objeto a la posición de empaquetado. A continuación, el robot deberá subir de nuevo a la posición de pre_place con un movimiento lineal y velocidad lenta.
13. Por último, volverá a la posición de origen con una velocidad alta.
14. Como es una aplicación continua, el robot deberá volver a ejecutar este procedimiento de forma indefinida.

En la parte de inicialización del programa RAPID nos tenemos que asegurar de llevar el robot a la posición de origen. Para ello se deberá programar un movimiento punto a punto. Además, en esta inicialización el programa deberá asegurarse que todas las entradas que indican el tipo de objeto (*objeto_Lata*, *objeto_Brick*, etc. están a nivel bajo).

Recordad que con la programación de los botones del FlexPendant, la señal *objeto_Lata* u *objeto_Brick* pasan a nivel alto. Por ello el programa del robot será en encargado de poner a nivel bajo las señales del tipo de objeto una vez se ha hecho el pick del objeto. De esta forma el sistema estará preparado para poder procesar bien el siguiente objeto que llegue por la cinta.

En la práctica se deberá programar y simular primero la resolución del problema de automatización con una estación robótica basada en RobotStudio. Una vez verificado que funciona correctamente en la simulación, se transferirá el programa RAPID desarrollado al robot real para comprobar su correcto funcionamiento sobre el sistema real.

4. Trabajo a realizar

En la práctica se deberá programar y simular primero la resolución del problema de automatización con una estación robótica simulada basada en RobotStudio. Para ello se deberá abrir la estación *p2_celula_pick&place_R2021.rsstn* que está en el fichero comprimido *p2_RobotStudio.zip* del PoliformaT de la asignatura. En esta estación se tiene modelado el entorno del robot (las mesas, el soporte, la cinta giratoria, la cinta transportadora, la ventosa, etc.). Tal como se comentará posteriormente, en esta estación hay que hacer algunas etapas de configuración para poderla utilizar con la simulación de la aplicación de pick&place.

Una vez que la estación de RobotStudio esté configurada se tendrá que cargar el programa base de RAPID en la controladora del robot, y se tendrá que completar la programación de la aplicación.

En el archivo comprimido *p2_rapid_simulada.zip* del PoliformaT hay 3 ficheros:

- *p2_2023_simulada.pgf*: programa RAPID de la práctica
- *Module2023_simulada.mod*: módulo que tiene la definición de las posiciones para el robot simulado
- *MainModule.mod*: módulo con la rutina main y que se tendrá que completar con la funcionalidad de la aplicación de pick&place.

La programación de los movimientos y la lógica de la aplicación se deberá hacer íntegramente en *MainModule.mod*, por lo que no se deberá modificar nada en *Module1.mod*.

Una vez se haya programado la aplicación de pick&place y se haya verificado que ésta funciona correctamente en la simulación, se deberá transferir el programa RAPID desarrollado al robot del laboratorio IRB140 para ejecutarlo en la célula real. Para ello se tendrá que descargar el archivo comprimido del PoliformaT *p2_rapid_real.zip*. Después de descomprimir este archivo se podrán ver dos ficheros:

- *p2_2023_real.pgf*: programa RAPID de la práctica
- *Module2023_real.mod*: módulo con las posiciones del robot real.

Lo único que se tendrá que hacer es añadir en la carpeta donde están estos ficheros el módulo *MainModule.mod* realizado para el robot simulado y descargar el programa en la controladora del robot real.

A continuación se detallan los pasos que hay que hacer para realizar esta aplicación.

4.1. Robot simulado

4.1.1. Desarrollo célula robotizada simulada

Como se ha comentado en el apartado anterior, para poder realizar la simulación del sistema se debe partir de la estación de RobotStudio *p2_celula_pick&place_R2022.rsstn* que se encuentra en la carpeta comprimida *p2_RobotStudio.zip* de la 2ª práctica del PoliformaT de la

asignatura. Para ello, descargar dicho fichero, descomprimirlo y abrir la estación de RobotStudio.

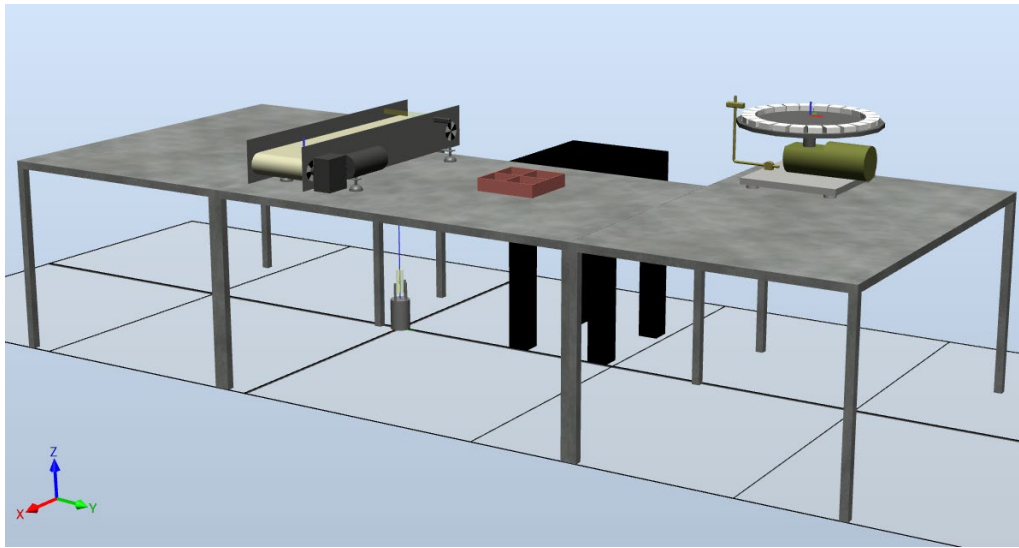


Fig. 6: Estación de RobotStudio para la realización de la aplicación

En la hoja de *Diseño* del menú *Posición Inicial* se puede ver los distintos elementos que componen la estación: la ventosa, las mesas de trabajo, el soporte donde irá subido el robot, las cintas y los objetos a manipular.

A continuación añadir un robot y una unidad de control. Para ello se deberá acceder al menú *Biblioteca ABB* y seleccionar el robot IRB140. Para poder subir el robot al soporte se tiene que seleccionar el robot en la hoja *Diseño*, y haciendo click con el botón derecho del ratón, indicar que se quiere fijar su posición. La nueva posición deberá ser X: -250; Y: 750; Z: 760.

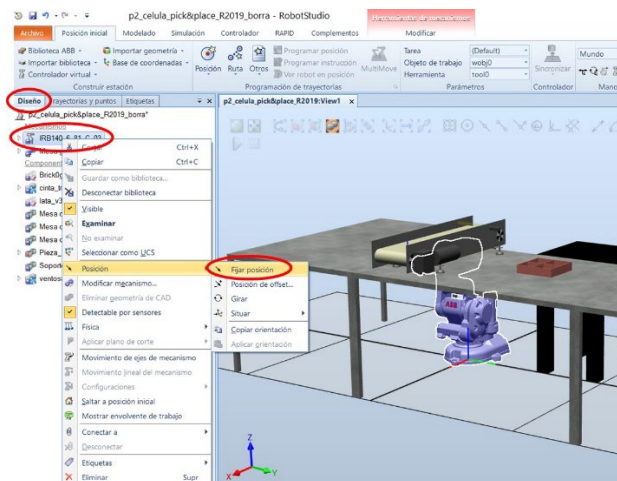


Fig. 7: Cambio de posición del robot IRB140 en la célula robotizada

En la lista de componentes de la estación se tiene la librería *ventosa_giu* que permitirá simular el elemento terminal del robot. Para poderla utilizar, primero se deberá conectar al robot. Eso se consigue arrastrando con el ratón la ventosa al mecanismo del robot.

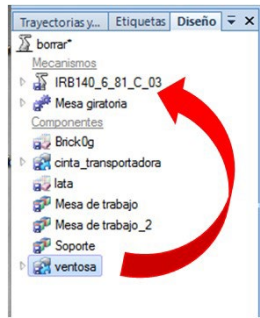


Fig. 8: Conexión del componente *ventosa* en el robot

Después de confirmar afirmativamente que se desea actualizar la posición de la ventosa, se podrá comprobar que ésta se ha posicionado en el extremo del robot.

Para acabar con la configuración de la célula robótica se debe añadir un controlador. Para ello en el menú *Posición Inicial* se debe seleccionar el botón *Controlador Virtual* y, en la ventana que aparece, se debe escoger la opción *Desde diseño*, para crear un controlador a partir del diseño existente. RobotStudio detectará que en la estación hay tres mecanismos: la mesa giratoria, la cinta transportadora y el robot IRB140_6_81_C03. Para la práctica se debe tener marcado únicamente el robot, verificando que tanto la cinta como la mesa estas sin seleccionar.

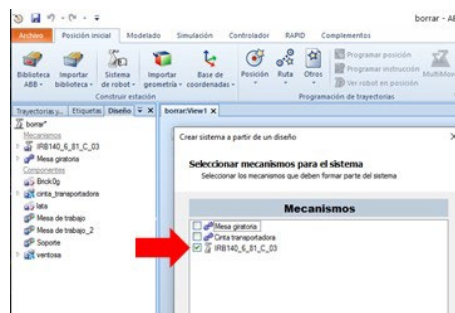


Fig. 9: Ventana de la generación de la controladora del robot

Antes de poder realizar la simulación de la célula robotizada, el robot virtual deberá estar configurado con las entradas y salidas digitales necesarias para la programación de la aplicación industrial. Para ello deberá tener exactamente las mismas señales que el sistema real del laboratorio. En concreto, el robot tiene en la tarjeta DeviceNet de entradas y salidas las siguientes salidas digitales físicas:

- *CONVEYOR_FWD*
- *GRIPPER_CLOSE*

Además, cuenta también con la entrada digital:

- *CONVEYOR_OBJ_SEN*

Debido a esto, en el controlador de la estación simulada de RobotStudio se deberán dar de alta estas señales.

Además de las señales físicas, el robot del laboratorio tiene dadas de alta una serie de señales virtuales. Por ello en el robot de la estación simulada se deberán tener las mismas señales. En concreto se deberán generar dos entradas digitales:

- *objeto_Lata*
- *objeto_Brick*

y dos salidas virtuales:

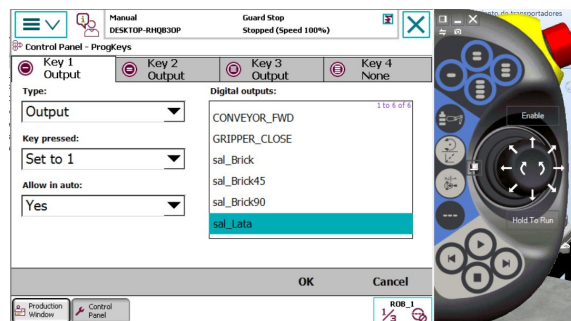
- *sal_Lata*
- *sal_Brick*

Una vez generadas las entradas y salidas virtuales, será necesario realizar una conexión cruzada (*cross connection*) entre ellas de forma que, cada vez que se active la salida virtual, provocará la activación de la entrada. En concreto para esta práctica se proponen las conexiones cruzadas siguientes:

- *Conexión 1:*
Resultante: objeto_Lata
Actor 1: sal_Lata
- *Conexión 2:*
Resultante: objeto_Brick
Actor 1: sal_Brick

A continuación, para poder manipular estas señales virtuales de una forma cómoda y directa se deben programar las teclas del FlexPendant, utilizando la siguiente asignación:

- *ProgKey 1:*
Type: Output
Digital output: sal_Lata
Key pressed: Set to 1
Allow in auto: Yes
- *ProgKey 2:*
Type: Output
Digital output: sal_Brick
Key pressed: Set to 1
Allow in auto: Yes



Lo último que es necesario hacer con las señales de entrada y salida del sistema es establecer la conexión lógica entre las distintas señales. Para ello, en el menú *Simulación*, se debe escoger la opción *Lógica de Estación* que se encuentra en la pestaña *Lógica de simulación*. Seleccionando *Diseño* se podrán ver los componentes de la estación. Para establecer las conexiones hay que pulsar sobre la ventana del combo del sistema del robot (*Sistema8* en este caso), e ir añadiendo una a una las señales del robot que se necesiten. Una vez se tienen, se deberán unir estas señales de la estación. Para ello se puede utilizar el ratón de una forma análoga a como se hace con Simulink o Labview, e ir uniando las entradas de un elemento de la célula con las salidas de las otras.

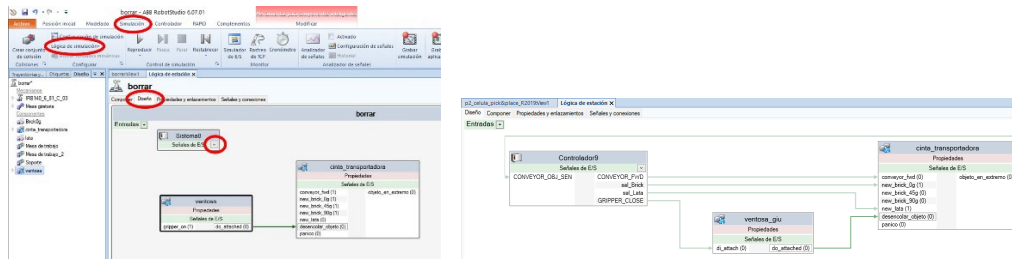


Fig. 10: Configuración de las señales de la estación

Tal y como se aprecia en la Fig. 10, las conexiones que se necesitan son:

- *CONVEYOR_FWD* (robot) \Rightarrow *conveyor_fwd* (cinta)
- *sal_Lata* (robot) \Rightarrow *new_lata* (cinta)
- *sal_Brick* (robot) \Rightarrow *new_brick_0g* (cinta)
- *GRIPPER_CLOSE* \Rightarrow *di_attach* (ventosa_giu)
- *Objeto_en_extremo* (cinta) \Rightarrow *CONVEYOR_OBJ_SEN*

4.1.2. Desarrollo de la práctica robot virtual

Una vez realizada la configuración de la célula robótica ya se puede empezar con la programación en RAPID que realice la aplicación de la automatización del empaquetado de objetos.

Para ello se debe descargar y descomprimir el fichero *p2_rapid_simulada.zip*. En él hay 3 ficheros:

- *p2_2023_simulada.pgf*
- *Module2023_simulada.mod*
- *MainModule.mod*

El primero es el fichero que indica a la controladora qué módulos debe cargar para la ejecución del programa.

El fichero *Module2023_simulada.mod* contiene las posiciones para el robot virtual de RobotStudio. Estas posiciones son las siguientes:

- *Pos_origen*: será la posición de origen (descanso) del robot.
- *Pos_pick_Lata*: será la posición donde se encuentran las latas se deben empaquetar.
- *Pos_pick_Brick*: será la posición donde se encuentran los bricks se deben empaquetar.
- *Pos_prepick_Lata*: será una posición que se encuentra 80mm por encima de *Pos_pick_Lata*.
- *Pos_prepick_brick*: será una posición que se encuentra 80mm por encima de *Pos_pick_Brick*.
- *Pos_place_Lata*: será la posición de empaquetado donde hay que llevar las latas de conserva.
- *Pos_preplace_Lata*: será una posición que se encuentra 80mm por encima de *Pos_place_Lata*.
- *Pos_place_Brick*: será la posición de empaquetado donde hay que llevar los bricks.
- *Pos_preplace_Brick*: será una posición que se encuentra 80mm por encima de *Pos_place_Brick*.

El fichero *MainModule.mod* es el fichero en el que se debe programar la rutina RAPID para la realización de la aplicación de pick&place del robot.

A continuación, hay que cargar el programa en la controladora del robot. Para ello se debe pulsar, teniendo seleccionado el menú RAPID) con el botón de la derecha del ratón sobre la tarea T_ROB1 de RAPID e indicar la opción “Cargar programa” e indicar el fichero *p2_2023_simulada.pgf* que se acaba de descomprimir.

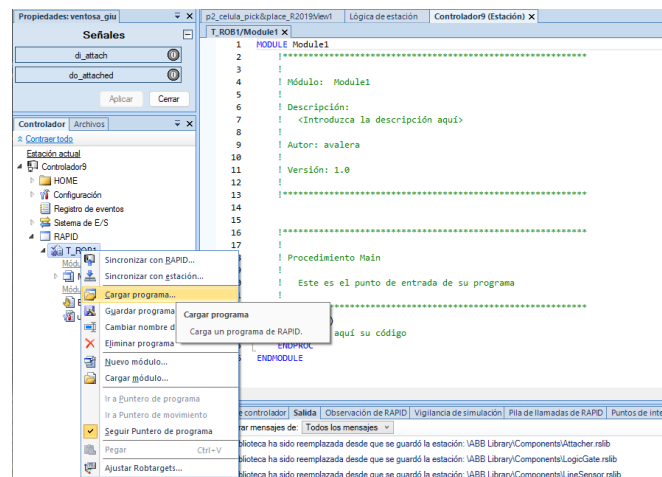


Fig. 11: Carga del programa en el controlador

Una vez completado el programa, para simular la estación hay que pulsar sobre el botón *Reproducir* del menú *Simulación*. Una vez arrancada la simulación hay que pulsar sobre el botón programable (I) del FlexPendant del robot si se desea que aparezca una lata. Al hacer esto, además de generar una lata nueva, pondrá a nivel alto la señal *objeto_Lata*, por lo que el controlador del robot podrá saber que se trata de una lata. Si en vez de una lata se desea que aparezca un brick se deberá pulsar sobre el botón (II) del FlexPendant, activándose a nivel alto la señal *objeto_Brick*.

Cuando el objeto llegue al final de la cinta se activará la entrada digital *CONVEYOR_OBJ_SEN*. Cuando esto ocurra el programa del robot deberá parar la cinta e ir a la posición de agarre en función de la clase de objeto. El tipo de objeto estará definido por qué señal esté a nivel alto (*objeto_Lata* u *objeto_Brick*).

Una vez el robot ha cogido el objeto deberá dejarlo en la posición de place asociada al tipo de objeto (*Pos_place_Lata* o *Pos_place_Brick*).

De esta forma, el trabajo que hay que hacer es programar el robot virtual para que la célula robotizada realice el empaquetado individual de los objetos.

4.2. Robot real

Una vez verificado el correcto funcionamiento del robot virtual hay que descargar y descomprimir el fichero *p2_rapid_real.zip*. En él hay 2 ficheros (*p2_2023_real.pgfy* *Module2023_real.mod*). En el módulo están declaradas las mismas posiciones que para el robot virtual, lo único que cambia son las ubicaciones físicas de las posiciones. Debido a que en este módulo están las posiciones a las que irá el robot real, no se debe hacer ningún cambio en este fichero porque podría provocar que el robot colisionase con algún elemento de la célula (laterales de la cinta, con la mesa, etc.).

A continuación, hay que añadir (en la carpeta donde se haya descomprimido estos ficheros) el programa *MainModule.mod* desarrollador para el robot virtual del apartado anterior. Por último, cargar este programa en el robot real y ejecutarlo.

4.2.1. Desarrollo de la práctica con el robot real

Los pasos a seguir en esta 2ª parte de la práctica serán los siguientes:

1. Cargar el programa RAPID en la unidad de control del robot real
2. En modo MANUAL, ejecutar instrucción a instrucción y verificar que el robot puede realizar correctamente todos los movimientos sin colisionar con nada del entorno.
3. Una vez validado, cambiar el modo de funcionamiento del robot a AUTO, y verificar que el robot funciona correctamente en modo continuo.

4.3. Empaquetado de 4 latas

En este apartado se va a considerar que, en vez de empaquetar cada lata de forma individual, se disponen de cajas donde caben 4 latas. Modificar el programa para que rellene las cajas considerando que la separación entre una ubicación y la siguiente deber ser de 115mm.

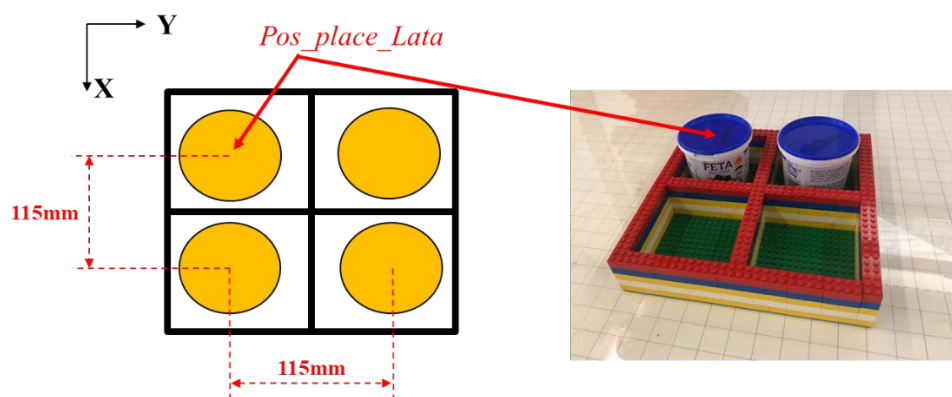


Fig. 12: Propuesta empaquetado 4 latas

4.4. Empaquetado de 8 latas

En este apartado se debe considerar que para empaquetar las latas se disponen de cajas donde caben 8 unidades. La caja tiene las 4 ubicaciones de la Ampliación 1, pero en este caso hay que apilar 2 latas por cada hueco. Modificar el programa para que rellene el primer nivel y luego empaque el segundo nivel. Para ello hay que considerar la altura de las latas (la separación entre una ubicación y la siguiente en el plano Z) deber ser de 80mm.

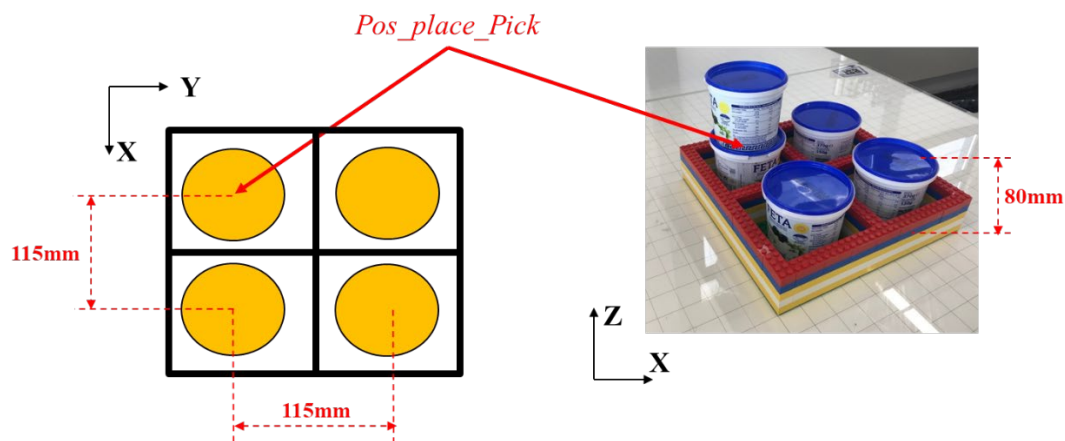


Fig. 13: Propuesta empaquetado 8 latas

4.5. Ampliación 1

Hasta ahora se ha considerado que los bricks llegaban al sensor *CONVEYOR_OBJ_SEN* con una orientación de 0°, por lo tanto, el robot podía cogerlos y llevarlos a la posición de empaquetado directamente. En esta ampliación se va a considerar que el objeto puede llegar con orientaciones diferentes. Si se tuviera operativo un sistema de visión para realizar el control de calidad podría indicar el tipo de objeto (lata o brick) y la orientación con que llegan los bricks.

En esta ampliación se va a considerar que los bricks pueden llegar con 3 orientaciones: 0°, 45° o 90°.



Fig. 14: Orientaciones de los bricks

Para poder especificar a la controladora del robot con qué orientación llegan se tienen que crear 2 entradas y 2 salidas digitales virtuales nuevas. Las entradas digitales serán:

- *objeto_Brick45*
- *objeto_Brick90*

Las salidas digitales nuevas serán:

- *sal_Brick45*
- *sal_Brick90*

Para poder trabajar con ellas se va a hacer lo mismo que en el apartado 4.1.1: se tendrá que hacer 2 enlaces cruzados entre las entradas y las salidas y luego se tendrán que programar los botones del FlexPendant.

Los enlaces cruzados serán los siguientes:

- *Conexión 3:*
Resultante: objeto_Brick45
Actor 1: sal_Brick45
- *Conexión 4:*
Resultante: objeto_Brick90
Actor 1: sal_Brick90

Para esta ampliación, además de las teclas 1 y 2 que indicarán que el objeto es una lata o un brick (con una orientación de 0°), se deberá realizar la programación de las teclas del FlexPendant siguiente:

- *ProgKey 3:*
Type: Output
Key pressed: Set to 1
Allow in auto: Yes
Digital output: sal_Brick45
- *ProgKey 4:*
Type: Output
Key pressed: Set to 1
Allow in auto: Yes
Digital output: sal_Brick90

Por lo tanto, ya tenemos la libertad de escoger la orientación de llegada de los objetos, y mediante el FlexPendant podemos especificar a la controladora dicha orientación en cuestión. Ahora lo que hace falta es ver cómo se puede programar el robot para que, independientemente de la orientación de llegada, el empaquetado se realice correctamente y los objetos se dejen con una orientación de 0°.

Como se sabe, a la hora de programar las orientaciones RAPID utiliza los cuaterniones. Éstos se pueden calcular mediante el giro de un ángulo θ de un vector $[x_0, y_0, z_0]$ a partir de las expresiones siguientes:

$$q_0 = \cos \frac{\theta}{2} \quad q_1 = x_0 \cdot \sin \frac{\theta}{2} \quad q_2 = y_0 \cdot \sin \frac{\theta}{2} \quad q_3 = z_0 \cdot \sin \frac{\theta}{2}$$

Para que la herramienta del robot (ventosa) pueda coger correctamente los objetos de la cinta, ésta deberá tener orientación normal (perpendicular) al plano X-Y de la cinta, de forma que el eje Z apunte hacia el suelo. Esta orientación se puede lograr haciendo un giro de 180° respecto del eje Y. El vector será $[0, 1, 0]$, por lo que el valor de los cuaterniones deberá ser el siguiente:

$$q_0 = \cos \frac{180^\circ}{2} = 0 \quad q_1 = 0 \cdot \sin \frac{180^\circ}{2} = 0 \quad q_2 = 1 \cdot \sin \frac{180^\circ}{2} = 1 \quad q_3 = 0 \cdot \sin \frac{180^\circ}{2} = 0$$

El caso de los bricks que lleguen con una orientación distinta de 0° esto es un poco más complicado puesto que además del giro de 180° alrededor del eje Y, será necesario también considerar la orientación con que llegan estos objetos por la cinta, ya que independientemente de esta orientación original, los objetos se deberán dejar todos en el mismo sentido.

La orientación con que llegan los bricks por la cinta se puede representar como un giro de un ángulo α respecto del eje Z, lo que proporcionará los siguientes valores de cuaterniones:

$$q_0 = \cos \frac{\alpha}{2} \quad q_1 = 0 \cdot \sin \frac{\alpha}{2} = 0 \quad q_2 = 0 \cdot \sin \frac{\alpha}{2} = 0 \quad q_3 = 1 \cdot \sin \frac{\alpha}{2} = \sin \frac{\alpha}{2}$$

Por lo tanto, para calcular la orientación que nos permita realizar el giro de 180° alrededor del eje Y y posteriormente el giro del ángulo α alrededor del eje Z, se tendrá que calcular el producto de los dos cuaterniones relacionados con estos giros. El producto de dos cuaterniones se puede obtener mediante la expresión siguiente:

$$\begin{aligned} [a, b, c, d] \cdot [e, f, g, h] = \\ = [ae - bf - cg - dh, af + be + ch - dg, ag + ce + df - bh, ah + de + bg - cf] \end{aligned}$$

Para el caso de esta práctica, el resultado será el siguiente:

$$\left[\cos \frac{\alpha}{2}, 0, 0, \sin \frac{\alpha}{2} \right] \cdot [0, 0, 1, 0] = \left[0, \sin \frac{\alpha}{2}, \cos \frac{\alpha}{2}, 0 \right]$$

Por lo tanto, si el objeto llega al punto de agarre con giro α , el robot deberá tener en cuenta esta orientación para poder dejarlo de una forma correcta. Una forma simple de hacerlo es hacer que el robot llegue a la posición de agarre de los bricks con la orientación calculada en la ecuación anterior. Eso provocará que cuando la ventosa llegue a la posición de agarre, éste estará girada con un ángulo α . Para dejarlo en la posición de empaquetado con la orientación adecuada no habrá que hacer nada más, puesto que en esta posición ya tiene programada una orientación de dejada de 0° .

4.6. Ampliación 2

Hasta ahora se ha considerado que los bricks se dejaban en la misma ubicación. En esta ampliación se deberá modificar el programa RAPID para que se empaqueten en pallets de 4 objetos: se tendrán 2 objetos por piso, y una altura de 2 pisos.

Los objetos del piso inferior estarán orientados con un ángulo de 0° . Sin embargo, los objetos del piso superior deberán orientarse con un ángulo de 90° .

Para realizar el empaquetado hay que tener en cuenta que el tamaño de los bricks es: 195x90x60. Además, se deberá dejar una distancia de seguridad entre un brick y el contiguo de 15mm.



Fig. 15: Propuesta empaquetado 4 bricks