



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

Trabajo Fin de Grado
CURSO 2017/18

*SIMULACIÓN DE UN PROCESO INDUSTRIAL PARA
LA INCORPORACIÓN DE ROBOTS*

Grado en Ingeniería Mecánica

ALUMNA

Rosalía Chans Veres

TUTOR

Dr. Diego Crespo Pereira

FECHA

JUNIO 2018

Resumen

En el presente proyecto se recoge el estudio de la incorporación de robots reales en el software de simulación FlexSim. Para ello se realizará un previo estudio de las características que definen a los robots industriales así como de su cinemática. Se realizará la incorporación del robot Baxter a un entorno en el que realizará una tarea concreta y se compararán los resultados obtenidos en FlexSim con los reales del propio robot realizando la misma labor. Además, se implementará un caso real de una línea de producción de empaquetado de kits en la cual se introducirán modelos de brazos robóticos KUKA reales y se estudiarán las posibilidades para mejorar dicho proceso. Finalmente se extraerán los resultados obtenidos.

No presente proxecto recóllese o estudo da incorporación de robots reais no software de simulación FlexSim. Para iso realizarase un previo estudo das características que definen aos robots industriais así como da súa cinemática. Realizarase a incorporación do robot Baxter a un entorno no que se realizará unha tarefa concreta e compararanse os resultados obtidos en FlexSim cos reais do propio robot realizando a mesma labor. Ademáis, implementarase un caso real dunha liña de produción de empaquetado de kits na cal se introducirán modelos de brazos robóticos KUKA reais e estudaranse as posibilidades para mellorar dito proceso. Finalmente extraeranse os resultados obtidos.

This bachelor thesis summarizes the study of the incorporation of real robots in the simulation software FlexSim. It presents a previous study of the characteristics which define the industrial robots and its kinematics. Then, it describes a Baxter robot model developed in Flexsim that performs a specific task. The same task was implemented in a real Baxter so the task durations in both reality and simulation could be compared. In addition, a real case of a pick and place production line was simulated to evaluate the implementation of KUKA robotic arms to automate it. Finally, the conclusions section discusses the simulation results and future lines of work.



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE GRADO
CURSO 2016/17**

*SIMULACIÓN DE UN PROCESO INDUSTRIAL PARA
LA INCORPORACIÓN DE ROBOTS*

Grado en Ingeniería Mecánica

Documento

MEMORIA

ÍNDICE

Índice de figuras	8
índice de tablas.....	10
1 Introducción	11
2 Objetivos.....	13
3 Antecedentes	15
3.1 Simulación	15
3.2 Robótica industrial	18
3.3 Simulación de robots.....	22
4 Robot en FlexSim	29
4.1 Descripción robot.....	29
4.2 Movimiento de flowitems con el robot.....	30
4.3 Otras propiedades del robot	32
5 Configuración del robot colaborativo baxter	35
5.1 Descripción del Baxter	35
5.2 Introducción al Baxter en FlexSim	37
5.3 Obtención de resultados	40
6 Introducción brazo robótico en proceso de empaquetado de kits.....	45
6.1 Metodología	45
6.1.1 Descripción del proceso	45
6.1.2 Elección brazo robótico	46
6.1.3 Implementación Quantec Ultra.....	49
6.1.4 Funcionamiento de cada puesto	52
6.2 Obtención de resultados	64
6.2.1 Resultados del escenario principal.....	64
6.2.2 Análisis de sensibilidad.....	67
6.2.3 Cambio de disposición.....	68
6.3 Seguridad.....	72
6.4 Realidad virtual.....	73
7 Conclusiones.....	75
8 Bibliografía.....	77

ÍNDICE DE FIGURAS

Figura 1. Componentes robot industrial. (http://educalab.es/intef).....	14
Figura 2. Tipos de articulaciones en robótica industrial (https://roboticapuj.es).	20
Figura 3. Campo de trabajo de un robot industrial (ABB Robotics 2017).	21
Figura 4. Software de simulación Tecnomatix. Visualización del entorno 3D (SIEMENS 2017).	23
Figura 5. Software de simulación robótica Delmia. Sketcher (Catia Community 2018).	24
Figura 6. Software de simulación robótica Delmia. Robot industrial y su trayectoria (Catia Community 2018).....	25
Figura 7. Gazebo. Robot industrial modelado (Fetchrobotics 2016).	26
Figura 8. Ejemplo de implementación de un robot en KUKA.Sim (KUKA Robotics 2018).....	26
Figura 9. Vistas Robot en FlexSim y eje de coordenadas (Elaboración propia).	29
Figura 10. Herramienta FlexSim Define Motion Paths (Elaboración propia).	31
Figura 11. Herramienta FlexSim Geometry (Elaboración propia).	32
Figura 12. FlexSim propiedades Robot. General (Elaboración propia).	33
Figura 13. Robot Baxter (Baxter España 2018).....	35
Figura 14. Vistas Baxter. Módulo Animations FlexSim (Elaboración propia).	36
Figura 15. Módulo 3D de FlexSim. Resalte articulación 1 (Elaboración propia).	36
Figura 16. Articulación 2 en SketchUp (Elaboración propia).....	38
Figura 17. Robot predeterminado de FlexSim. Animations (Elaboración propia).	39
Figura 18. Parámetros del robot. Explore Tree (Elaboración propia).	39
Figura 19. Posiciones 1, 2 y 3 Baxter (Elaboración propia).	40
Figura 20. Sistema de coordenadas Baxter (Elaboración propia).	41
Figura 21. Process Flow Baxter (Elaboración propia).....	42
Figura 22. Posiciones 1, 2 y 3 Baxter en FlexSim (Elaboración propia).	42
Figura 23. Dimensiones KR Quantec Ultra (KUKA Robotics 2018).	46
Figura 24. Dimensiones KR Quantec Extra (KUKA Robotics 2018).	47
Figura 25. Dimensiones del brazo robótico KUKA 2102-2 (KUKA Robotics 2018).	48
Figura 26. Vista en planta de un puesto de trabajo del proceso (Elaboración propia).....	49
Figura 27. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 1 (Elaboración propia).....	50
Figura 28. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 2 (Elaboración propia).....	50
Figura 29. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 3 (Elaboración propia).....	51
Figura 30. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 4 (Elaboración propia).....	51
Figura 31. Vista en planta de todos los puestos (Elaboración propia).	52
Figura 32. Diagrama Comprobación Colocación (Elaboración propia).	54
Figura 33. Diagrama de flujo general de la función del brazo robótico (Elaboración propia).	55

Figura 34. Posición inicial brazos robóticos (Elaboración propia).....	56
Figura 35. Travel Ida (Elaboración propia).	56
Figura 36. Sistema de coordenadas de una pila de piezas (Elaboración propia).	57
Figura 37. Travel vuelta (Elaboración propia).	59
Figura 38. TravelNoChoque (Elaboración propia).	59
Figura 39. Momento crítico de colisión (Elaboración propia).....	60
Figura 40. Primera prueba trayectorias (Elaboración propia).....	61
Figura 41. Segunda prueba trayectorias (Elaboración propia).....	61
Figura 42. Tercera prueba trayectorias (Elaboración propia).	62
Figura 43. Process Flow parte brazo robótico (Elaboración propia).	63
Figura 44. Gráficos ocupación robots inferiores (Elaboración propia).	65
Figura 45. Gráficos ocupación robots superiores (Elaboración propia).....	65
Figura 46. Gráfica TviajeRobot en función del tiempo (Elaboración propia).	66
Figura 47. Valores TviajeRobot (Elaboración propia).....	66
Figura 48. Gráfica tasa de producción de cajas (Elaboración propia).	67
Figura 49. Valores de salida de kits por unidad de tiempo (Elaboración propia).	67
Figura 50. Puesto con un solo robot (Elaboración propia).....	69
Figura 51. Tasa de producción con un solo robot (Elaboración propia).	69
Figura 52. Nueva disposición puestos (Elaboración propia).	70
Figura 53. Disposición de las barreras para los puestos 1 y 2 (Elaboración propia).	72
Figura 54. Disposición de las barreras para los puestos 3, 4, 5 y 6 (Elaboración propia).	73
Figura 55. Prueba con las gafas de realidad virtual (Elaboración propia).....	74

ÍNDICE DE TABLAS

Tabla 1. Velocidades máximas articulaciones Baxter (Elaboración propia).....	37
Tabla 2. Comparación resultado tiempos Baxter (Elaboración propia).....	43
Tabla 3. Velocidades de giro KR Quantec Ultra (KUKA Robotics 2018).....	46
Tabla 3. Velocidades de giro KR Quantic Extra (KUKA Robotics 2018).	47
Tabla 5. Dimensiones adicionales y volumen de brazo robótico KUKA 2102-2 (KUKA Robotics 2018).48	
Tabla 6. Velocidades angulares de KR2102-2 para cada articulación (KUKA Robotics 2018).	48
Tabla 7. Distancias en mm entre robots y pilas (Elaboración propia).....	49
Tabla 8. Velocidades angulares de KR Quantec Ultra aplicadas a cada una de las articulaciones (Elaboración propia).....	51
Tabla 9. Aceleraciones y deceleraciones angulares de KR2102-2 aplicadas a cada una de las articulaciones (Elaboración propia).....	52
Tabla 10. Valores velocidades, aceleraciones y deceleraciones caso 1 (Elaboración propia).	67
Tabla 11. Valores velocidades, aceleraciones y deceleraciones caso 2 (Elaboración propia).	67
Tabla 12. Valores velocidades, aceleraciones y deceleraciones caso 3 (Elaboración propia).	67
Tabla 13. Tasas de producción en los diferentes casos (Elaboración propia).....	68
Tabla 14. Distancias en mm entre robots y pilas para la nueva distribución (Elaboración propia).	70
Tabla 15. Comparación diferentes escenarios (Elaboración propia).	70

1 INTRODUCCIÓN

El estudio del funcionamiento de una línea de producción es un proceso costoso que requiere tomar muchas decisiones que tendrán un impacto crítico en su rendimiento. El coste de una decisión errónea es elevado y, por ello, se han de emplear métodos y herramientas que ayuden a minimizar estos riesgos. El uso de las tecnologías de modelado y simulación, como en el caso estudiado en este proyecto, permite realizar modificaciones en un proceso, o incluso para diseñarlo desde un principio, reduciendo costes y plazos.

El software de simulación permite visualizar el proceso tal y como se llevaría cabo en la realidad. Contiene un gran número de comandos y herramientas que permitirán hacer el entorno lo más realista posible. Si se detectan fallos de cualquier tipo podrán solucionarse y además será posible ver con más claridad la posibilidad de introducir mejoras. Tal es el caso de la introducción de robots en el proceso, a pesar de que muchas tareas se siguen realizando de forma manual en la industria. Especialmente, una gran variabilidad en los procesos y en los medios de producción dificultan la incorporación de robots, aunque si se logra conlleva un gran salto en este tipo líneas de producción.

Actualmente, el concepto de industria 4.0 proporciona un conjunto de técnicas innovadoras en el contexto de procesos de producción. Los avances producidos en el ámbito de la robótica han permitido que, a pesar de existir procesos complejos que contengan gran variedad de tareas específicas con diferentes elementos implicados en las mismas, sea menos costosa su automatización. Los robots serán cada vez más capaces de sustituir a operarios mejorando los tiempos y siendo en muchos casos más precisos.

En este trabajo se introducirá un modelo real de robot Baxter en el software de simulación FlexSim de manera que se modificarán los parámetros necesarios para que realice un movimiento tal y como lo haría en la realidad. Se realizará una tarea concreta con el Baxter de manera experimental y se compararán los resultados obtenidos. Posteriormente, se compararán dichos resultados con los del robot implementado en FlexSim al realizar el mismo movimiento.

Además, en este trabajo se ha estudiado un proceso de empaquetado de kits para la introducción de brazos robóticos en el mismo. Para ello ha sido necesario programarlos de manera que ejecuten las tareas específicas para las cuales han sido encomendados. Se ha descrito la metodología llevada a cabo para realizar la mejora del proceso en el software de simulación FlexSim. Además se han extraído resultados, partiendo de un escenario con datos reales, de manera que se proyecten los avances que conlleva el hecho de haber introducido robots en la línea de producción.

2 OBJETIVOS

El objetivo del presente proyecto es estudiar las posibilidades que ofrece el software de simulación FlexSim a la hora de introducir robots. Se estudiarán los procedimientos necesarios para reflejar de manera real la cinemática propia de los brazos robóticos y las opciones que ofrece a la hora de programar las tareas que se le quieran encomendar a los mismos.

Se llevarán a cabo dos tareas, la primera de ellas consistirá introducir un robot Baxter en el entorno de simulación del software FlexSim. Se le encomendará una tarea que será realizada por el propio robot en la realidad y se compararán los resultados reales con los de la simulación. De esta manera se podrá comprobar hasta qué punto un modelo de un robot en Flexsim es capaz de predecir tiempos reales de operación.

Por otro lado, se implementarán modelos de brazos robóticos reales en una línea de empaquetado de kits de manera que se introduzcan los parámetros que los definen y puedan obtenerse unos resultados reales en cuanto a la productividad del proceso.

3 ANTECEDENTES

3.1 Simulación

INTRODUCCIÓN

La simulación es un método de resolución de problemas que es de gran ayuda para la toma de decisiones sobre procesos industriales en cuanto al diseño y mejora de procesos y productos. Ofrece una alternativa al proceso de experimentación con el sistema, en el cual los factores de costos y seguridad, entre otros, dan lugar a que esta opción no sea viable en muchos casos. Con el fin de superar estos inconvenientes, en la simulación se sustituye el sistema real por un modelo. Dicho modelo es una representación simplificada de un sistema elaborado para comprender, predecir y controlar el comportamiento del mismo.

La evolución que se ha dado en el área de la informática en estos últimos años es la que ha permitido que la técnica de simulación cobre tanta importancia dentro del desarrollo de este tipo de modelos. Con la ayuda del software informático correspondiente, el modelo de simulación es capaz de considerar complejas tareas interrelacionadas y proyectarlas mediante la realización de muchas combinaciones alternativas sin necesidad de actuar sobre el sistema real. De esta manera se alcanzarán los objetivos planteados de la mejor manera posible gracias a la adecuada toma de decisiones que nos va a permitir este método.

Por todas sus ventajas, el proceso de simulación puede ser aplicado en gran cantidad de áreas como plantas industriales, sistemas públicos, construcción, diseño de maquinaria, procesos de manufactura, educación, etc.

Se puede decir, por tanto, que la simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, con la finalidad de aprender el comportamiento del sistema o de evaluar diversas estrategias para el funcionamiento del mismo.

ETAPAS

Debido a su complejidad, cabe diferenciar diferentes etapas dentro del propio proceso de simulación:

- **Formulación del problema**
Como su propio nombre indica, en esta etapa se definirá perfectamente el problema al que habrá que enfrentarse, es decir, concretar de manera precisa el objeto de la simulación.
- **Definición del sistema**
Se conoce como sistema al conjunto de objetos o ideas que están interrelacionados entre sí para la consecución de un fin. Habrá que definir el sistema del proyecto a realizar concretando todos los detalles necesarios para poder proseguir con la simulación.
- **Formulación del modelo**
Consiste en la definición de un modelo de simulación base que contenga los aspectos más relevantes del sistema real, igual que en la etapa anterior, debe quedar todo

perfectamente definido. Se traducirá a lenguaje de simulación la información obtenida en las etapas anteriores y se creará el gráfico que represente al sistema en caso de que sea necesario.

- **Recolección y análisis de datos**
Se recolectará y analizará la información necesaria para el proceso, que puede ser sacada de registros históricos, experimentos de laboratorio o mediciones realizadas en el sistema real (en caso de que lo haya).
- **Generación del modelo preliminar**
En esta etapa se integra toda la información obtenida anteriormente para crear el modelo definido que será lo más próximo a la realidad dentro de lo posible.
- **Verificación del modelo**
Consiste en la comprobación del modelo generado mediante la búsqueda de errores y la comprobación de los parámetros, de manera que todo funcione correctamente.
- **Validación del modelo**
En la validación se llevarán a cabo una serie de pruebas, para ello, se introducirá información de entrada real, de esta manera analizaremos el comportamiento del modelo y se analizarán los resultados.
- **Diseño de experimentos**
Se decidirán y diseñarán los experimentos que van a ser simulados con nuestro modelo para la obtención de los diferentes resultados.
- **Experimentación**
Se simularán los experimentos diseñados en la etapa anterior recogiendo los resultados obtenidos.
- **Interpretación**
Se analizará la sensibilidad del modelo en función de los parámetros que tienen asociados la mayor incertidumbre.
- **Implementación**
Como su propio nombre indica, consiste en implementar el modelo y mostrárselo al cliente.
- **Documentación**
Elaboración de los documentos y manuales necesarios de uso.

VENTAJAS Y DESVENTAJAS DE LA SIMULACIÓN

Como todos los métodos y técnicas de diseño y toma de decisiones, la simulación presenta una serie de ventajas y desventajas a la hora de ejecutarla.

Algunas de las ventajas más significativas son las siguientes:

- Permite anticipar la respuesta del sistema ante los cambios de manera que se puede valorar la validez de nuevas situaciones. De esta manera también evitaremos el cambio en el proceso real directamente, lo cual sería menos económico.

- Se pueden valorar diferentes escenarios para encontrar la mayor eficiencia del proceso así como las mejores condiciones de trabajo.
- Genera mejores soluciones en procesos de mayor complejidad.
- Es un proceso efectivo para el control de costes.
- Permite la modelación de sistemas completos, por lo que se obtienen soluciones totales.

Como desventajas de esta técnica podemos citar las siguientes:

- En problemas sencillos, la simulación puede resultar un método costoso.
- Es necesario un largo período de tiempo para realizar un completo estudio de simulación.
- Se necesita una persona cualificada que domine el software correspondiente.

TIPOS DE MODELOS

Como ya sabemos, a la hora de realizar una simulación es necesario crear un modelo aproximado a la realidad. Existen diferentes tipos de modelos que se clasifican en los siguientes grupos en función a sus características:

- **Estático/Dinámico**
En los modelos estáticos se representan sistemas que no dependen del tiempo, mientras que los dinámicos sí dependen.
- **Determinísticos/Estocásticos**
En los modelos de simulación determinísticos, la solución se obtiene aplicando siempre unas condiciones determinadas, que no dependen del azar, de esta manera si introducimos las mismas entradas, siempre vamos a obtener na solución única.
Por otro lado, en los modelos estocásticos va a haber valores que se obtienen al azar.
- **Discreto/Continuo**
Los modelos de simulación discretos contienen variables que varían a saltos, de forma discontinua.
Sin embargo, en los modelos continuos los cambios son graduales, es decir, las variables varían de forma discontinua.

SOFTWARES DE SIMULACIÓN

Existen un gran número de aplicaciones de simulación, muchos son creados para simular diseños de máquinas o piezas, otros para el diseño de procesos, etc. A continuación se citarán algunos de los más relevantes:

- **ARENA**
Es un software diseñado para la simulación de eventos discretos y para la optimización de procesos complejos. Con él se representa el comportamiento de un sistema complejo con una serie de eventos bien definidos y ordenados en el tiempo. De esta manera se analizará el comportamiento de un proceso.
- **FLEXSIM**
Este software es el que usaremos para llevar a cabo este proyecto y tiene la capacidad de simular eventos discretos de manera que permite modelar, analizar, visualizar y optimizar cualquier elemento industrial, desde procesos de manufactura hasta cadenas de suministro. Contiene un entorno 3D que hace al programa más visual y fácil de comprender, de manera que evita la programación a mano en muchas situaciones.

De este mismo tipo nos encontramos también programas como el PROMODEL.

➤ **ABAQUS Y SOLIDWORKS**

Son software de simulación por elementos finitos que permiten el diseño de piezas y maquinaria con un entorno 3D muy visual.

➤ **DYNAMO**

Es el software más primitivo en materia de simulación dinámica que ha servido de referencia para la creación de otros programas de simulación más avanzados. Permite crear algoritmos personalizados a fin de procesar datos y crear geometrías.

La importancia de este método es tal que gracias a él se han desarrollado trabajo de gran importancia como el regreso del Apolo 13, el cual logro retornar a pesar de sus grandes averías en parte gracias a aplicar este método; las simulaciones de vuelos para entrenar pilotos; o el Proyecto Monte Carlo, en el cual Von Newman y Ulam emplearon las técnicas de simulación para estudiar reacciones nucleares.

A pesar de que ya se han conseguido grandes logros con esta técnica, a día de hoy es necesario seguir con la evolución de la misma para perfeccionar su método y poder llegar a soluciones perfectas que nos permitan alcanzar los objetivos con costes más bajos a los actuales y en plazos más cortos de tiempo.

3.2 Robótica industrial

INTRODUCCIÓN

El desarrollo de la robótica desde su aparición en el siglo XX es debido a las necesidades de automatización dentro de la industria, ya que es un elemento fundamental dentro de las líneas de fabricación en serie y también fue de gran importancia para la manipulación en entornos de trabajo hostiles para el hombre como, por ejemplo, el manejo de materiales radioactivos.

Algunos de los factores clave que han permitido la invención de la robótica han sido los avances tecnológicos de los ordenadores, el desarrollo de la microelectrónica, la disponibilidad de servomecanismos electromecánicos e hidromecánicos fiables y el desarrollo de la teoría de control y de máquinas de control numérico.

La máquina textil programable mediante tarjetas perforadas inventada por Joseph Jacquard en 1801 puede ser considerada como el primer robot industrial, aunque el primer prototipo el primer prototipo de máquina de control numérico en el MIT (Massachusetts Institute of Technology) para controlar una máquina fresadora de 3 ejes fue más relevante.

Los avances que se han dado lugar durante estos últimos años en las herramientas de CAD (Computer Aided Design) han sido de gran importancia para el diseño y desarrollo de la de la robótica.

En general, los robots son herramientas de gran efectividad a la hora de realizar tareas repetitivas y automatizadas, pero el modelo de robot como maquinaria universal que pueda realizar un gran número de tareas diferentes es una idea que a día de hoy aún no se ha conseguido.

DEFINICIÓN

La definición del término “Robot industrial” ha sido realizada por diversas instituciones, aunque una de las más aceptadas es la que redacta la RIA (Robot Institute of America) y dice lo siguiente:

“Un robot es un manipulador reprogramable y multifuncional, diseñado para mover cargas, piezas, herramientas o dispositivos especiales, según varias trayectorias, programadas para realizar diferentes trabajos”.

Es decir, el robot ejecutará un programa almacenado en su memoria que dará lugar a los movimientos de sus partes mecánicas para realizar su función correspondiente. Para cambiar el trabajo que realice el mismo bastará con modificar dicho programa, siempre y cuando pueda realizarlo dentro de sus restricciones mecánicas.

COMPONENTES Y ESTRUCTURA

La estructura de los robots industriales suele estar formada por el mismo conjunto de componentes mecánicos. Tenemos una serie de elementos rígidos conocidos como eslabones y que estarán unidos por las articulaciones. De entre dichos componentes podemos diferenciar, en primer lugar, la base o cuerpo, que será el eslabón fijo; por otro lado, tenemos los eslabones no fijos, de los cuales podemos diferenciar el brazo, la muñeca y el elemento final (gripper). El brazo se encarga del posicionamiento adecuado, mientras que la muñeca se ocupa de la orientación. Finalmente, el elemento final tendrá la función de agarrar, cortar, doblar, agujerear o la operación que le corresponda al objeto procesado.

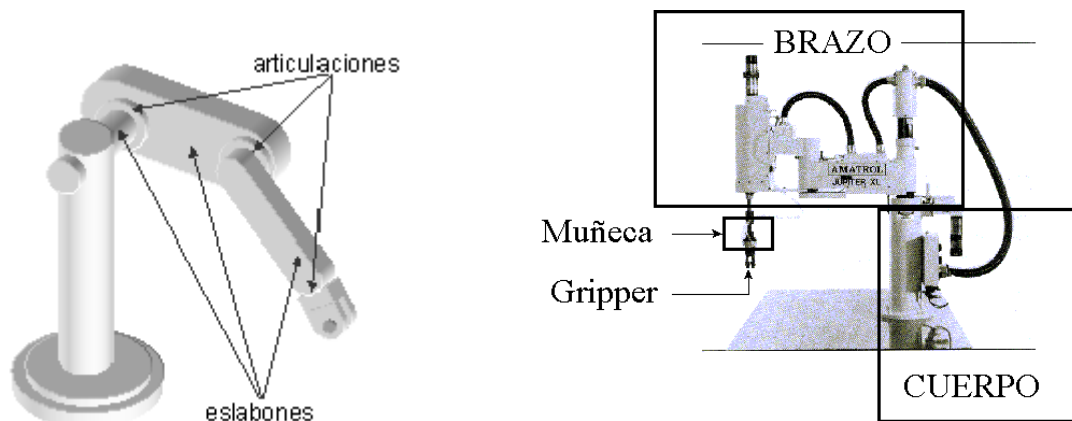


Figura 1. Componentes robot industrial (<http://educalab.es/intef>).

Además, cabe citar que dicho brazo estará compuesto a su vez por los siguientes elementos:

- **Carcasa o chasis:** es la parte externa del robot que estará formada generalmente de acero o aluminio habitualmente.
- **Sistema de accionamiento:** para permitir el movimiento de los eslabones es necesario un sistema de accionamiento que puede ser mediante actuadores, transmisiones o sensores de posición y velocidad.
- **Otros elementos:** como cableado, conectores, fines de carrera, etc.

Existe otra clasificación de los sistemas de accionamiento relacionada con la energía que utilizan, por lo que podemos diferenciar sistemas neumáticos hidráulicos y eléctricos. Para seleccionar el que más nos conviene en cada caso habrá que tener en cuenta parámetros como la potencia, la velocidad, la precisión, etc.

Como bien hemos dicho anteriormente, las articulaciones son los elementos que unen los eslabones entre sí permitiendo el movimiento relativo entre los mismos, dicho movimiento

puede ser de desplazamiento, giro, o una combinación de ambos. Podemos diferenciar pues los siguientes seis tipos de articulaciones:

Tipos de Articulaciones

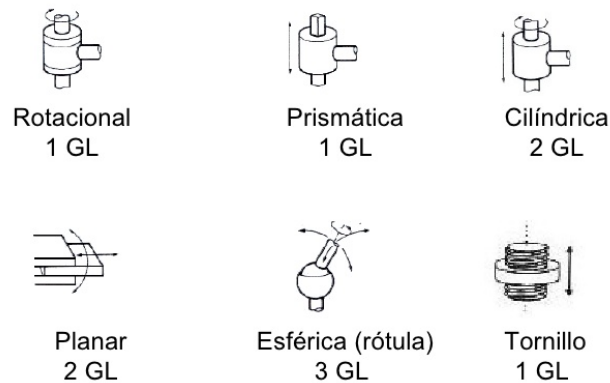


Figura 2. Tipos de articulaciones en robótica industrial
(<https://roboticapuj.es>).

- **Rotacional:** permite el giro relativo entre dos eslabones.
- **Prismática:** permite la traslación relativa entre dos eslabones.
- **Cilíndrica:** permite el giro y la traslación (Rotacional + Prismática).
- **Planar:** permite el desplazamiento en un solo eje de una pieza.
- **Esférica:** permite la rotación en tres direcciones.
- **Tornillo:** el giro de una de las piezas da lugar a la traslación de la otra y viceversa.

A pesar de que son necesarios los 6 grados de libertad para que el robot pueda realizar todos los movimientos posibles, muchas veces no son necesarios debido a la funcionalidad que debe cumplir. Por ello, los dos tipos de articulaciones más comunes son la rotacional y la prismática, ya que con ellas dos los movimientos son más que suficientes en muchos de los robots industriales existentes.

Se conoce como configuración del robot a la combinación de los diferentes tipos de articulaciones en los primeros tres ejes del robot, para nombrar dichas configuraciones se escriben las tres iniciales de las articulaciones correspondientes en orden (de la base a la muñeca). Por ejemplo, RPR (rotación, prismática, rotación).

UNIDAD DE CONTROL Y PROGRAMACIÓN

La unidad de control y programación es la parte del robot industrial que va a dar capacidad al mismo de realizar sus funciones. Una vez que se haya escrito el código necesario para el movimiento, habrá que actuar sobre esta unidad únicamente para dar lugar al arranque y al paro del propio robot.

Además de controlar la dinámica del robot, la unidad de control y programación se encarga del soporte del sistema de programación del robot, del control del equipamiento de la célula de fabricante flexible y del procesamiento de información proveniente de los sensores externos, los cuales son de gran importancia ya que permiten detectar las posiciones de las diferentes articulaciones del robot, por ello suelen estar situados en cada eslabón.

ESPECIFICACIONES COMERCIALES DE UN ROBOT INDUSTRIAL

A la hora de adquirir un robot o, del caso contrario, de fabricarlo y venderlo, es de vital importancia la especificación de sus características para poder definirlo completamente. Dichas características aportarán una idea de la dinámica del robot industrial, así como de las capacidades de control del mismo y sus dimensiones. Por todo ello, en este apartado se explicarán las especificaciones que son de vital importancia para definir a cualquier robot.

En primer lugar tenemos las especificaciones que se refieren a la mecánica del brazo del robot, las cuales permitirán que el usuario se haga una idea acerca de la exactitud con la que se puede llegar a situar. Las **dimensiones de las articulaciones** es un parámetro de vital importancia. Para el extremo de la muñeca del robot habrá que definir tres conceptos: la **resolución**, que es el incremento mínimo de movimiento que puede generar la unidad de control; la **precisión**, es decir, la distancia entre el punto programado y el punto en el que realmente está posicionado el robot y, por último, la **repetibilidad**, que es el grado de exactitud entre la repetición de movimientos cuando el robot intenta acceder un punto para el que previamente ha sido programado.

Por otro lado, tenemos otro tipo de especificaciones referidas a las limitaciones del robot industrial tales como el **campo de trabajo**. Con esta característica se define el campo espacial dentro del cual se puede trabajar, que estará limitado por la geometría del brazo del robot y de su capacidad de giro. Normalmente los giros de la muñeca nunca llegan a los 360° debido a su diseño mecánico. Al definir el campo de trabajo ya se refleja el alcance máximo, que es una característica que puede llegar a ser crítica a la hora de realizar la elección del robot.

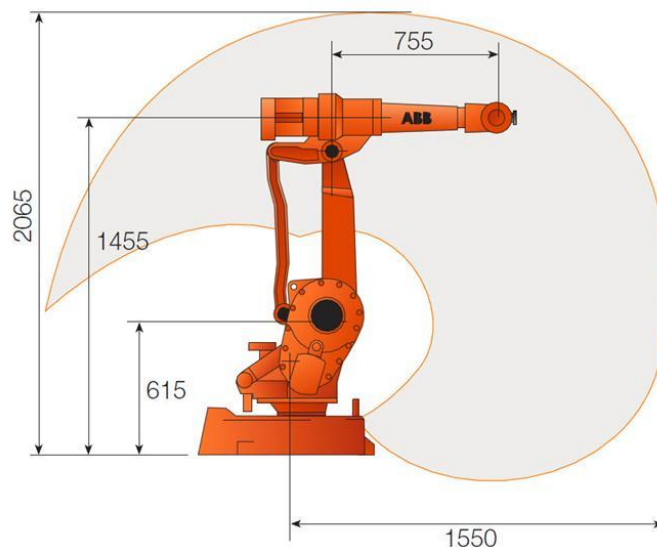


Figura 3. Campo de trabajo de un robot industrial (ABB Robotics 2017).

La **capacidad de carga** es una especificación de gran interés ya que nos permite saber el peso máximo en kg que puede llegar a transportar o manejar el robot. De esta manera podremos saber cuál es el peso máximo que debe existir teniendo en cuenta la pieza trabajada y la herramienta. Esta característica está condicionada por el tamaño del robot así como por su accionamiento. Normalmente, los valores asociados rondan entre los 5 y los 120 kg, aunque existen robots industriales que pueden llegar a soportar mucho más peso.

También es de gran importancia incluir la **velocidad** y la **aceleración** que puede tener el brazo del robot industrial. La primera de ellas puede expresarse de dos formas, como la velocidad de cada articulación o como la velocidad media de la muñeca. Muchas veces en lugar de expresar la velocidad como tal se indica el tiempo que tarda el robot en realizar un movimiento determinado. Por otro lado, la aceleración también se especifica ya que suele ser muy

relevante en movimientos cortos, esto es debido a que en ellos el arranque y la parada son de gran importancia. En el caso de este tipo de robots, cuánto más alta sea mejor.

Por último, queda citar las **características de la unidad de control** que nos darán una idea acerca de la potencia del sistema, de las posibilidades de actuación, memoria disponible, método de programación, etc.

APLICACIONES INDUSTRIALES CONSOLIDADAS

En este apartado se va a hablar de las numerosas aplicaciones que tienen los robots industriales que cada vez son más debido al constante desarrollo que se está llevando a cabo en este campo. Teóricamente, la aplicación de los mismos se podría extender a casi cualquier ámbito en el que se necesiten tareas mecánicas de líneas de producción, algunas de ellas son las siguientes:

- **Manipulación**
Se incluye en estas tareas aquellas en las que el robot mueve las piezas de un lugar a otro. Por ello, podemos incluir en este grupo todos aquellos cuya función sea la paletización, el empaquetado, pick and place (transferencia de una pieza de un punto a otro), el ensamblado, etc. Son tareas simples y repetitivas normalmente inviábiles para su ejecución por una persona.
- **Procesado**
En este tipo de operaciones el robot realiza modificaciones sobre la pieza, por lo que ya toma parte dentro del propio proceso de fabricación. Se incluyen en este grupo aquellos robots industriales que tienen la función de soldar (por puntos o en arco), cortar (por láser o por chorro de agua), de pulverizar como sprays, aplicar fluidos, mecanizar, etc.

3.3 Simulación de robots

Como bien se ha dicho en los apartados anteriores, la simulación y la robótica son unas herramientas de vital importancia en la industria a día de hoy. En este apartado se va a hablar de la combinación de ambos conceptos, es decir, de la simulación de robots.

La necesidad de los métodos de simulación en los procesos industriales ya las hemos visto, pero concretando al ámbito de los robots, se puede decir que esta herramienta nos aporta una solución visual de manera que podamos predecir posibles fallos y daños en el mecanismo, así como definir sus trayectorias, detectar colisiones y ver los alcances del robot de una manera sencilla.

La mayor parte de las herramientas de software para la simulación son diseñadas por los propios fabricantes de robots, la razón es que si simulamos los movimientos de dichos robots podremos asegurar si el modelo estudiado es válido y el comprador sabrá con anterioridad a la compra si el robot cumple las necesidades o no, lo cual va a facilitar las ventas en este sector.

A día de hoy, las características con las que cuenta el software de simulación son tan avanzadas que permiten ajustar los parámetros del robot con una precisión muy alta. Dichos parámetros pueden ser las velocidades de cada articulación, las aceleraciones, las dimensiones, etc. Aunque depende siempre del software de simulación con el que se trabaje, ya que unos serán más avanzados y precisos que otros.

Por todo ello, vamos a diferenciar los siguientes tipos de softwares de simulación robótica:

- **Software orientado a la optimización de los procesos de fabricación:**

Este tipo de software también es conocidos como PLM (Product Lifecycle Management) y se centran en encontrar la manera óptima de llevar a cabo un proceso de fabricación. Para ello, se usarán programas muy visuales que permitan resolver los problemas de una manera más rápida e intuitiva, modificando un gran número de parámetros de los que dependen todos los elementos de la simulación.

Este tipo de programas no están especializados en las características del robot industrial, por lo que es posible que en este ámbito estén más limitados que otro tipo de software que se centra únicamente en ellos.

A continuación se citarán algunos ejemplos:

- Tecnomatix

Es un software desarrollado por Siemens que ofrece grandes capacidades de arquitectura y modelado, de manera que permita al cliente crear sistemas altamente complejos y así sean lo más próximos a la realidad posible. Incluye una serie de bibliotecas en las cuales se especifican procesos de negocio específicos tales como ensamblado, fabricación de cuerpos automotrices, etc. Muchos de los recursos que contiene este programa están ya predefinidos, por lo que facilita en gran medida su uso.

Cabe mencionar que se pueden generar gráficas y tablas que representen los resultados obtenidos en la simulación, este hecho facilita el análisis y la corrección del sistema.

Los modelos pueden visualizarse en un entorno 2D y en un ambiente virtual 3D que da un aspecto más realista al entorno.



Figura 4. Software de simulación Tecnomatix. Visualización del entorno 3D (SIEMENS 2017).

- Delmia

Delmia es un software PLM europeo desarrollado por la empresa Dassault Systems.

Al igual que con el software anterior, con Delmia se puede experimentar de forma virtual la producción completa de la fábrica.

A diferencia de otros software de simulación robótica, Delmia presenta como gran ventaja el hecho de que permite al cliente centrarse en el elemento que se desea estudiar de manera que, a pesar de que se simule por completo el proceso de producción entero, se obtendrán unos resultados centrados en el elemento que se desee, obteniendo como resultado su operatividad, de esta manera se tendrá en cuenta todo tipo de detalles.

Además contiene un módulo de trabajo denominado Sketcher con el cual se pueden modelar los diferentes elementos de la simulación desde cero. En él aparecen comandos de línea, punto, polígonos; relaciones entre las diferentes geometrías, extrusiones, revoluciones y todo lo necesario para crear las piezas

que se precisen en 3D. A partir de dichas piezas se procederá al ensamblaje, dando lugar a los mecanismos que queramos obtener para nuestro modelo. Sus movimientos también podrán ser definidos mediante las trayectorias. Además, también se podrán generar los planos de las piezas creadas.

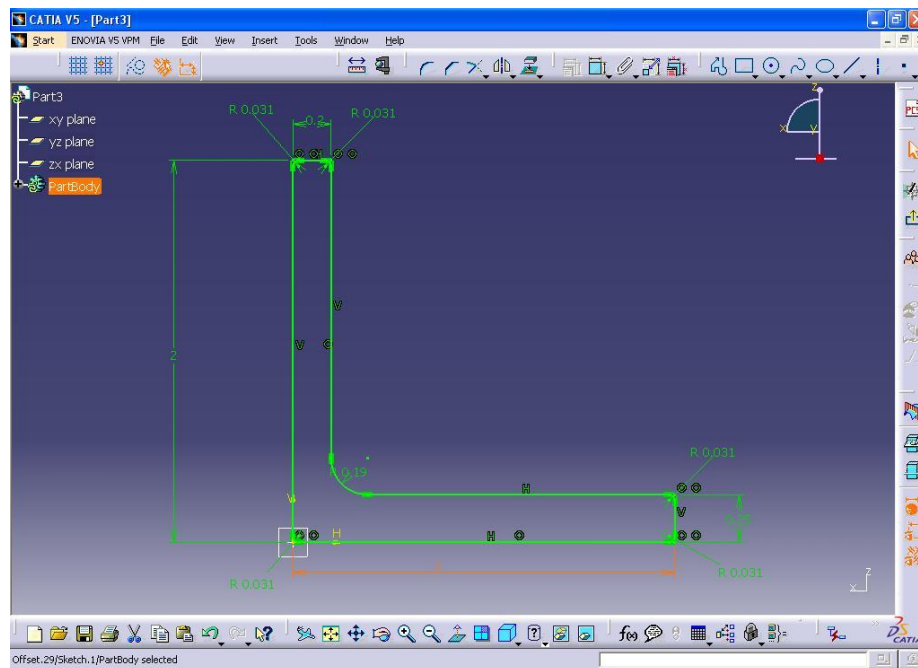


Figura 5. Software de simulación robótica Delmia. Sketcher (Catia Community 2018).

Esta capacidad diferencia a este software de simulación de su competencia, ya que con el modelaje de las piezas se aporta gran realismo al modelo y todos los elementos van a poder semejarse a los reales.

Si del caso contrario no queremos dedicar tanto tiempo a la modelización de los elementos, el propio programa incluye bibliotecas en las cuales podemos descargar diferentes piezas y mecanismos que se usan habitualmente en procesos de producción.

Cabe mencionar que Delmia cuenta con un módulo conocido como DELMIA ROBÓTICS V5 que contiene los siguientes comandos que nos permitirán controlar los robots industriales:

- Device task definition: herramienta para la programación de cada uno de los robots.
- Arc Welding: permite hacer la simulación de procesos industriales como soldadura, unión de piezas, etc.
- Workcell sequency: se usa para la programación completa de los robots, generando la secuencia de movimientos deseada para la generación de un proceso.

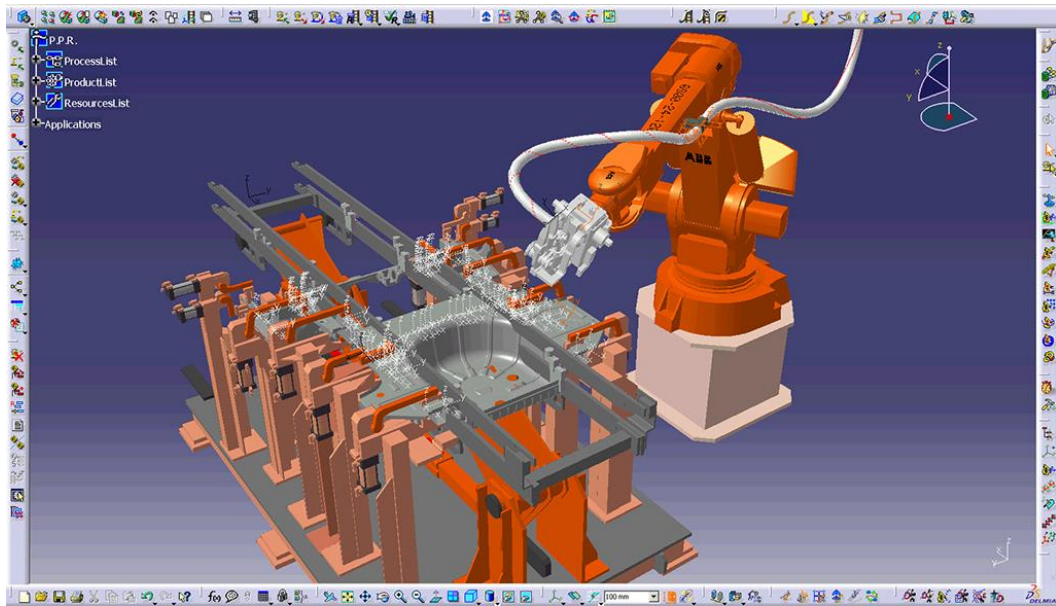


Figura 6. Software de simulación robótica Delmia. Robot industrial y su trayectoria (Catia Community 2018).

Por todas estas funciones, Delmia es un software más preciso y avanzado, con el que se obtendrán resultados más reales, aunque también habrá que invertir más tiempo en llevar a cabo la simulación completa del sistema de producción.

- **Softwares orientados al robot industrial:**

Este tipo de softwares van a estar centrados en el robot industrial, pudiendo definir por completo su geometría, trayectoria y funcionalidad. Ya no se simulará el proceso industrial completo como en el apartado anterior, sino que únicamente nos centraremos en el elemento robot como tal. Este tipo de programas han sido creados en su mayoría por los propios fabricantes de los robots industriales. A continuación se citarán algunos de los más relevantes:

o Gazebo

Gazebo es un software que permite el control de robots en un entorno 3D. Para ello, podrán ser diseñados mediante un módulo CAD que integra el propio programa, aunque tiene unas capacidades bastante limitadas. Además, cuenta con librerías con robots predeterminados. Existe también la opción de importar archivos 3D creados en otros entornos, de esta manera podremos tener modelos más complejos que no se vean limitados por el módulo simple que integra el propio programa.

Gazebo nos va a aportar mucha más información sobre el robot debido a que nos permite realizar muchas más funciones y controlar más parámetros como, por ejemplo, los sensores y la información que recogen los mismos.

En este programa el control de las trayectorias y acciones de los robots industriales tendrán que llevarse a cabo mediante la programación. Este hecho ralentiza el proceso de simulación ya que hay que controlar el código así como las diferentes funciones que trae el programa.

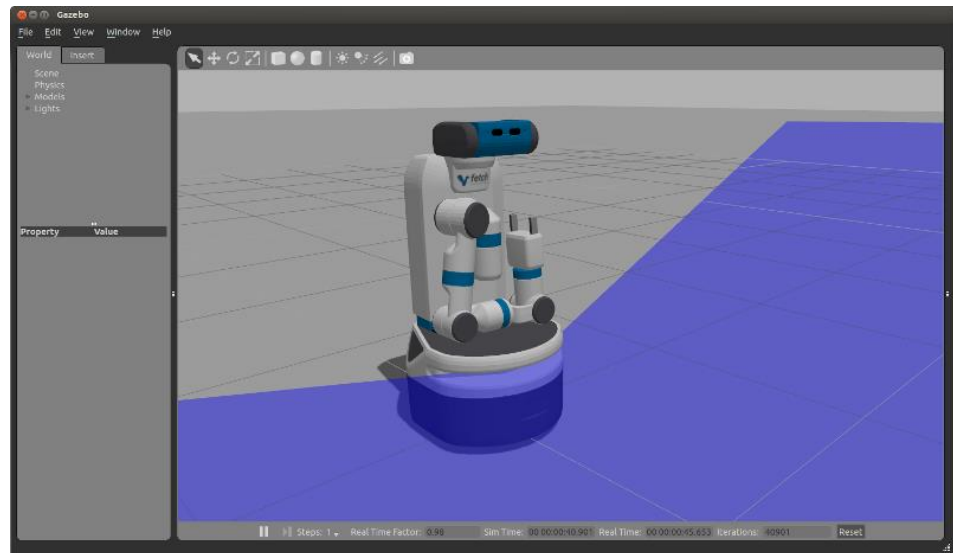


Figura 7. Gazebo. Robot industrial modelado (Fetchrobotics 2016).

○ KUKA.Sim

Este software fue creado por la propia empresa fabricante de robots KUKA. Por ello, ofrece facilidades a la hora de simular procesos que contienen robots industriales de su propia marca, ya que contiene un catálogo con los distintos modelos que se encuentran a la venta.

Incluye, como todos los anteriores, un entorno muy visual en el que, además de robots, nos encontramos con todo tipo de elementos de apoyo para que se completen las funciones que se requieran.

El control se realizará mediante programación offline en un lenguaje conocido como KUKA Robot Language (KRL).

Los programas creados podrán transmitirse directamente del robot virtual al real garantizando una productividad inmediata.

Este tipo de software ofrecen grandes ventajas en el caso de la compra de productos de su propia marca, ya que aportan grandes facilidades para su manejo, que es lo que se pretende, de esta manera se promueve a la venta de estos productos.

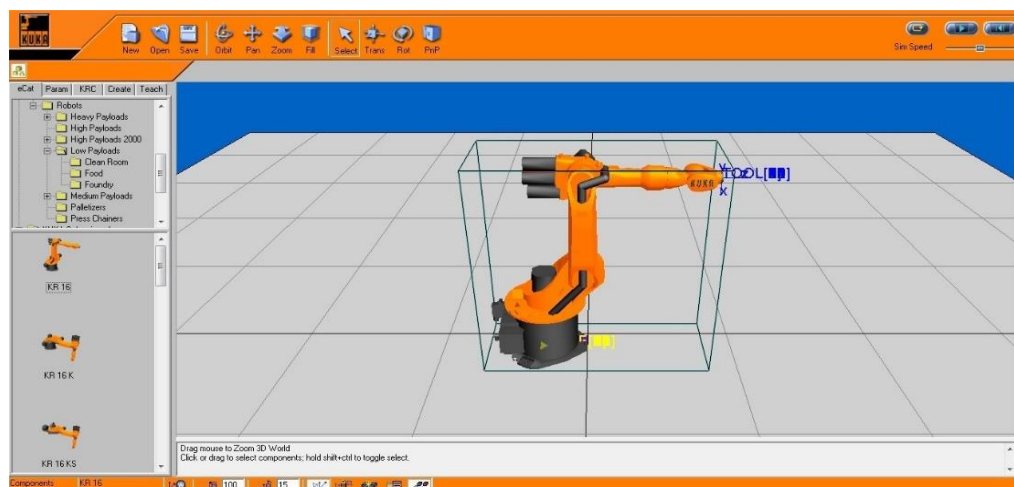


Figura 8. Ejemplo de implementación de un robot en KUKA.Sim (KUKA Robotics 2018).

FlexSim va a ser el programa con el que se trabaje en este proyecto y se puede considerar una combinación de los dos tipos de softwares citados anteriormente ya que, por un lado, permite la optimización de los procesos de fabricación pero, además, también tiene la capacidad de profundizar en los diferentes parámetros que caracterizan a los brazos robóticos.

Gracias al propio entorno 3D que incluye, así como los diagramas de flujo que permite crear para que los procesos sigan la lógica que se desee, FlexSim tiene la capacidad de reproducir los procesos con resultados muy próximos a la realidad. De esta manera se pueden obtener tiempos, tasas de producción, gráficas, histogramas o cualquier variable de la que dependa el propio proceso. Además, este programa permite profundizar en la lógica de cada uno de los elementos que se incluyen en el entorno, de manera que pueda cambiarse su aspecto, cinemática o prácticamente cualquier característica del mismo.

4 ROBOT EN FLEXSIM

4.1 Descripción robot

El tipo de robots industriales que se integran en FlexSim constan de una base fija al suelo, seis eslabones y dos manecillas en sus extremos. Tiene además 6 grados de libertad, a los que conoceremos como articulaciones. Vamos a establecer un eje de coordenadas y una posición inicial, tal y como se ve en la Figura 9, para el que el eje z (azul) será vertical hacia arriba, el eje y (verde) será en horizontal en dirección opuesta al sentido del brazo teniendo en cuenta la posición inicial de la figura y el eje x (rojo) horizontal hacia la derecha viendo el robot de frente (primera casilla).

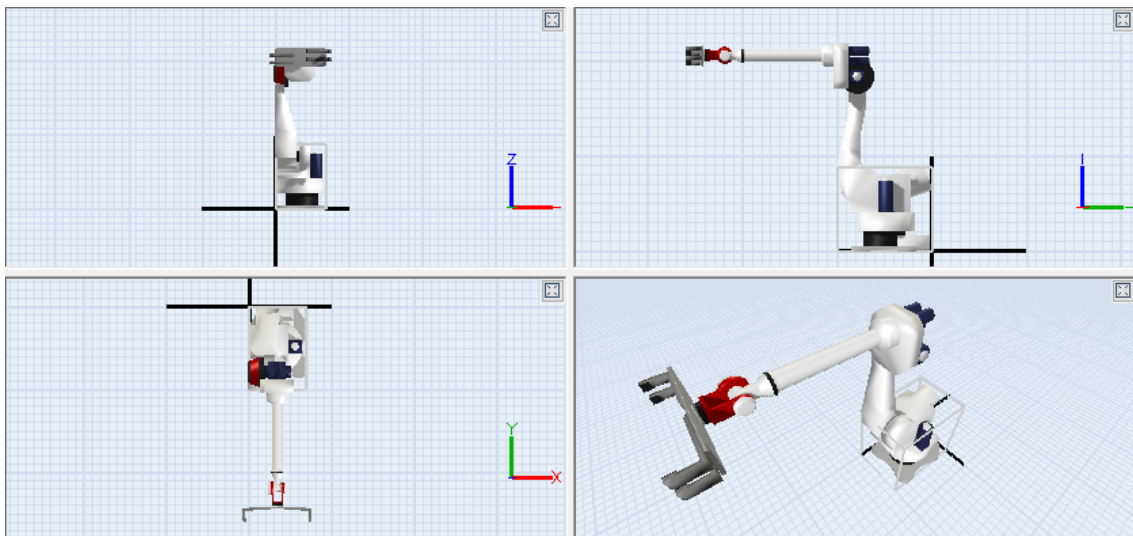


Figura 9. Vistas Robot en FlexSim y eje de coordenadas (Elaboración propia).

Los eslabones del robot son los siguientes:

- Base
La base es la parte del robot que se encuentra en contacto con el suelo y, que en este caso, estará fija.
- Eslabón 1
Es la parte del robot que se encuentra encima de la base y no está fija, va a girar.
- Eslabón 2
Es el primer brazo del robot, está a continuación del eslabón 1. En la figura 9 se encuentra en vertical.
- Eslabón 3
Es el segundo brazo del robot, a continuación del eslabón 2. Se encuentra en horizontal en la figura 9.
- Eslabón 4
Es la parte del robot que se encuentra en la punta del eslabón 3 y que girará de manera diferente al mismo.
- Eslabón 5

Está a continuación del eslabón 4 y unidos al mismo gracias a un eje. En color rojo en la Figura 9.

- Eslabón 6
Es la placa que se encuentra en el extremo del brazo robótico sobre la que se situarán las manecillas.
- Manecillas
Hay 2 sobre el eslabón 6, se diferenciarán entre derecha e izquierda. Las articulaciones, ordenadas de base a extremo, son las siguientes:
- Articulación 1
Es la que permite el movimiento alrededor del eje z, de manera que el robot gire sobre sí mismo. Gira todo menos la base.
- Articulación 2
Esta articulación permitirá el giro alrededor del eje x en el punto de unión entre el eslabón 1 y el 2. Por ello, girarán todo el robot exceptuando base y Eslabón 1.
- Articulación 3:
Al igual que el anterior, permite el movimiento alrededor del eje x, aunque el punto de giro se encuentra en la unión entre ambos brazos, es decir, entre el eslabón 2 y 3.
- Articulación 4:
Giro del eslabón 4 sobre sí mismo alrededor de la dirección del segundo brazo (eslabón 3).
- Articulación 5:
Giro del eslabón 5 sobre el eje de unión entre el eslabón 4 y el 5.
- Articulación 6:
Giro del eslabón 6 alrededor del eje perpendicular al propio plano de la placa.

Además, habrá otro movimiento de las manecillas del extremo del brazo robótico que será necesario para enganchar el objeto que vaya a manipular el robot.

Existe una herramienta que contiene FlexSim llamada Animations que nos permite cambiar los eslabones del robot, o bien por formas predeterminadas que contiene el propio programa (cubos, esferas, etc), o bien importando archivos 3D.

Dichos eslabones podrán ser modificados una vez importados variando sus dimensiones y posiciones. Esto lo podemos hacer gracias a la herramienta Explore Tree.

De esta manera podemos obtener la apariencia que nosotros necesitemos para adaptar la simulación a un entorno más real.

Una vez que tenemos definido completamente el brazo robótico podemos empezar a programar el movimiento del mismo. Para ello, FlexSim cuenta con una serie de comandos.

4.2 Movimiento de flowitems con el robot

Una vez tengamos creado nuestro espacio de simulación, podremos definir la acción que queramos que nuestro robot realice. Para ello, se cuenta con cuatro modos de operación:

- Define Move Time (definir tiempo de movimiento):
Esta opción es la que viene establecida por defecto, con ella, podemos definir el tiempo que el robot tarda en realizar el movimiento deseado. Puede ser un tiempo fijo, seguir una distribución estadística, etc.
- Use Simple Motion Method (método de movimiento simple):
Esta opción permite definir la velocidad de rotación en el eje z, en el y y la velocidad de extensión del brazo.
- Use Join Speeds (Velocidad de los eslabones):

Permite definir la velocidad, aceleración y deceleración para cada uno de los eslabones.

- Define Motion Paths (definir rutas de movimiento):

Es la opción más avanzada.

Cuenta con una tabla para definir las posiciones del robot por las que queremos que pase durante su recorrido. Para establecer dichas posiciones hay que definir los 6 ángulos correspondientes a cada articulación. Dicha tabla cuenta con una última columna llamada Move Time (Tiempo de movimiento).

Para definir el tiempo de movimiento del robot tenemos varias opciones.

Por un lado tenemos la casilla de Gripper Action (acción de agarre), la cual definirá tiempos añadidos al final del movimiento.

Por otro lado tenemos la casilla de Path Cycle Time (tiempo de ciclo de ruta), en función de su valor nos encontraremos con las siguientes situaciones:

○ > 0

Se puede especificar la duración de tiempo que le lleva al robot completar el movimiento. En este caso, la columna Move Time de la tabla afecta a la distribución del tiempo de ciclo de ruta retrasándolo de la siguiente forma:

- Sin ponderación: cubrir la columna Move Time con ceros en todas las casillas o valores iguales, de esta manera el tiempo de ciclo de ruta será dividido uniformemente entre el número de posiciones.
- Con ponderación: ajustar los valores en la columna Move Time para asignar el tiempo que se quiera a cada posición establecida previamente.

○ = 0

Cubriendo los valores de la columna Move Time con valores positivos, el robot va a usar la suma de los valores de dicha columna para definir la duración del movimiento del mismo.

Este método puede ser usado para insertar una pausa durante el movimiento, para ello es necesario colocar dos filas seguidas en la misma posición, de manera que el Move Time será el tiempo de espera.

○ < 0

Los valores de la columna Move Time solo afectarán a la animación.

The screenshot shows the 'Define Motion Paths' interface in FlexSim. At the top, there are controls for 'Path Cycle Time' (set to 4), 'Gripper Action' (Close Time: 0.00, Open Time: -1.00, Clamp Orientation: Z+ by Y), and 'Manipulation Tools' (Planner Item, Sync To Flow Item). Below these are 'Current Positions' for joints J1 through J6 (0.0, 19.4, -13.7, 224.4, -38.7, -2.3). The main part of the interface is a table with 6 columns (J1-J6) and a 'MoveTime' column. The table contains 6 rows of data.

	J1	J2	J3	J4	J5	J6	MoveTime
Row1	-58.1	10.5	-1.3	180	-80.9	-31.9	0.5
Row2	-0.8	6.3	13.1	180	-70.6	-89.2	1
Row3	-1.0	-8.9	35.2	175.6	-10.7	-85.0	0.5
Row4	-1.0	-8.9	35.2	175.6	-10.7	-85.0	1
Row5	53.3	-1.1	11.0	180	-80.1	-143.3	1.5
Row6	53.3	0.2	19.5	180.0	-70.3	-143.3	0.5

Figura 10. Herramienta FlexSim Define Motion Paths (Elaboración propia).

Cabe decir que a mayores de estos cuatro modos de operación existen unos diagramas de flujo que se pueden añadir en el propio programa para que el brazo robótico siga una lógica determinada. En ellos existen multitud de comandos que, combinándolos, se puede referenciar al robot e indicarle que realice movimientos determinados.

4.3 Otras propiedades del robot

Dentro de la pestaña de propiedades del Robot, además de las opciones explicadas en el apartado anterior, nos encontramos con más comandos para facilitar el manejo del brazo robótico.

En Geometría, contamos con seis filas, cada una corresponde a una articulación. En esta pestaña podemos definir las velocidades relativas de cada uno de ellos así como la posición de inicio. Para visualizar las posiciones de una manera más sencilla, a la derecha aparecen unas barras regulables que, al moverlas varía la posición de cada articulación.

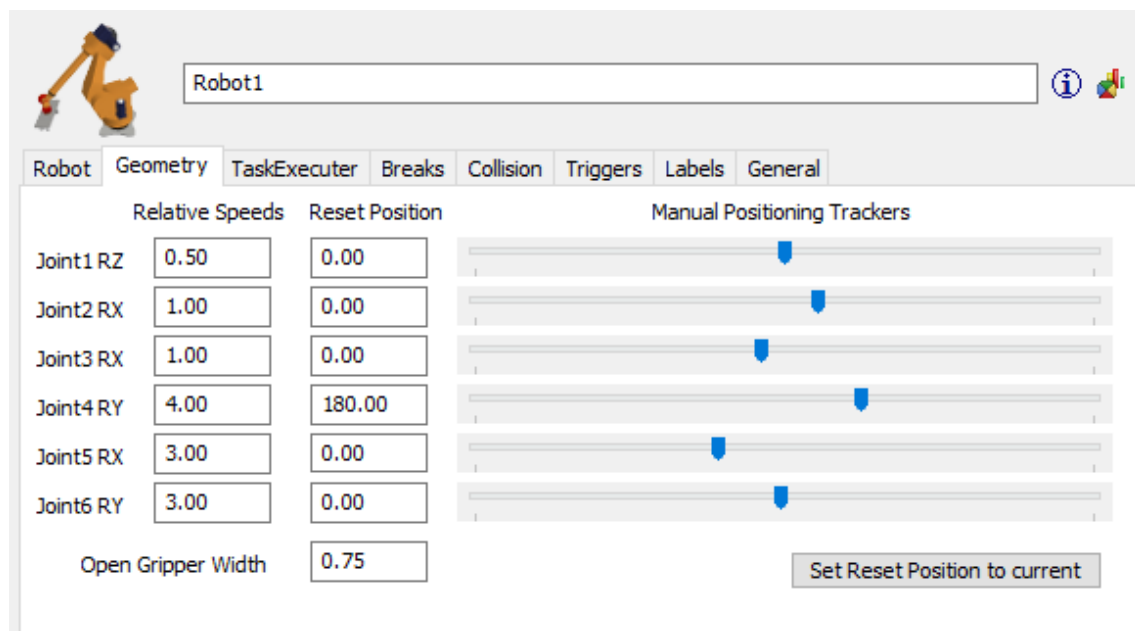


Figura 11. Herramienta FlexSim Geometry (Elaboración propia).

La siguiente pestaña es la TaskExecuter (ejecución de tareas).

Aquí se pueden definir parámetros como la capacidad del robot, que es el número de elementos que va a poder transportar por viaje, su velocidad máxima, aceleración, deceleración, el tiempo durante el que está cargado y descargado, etc.

En la pestaña Breaks (descansos), como su propio nombre indica, vamos a poder modificar todo lo referente a los descansos que realice el brazo robótico.

En la pestaña Collision (colisión) se van a poder contabilizar el número de colisiones que se producen durante el recorrido del robot así como el tiempo transcurrido entre las mismas. Además, se especificarán cuáles son los elementos afectados por dicha colisión.

La siguiente pestaña es la de Triggers, éste comando es común a todos los elementos de FlexSim e incluye una serie de pestañas en las cuales podemos modificar parámetros relacionados al objeto que va a manipular el Robot (su frecuencia de llegada, de salida, tipo de objeto, etc).

En la pestaña Labels (etiquetas) vamos a poder crear etiquetas, las cuales pueden llegar a ser útiles para llevar a cabo alguno de los comandos explicados con anterioridad.

Por último, en la pestaña General, tenemos cuatro cuadrantes diferentes.

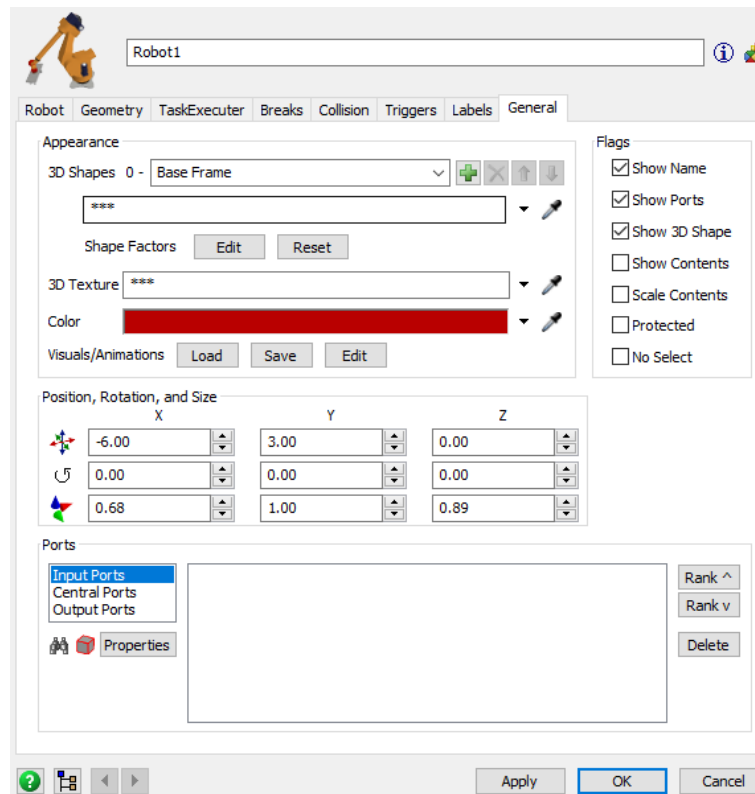


Figura 12. FlexSim propiedades Robot. General (Elaboración propia).

El primero de ellos es el referente a la apariencia, en él vamos a poder modificar parámetros tales como el color y la textura del Robot.

En segundo lugar, tenemos el cuadrante de Posición, rotación y forma. Gracias a él se puede modificar, como su propio título indica, la posición, la rotación y la forma del Robot en cada uno de sus tres ejes.

A continuación tenemos el cuadrante de puertos. Se entienden como puertos a todos aquellos elementos que están directamente relacionados con el Robot. Por ello, en esta opción podemos seleccionar a aquellos que consideremos adecuados para ejecutar nuestra simulación de manera adecuada.

Por último, en la parte derecha, aparece el cuadrante banderas. Las banderas son aquellos contenidos que queremos que se visualicen en el entorno 3D.

5 CONFIGURACIÓN DEL ROBOT COLABORATIVO BAXTER

5.1 Descripción del Baxter

El robot Baxter es un robot industrial para manipulación de elementos que ha sido diseñado por Rethink Robotics en 2013. Su principal función es la de llevar a cabo tareas repetitivas y poder ayudar en los procesos de producción de pequeñas y medianas empresas.

Este robot está capacitado para ejecutar procesos de “pick and place”, de empaque u otros similares que no requieran de gran dificultad. A diferencia de la mayoría de robots industriales, sus movimientos no están alimentados por un motor, sino que utiliza unos resortes que dan lugar a unos movimientos más suaves. Cuenta además con cinco cámaras que le ayudarán a guiarse y ajustarse a las posiciones marcadas.

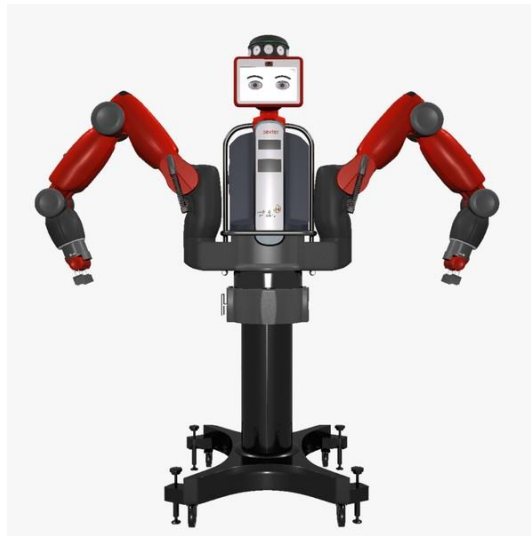


Figura 13. Robot Baxter (Baxter España 2018).

Uno de sus grandes beneficios recae en que no requiere de sistemas de protección como barreras o rejillas. Esto es debido a que el propio robot responde a variaciones en el espacio o a la colisión con obstáculos o personas sin necesidad de programar este hecho de ninguna forma. Todo ello es posible gracias a un dispositivo sonoro con el que cuenta de manera que pueda situar el lugar de cada persona en el espacio que le rodea. Por ello, el robot Baxter puede estar en funcionamiento al lado de los operarios sin ningún tipo de inconveniente. Además, su instalación es sencilla, por lo que podrá montarse y desmontarse con rapidez.

Baxter requiere ser entrenado para que realice una actividad. A diferencia de otros, su configuración es sencilla porque no requiere de dispositivos externos para la ejecución de una tarea. Basta con mover manualmente sus articulaciones a lo largo de una trayectoria determinada y presionar un botón de navegación. De esta manera el robot Baxter repetirá el movimiento deseado.

Su estructura cuenta con una base que puede estar fija atornillada al suelo o, por el contrario, puede ser móvil, ya que cuenta con unas ruedas en caso de que se prefiera. En el caso de la

base fija, la altura del Baxter será de 1,78 m, mientras que si se le añaden las ruedas la altura ascenderá a 1,87 m, que será la que tomaremos para obtener los resultados.

En la parte central superior, tal y como se ve en la Figura 14, cuenta con una pantalla. Su presencia no es simplemente estética, sino que cumple la función de representar el “estado de ánimo” del robot. Es decir, muestra su reacción a lo ocurrido, por ejemplo, si ocurre algo inesperado adquirirá expresión de sorpresa.

Pesa alrededor de los ochenta kilos. En comparación a otros robots de este tipo se puede afirmar que es pequeño.

Cuenta con dos brazos articulados, los cuales son iguales entre sí y podrán cumplir el mismo tipo de funciones.

En cada brazo hay seis articulaciones además de dos pinzas para el agarre de los elementos manipulados.

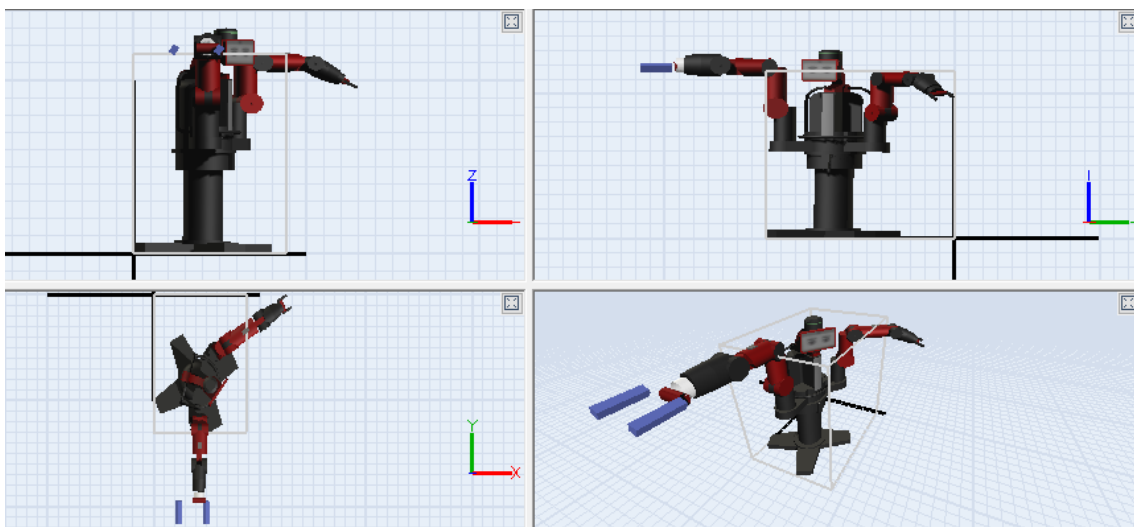


Figura 14. Vistas Baxter. Módulo Animations FlexSim (Elaboración propia).

Para explicar el movimiento que realiza cada una de las articulaciones definiré un eje de coordenadas orientado tal y como se ve en la figura 24. Donde Z se ve en azul, Y en verde y X en rojo.

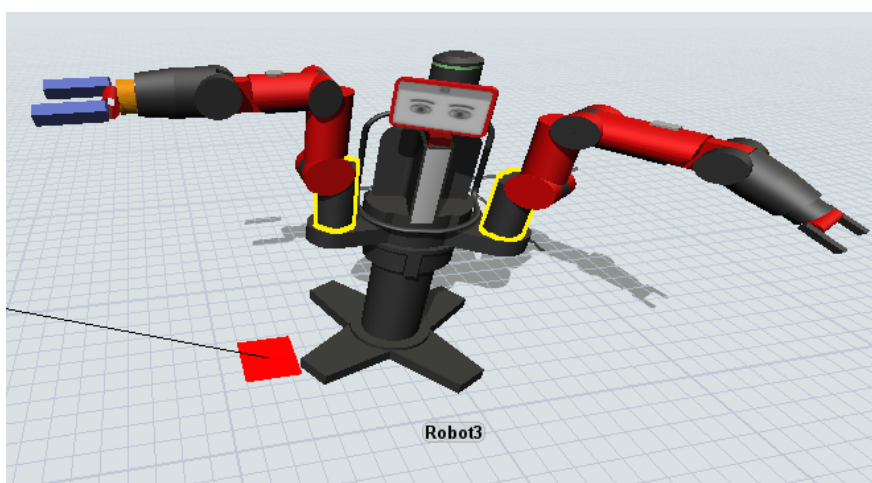


Figura 15. Módulo 3D de FlexSim. Resalte articulación 1 (Elaboración propia).

La primera articulación (de base a extremo del brazo) es la seleccionada en la figura (en amarillo). Está apoyada sobre la base del Baxter, concretamente sobre una plataforma que sobresale sobre el mismo. Es un cuerpo cilíndrico con un saliente para el alojamiento de la segunda articulación. El movimiento que realiza es un giro alrededor del eje z, al ejecutarse dicho movimiento se moverán solidariamente el resto de las articulaciones.

La segunda articulación es la que continúa a la primera, es decir, la que está apoyada sobre la misma. En la imagen se representa de color rojo. Su eje de rotación pasa por el cilindro en contacto con la primera articulación en dirección x (ver Figura 15).

La tercera articulación es prácticamente igual a la segunda, ya que el movimiento que realiza es igual. Sin embargo, en su extremo contiene una pestaña que servirá de apoyo para la siguiente articulación.

La cuarta articulación es análoga a las dos anteriores, gira sobre la pestaña de la tercera también alrededor del eje x.

A continuación de la anterior está la quinta articulación, la cual va a ser completamente diferente a las anteriores, ya que va a girar alrededor del eje y (ver Figura 15) de manera que permita a las pinzas agarrar un objeto orientado de cualquier forma.

Por último, el robot Baxter incluye unas pinzas, las cuales serán la parte que contacte con el elemento a manipular. En función de la tarea que tenga que realizar el robot podrán ser modificadas para que se adapten mejor a su trabajo.

El alcance máximo de este robot es de

Las velocidades máximas de cada articulación vienen especificadas en la siguiente tabla.

Tabla 1. Velocidades máximas articulaciones Baxter (Elaboración propia).

	Velocidades (°/s)
Articulación 1	114,5
Articulación 2	114,5
Articulación 3	114,5
Articulación 4	229,2
Articulación 5	229,2

5.2 Introducción al Baxter en FlexSim

Para introducir el modelo del robot Baxter en FlexSim habrá que partir del robot industrial que viene implementado en el programa e ir modificando sus parámetros hasta obtener los movimientos y apariencia que deseamos.

En primer lugar, habrá que obtener un modelo 3D del robot Baxter.

En la página web 3D Warehouse (<https://3dwarehouse.sketchup.com/>) existen infinidad de modelos de objetos y estructuras entre las cuales se encuentra la de nuestro robot. El archivo descargado es de formato SketchUp y contiene el modelo del Baxter entero como una pieza única.

Para introducirlo en FlexSim habrá que explotar el modelo del robot en el programa SketchUp de manera que se obtengan cada una de las articulaciones por separado así como la base del mismo. En esta tarea se tomará el brazo izquierdo del robot como parte de la base

de manera que solo tenga capacidad de movimiento el brazo derecho. Esto es debido a que el robot que incluye el FlexSim solo contiene un brazo articulado por lo que no será posible implementar el Baxter de manera que se muevan las dos extremidades.

Una vez que se tiene cada una de las partes separadas, se exportará cada una de ellas en formato “.dae” para que pueda reconocerlo FlexSim. De esta manera contaremos con siete archivos que contendrán sus correspondientes figuras en formato 3D por separado. Uno para cada articulación, otro para la base y otro para la pinza.

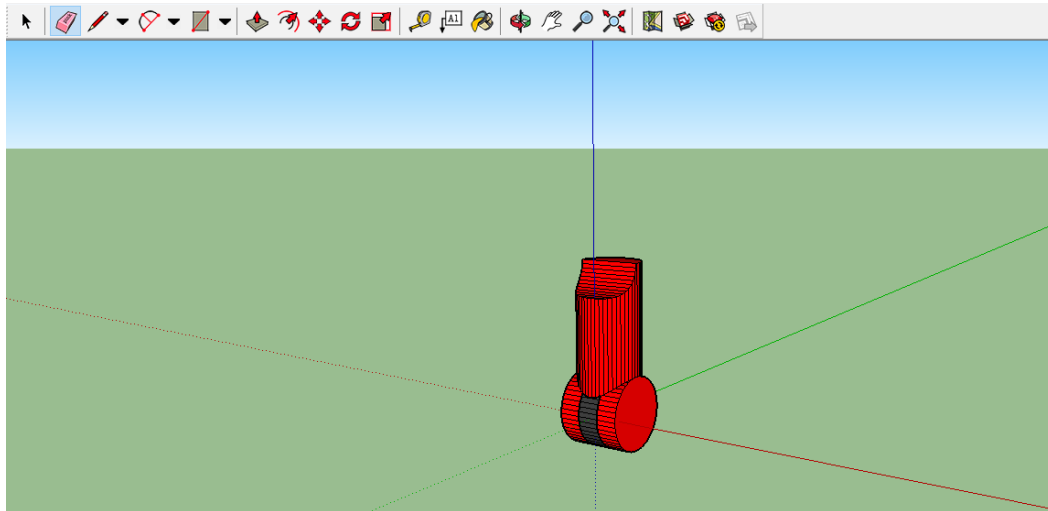


Figura 16. Articulación 2 en SketchUp (Elaboración propia).

El siguiente paso consistirá en importar cada uno de estos siete archivos en FlexSim de manera que correspondan con cada parte del robot que contiene el programa para que pueda realizar los movimientos de manera correcta. Para ello, en primer lugar, habrá que asociar cada una de las partes del Baxter con las partes del robot que incorpora FlexSim identificando la semejanza de los movimientos de uno y de otro. La base no se mueve en ninguno de los dos casos, por lo que se sustituirá la base del robot de FlexSim por la del Baxter. Las articulaciones 1, 2 y 3 de ambos también se corresponden, ya que los ejes de giro son los mismos en ambos casos. La articulación 4 del robot de FlexSim es inexistente en el caso del Baxter. Las dos últimas articulaciones de ambos robots también se corresponden. Por último, las pinzas también se sustituirán unas por otras.

Para proceder a importar las diferentes partes del robot Baxter habrá que introducir en primer lugar el modelo de robot que contiene FlexSim en el espacio 3D. Dicho modelo se encuentra en el grupo *Task Executors* y bastará con arrastrarlo hacia el área de trabajo.

Desplegando el menú del robot se entrará en *Edit>Animations*. Se abrirá una pestaña con el nombre del robot introducido, en ella se podrá ver al robot desde diferentes perspectivas.

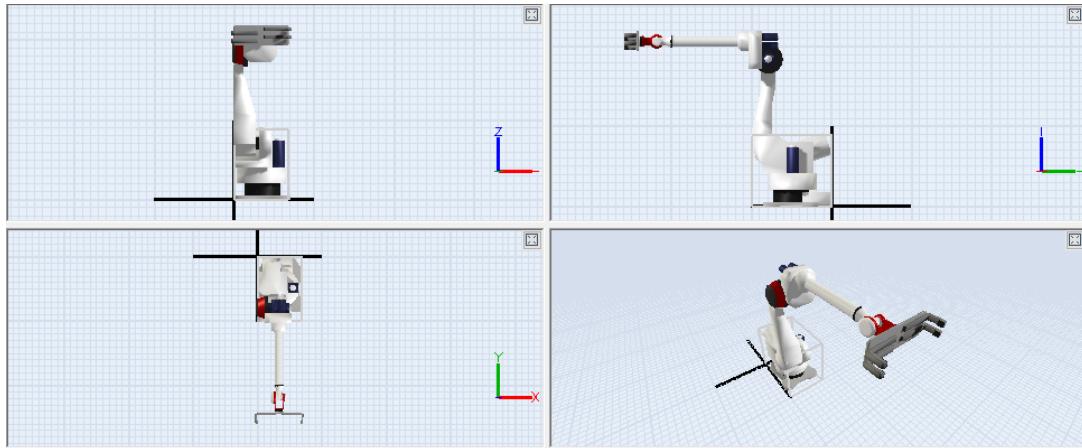


Figura 17. Robot predeterminado de FlexSim. Animations (Elaboración propia).

El proceso que se llevará a cabo para ir introduciendo cada una de las partes del robot se hará en orden (de base a extremo del brazo). Por ello, habrá que seleccionar en primer lugar la base del robot en el área gráfica. Al hacerlo, en la ventana *General Properties* nos dará la opción de importar un archivo 3D para sustituirlo por la parte del robot seleccionada.

El siguiente paso a realizar una vez importada la pieza del Baxter va a ser orientarla y dimensionarla. Para ello, habrá que volver que salir de la pestaña *Animations* para entrar en *Explore Tree*. Esta ventana permitirá modificar la mayor parte de los parámetros del robot. Está ordenada de manera esquemática y se ven numerados las diferentes variables del robot así como su correspondiente valor, entre otras cosas.

Para modificar la posición y dimensiones de la articulación en cuestión se tendrá que entrar en *Visual>resizeinfo>bodies*. Al desplegar la pestaña de cada una de las articulaciones aparecerán los parámetros de las coordenadas (x,y,z), los de la rotación (rx,ry,rz) y los de las dimensiones de la pieza (sx,sy,sz). La rotación es un parámetro que no se permite modificar, por lo que será necesario orientar cada una de las piezas correctamente en el paso previo en el que usamos SketchUp.

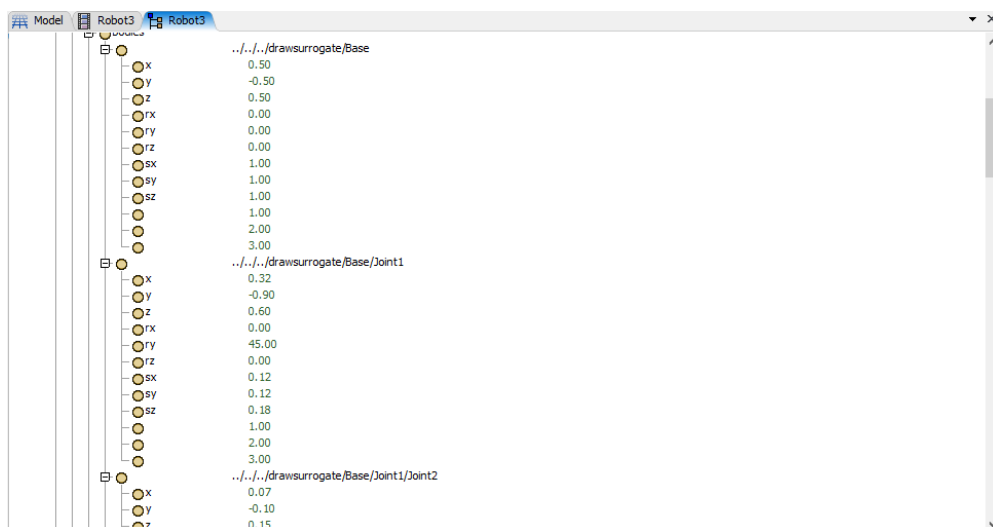


Figura 18. Parámetros del robot. Explore Tree (Elaboración propia).

Una vez que se introduzcan las orientaciones y medidas de la base, al presionar *Reset* se actualizarán los valores de manera que ya se verá la posición y tamaño correctos en el entorno 3D.

Para introducir el resto de articulaciones, el procedimiento será análogo. Es importante realizar la introducción de las diferentes partes por orden, ya que la posición de la articulación más próxima al extremo va a depender de la anterior. Es decir, si movemos la articulación 2, se moverán también la 3, la 4, la 5 y la 6. Esto es debido a que cada una de las articulaciones tendrá su propio sistema local de coordenadas que será relativo al de la anterior articulación.

El caso ideal sería que el Baxter tuviese el mismo número de articulaciones que el robot del FlexSim y que, además, éstas se moviesen de la misma forma. De esta manera, no habría nada más que hacer y el robot ya se movería de la manera deseada. Sin embargo, el Baxter tiene una articulación menos, que se corresponde con la 4 del robot de FlexSim. Por ello, para poder ver un movimiento real como el que llevaría a cabo el Baxter, habrá que crear una trayectoria en la que dicha articulación 4 no rote o simplemente ignorar el movimiento de la misma.

5.3 Obtención de resultados

En este apartado realizará una tarea concreta con el Baxter y se compararán los tiempos obtenidos en la realidad con los del Baxter introducido en FlexSim.

Para ello, por un lado se programará el Baxter en el laboratorio para que realice una tarea concreta. Tal y como se ha explicado con anterioridad, basta con mover manualmente las articulaciones a las posiciones por las que se quiere que pase el Baxter. La tarea que se le ha encomendado consiste en coger un cilindro, levantarlo y volver a situarlo en el mismo sitio.

El Baxter partirá de una posición inicial (posición 1), a partir de ahí pasará a una posición intermedia (posición 2) y después alcanzará la posición inferior (posición 3) en la cual adquirirá el cilindro. Estas serán los movimientos que realice el robot descargado. Se representan en la siguiente figura por orden.

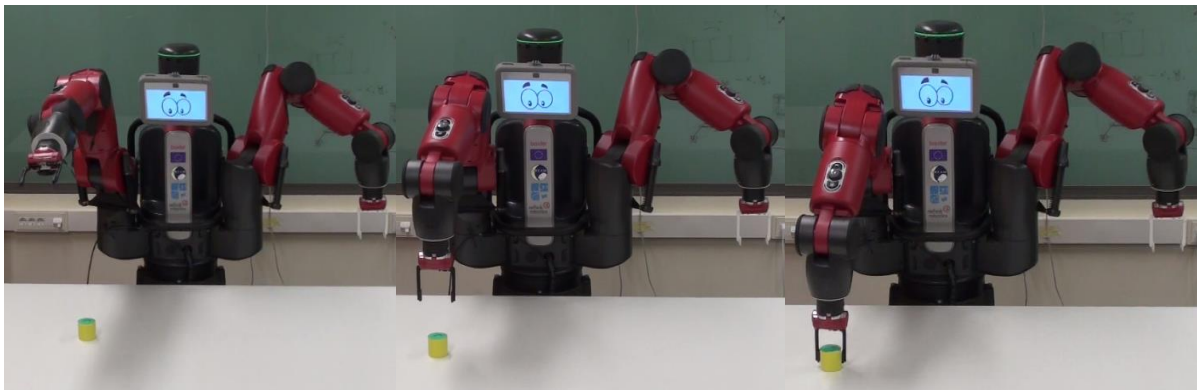


Figura 19. Posiciones 1, 2 y 3 Baxter (Elaboración propia).

Los viajes que realizará el Baxter cargado pasarán por las mismas posiciones aunque en diferente orden. Partirá de la posición 3, donde se cogerá el cilindro, pasará a la posición 2, subirá hasta la posición 1, volverá de nuevo a la posición 2 y finalmente acabará en la posición 3 donde de nuevo soltará el cilindro.

Se tomarán los datos de las diferentes coordenadas a las que viaja el robot para adquirir las tres posiciones. Habrá que tener en cuenta que el origen del sistema de coordenadas del Baxter en el centro del mismo a una altura de 0,968 sobre la superficie del suelo, tal y como se representa en la Figura 20.

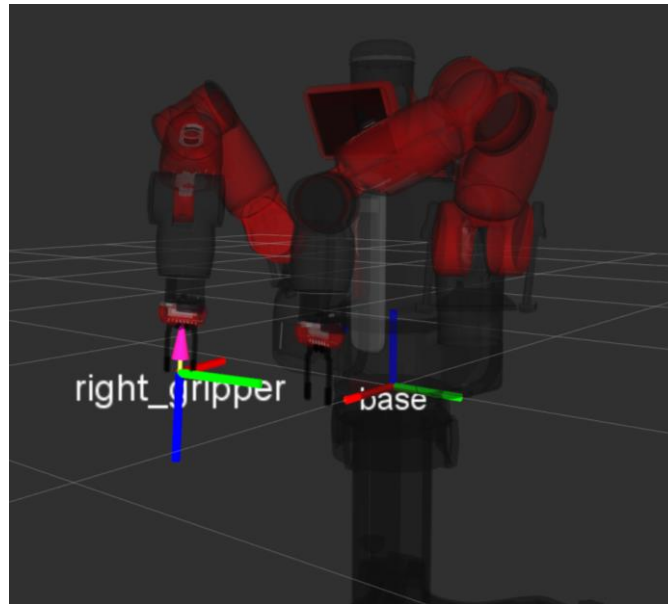


Figura 20. Sistema de coordenadas Baxter (Elaboración propia).

Por otro lado, la mesa estará situada a una altura de 0,928 m sobre la superficie del suelo.

El origen del sistema de coordenadas en FlexSim será diferente, ya que se tomará el punto más bajo del Baxter, es decir, a la altura del suelo y en el centro del mismo. Esto será necesario tenerlo en cuenta ya que en FlexSim lo que se hará será indicarle al robot las diferentes coordenadas a las que se tiene que situar para adquirir las posiciones indicadas anteriormente.

El cilindro con el que se trabaja en esta tarea tiene una altura y un diámetro de 6 cm.

Todos estos datos ya son suficientes para reproducir el movimiento en FlexSim. Para ello, tendremos en cuenta las velocidades máximas aportadas por el catálogo del Baxter especificadas en la Tabla 1. Para poder indicarle al FlexSim que la trayectoria y acciones del Baxter, habrá que recurrir a un diagrama de flujo, denominado en FlexSim *Process Flow*. En ellos se podrán referenciar a los diferentes elementos del entorno 3D creado gracias a una serie de comandos que se encuentran en la librería. Cada uno de ellos se representará dentro de una caja y contendrá un código que se ejecutará cuando un elemento *token* llegue a dicha caja. Los *tokens* van a ser cada uno de los elementos que circulen por el *Process Flow*, se ven representado como círculos verdes.

La lógica empleada en esta tarea se ve representada en la siguiente figura.

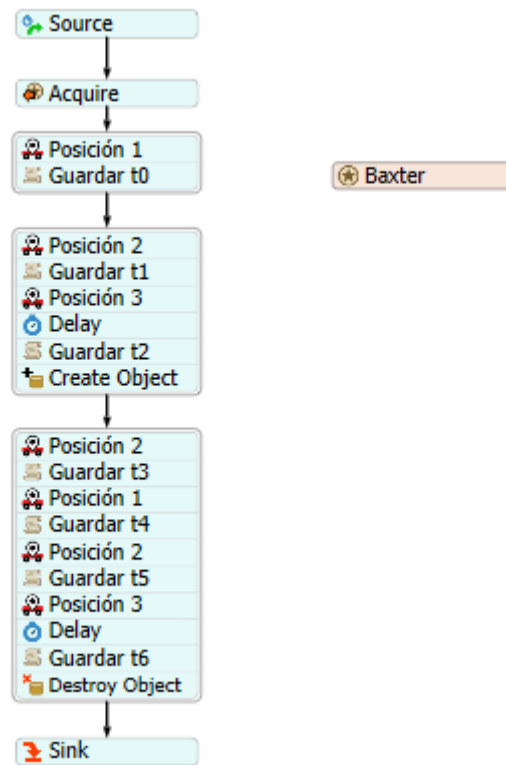


Figura 21. Process Flow Baxter (Elaboración propia).

“Source” será la casilla que cree los *tokens*, en este caso bastará con que cree uno, ya que el ciclo solo se repetirá una vez. La casilla “Acquire” y “Baxter” van a permitir referenciar al robot para que ejecute las tareas. Las casillas “Posición 1”, “Posición 2” y “Posición 3” serán las que contengan las coordenadas a las que se tiene que mover el brazo robótico en las tres posiciones. En “Guardar ti” se guardarán los diferentes tiempos para poder compararlos con los obtenidos en el laboratorio. Se han añadido dos “Delay” que simularán el tiempo en el que las pinzas agarran y sueltan al cilindro, estos tiempos son de 0,72 segundos. Las casillas “Create Object” y “Destroy Object” son para que aparezca y desaparezca el cilindro en el entorno 3D, respectivamente.

En la Figura 22 se representan las posiciones 1, 2 y 3 del Baxter en FlexSim.



Figura 22. Posiciones 1, 2 y 3 Baxter en FlexSim (Elaboración propia).

Los tiempos en segundos obtenidos en el laboratorio y en FlexSim se representan en la siguiente tabla. Además, también se introducirán los porcentajes de error obtenidos en cada uno de los tramos.

Tabla 2. Comparación resultado tiempos Baxter (Elaboración propia).

	Laboratorio	FlexSim	Error
t1	1,82	1,95	7,14%
t2	3,44	3,22	-6,40%
t3	4,34	3,77	-13,13%
t4	6,16	5,72	-7,14%
t5	7,98	7,67	-3,88%
t6	9,6	8,93	-6,98%

Donde t_i es el tiempo que le lleva al Baxter adquirir las diferentes posiciones por orden:

t1: de Posición 1 a Posición 2 (descargado).

t2: de Posición 2 a Posición 3 (descargado).

t3: de Posición 3 a Posición 2 (cargado).

t4: de Posición 2 a Posición 1 (cargado).

t5: de Posición 1 a Posición 2 (cargado).

t6: de Posición 2 a Posición 3 (cargado).

Como se puede observar, los valores de error obtenidos son bastante bajos, por lo que se puede afirmar que la introducción del Baxter en FlexSim aporta resultados muy próximos a la realidad.

6 INTRODUCCIÓN BRAZO ROBÓTICO EN PROCESO DE EMPAQUETADO DE KITS

6.1 Metodología

6.1.1 Descripción del proceso

El objetivo principal del proceso estudiado en este trabajo es el empaquetado de kits en cajas basado en un proceso real.

Dentro del propio proceso existen dos fases diferenciadas. Por un lado existe el proceso de gestión de stock en el almacén y por otro tenemos la fase principal en la cual se formarán los kits colocando los ítems en cada caja.

Los diferentes elementos que contiene el proceso son las cajas de cartón, las piezas o ítems y los palés. Se conoce como kit al conjunto formado por la caja de cartón junto con las piezas que ésta contiene. Debido a que la empresa cuenta con una gran variedad de productos, el número y tipo de piezas de cada kit variará. Los kits se agruparán por lotes y el número de elementos por cada lote también variará. Para transportar todos los componentes y cajas que participan en el proceso se usarán carretillas elevadoras, es por ello que serán necesarios palés para servir de base.

El proceso de gestión de stock consiste en almacenar los palés con las pilas de elementos en diferentes estanterías. Los operarios tendrán diversas funciones que serán vitales para llevar a cabo este proceso. Tendrán que tomar los palés con sus respectivas piezas en las estanterías correspondientes una vez lleguen. Además, cuando sea necesario, los operarios irán a buscar dichas pilas con la ayuda de carretillas elevadoras, de esta manera alimentarán la línea colocándolas en las mesas de rodillos. Por último, también será necesario reponer aquellos palés que estén vacíos con más pilas de ítems.

El proceso de colocación de las piezas se realizará con la ayuda de una cinta transportadora, en ella se irán introduciendo las cajas de cartón sobre las cuales se colocarán los diferentes elementos del kit. Además, se contará con diferentes puestos constituidos por robots y mesas de rodillos. Las cajas viajarán por la cinta transportadora hasta detenerse en cada uno de los puestos. Por las mesas de rodillos llegan las piezas apilados encima de un palé y, una vez que acaban el recorrido sobre la misma, los robots industriales correspondientes a cada puesto introducirán los ítems que correspondan dentro de las cajas. Una vez que las cajas atraviesen la cinta transportadora hasta su fin, ya contendrán todos los elementos del kit y estarán listas.

Este proceso había sido estudiado con anterioridad empleando software de simulación pero sin detallar el puesto robótico y sin modelar la cinemática del robot. La operación de los robots se simplificaba mediante la consideración de un tiempo de ciclo fijo y constante que se había estimado por comparación con otro tipo de procesos del mismo tipo. En dicho estudio previo había una gran incertidumbre sobre los tiempos de ciclo de los robots por lo que se probaron distintos escenarios de tiempo. En dicho trabajo se estimaba como escenario más probable que los tiempos de ciclo de los robots fueran de entre 10 y 12 segundos por pieza colocada. Debido a que estos tiempos eran demasiado altos no se superaba una tasa de producción de 7 cajas por minuto, lo cual es el objetivo de este estudio. Se consideró que la solución al

problema sería duplicar el número de brazos robóticos por puesto de manera que dicho tiempo se redujese a la mitad. Por todo ello, en este apartado se va a simular este escenario en el software FlexSim para ver si dichas conclusiones eran válidas detallando las características del puesto robótico. Para ello fue necesario completar dos tareas fundamentales:

1. Diseñar un puesto robótico geoméricamente válido en el cual se garantice el alcance de los robots a todas las piezas
2. Introducir una cinemática de robots reales garantizando que no se produzcan colisiones con los elementos del proceso.

6.1.2 Elección brazo robótico

En el estudio de este proceso realizado con anterioridad, habían sido propuestos dos modelos de brazos robóticos, el KR Quantec Extra y el KR Quantec Ultra. Ambos habían sido considerados válidos para realizar la función que se les encomienda en cada puesto. Además de estos dos modelos, se comprobará la validez del modelo KR 2102-2 serie. Se introducirá cada uno de ellos en la línea de producción de manera que se pueda valorar cuál de las opciones es la más adecuada.

Las especificaciones que se tendrán en cuenta para la selección del brazo robótico serán las dimensiones, las velocidades de giro y el alcance máximo.

Los tres modelos que se van a estudiar tendrán seis ejes de rotación.

KR QUANTEC ULTRA

Las dimensiones con las que cuentan este tipo de brazos robóticos se suelen representar en ilustraciones debido a la complejidad de la geometría tal y como se ve en a Figura 19.

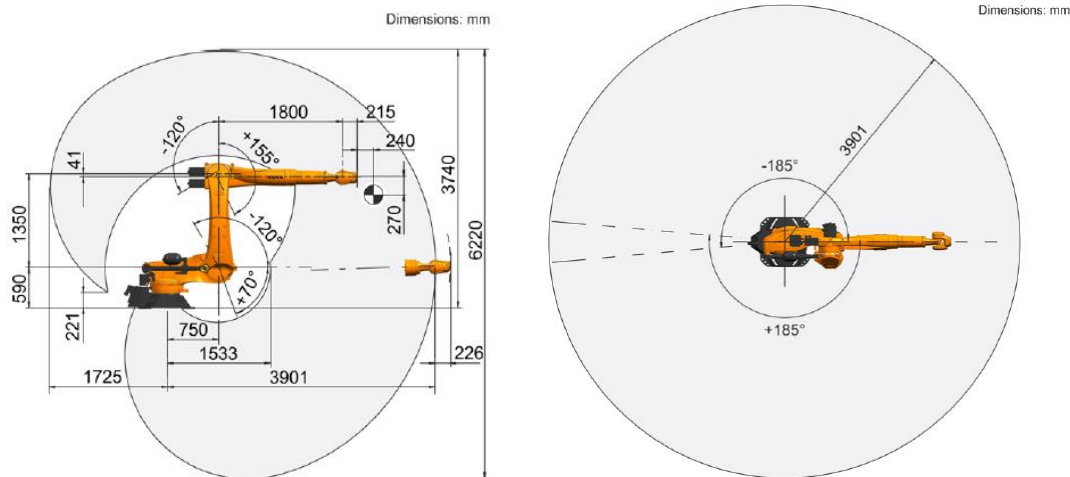


Figura 23. Dimensiones KR Quantec Ultra (KUKA Robotics 2018).

En esta figura también se representan los ángulos de alcance de maniobra de cada una de las articulaciones. De esta manera se podrá saber de una manera simple si el brazo robótico cumple con los requerimientos en cuanto al alcance.

La capacidad de carga de este modelo será de 120 kg y el alcance máximo de 3900 mm.

Las velocidades de giro con carga nominal se especifican en la siguiente tabla.

Tabla 3. Velocidades de giro KR Quantec Ultra (KUKA Robotics 2018).

A1	105 °/s
A2	101 °/s
A3	107 °/s
A4	292 °/s
A5	258 °/s
A6	284 °/s

Las velocidades especificadas son las que tiene el robot alrededor de la articulación correspondiente, es decir, la velocidad de giro A1 se refiere a la que tiene el robot alrededor del eje de la articulación 1.

KR QUANTEC EXTRA

Las dimensiones de este modelo se ven representadas en la siguiente figura.

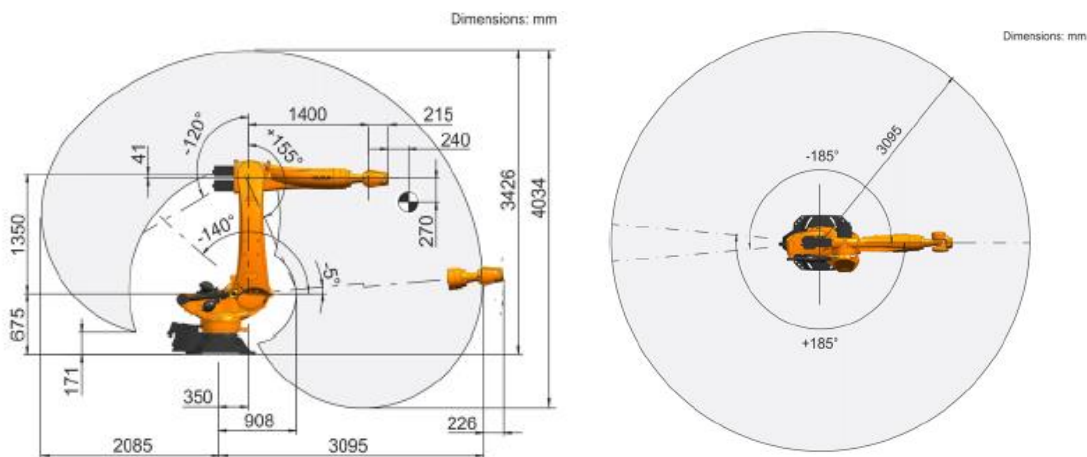


Figura 24. Dimensiones KR Quantec Extra (KUKA Robotics 2018).

La capacidad de carga de este modelo es de 90 kg y el alcance máximo de 3100 mm.

Las velocidades de giro con carga nominal se especifican en la siguiente tabla.

Tabla 3. Velocidades de giro KR Quantec Extra (KUKA Robotics 2018).

A1	105 °/s
A2	101 °/s
A3	107 °/s
A4	292 °/s
A5	258 °/s
A6	284 °/s

KR 2102-2 Serie 2000

Las dimensiones de este modelo son las representadas en la Figura 25.

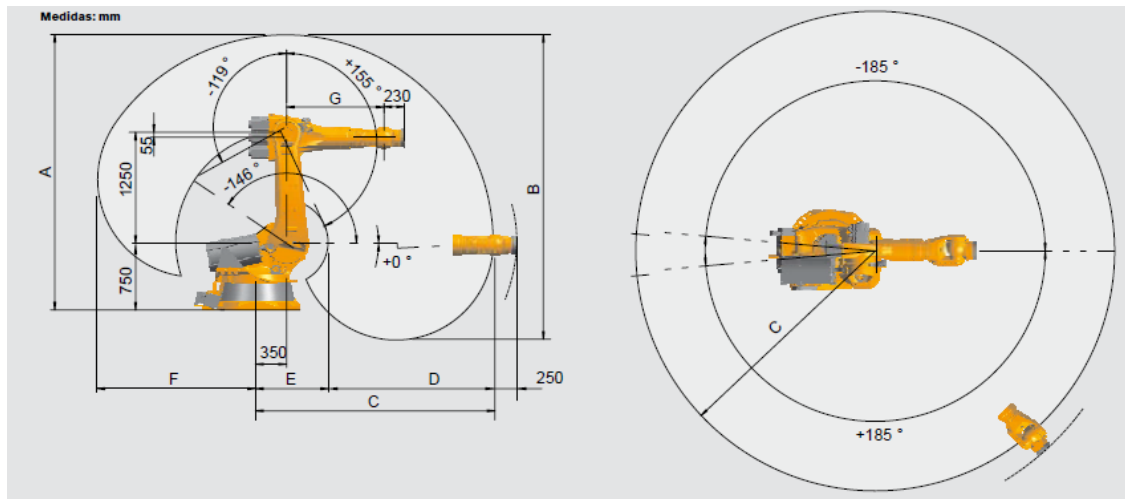


Figura 25. Dimensiones del brazo robótico KUKA 2102-2 (KUKA Robotics 2018).

Las medidas especificadas con letras en la Figura 25 se representan en la siguiente tabla.

Tabla 5. Dimensiones adicionales y volumen de brazo robótico KUKA 2102-2 (KUKA Robotics 2018).

Campo de trabajo	Medidas ⁽¹⁾							Volumen
	A	B	C	D	E	F	G	
KR 210-2	3.100 mm	3.450 mm	2.700 mm	1.875 mm	825 mm	1.788 mm	1.100 mm	55,0 m³

La capacidad de carga de este modelo es de 210 kg y el alcance de 2700 mm.

Las velocidades de giro máximas se representan en la Tabla 6 en función de la carga nominal que soporte el brazo robótico.

Tabla 6. Velocidades angulares de KR2102-2 para cada articulación (KUKA Robotics 2018).

Datos de los ejes	Campo (software)	Velocidad con carga nominal		
		210 kg	180 kg	150 kg
Eje 1 (A1)	±185°	86°/s	86°/s	86°/s
Eje 2 (A2)	0°/-146°	84°/s	84°/s	84°/s
Eje 3 (A3)	+155°/-119°	84°/s	84°/s	81°/s
Eje 4 (A4)	±350°	100°/s	100°/s	100°/s
Eje 5 (A5)	±125°	110°/s	110°/s	110°/s
Eje 6 (A6)	±350°	184°/s	184°/s	184°/s

Tras estudiar los diferentes modelos, se puede decir existen ventajas e inconvenientes en cada uno de ellos. La característica que va a ser crítica para la elección del robot va a ser el alcance máximo, ya que el resto de especificaciones son suficientes para realizar el proceso de la línea de producción en los tres modelos.

Será conveniente establecer las distancias existentes entre los elementos de cada puesto de manera que así se pueda ver si los alcances son suficientes.

En este escenario principal, las cuatro pilas de piezas se sitúan tal y como se ve en la Figura 26. Si solo uno de ellos tiene alcance para tomar las piezas de una pila se ralentizaría el proceso y, si ninguno de los dos llega, se detendría el flujo de cajas. Por ello, la situación ideal sería que ambos robots llegasen a todas las pilas, por lo que la distancia entre los elementos tiene que ser menor al alcance máximo.

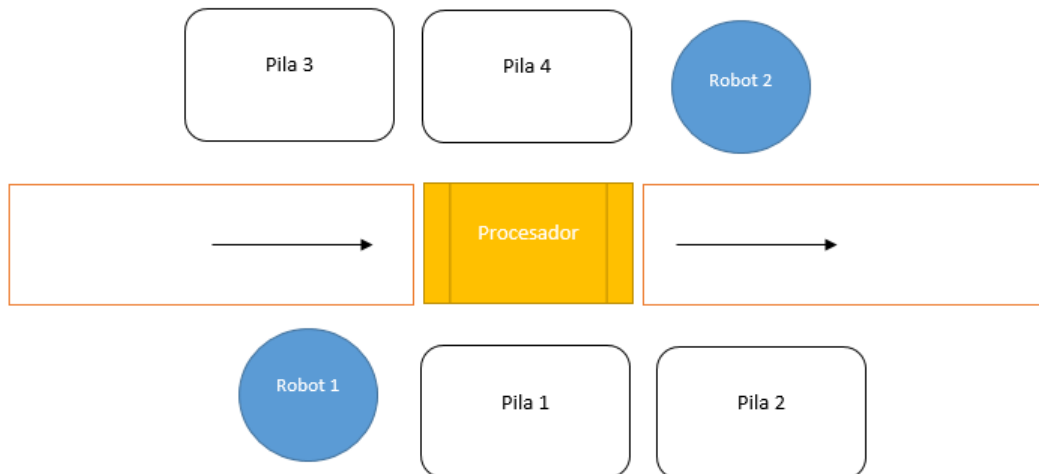


Figura 26. Vista en planta de un puesto de trabajo del proceso (Elaboración propia).

Las distancias entre los brazos robóticos y las pilas de pizzas se indican en la siguiente tabla. Cabe decir que las distancias en el resto de puestos serán las mismas.

Tabla 7. Distancias en mm entre robots y pilas (Elaboración propia).

(mm)	Robot 1	Robot 2
Pila 1	1400	2360
Pila 2	2600	1900
Pila 3	1900	2600
Pila 4	2360	1400

Teniendo en cuenta estas distancias, se puede concluir que:

- KR Quantec Ultra: cuenta con un gran alcance máximo, sin embargo, debido a sus dimensiones y demás características va a ser el más costoso de los tres modelos.
- KR Quantec Extra: cuenta con un alcance máximo menor, lo cual va a producir ciertos choques entre el robot y las pilas de piezas en ciertas ocasiones durante el proceso.
- KR 2102-2: es el que cuenta con menor alcance de los tres, por lo que teniendo en cuenta la distribución de los elementos del puesto no va a tener alcance para llegar a ciertas piezas y, además, se van a producir numerosos choques durante el proceso.

Por todo ello, el brazo robótico seleccionado ha sido el KR Quantec Ultra.

6.1.3 Implementación Quantec Ultra

El brazo robótico implementado en el proceso es el modelo KR Quantec Ultra.

Cuenta con una serie de ventajas que van a permitir llevar a cabo el proceso de manera adecuada:

- Cuenta con las dimensiones adecuadas para situarse en el espacio del modelo.
- El alcance del brazo es suficiente para realizar la función para la cual está encomendado.

- Cuenta con las articulaciones adecuadas para poder realizar su función con facilidad.
- Las velocidades angulares son suficientes para poder realizar las trayectorias en el tiempo esperado.
- Cuenta con una capacidad de carga suficiente para transportar los diferentes elementos con los que cuenta el proceso.
- Tiene una gran precisión.

Se ha elevado la base de los robots con respecto del suelo una altura de 850 mm. De esta manera alcanzará con mayor facilidad la parte superior de las pilas de piezas.

Las figuras 27, 28, 29 y 30 mostrarán la manera en la que se tendrá que posicionar el brazo robótico para poder alcanzar las cuatro pilas. Bastará con mostrar la posición para uno de los robots del puesto, ya que el otro tendrá que realizar los mismos movimientos.

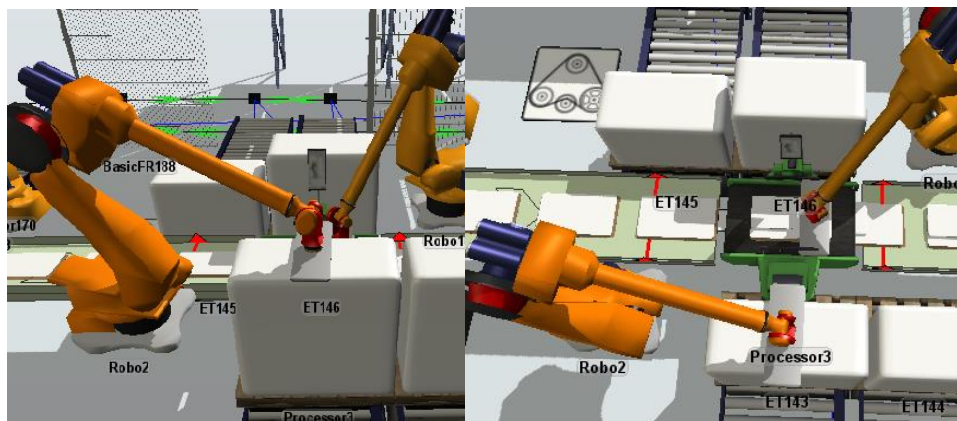


Figura 27. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 1 (Elaboración propia).

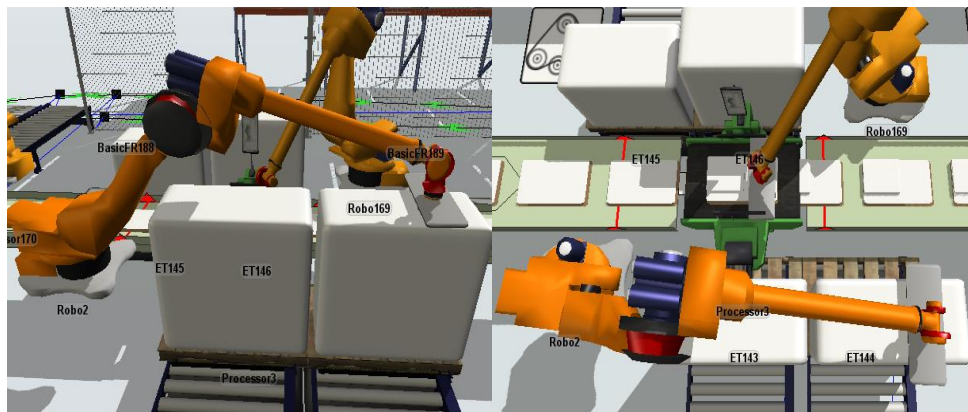


Figura 28. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 2 (Elaboración propia).

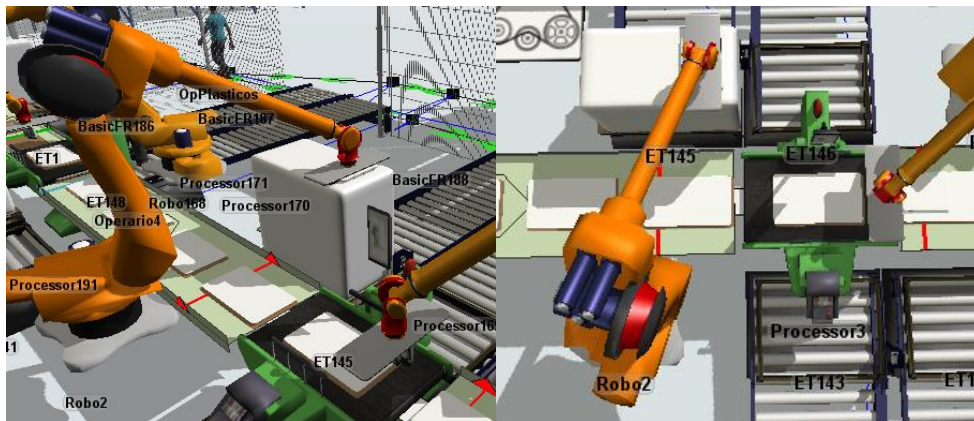


Figura 29. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 3 (Elaboración propia).

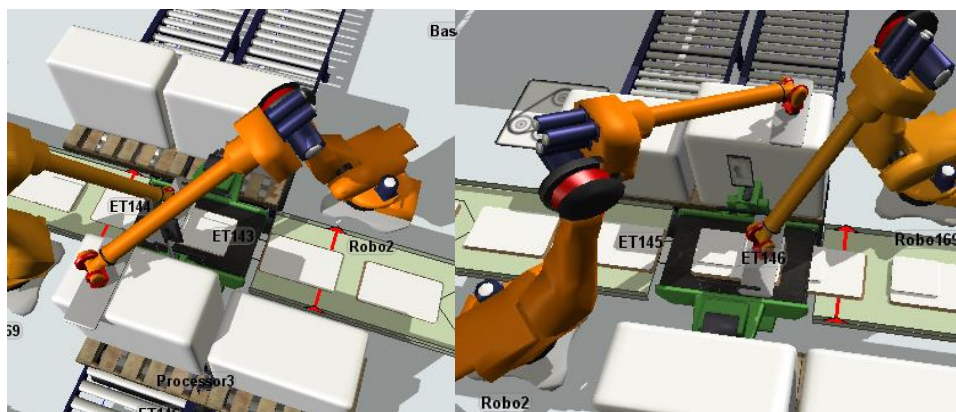


Figura 30. Vistas del brazo robótico KR Quantec Ultra alcanzando la Pila 4 (Elaboración propia).

Para este trabajo, teniendo en cuenta el tipo de elementos que se transportan así como su peso, las velocidades de giro que se han considerados más adecuadas se especifican en la Tabla 9 y su valor será la mitad del valor de las velocidades máximas especificadas en la Tabla 8. Se indican las velocidades también en mm/s debido a que son en estas unidades en las cuales se introducen las velocidades en FlexSim. Para introducir este parámetro en FlexSim, en las propiedades del robot habrá que marcar la opción de *Use Join Speeds*. De esta manera se podrán especificar las velocidades con las que actúe cada una de las articulaciones.

Para pasar de $^{\circ}/s$ a mm/s será necesario conocer los radios de giro de cada una de las articulaciones y usar los factores de conversión necesarios para llevar a cabo el cambio de unidades. Los radios de giro tomados en cada caso para cada articulación también se especifican en la Tabla 8.

Tabla 8. Velocidades angulares de KR Quantec Ultra aplicadas a cada una de las articulaciones (Elaboración propia).

Ejes	Velocidades angulares ($^{\circ}/s$)	Velocidades angulares (mm/s)	Radios de giro (mm)
Eje 1	52,5	448	489
Eje 2	50,5	1101	1250,00
Eje 3	53,5	1400	1500,00
Eje 4	146	285	112
Eje 5	129	346	154
Eje 6	142	867	350

Además de las velocidades, en la pestaña *Use Join Speeds* también habrá que especificar las aceleraciones y deceleraciones de cada articulación. Estos parámetros van a ser de mayor importancia que las velocidades en este proyecto. Esto es debido a que el brazo robótico realiza trayectos muy cortos y con muchos cambios de sentido durante el proceso y tendrá que acelerar y decelerar un gran número de veces. Se ha seleccionado un valor diez veces mayor al de la velocidad de manera que los brazos robóticos lleguen a alcanzar dicha velocidad en 0,1 segundos.

Tabla 9. Aceleraciones y deceleraciones angulares de KR2102-2 aplicadas a cada una de las articulaciones (Elaboración propia).

Ejes	Aceleración (mm/s ²)	Deceleración (mm/s ²)
Eje 1	4480	4480
Eje 2	11010	11010
Eje 3	14000	14000
Eje 4	2850	2850
Eje 5	3460	3460
Eje 6	8670	8670

6.1.4 Funcionamiento de cada puesto

En total habrá seis puesto de trabajo los cuales constarán de dos brazos robóticos, un procesador, cuatro mesas de rodillos y las cintas transportadoras de entrada y de salida.



Figura 31. Vista en planta de todos los puestos (Elaboración propia).

La situación de cada uno de los puestos de trabajo va a ser análoga, las piezas van a llegar apiladas a través de las mesas de rodillos hasta el final de las mismas. A su vez, las cajas llegarán a través de la cinta transportadora de entrada de cada puesto y se irán situando en el procesador, donde permanecerán inmóviles hasta que se le sitúe la pieza o piezas correspondientes encima. Una vez que tenga la pieza que le corresponde pasará a la cinta transportadora de salida y continuará hacia el siguiente puesto.

Los brazos robóticos comenzarán a tomar una a una las piezas de la pila que corresponda. En cada puesto puede llegar a colocarse una pieza por caja, más de una o ninguna. De hecho, en los puestos finales muchas veces ya no se colocan piezas encima debido a que la caja ya estaría completa con todas las piezas que le corresponden.

Teniendo en cuenta que la función de los brazos robóticos consiste en tomar cada una de las piezas de la pila y situarlas en la caja de cartón que se situará en el procesador, van a existir varias posibilidades para programarlos.

- En primer lugar podría realizar todos los transportes de piezas el mismo robot y solo se recurriría al segundo robot cuando el primer robot no tenga alcance suficiente para llegar a la pila de piezas. Esta opción se ha descartado ya que no se estaría haciendo el uso más efectivo de los robots, se estaría perdiendo tiempo y malgastando la posibilidad de actuación del segundo robot, que permanecería inmóvil la mayor parte del tiempo, además teniendo en cuenta que en este caso cada robot tiene alcance para llegar a las cuatro pilas, el segundo robot no actuaría en ningún momento.
- La segunda opción consiste en alternar los robots, cada uno coge una pieza de manera que el primero coja la primera, el segundo la segunda, el primero de nuevo la tercera y así sucesivamente. En caso de que se tengan que colocar más de una pieza por caja, esto lo realizaría el mismo robot. Esta opción también se ha descartado debido a que en el caso de que exista más de una pieza por caja estamos perdiendo tiempo, ya que los robots no se alternan.
- Por último, existe la posibilidad de que los robots se alternen en todo caso, incluso cuando existe más de una pieza por caja. Esta opción ha sido la elegida porque es la que se ha considerado que hace un mayor uso de los recursos, de manera que es la más rentable. Ello es debido a que mientras un robot está haciendo un viaje, el otro puede ir moviéndose de manera que se ahorre ese tiempo que, en caso de usar un solo robot se perdería.

CINTAS TRANSPORTADORAS

En cuanto a las cintas transportadoras, va a haber un total de siete, las cuales van a estar separadas entre sí por los procesadores. Todas ellas estarán situadas a la misma altura. La longitud de las mismas va a depender de la separación existente entre los puestos.

A las cintas habrá que imponerle una velocidad de flujo, la cual, en este caso ha sido de 250 mm/s. Este valor ha sido seleccionado de manera que no sea lo suficientemente bajo como para que ralentice el proceso y que al mismo tiempo no sea lo suficientemente alto como para que a los robots no les dé tiempo de realizar sus viajes antes de que llegue otra pieza.

Además, habrá que imponer una separación entre las cajas que fluyen sobre las cintas, la cual se ha considerado adecuada con un valor de 200 mm.

A lo largo de las cintas transportadoras se van a situar unos elementos llamados *PhotoEyes*. Desempeñarán la función de unos sensores de posición, de manera que si tienen una caja encima enviarán una señal al *Process Flow* en el que se encuentren referenciados.

Las cajas avanzarán por la cinta transportadora siempre y cuando el robot tenga material disponible en las pilas. En caso de que no haya una pila de piezas disponible en las mesas de rodillos porque aún no ha llegado, el robot permanecerá parado y, en consecuencia, el flujo de cajas también. Este hecho será posible gracias a la lógica creada en el siguiente diagrama.

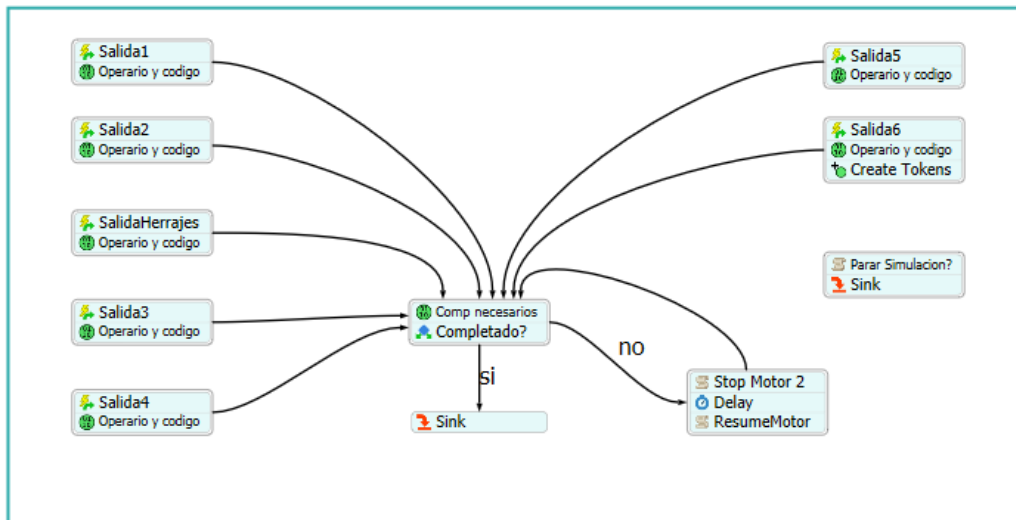


Figura 32. Diagrama Comprobación Colocación (Elaboración propia).

El movimiento de cajas a través de las cintas transportadoras se produce gracias a la existencia de unos motores. Habrá un motor por cada cinta transportadora. Si no hay material disponible en las mesas de rodillos para que el robot pueda situar las piezas en las cajas, será el motor el que parará el flujo de cajas de manera que tenga velocidad 0. Dichas cajas se pararán una vez lleguen al PhotoEye que se encuentra próximo a la salida de cada procesador.

Además, cabe mencionar que en caso de que esto ocurra, no se irán acumulando las cajas a lo largo de la cinta transportadora reduciéndose el espacio entre las mismas, sino que permanecerán los 200 mm de distancia entre ellas y simplemente se pararán.

PROCESADORES

Los procesadores son los elementos sobre los que se situarán las cajas a la espera de que se le coloquen las piezas correspondientes encima. Las cajas permanecerán inmóviles hasta que estén completas y pasen a la cinta transportadora de salida. La capacidad del procesador será de una sola caja.

El tiempo que cada caja permanecerá inmóvil en los procesadores se va a definir con una variable global. Se hará de esta manera ya que habrá que ajustar la velocidad de flujo de las cintas transportadoras con este tiempo de espera de manera que al robot le sea posible realizar su función a tiempo sin que se le acumulen cajas a la entrada del puesto. Se le ha dado un valor de 3 segundos de espera en cada procesador.

PROGRAMACIÓN DEL ROBOT

Los robots de cada puesto deben estar programados para que puedan cumplir la función que se les encomienda. La función que cumplen se ve representada de manera simplificada en el siguiente diagrama de flujo.



Figura 33. Diagrama de flujo general de la función del brazo robótico (Elaboración propia).

La llegada de la pieza al procesador va a ser la primera acción de relevancia para el brazo robótico, ya que será en este momento cuando tenga que comenzar a realizar su recorrido.

En FlexSim, por cada una de las piezas de la pila se creará un *token*. De esta manera, el *token* irá recorriendo el diagrama de flujo de manera que por cada pieza existente se ejecutarán todas las acciones del diagrama de la Figura 33.

La segunda caja del diagrama corresponde a la acción “Tomar robot”. En esta parte del diagrama se decidirá cuál de los dos robots del puesto será el encargado de realizar las siguientes tareas. Teniendo en cuenta que los robots se alternarán, bastará con crear una variable global que cambie de valor cada vez que un *token* llegue a esta parte del diagrama. Dichos valores serán siempre 0 o 1 y cada uno de ellos se asociará a uno de los brazos robóticos.

El siguiente bloque del diagrama corresponde al *Travel Ida*. Este viaje es el que realiza el robot desde su posición inicial hasta la pieza correspondiente de la pila, durante este trayecto permanecerá descargado. Se va a descomponer en tres viajes diferentes.

- *TravelLadoProc*: este trayecto tiene como destino una posición situada a 700 mm al lateral del procesador, en dirección de la cinta transportadora hacia fuera. Es decir, en el caso del robot inferior viajara a 700 mm a la izquierda del procesador y en el caso

del superior a 700 mm a la derecha. En ambos casos la altura a la que se trasladará será de 1500 mm por encima de la superficie del suelo. En el caso de que sea el primer viaje del robot de todo el proceso, la posición inicial de la que partirá será la que se muestra en la Figura 34, si, por el contrario, nos encontramos en cualquier otro momento del proceso que no sea el principio, partirá desde la posición *TravelNoChoque*, ya que es la última del ciclo, tal y como se representa en la Figura 35.

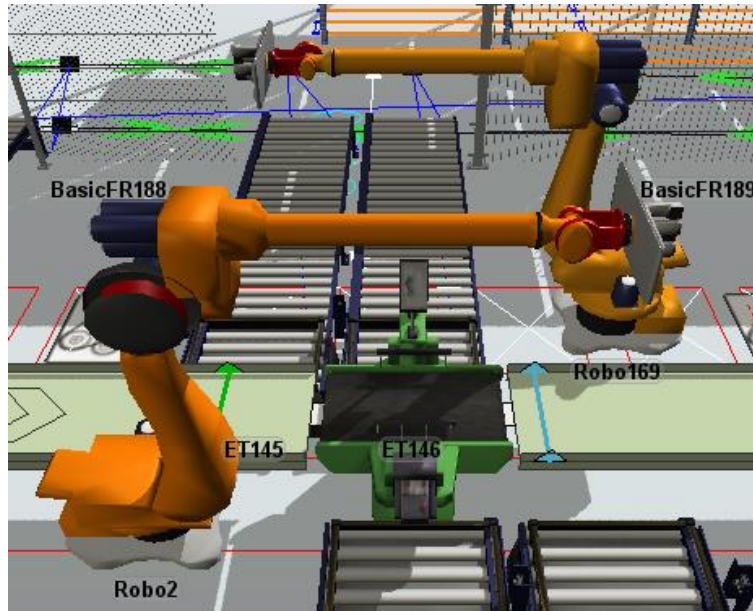


Figura 34. Posición inicial brazos robóticos (Elaboración propia).

- *TravelArribaPila*: viaje desde la posición final del anterior trayecto hasta 300 mm por encima de la pieza correspondiente.
- *Travel 1*: viaje desde la posición final del *TravelArribaPila* hasta la propia pieza.

Estos tres trayectos se representan en la siguiente figura, el *TravelLadoProc* en azul, el *TravelArribaPila* en naranja y el *Travel 1* en amarillo.



Figura 35. Travel Ida (Elaboración propia).

Para referenciar la posición de cada una de las piezas en la pila que corresponda habrá que tener en cuenta que al ir recogiendo elementos de la pila, la altura de la misma irá bajando y el robot tendrá que adaptarse a esta nueva altura. Además, cada nivel de la pila contiene seis

piezas (dos filas y tres columnas), por lo que el robot tendrá que viajar justo a la pieza que tenga que coger dentro del nivel más alto de la pila. Para ello habrá que especificarle al programa tres coordenadas correspondientes a las del centro de cada pieza a las que se les llamará xloc, yloc y zloc. Para poder indicarle esto al brazo robótico habrá que crear un código para cada una de las tres coordenadas, en FlexSim a dichos códigos se les conoce como *Custom Codes*.

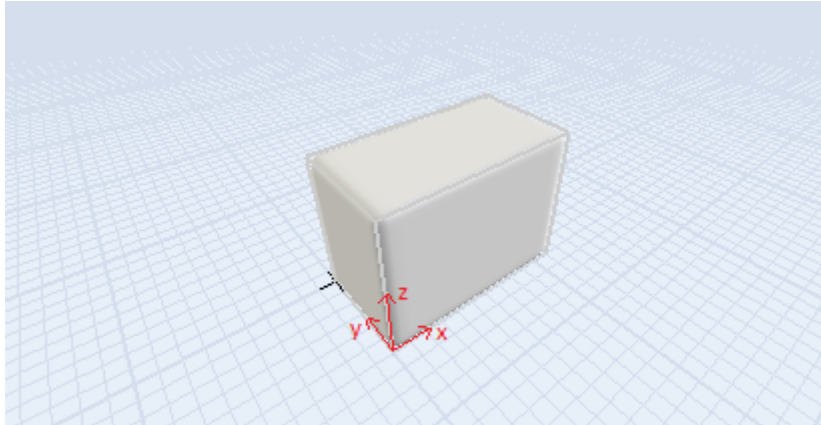


Figura 36. Sistema de coordenadas de una pila de piezas (Elaboración propia).

La coordenada z, tal y como se ve en la Figura 36, es la vertical, la x será la que siga la pila en su parte más larga y la y será la del ancho de la pila.

- **Coordenada z**

La coordenada z que le hay que indicar al brazo robótico es la correspondiente a la altura de la pila de piezas en ese momento. Será necesario utilizar la Ecuación 1, ya que esta coordenada variará a lo largo de los ciclos.

$$[zloc = pale.nElementos / (contenido.numX * contenido.numY) * contenido.dimZ + 600] \text{ Ec 1}$$

Con pale.nElementos se refiere al número de piezas que contiene la pila en total.

Con contenido.numX y contenido.numY se refiere al número de elementos que hay en cada fila y columna de un nivel, respectivamente.

Con contenido.numZ se especifica el número de niveles que hay en la pila.

El valor que se suma al final de la ecuación es la altura existente entre el suelo y la parte superior del palé.

- **Coordenada x**

La coordenada x que habrá que indicar, tal y como se ve en la Figura 18, será la correspondiente al centro de la pieza en sentido longitudinal de la pila.

$$[posX = ((pale.nElementos / contenido.numX) - 1) \% contenido.numY] \text{ Ec 2}$$

La Ec 2 devolverá el valor 0, 1 o 2 debido a que va a haber tres elementos por fila. Si se devuelve un 0 el robot tendrá que tomar la primera pieza de la fila, si se devuelve un 1 la segunda y si se devuelve un 2 la tercera. La Ec 3 servirá para asignar las coordenadas x a las que tendrá que viajar el brazo robótico a cada uno de los números obtenidos en la Ec 2.

El “%” indica la operación en la que se obtendrá el resto.

$$xloc = estacion.location.x + contenido.dimY * (0.5 + posX); \text{ Ec 3}$$

- **Coordenada y**

La coordenada y corresponderá a la del centro de cada pieza en el eje y. Habrá dos elementos por columna, por lo que, análogamente al caso de la coordenada x, las ecuaciones serán las siguientes.

$$[\text{posY}=(\text{pale.nElementos}-1)\%\text{contenido.numX}] \quad \text{Ec 4}$$

Con la Ec 4 se obtendrá u 0 o un 1 dependiendo de si el robot tiene que tomar el primer elemento de la columna o el segundo. La Ec 5 servirá para asignar la coordenada y del centro de cada pieza en función de si en la Ec 4 se ha obtenido un 0 o un 1.

$$[\text{coordY}=\text{estacion.location.y}-\text{contenido.dimX}*(0.5+\text{posY})] \quad \text{Ec 5}$$

Bastará con indicar en el *Process Flow* que estas tres coordenadas son las de destino del *Travel 1*. Para el caso del *TravelArribaPila*, las coordenadas serán las mismas a diferencia de que a zloc habrá que sumarle 300 mm.

El siguiente paso del diagrama es el momento en el que el robot captura la pieza de la pila. En la realidad, esta acción se llevaría a cabo con una ventosa en el extremo del brazo robótico o con otro tipo de elemento de captura. Para simular este momento, habrá que tener en cuenta el tiempo durante el cual se cogería la pieza, el cual, en base a otros procesos, se ha considerado que tenga una duración de 1 segundo.

El siguiente paso consiste en el viaje hasta el procesador, al que se le ha llamado *Travel Vuelta*. En este viaje, el brazo robótico partirá cargado desde la pieza situada en la pila hasta el procesador. Este trayecto se dividirá en seis tramos tal y como se ve en la Figura 37:

- *TravelArribaPila* (en azul): igual que el anterior *TravelArribaPila*. De nuevo el brazo robótico sube hasta 300 mm por encima de la pieza correspondiente.
- *TravelLadoPila* (en naranja): viaje desde la posición final de *TravelArribaPila* hasta uno de los lados de la pila en sentido longitudinal de la misma y con una altura de 2000 mm por encima de la superficie del suelo. En el caso del robot inferior viajará al lado izquierdo de la pila y en el del robot superior al lado derecho.
- *TravelLadoProc*: (en verde): la posición final es igual a la del *TravelLadoProc* anterior, el robot irá desde la posición final del *TravelLadoPila* hasta una posición situada a 700 mm al lateral del procesador, en dirección de la cinta transportadora hacia fuera. La altura a la que se trasladará será de 1500 mm por encima de la superficie del suelo.
- *TravelNoChoque* (en amarillo): es similar al *TravelLadoProc* a diferencia de que la posición inicial estará situada a menor altura, a 1100 mm por encima de la superficie, que corresponde con una altura un poco superior a la de la superficie del procesador. Esta posición se representa en la siguiente figura.
- *Travel 2* (en rojo): desde la posición final de *TravelNoChoque*, el brazo robótico se trasladará al centro del procesador a la altura de la superficie del mismo.

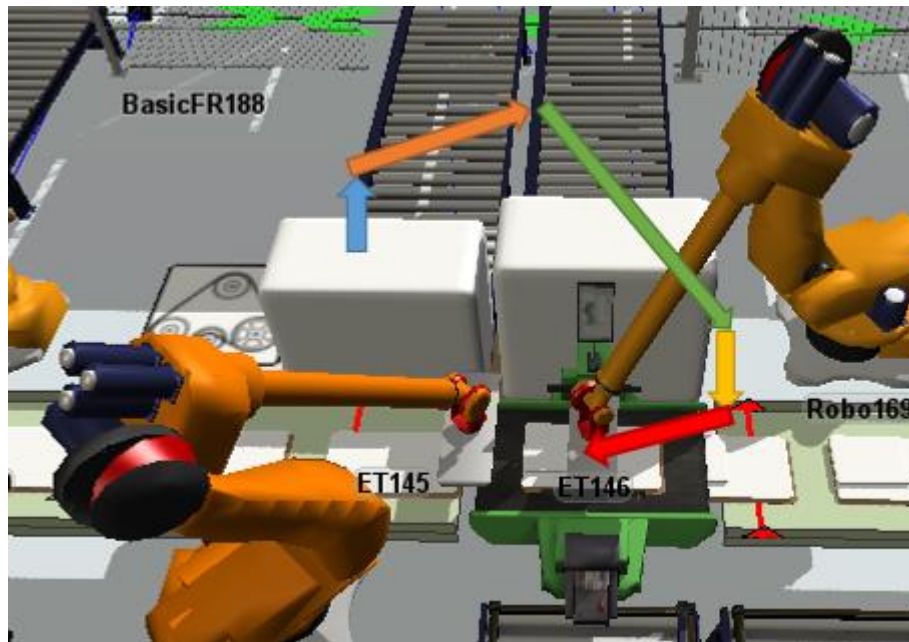


Figura 37. Travel vuelta (Elaboración propia).

A continuación, se soltará la pieza que porte el brazo robótico en ese momento. Al igual que para capturarla, será necesario añadir una espera que represente el tiempo que le lleva al robot soltar la pieza. Tendrá un valor de 1 segundo, al igual que para capturarla.

De nuevo se volverá a realizar el *TravelNoChoque*, aunque esta vez el brazo robótico ya estará descargado. Este trayecto se representa en la siguiente figura.

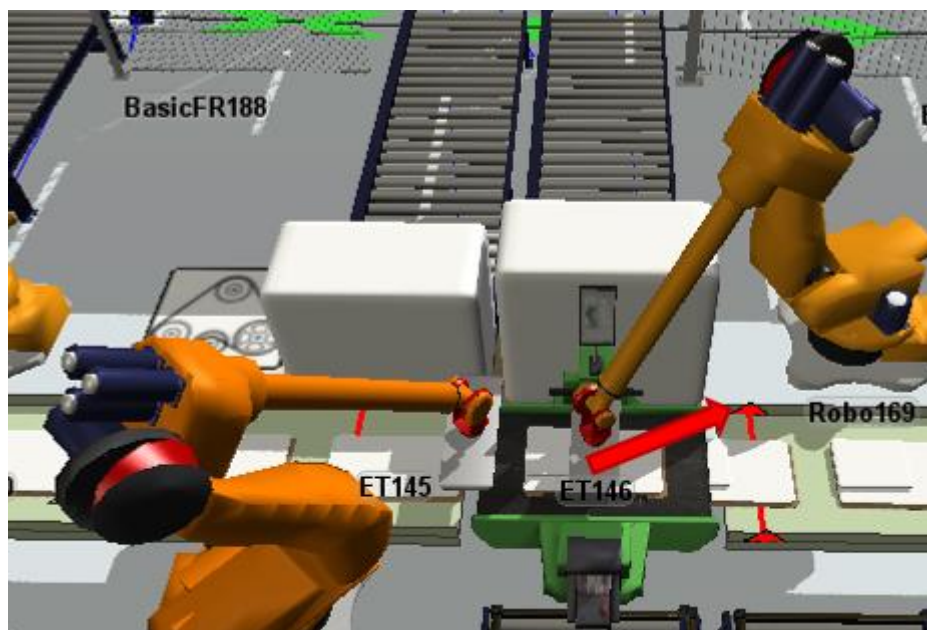


Figura 38. TravelNoChoque (Elaboración propia).

La posición *TravelNoChoque* puede considerarse la final del robot, ya que será de la que parta el brazo robótico para realizar de nuevo el *Travel 1*.

Para el brazo robótico inferior, los trayectos serían análogos a los de la figura teniendo en cuenta que el *TravelLadoPila*, el *TravelLadoProc* y el *TravelNoChoque* serían hacia la izquierda en lugar de hacia la derecha.

Habr  que tener en cuenta que se han programado los brazos rob ticos para que no realicen en *Travel Ida* ni el *Travel Vuelta* al mismo tiempo, sino que se alternen y cuando uno acabe pueda empezar el otro.

Existir  una situaci n cr tica en la cual el brazo rob tico colisionar  con una de las pilas de piezas. Esta situaci n se dar  cuando se cumplan al mismo tiempo las siguientes condiciones:

- La pila de piezas que situada al lado del robot, es decir, la pila 1, se encuentre en su altura m xima de 1 m.
- El brazo rob tico tenga que transportar las piezas que se encuentran en la pila 2.
- El brazo rob tico tenga que alcanzar la pieza m s alejada del mismo dentro de la misma fila de la pila 2.

La situaci n tambi n podr  darse cuando la diferencia de altura entre la pila 1 y la pila 2 sea de 670 mm o m s (siendo la pila 1 la m s alta) y el robot tenga que transportar cualquiera de las piezas de la pila 2.

En el caso de que se presente esta situaci n el software de control del robot seleccionar  al otro robot aunque tenga que repetir su tarea. Cabe decir que este caso se presenta con muy poca frecuencia durante el proceso que se est  estudiando. En la Figura 39 se representa el momento en el que se produce este caso cr tico.

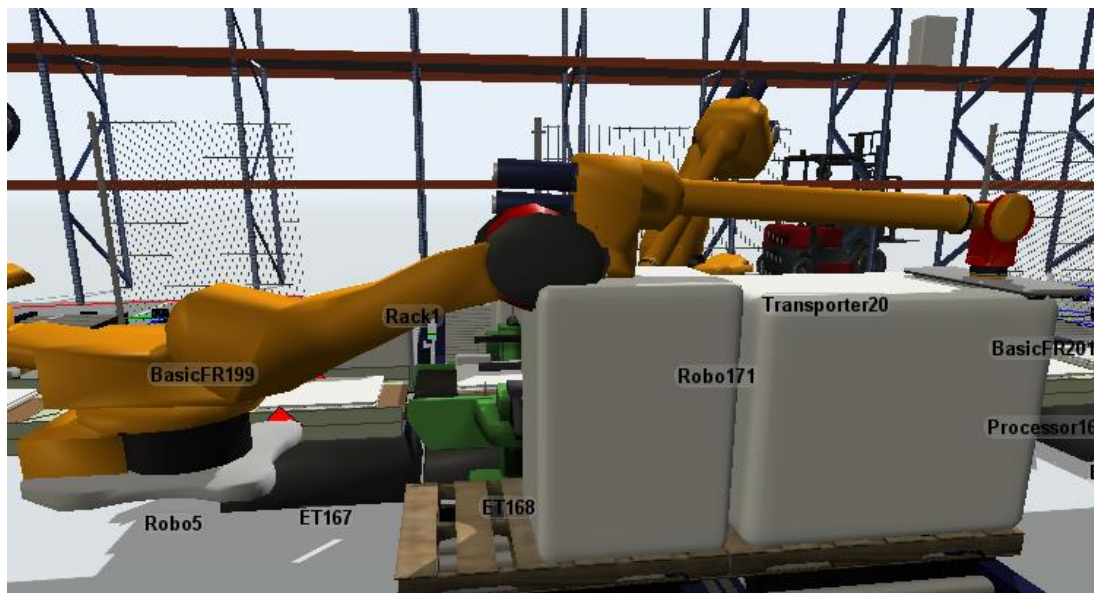


Figura 39. Momento cr tico de colisi n (Elaboraci n propia).

En conclusi n, se puede afirmar que con este conjunto de movimientos se evitar  la colisi n entre ambos brazos rob ticos ya que ambos van y vuelven por unas trayectorias distintas de manera que no se toquen en ning n momento.

Para llegar a este resultado, se ha realizado un estudio en el cual se han descartado otros trayectos m s simplificados que daban lugar a error. Dichos trayectos son los siguientes:

- La trayectoria de la que se parti  era una en la cual ambos brazos rob ticos simplemente ten an dos tramos de viaje, uno de ellos descargado y otro cargado (*Travel 1* y *Travel 2*). En ambos trayectos el brazo rob tico segu a una l nea recta desde el procesador hasta la pieza contenida en la pila correspondiente. Esta primera hip tesis daba lugar a un gran n mero de fallos. Por un lado ambos brazos rob ticos colisionaban en la gran mayor a de los ciclos debido a que la ida y la vuelta se realizaba por el mismo trayecto o uno muy similar. Por otro lado, los brazos rob ticos a su vez

también colisionaban con las pilas de piezas ya que las atravesaban. Por todo ello este recorrido fue descartado. Esta situación se representa en la siguiente figura.

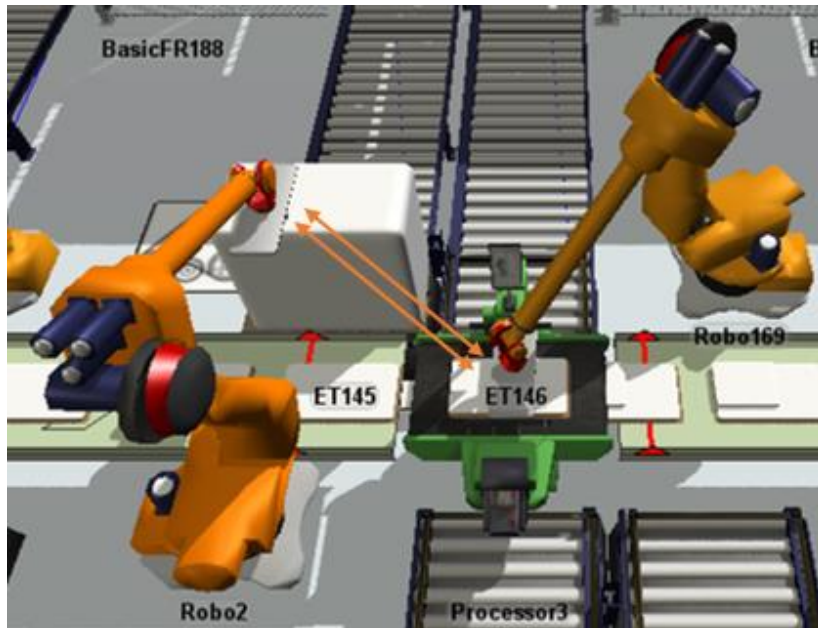


Figura 40. Primera prueba trayectorias (Elaboración propia).

- El siguiente trayecto que se diseñó fue igual que el anterior a diferencia de que se le añadió el *TravelLadoPila* después del *Travel 1* (aunque esta vez con una altura igual a la de la superficie de la pila correspondiente) y el *TravelNoChoque* al final del ciclo de manera que las trayectorias describen un ocho tal y como se ve en la figura.

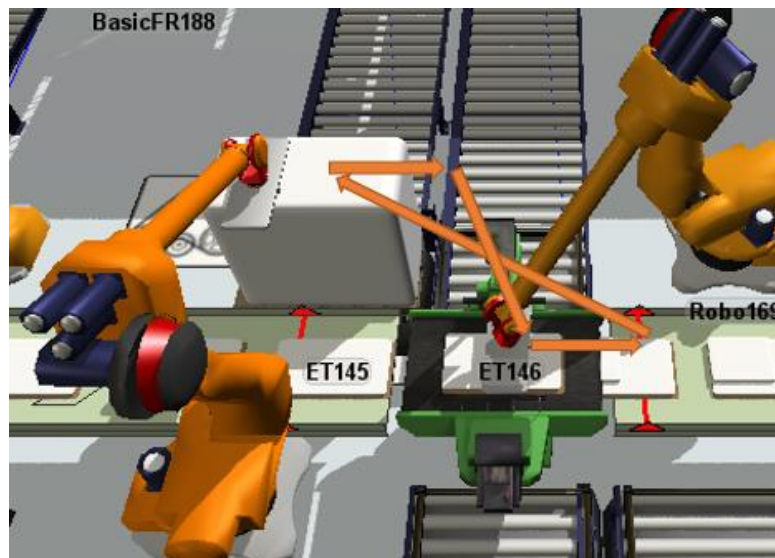


Figura 41. Segunda prueba trayectorias (Elaboración propia).

Sin embargo, esta situación también dio lugar a problemas debido a que los brazos robóticos continuaban chocando con las pilas a las que tenían que acceder en cada momento. Además, también chocaban con las pilas que se encontraban al lado al realizar el *TravelLadoPila*.

- El siguiente recorrido que se probó fue uno en el cual se añadían los *TravelArribaPila* con la intención de que los brazos robóticos dejaran de colisionar con la pila

correspondiente. Además, se añadió otro tramo similar al *TravelLadoPila* aunque se diferenciaba en que estaba a la misma altura que el *TravelArribaPila*, es decir, a 300 mm por encima de la superficie de la pila. El resto de tramos eran iguales al del caso anterior.

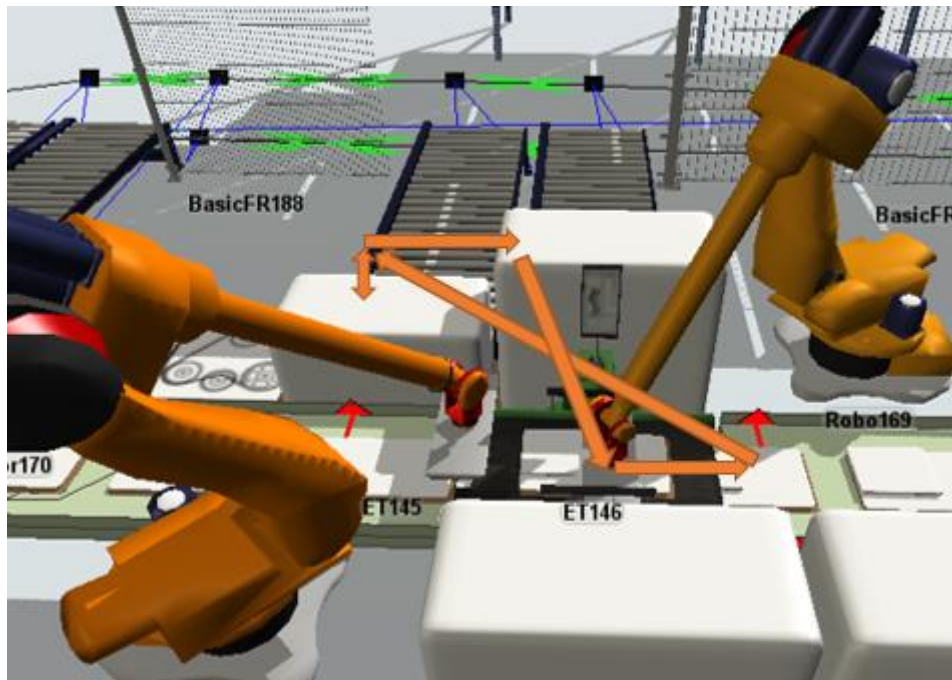


Figura 42. Tercera prueba trayectorias (Elaboración propia).

Con este caso, los brazos robóticos seguían chocando con las pilas de al lado. Además, chocaban entre ellos en algunas situaciones a la hora de realizar el *Travel 1* y el *Travel 2*, ya que muchas veces en estos trayectos se cruzaban.

- Finalmente, la siguiente prueba de trayectorias que se realizó fue la definitiva, en la cual los *Travel 1* y *Travel 2* tienen trayectorias diferentes para cada brazo robótico de manera que no se entrecruzan en ningún momento y, además el *TravelLadoPila* se elevó a una altura de 20000 mm sobre la superficie del suelo de manera que en ningún momento ninguno de los robots choque con la pila de al lado.

A continuación se describirán los comandos necesarios para programar al robot en el software FlexSim. Para poder programar las tareas y trayectorias que llevarán a cabo los brazos robóticos se recurrirá a un Process Flow.

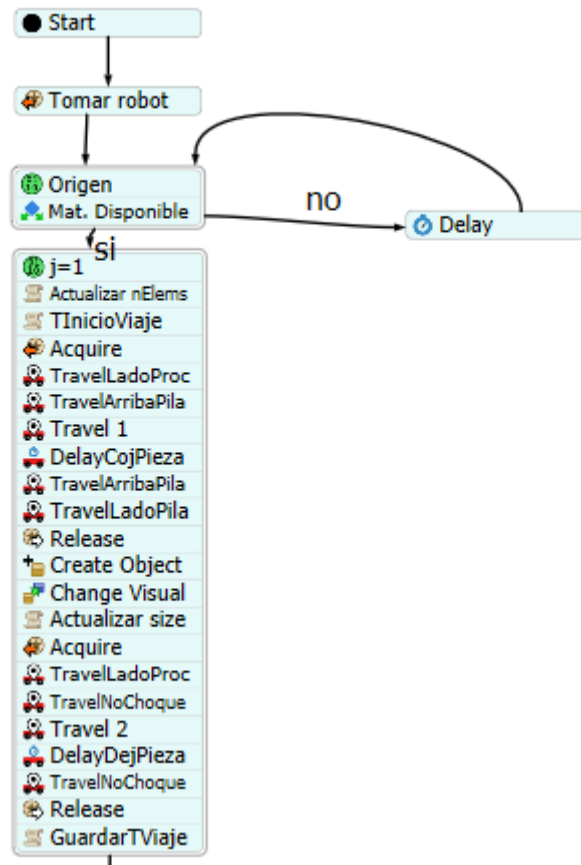


Figura 43. Process Flow parte brazo robótico (Elaboración propia).

El esquema es análogo al representado en el diagrama de flujo simplificado de la Figura 3. Sin embargo, habrá que añadirle diferentes casillas a mayores para que el proceso pueda llevarse a cabo de manera adecuada.

A través de la casilla “Start” entrarán los diferentes *tokens* correspondientes a cada una de las piezas. A continuación, en “Tomar robot” se va a seleccionar el robot de cada puesto que actuará a la hora de colocar la pieza que llega en ese momento.

La casilla “Origen” contiene un código necesario para saber cuál será la pila de piezas a la que se tiene que trasladar el brazo robótico. Junto a este comando se encuentra un bucle que tendrá la función de comprobar si existe material disponible para continuar con el proceso, de lo contrario, se ejecutará una espera deteniéndose el flujo de cajas.

El siguiente bloque de acciones va a ser el que contenga los trozos de código con los cuales se ejecutarán los viajes del robot.

En la primera casilla de este bloque “j=1”, se declararán unas variables necesarias para ejecutar diferentes códigos posteriores tales como las coordenadas a las que se encuentran las diferentes piezas explicadas anteriormente (xloc, yloc y zloc). A continuación, la casilla “Actualizar nElements” será un código necesario para devolver el valor actualizado del número de elementos de la pila, ya que ésta se irá reduciendo debido a que se irán retirando las piezas con el paso del tiempo. La casilla “TInicioViaje” tendrá el objeto de guardar el tiempo actual en ese momento, el cual será el del inicio del *TravelArribaPila*.

Lo siguiente será la propia ejecución del *Travel Ida* tal y como se ha explicado con anterioridad, así como la correspondiente espera del momento en que se capta la pieza representada en el *Process Flow* con la casilla “DelayCojPieza”.

Las dos casillas que se encuentran a continuación son necesarias para la visualización de la simulación, ya que servirán para representar la pieza que será transportada por el brazo robótico.

La caja de “Actualizar size” tiene la función de actualizar el valor de la altura de la pila correspondiente, ya que irá disminuyendo con el paso del tiempo.

“Acquire” y “Release” van a ser trozos de código que evitarán que los dos brazos robóticos de un mismo puesto realicen el traslado de piezas al mismo tiempo y, en consecuencia, choquen.

Se ejecutará después el *Travel Vuelta*, así como la espera del robot al soltar la pieza en el procesador denominada en el *Process Flow* “DelayDejPieza”. Finalmente, se ejecutará el último *TravelNoChoque* y se guardará el tiempo que le ha llevado al brazo robótico realizar las acciones de este bloque restándole al tiempo actual en ese momento el “InicioViaje”. A este tiempo se le dará el nombre de “TViajeRobot” y será de gran importancia a la hora de obtener los resultados.

6.2 Obtención de resultados

6.2.1 Resultados del escenario principal

A la hora de obtener los resultados, es importante tener en cuenta que se busca el máximo rendimiento. Es decir, se pretende obtener un flujo de cajas lo más rápido posible aprovechando los recursos de los que se disponen. Para ello habrá que tomar los valores adecuados de las diferentes velocidades, las disposiciones de los elementos que más convengan y el mayor aprovechamiento de ambos brazos robóticos.

Todo esto ha sido tratado previamente, por lo que solo falta obtener los resultados correspondientes a las elecciones tomadas.

La ocupación de los diferentes puestos robóticos es un parámetro fundamental en este proyecto.

Se diferenciarán cuatro estados para cada robot. “*Idle*” será el tiempo que permanecen desocupados, por el contrario, “*busy*” será cuando se encuentren ocupados. Finalmente, “*offset travel empty*” y “*offset travel loaded*” serán los tiempos en los que se encuentran haciendo los viajes sin carga y con carga. Los viajes sin carga serán los de ida y los viajes sin carga los de vuelta.

En la Figura 44 se representarán los gráficos correspondientes a la ocupación de los robots situados en la parte inferior de cada puesto teniendo en cuenta la vista en planta ordenados desde el primer puesto hasta el último. Por otro lado, en la Figura 45 se representará la ocupación de los situados en la parte superior ordenados de la misma manera. Cabe destacar que los dos últimos puestos robóticos no se usan en este estudio ya que en el escenario configurado de demanda no son necesarios pero la empresa los desea mantener para otros formatos de producto.

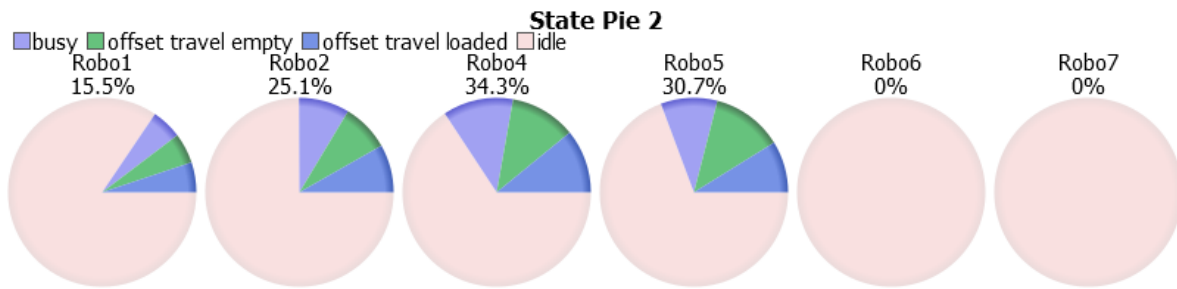


Figura 44. Gráficos ocupación robots inferiores (Elaboración propia).

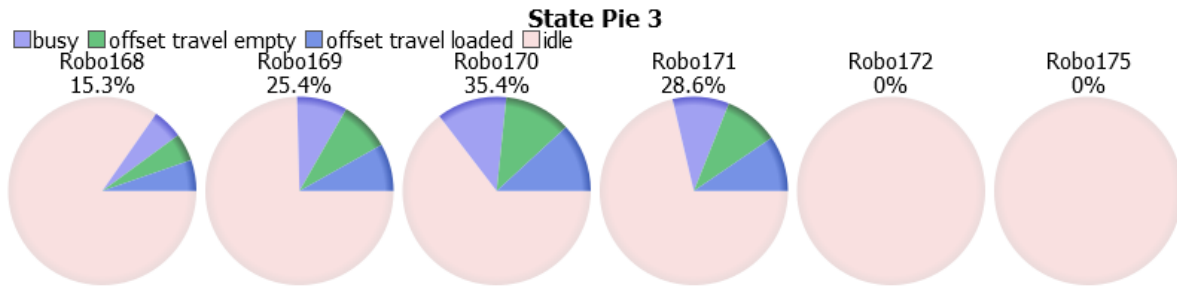


Figura 45. Gráficos ocupación robots superiores (Elaboración propia).

El “TViajeRobot”, como bien se ha explicado anteriormente, va a ser otra de las variables de interés. Cuanto menor sea el valor de esta variable mejor va a ser el rendimiento del proceso, ya que el brazo robótico desempeñará su función con más rapidez y, en consecuencia, las cajas viajarán de manera más fluida.

Esta variable variará en función de:

- **La pila de piezas que los robots alcancen.** El mismo robot obtendrá tiempos diferentes en función de si tiene que alcanzar una pila u otra debido a que éstas se encuentran a diferentes distancias del mismo.
- **El robot que esté tomando la pieza en ese momento.** Teniendo en cuenta una misma pila, dependiendo del robot que tome la pieza habrá un tiempo u otro, ya que ambos se encontrarán a diferentes distancias de la misma.
- **La posición de la pieza dentro de la misma pila.** Dentro de la misma pila y teniendo en cuenta al mismo robot, se tendrán que tomar piezas posicionadas en partes diferentes y variarán ligeramente los tiempos, aunque en un orden mucho menor que en los anteriores casos.

Por todo ello, la gráfica obtenida con la variación de “TViajeRobot” en función del tiempo tomará diferentes valores correspondientes a cada caso tal y como se ve en la Figura 46.

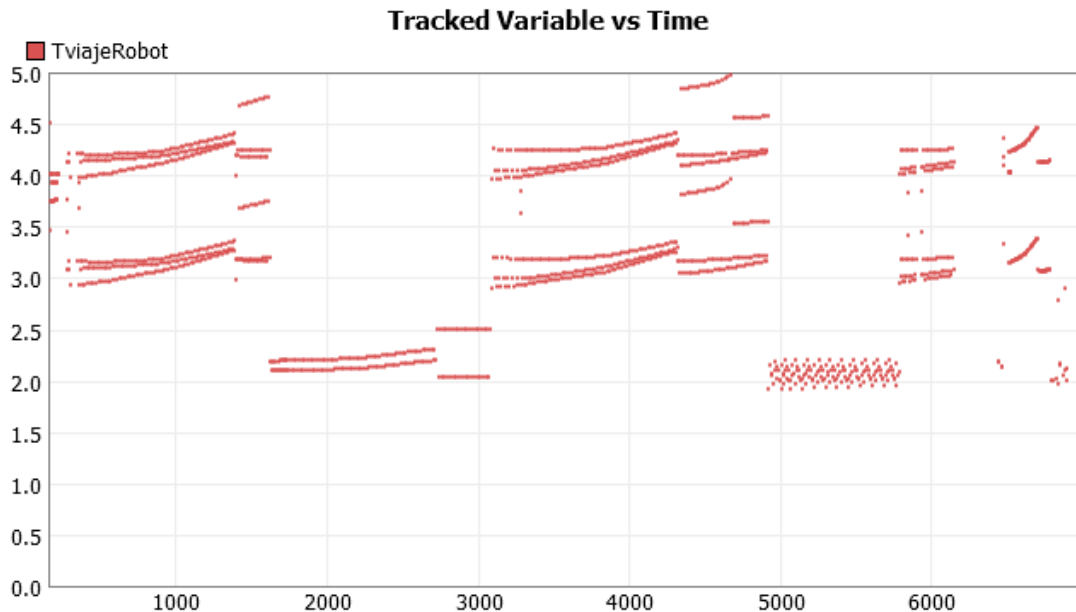


Figura 46. Gráfica TviajeRobot en función del tiempo (Elaboración propia).

En la siguiente figura se representa el mismo valor de la variable TviajeRobot pero solo indicando el valor en ese instante, el máximo, el mínimo y el medio.

TviajeRobot			
Actual	Mínimo	Máximo	Medio
2,02	1,92	4,99	3,40

Figura 47. Valores TviajeRobot (Elaboración propia).

Otra variable de gran importancia a la hora de obtener los resultados va a ser el flujo de cajas por minuto. Cuanto mayor sea esta variable, mejor resultado se obtendrá ya que saldrán más productos terminados por unidad de tiempo de manera que el rendimiento del proceso es mayor.

En la siguiente gráfica se observa la salida de cajas en función del tiempo para el último procesador. Cada pulso que aparece representado con una línea vertical corresponde con la salida de una caja por dicho procesador. Hasta el segundo 375 no han empezado a salir cajas, lo cual es lógico ya que desde que empieza la simulación hasta que las cajas llegan al último procesador pasa un tiempo. Una vez transcurrido este período empiezan a salir los kits con una frecuencia prácticamente constante.

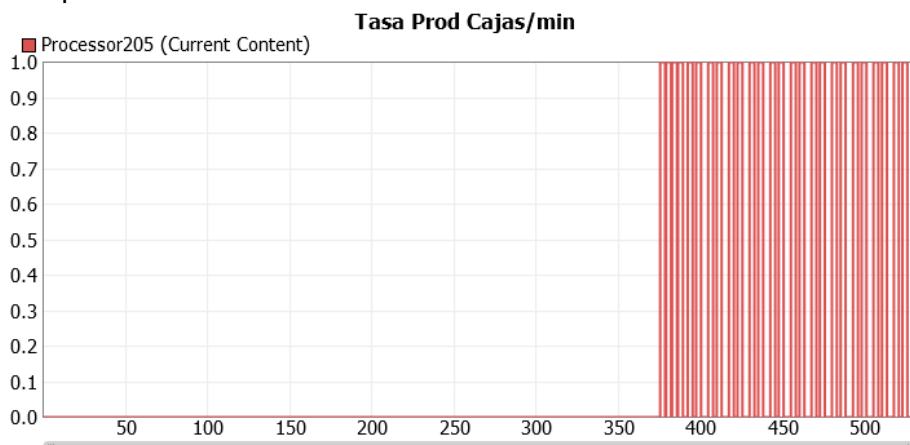


Figura 48. Gráfica tasa de producción de cajas (Elaboración propia).

El siguiente resultado de interés es el número de kits que salen por unidad de tiempo. Se puede observar el valor de cajas completas obtenidas por hora y por minuto. Estos valores variarán a lo largo del tiempo, aunque una vez que pasa un período determinado se empezarán a estabilizar alrededor de los valores que se ven en la Figura 44.

Cajas/hora	Cajas/minuto
462,9	7,7

Figura 49. Valores de salida de kits por unidad de tiempo (Elaboración propia).

6.2.2 Análisis de sensibilidad

En este apartado se van a comparar las tasas de producción teniendo en cuenta tres casos diferentes. El primero de ellos será la situación que se ha considerado hasta el momento en la cual los brazos robóticos tendrán unas velocidades angulares del 50% del valor de las máximas que aporta el fabricante y las aceleraciones y deceleraciones serán de un valor correspondiente a diez veces más el valor de dichas velocidades (caso 1). En el segundo caso se aumentará la velocidad de los robots, pasando a tener un valor del 80% de las velocidades máximas sin variar el valor de las aceleraciones (caso 2). Por último, se evaluará una situación en la cual se mantendrán las velocidades angulares al 50% del valor de las máximas y se disminuirán las aceleraciones y deceleraciones al mismo valor que las velocidades (caso 3). En las siguientes tablas se representan los valores de las velocidades, aceleraciones y deceleraciones para el caso 1, 2 y 3.

Tabla 10. Valores velocidades, aceleraciones y deceleraciones caso 1 (Elaboración propia).

	Velocidad máxima	Aceleración	Deceleración
Joint 1	448	4480	4480
Joint 2	1101	11010	11010
Joint 3	1400	14000	14000
Joint 4	285	2850	2850
Joint 5	346	3460	3460
Joint 6	867	8670	8670

Tabla 11. Valores velocidades, aceleraciones y deceleraciones caso 2 (Elaboración propia).

	Velocidad máxima	Aceleración	Deceleración
Joint 1	717	4480	4480
Joint 2	1763	11010	11010
Joint 3	2241	14000	14000
Joint 4	457	2850	2850
Joint 5	555	3460	3460
Joint 6	1388	8670	8670

Tabla 12. Valores velocidades, aceleraciones y deceleraciones caso 3 (Elaboración propia).

	Velocidad máxima	Aceleración	Deceleración
--	------------------	-------------	--------------

Joint 1	448	448	448
Joint 2	1101	1101	1101
Joint 3	1400	1400	1400
Joint 4	285	285	285
Joint 5	346	346	346
Joint 6	867	867	867

Las tasas de producción obtenidas en los diferentes casos pasados 7000 segundos de simulación se muestran en la siguiente tabla.

Tabla 13. Tasas de producción en los diferentes casos (Elaboración propia).

	Tasa de producción (cajas/min)
Caso 1	7,7
Caso 2	7,7
Caso 3	3,0

Cada viaje del robot está descompuesto en tres etapas, en la primera de ellas, el brazo robótico parte del reposo y aumenta su velocidad a la aceleración que se le indique hasta llegar al valor de velocidad escogido. En la segunda etapa, el robot mantendrá dicha velocidad constante. Por último, en la tercera etapa decelerará hasta llegar a de nuevo al reposo. Teniendo en cuenta que los tramos que realizan los brazos robóticos en este proyecto son muy cortos, cuando el robot deje de acelerar, pasará un intervalo de tiempo muy pequeño hasta que tenga que decelerar, por lo que se mantendrá una duración muy corta a la velocidad indicada. Es por ello que los valores de tasa de producción en el caso uno y dos son los mismos. Por el contrario, al reducir las aceleraciones y deceleraciones tal y como se hace en el caso 3, la tasa de producción disminuye significativamente.

6.2.3 Cambio de disposición

En este apartado se compararán los resultados de la tasa de producción teniendo tres escenarios diferentes.

El primero de ellos será uno en el cual hay un solo robot por puesto. De esta manera, se tendrá la desventaja de contar con un robot menos, de manera que el proceso se ralentizara. Sin embargo, al no existir un segundo robot podrá simplificarse el recorrido que realice. Se prescindirá de los *TravelNochoque* que habían sido añadidos en el escenario principal para evitar que ambos brazos robóticos colisionasen.

El modelo de brazo robótico seguirá siendo el Quantec Ultra ya que la distancia entre robot y pilas será la misma. Al haber solo un único ejemplar de brazo robótico se reducirán los costes con respecto a la configuración principal.

Por todo ello, la situación en cada puesto será la siguiente.

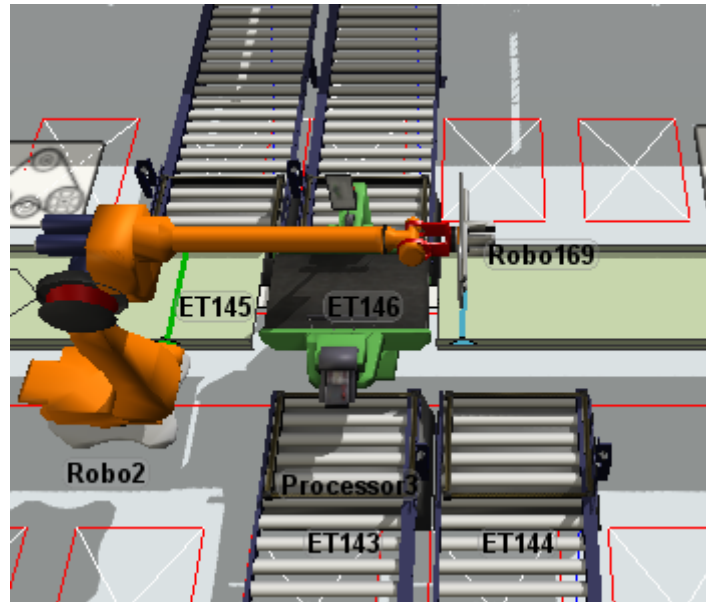


Figura 50. Puesto con un solo robot (Elaboración propia).

Teniendo en cuenta los cambios realizados, la tasa de producción obtenida es la siguiente.

Cajas/hora	Cajas/minuto
285,4	4,8

Figura 51. Tasa de producción con un solo robot (Elaboración propia).

El segundo escenario será el que hemos estudiado hasta ahora en el cual existen dos brazos robóticos modelo Quantec Ultra situados a los laterales de las mesas de rodillos. Para este caso ya se han extraído los resultados anteriormente.

Por último, el tercer escenario que se va a estudiar consistirá en un cambio de posición de los brazos robóticos así como de las mesas de rodillos de manera que cada puesto quede tal y como se ve en la siguiente figura.

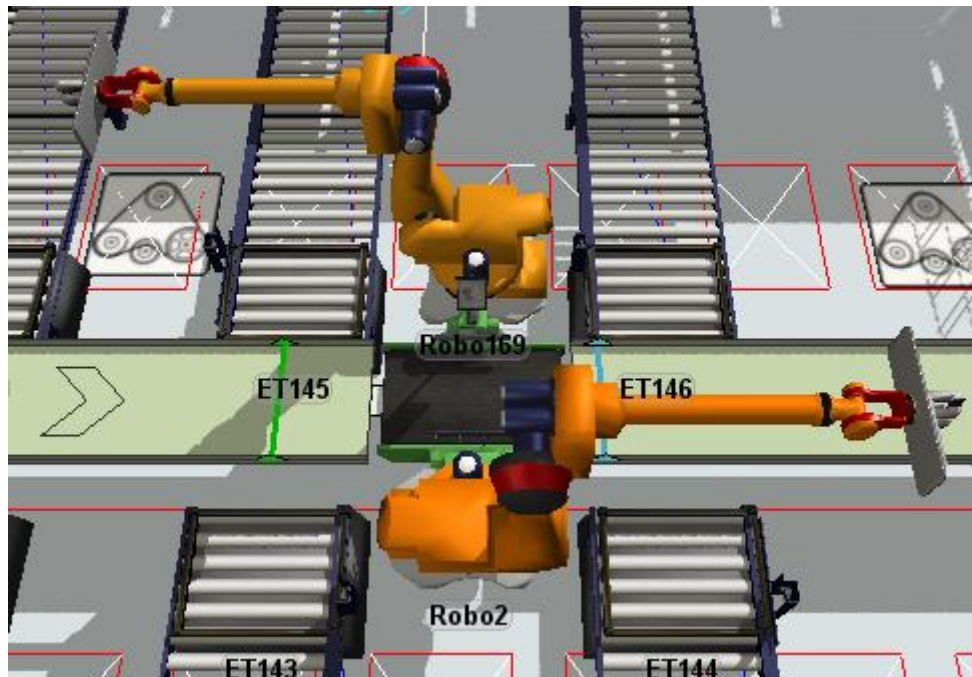


Figura 52. Nueva disposición puestos (Elaboración propia).

Se tendrá en cuenta que las pilas 1 y 2 serán las situadas en la parte inferior a la izquierda y a la derecha, respectivamente y las pilas 3 y 4 serán las de la parte superior también situadas a izquierda y derecha. Se le llamará Robot 1 al situado en la parte inferior y Robot 2 al situado en la parte superior. De esta manera, las distancias entre los robots y las pilas serán las indicadas en la siguiente tabla.

Tabla 14. Distancias en mm entre robots y pilas para la nueva distribución (Elaboración propia).

(mm)	Robot 1	Robot 2
Pila 1	1300	2385
Pila 2	1300	2385
Pila 3	2385	1300
Pila 4	2385	1300

Teniendo en cuenta las distancias a las que tiene que llegar el brazo robótico, se ha considerado que es conveniente cambiar el modelo de robot a uno con menor alcance. De esta manera se ahorrará en costes teniendo en cuenta que se usará un robot de menor tamaño.

El modelo escogido ha sido el Quantec Extra, el cual tiene un alcance máximo de 3100 mm. Se situará de nuevo a una altura de 850 mm de manera que pueda alcanzar con facilidad todas las pilas. Los trayectos que realizará serán los mismos que para la configuración principal.

En la siguiente tabla se comparan las características principales que definen a las tres distribuciones estudiadas así como los resultados obtenidos de la tasa de producción al cabo de 7000 segundos.

Tabla 15. Comparación diferentes escenarios (Elaboración propia).

	Primer escenario	Segundo escenario	Tercer escenario
Modelo de brazo robótico	Quantec Ultra	Quantec Ultra	Quantec Extra
Posición brazos robóticos	Un solo robot al lado de la mesa de rodillos inferior	Al lado de las mesas de rodillos	En medio de las mesas de rodillos
Espacio ocupado por cada puesto	3500 mm x 2700mm	3550 mm x 2700 mm	3500 mm x 2700mm
Tasa de producción	4,8 (cajas/min)	7,7 (cajas/min)	7,2 (cajas/min)

Para el caso de un solo robot, el valor de la tasa de producción que se obtiene es considerablemente menor que para los otros casos y no supera el valor de 7 cajas por minuto, lo cual es el objeto de este estudio. La tasa de producción mayor se obtiene con el segundo escenario. Para el caso de la tasa de producción obtenida con dos robots situados en el centro de las mesas de rodillos, la tasa de producción baja su valor debido a que los brazos robóticos tienen que hacer trayectos con más giros.

6.3 Seguridad

El 90% de los accidentes que se producen con robots industriales son durante su mantenimiento, programación o ajuste, sólo el 10% se producirán durante el funcionamiento de los mismos. Por ello se considera de gran importancia adoptar las medidas de seguridad adecuadas para evitar cualquier problema que se pueda ocasionar.

Existe una normativa de aplicación en toda la UE que dice que es necesario evitar el acceso de personas a la zona de trabajo del robot durante su funcionamiento. Para ello se instalarán unas barreras de seguridad alrededor del mismo cuyas dimensiones serán concordantes al tipo de riesgo existente y a las características del robot instalado. Dichas barreras deben estar cubiertas con una malla que impida permitir a los operarios traspasarla con los dedos o con las manos.

La barrera instalada en el espacio de trabajo se extenderá de manera que recoja el área de alcance del robot. En caso de ser necesaria la entrada a dicha área durante el funcionamiento del robot, éste deberá ser parado mediante algún tipo de acción.

Las normas armonizadas ISO 10218-1 e ISO 10218-2 contienen todos los requisitos mínimos necesarios para el funcionamiento seguro de estos robots industriales.

Esta norma será aplicable al caso estudiado en este trabajo del proceso de empaquetado de kits. Por ello, se han introducido una serie de barreras que, tal y como indica la norma, están lo suficientemente separadas de los brazos robóticos como para éstos no las alcancen al girar, debido a que el alcance del brazo robótico KR Quantec Ultra es de 3900 mm, se han distanciado las barreras a 4000 mm de manera que se esté del lado de la seguridad. La disposición se muestra en la siguiente figura.

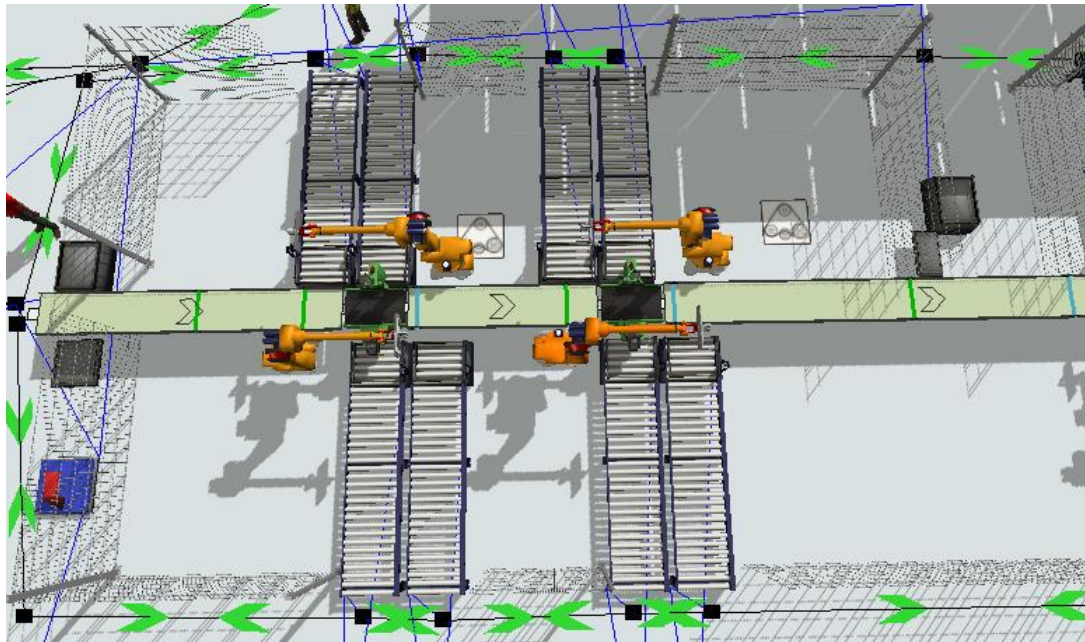


Figura 53. Disposición de las barreras para los puestos 1 y 2 (Elaboración propia).

Tal y como se ve en la parte derecha de la Figura 54, existe un puesto manual llamado ETHerraje al que tendrán que acceder operarios, por lo que debe estar aislado de los robots y por tanto debe haber un acceso entre las barreras.

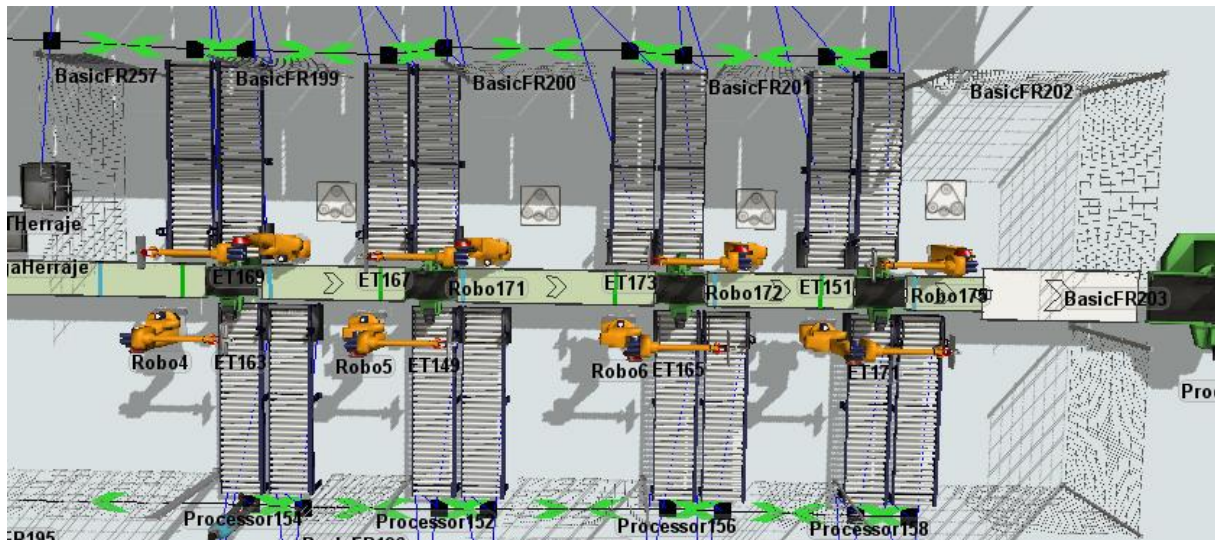


Figura 54. Disposición de las barreras para los puestos 3, 4, 5 y 6 (Elaboración propia).

Por otro lado, la existencia de robots colaborativos provoca que exista otra normativa aplicable en este tipo de casos. Este tipo de robots están diseñados para que puedan estar en funcionamiento en un espacio de trabajo compartido con los trabajadores sin correr ningún tipo de riesgo. Por ello, existe una nueva norma conocida como ISO/TS 15066:2016 “Robots colaborativos” que especifica los requisitos de seguridad para los sistemas con robots industriales colaborativos. En la propia norma se señala que la operación de colaboración es un campo en desarrollo que la nueva especificación técnica probablemente evolucionará en ediciones futuras.

El Baxter es un ejemplo de robot colaborativo, por lo que esta norma será aplicable a robots de este modelo.

6.4 Realidad virtual

La realidad virtual es una tecnología que permite visualizar un entorno 3D previamente creado de manera que el usuario tenga la sensación de estar inmerso en él. El dispositivo que hace posible este hecho son unas gafas o un casco de realidad virtual. Basta con que el usuario se los coloque para poder experimentar la sensación de estar dentro del entorno. Además, el software con el que se trabaje debe tener la capacidad de conectar este tipo de dispositivo. Es el caso del software con el que se trabaja en este proyecto, FlexSim.

Durante la realización de la introducción del brazo robótico en el proceso de empaquetado de kits, se ha recurrido a usar las gafas de realidad virtual. Esto ha permitido visualizar el entorno con más facilidad y detectar errores que con la visualización en pantalla no eran posibles de ver, pudiendo corregirlos a tiempo. Por ejemplo, se han detectado determinadas colisiones del robot con gran facilidad.



Figura 55. Prueba con las gafas de realidad virtual (Elaboración propia).

Por todo ello, puede afirmarse que la realidad virtual es un gran avance en este campo teniendo en cuenta que, en el caso de este proyecto, de no haber sido por el hecho de usarlas, podrían haberse dado lugar errores que supondrían un gran problema en la realidad.

En este proyecto se emplearon las gafas Oculus Rift que pueden ser integradas con Flexsim de manera que se facilitara la visualización en un entorno inmersivo de las cinemáticas programadas para el robot.

7 CONCLUSIONES

En este proyecto se ha usado el software de simulación FlexSim para poder simular un proceso industrial y obtener resultados acerca del mismo. Se ha incorporado un robot Baxter real al entorno de dicho programa y se han comparado los tiempos obtenidos con los reales del propio robot en el laboratorio llegando a la conclusión de que el programa no inducía a grandes errores a la hora de simular los movimientos de un brazo robótico.

En la siguiente parte del proyecto se partía de un proceso de empaquetado de kits que ya había sido estudiado previamente. Se había llegado a la conclusión de que dos robots por puesto era la mejor opción y se había propuesto dos modelos de brazos robóticos KUKA que cumplían los requisitos para realizar esta función. Sin embargo, no se había entrado en detalle en la cinemática del propio robot, lo cual sí se ha estudiado en profundidad en este proyecto.

Los dos robots que habían sido propuestos para introducir en el proceso eran los modelos KR Quantec Extra y KR Quantec Ultra. Teniendo en cuenta los diferentes trayectos que tenían que realizar los brazos robóticos para no colisionar entre ellos ni con las pilas, se pudo concluir que el modelo KR Quantec Extra no tenía alcance suficiente. Además, para evitar las colisiones, los robots no podían alcanzar las pilas y los procesadores simultáneamente, sino que se tenían que alternar.

Además, en los estudios realizados previamente, se han basado en procesos con manipuladores muy complejos, de manera que las velocidades aplicadas a los brazos robóticos eran más lentas de lo que realmente podían ser. En este proyecto las velocidades y aceleraciones introducidas han sido más realistas teniendo en cuenta el tipo de proceso estudiado.

Tras haber comparado diversos escenarios con configuraciones diferentes, se han obtenido resultados precisos de tasas de producción que pueden ser considerados muy próximos a los que habría en la realidad, los mejores resultados se han obtenido con el escenario principal de dos robots KR Quantec Ultra a los laterales de las mesas de rodillos. Como bien se había esperado en el anterior estudio, la tasa de producción obtenida aumenta considerablemente, aunque no al doble, ya que los trayectos de los brazos robóticos se alternan permaneciendo parados en diversas ocasiones. Por otro lado, los tiempos de ciclo de cada robot por pieza son mucho más bajos de los estimados en el estudio previo debido a la elección más realista de los valores de velocidades y aceleraciones, así como de los tiempos de tomar y soltar las diferentes piezas.

El valor de tasa de producción que se quería superar era de 7 cajas por minuto. Usando la configuración elegida con dos brazos robóticos por puesto, dicho valor asciende a 7,7, por lo que se puede considerar que ya es suficiente para que los objetivos propuestos.

Como línea futura de investigación, es interesante mencionar la existencia de algoritmos de control que permiten que los robots no colisionen entre sí. Este tipo de métodos puede que puedan llegar a simplificar los movimientos que realizan cada uno de los brazos robóticos pudiendo llegar a reducir los tiempos de ciclo.

8 BIBLIOGRAFÍA

LAW, 2009. How to build valid and credible simulation models. *IEEE Engineering Management Review*, vol. 37, pp. 57-57. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5235457>

KUKA Robotics, 2018. Pub Spez KR QUANTEC extra HA. *Spez KR QUANTEC extra HA*, V5.1.

KUKA Robotics, 2018. Pub Spez KR QUANTEC K ultra. *Spez KR QUANTEC K ultra*, V11.1.

UNE-EN ISO 10218-1:2012. Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Disponible en:

ISO/TS 15066:2016. Specifies safety requirements for collaborative industrial robot systems and the work environment, and supplements the requirements and guidance on collaborative industrial robot operation given in ISO 10218-1 and ISO 10218-2. Disponible en: <https://www.iso.org/standard/51330.html>

Rethink Robotics, 2016, Baxter Research Robot: Technical Specification Datasheet & Hardware Architecture Overview. <http://www.active8robots.com/wp-content/uploads/Baxter-Hardware-Specification-Architecture-Datasheet.pdf>.